Syntactic Knowledge: A Logic of Reasoning, Communication and Cooperation

Thomas Ågotnes and Michal Walicki
Department of Informatics, University of Bergen
PB. 7800, N-5020 Bergen, Norway
{agotnes,walicki}@ii.uib.no

Abstract

This paper presents a logic for reasoning about the explicit knowledge of agents who represent their knowledge as finite sets of logical formulae, and who can change their knowledge by reasoning and communicating. No assumptions about closure or consistency conditions on the sets, or about soundness or completeness of the reasoning mechanisms, are made. Traditional epistemic logic, based on modal logic, is not a good model for the explicit knowledge of such agents, among other things because of the logical omniscience problem. We are interested in agents who can make different choices about how to reason and communicate, and in reasoning about how the agents can use communication to achieve common goals. To this end, our logic is partly based on Alternating-time Temporal Logic. *The resulting logic allows the expression of properties about an agent's reasoning mechanism on the form "the agent knows modus ponens (MP)" – meaning that the agent's reasoning mechanism can perform the reasoning steps described by MP. Instead of a common closure condition such as "if the agent knows both p and $p \to q$, then he must also know q", the following holds in our logic: "if the agent knows p, $p \to q$ and MP, then he has a strategy to get to know q in the future".*

1. Introduction

In this paper we present a logic for modeling agents who represent their knowledge syntactically as formulae in a language, e.g. as strings stored in a database or written piece of paper[1]. It is assumed that the storage, e.g. the database or piece of paper, is *finite*, i.e. that it can store only finitely many formulae at the same time. We call the finite set of formulae known by an agent at a particular point in time the agent's (finite) *epistemic state*. Neither closure nor consistency conditions on epistemic states are assumed; agents may know contradictions and they will not know anything automatically.

Epistemic logic [12, 25] are used to reason about knowledge in groups of agents. If traditional epistemic logics, based on modal logic, are used as models for agents with finite epistemic states consisting of syntactic objects, however, the concept these logics describe is not what the agents *explicitly* know but rather what they *implicitly* know – what follows from their actual knowledge. Modal epistemic logic cannot be used to describe the explicit knowledge of such agents among other things because of what is known as *the logical omniscience problem (LOP)* [18, 28, 29]: knowledge is closed under logical consequence. Instead of making this assumption, we assume that an agent at a point in time in addition to a finite epistemic state has a reasoning *mechanism* which can be used to transform the epistemic state into a new one. This approach avoids the logical omniscience problem in a non-trivial way — the agents need not be "stupid" – as discussed by Duc [7]: the agents do not know anything automatically, but given a proper mechanism and enough time they can find *any* consequence of their knowledge. Also, they are not necessarily *infallible* – they may have unsound mechanisms.

In addition to reasoning, agents change their knowledge by making observations. Here, we focus on multi-agent systems, and view observation as *communication* – perhaps with an "environment" agent. Communication is similar to reasoning in many ways; communication involves sending information to someone based on one's current knowledge and reasoning can be seen as "communicating with oneself". Thus, we will model communication and reasoning in the same way.

We are interested in agents who can make different *choices* about how to reason and communicate, and in modeling situations where the future choices of the agents are unknown. Agents who can reason and communicate can cooperate strategically to e.g. reach certain epistemic states. There exists a logic which models such non-deterministic

1 Konolige [23] calls this concept "sentential", and uses the term "syntactic" for first-order approaches to epistemic logic.

cooperation: *Alternating-time Temporal Logic (ATL)* [6]. The logic presented in this paper is based on ATL, which is introduced below.

A formal model of epistemic states and mechanisms is presented in Section 2 before the logical language is presented in Section 3. In addition to the ATL operators expressing facts about what coalitions can achieve in the future, the language has operators expressing properties of the reasoning and/or communication mechanisms. Such properties are expressed in the form of *rules*, for example "agent i knows modus ponens". The satisfaction relation is defined in Section 4. In Section 5 examples and properties are discussed, before summary and conclusions in Section 6.

1.1. Alternating-time Temporal Logic

Alternating-time Temporal Logic (ATL) [6] is a generalization of the branching time temporal logic Computational Tree Logic (CTL), in which new path quantifiers $\langle\langle G\rangle\rangle$, one for each set of agents G, are introduced. For example, the formula $\langle\langle G\rangle\rangle\mathcal{F}\phi$ means that the coalition G can cooperate – or that they have a strategy – to ensure that ϕ is true in some future state.

The semantics of ATL is defined by concurrent game structures [5]. The following definition is a slight generalization of the original one.

Definition 1 (Concurrent Game Structure) A concurrent game structure is a tuple

$$(k, Q, \Pi, \pi, ACT, d, \delta)$$

where

- $k > 0$ is a natural number of *players*. The set of players is $\Sigma = \{1, \dots, k\}$.
- Q is a set of *states*.
- Π is a set of *propositions*.
- $\pi(q) \subseteq \Pi$ for each $q \in Q$; *the labeling function*.
- ACT is a set of *actions*.
- For each player $a \in \{1, \dots, k\}$ and state $q \in Q$, $d_a(q) \subseteq ACT$ is the non-empty set of actions available to player a in q. $D(q) = d_1(q) \times \cdots \times d_k(q)$ is the set of *joint actions* in q.
- For each joint action $v \in D(q)$ in a state $q \in Q$, $\delta(q, v) \in Q$; the *transition function*. $\square$

The following are the generalizations made in our definition of a concurrent game structure: the state space and the set of primitive propositions can both be infinite, and the d_a function maps to a general set of actions instead of a natural number. The latter generalization makes it possible to model infinitely many available actions in a state, and makes it more convenient to identify the same action in different states. These generalizations make concurrent game structures more applicable to the problem considered in this paper.

A *computation* λ is an infinite sequence of states; $\lambda = q_0 q_1 \cdots$, where for each $j \geq 0$ there is a joint action $v \in D(q_j)$ such that $\delta(q_j, v) = q_{j+1}$. We use $\lambda[j]$ to denote the element in λ with index j (q_j). A *strategy* for player i is a function $f_i : Q^+ \to ACT$ where $f_i(q_0 \cdots q_m) \in d_i(q_m)$, mapping any finite prefix of a computation to an action for player i. Given a state q and a set of strategies for a group of agents $G \subseteq \Sigma$, $\vec{f_G} = \{f_i : i \in G\}$, $out(q, \vec{f_G})$ denotes the set of possible computations starting in state q where the agents in G use the strategies $\vec{f_G}$ and each agent $a \in \Sigma \setminus G$ is only restricted, in state q', to choices from $d_a(q')$. The set of all strategies for a group G is denoted $Str(G)$.

The syntax of the ATL language is defined over Π and Σ. Π are formulae, and if ϕ_1, ϕ_2 are formulae, $G \subseteq \Sigma$ and $a \in \Sigma$ then $\neg\phi_1$, $\phi_1 \vee \phi_2$, $\langle\langle G\rangle\rangle \bigcirc \phi_1$, $\langle\langle G\rangle\rangle\Box\phi_1$ and $\langle\langle G\rangle\rangle\phi_1\mathcal{U}\phi_2$ are formulae. Satisfiability of a formula ψ in a state q of a concurrent game structure S is defined as follows, where $p \in \Pi$:

$$\begin{array}{l}
S, q \models p \Leftrightarrow p \in \pi(q) \\
S, q \models \neg\phi \Leftrightarrow S, q \not\models \phi \\
S, q \models \phi_1 \vee \phi_2 \Leftrightarrow S, q \models \phi_1 \text{ or } S, q \models \phi_2 \\
S, q \models \langle\langle G\rangle\rangle \bigcirc \phi \Leftrightarrow \\
\quad \exists_{\vec{f_G} \in Str(G)} \forall_{\lambda \in out(q, \vec{f_G})} S, \lambda[1] \models \phi \\
S, q \models \langle\langle G\rangle\rangle\Box\phi \Leftrightarrow \\
\quad \exists_{\vec{f_G} \in Str(G)} \forall_{\lambda \in out(q, \vec{f_G})} \forall_{j \geq 0} S, \lambda[j] \models \phi \\
S, q \models \langle\langle G\rangle\rangle\phi_1\mathcal{U}\phi_2 \Leftrightarrow \\
\quad \exists_{\vec{f_G} \in Str(G)} \forall_{\lambda \in out(q, \vec{f_G})} \\
\qquad \exists_{j \geq 0}(S, \lambda[j] \models \phi_2 \text{ and } \forall_{0 \leq k < j} S, \lambda[k] \models \phi_1)
\end{array}$$

The usual derived propositional connectives are used, including $\top$ for an arbitrary propositional tautology, in addition to $\langle\langle G\rangle\rangle\mathcal{F}\phi$ for $\langle\langle G\rangle\rangle\top\mathcal{U}\phi$.

1.1.1. Incomplete Information In the concurrent game structures of interest in this paper, the players have *incomplete information* about the current (global) state (an agent do not necessarily know the contents of another agent's epistemic state). Two natural restrictions on such structures, and their associated strategies, are the following:

1. An agent must have the same actions available in two states which are indiscernible for that agent.
2. A strategy must map indiscernible histories of states to the same action.

Alur et. al [5] propose a restriction on concurrent game structures for players with incomplete information, which includes the two mentioned restrictions, but only in the special case of turn-based synchronous structures. These conditions have recently [19, 31, 2, 20] been discussed in the special case of *Alternating-time Temporal Epistemic Logic (ATEL)* [30]. The following is a generalization, defining a general requirement on concurrent game structures with incomplete information.

Definition 2 A *game structure with incomplete information* is a concurrent game structure $(k, Q, \Pi, \pi, ACT, d, \delta)$ together with a set $\Pi_a \subseteq \Pi$ of *observable propositions* for each player a such that

$$d_a(q) = d_a(q')$$

for any states $q, q' \in Q$ such that

$$\pi_a(q) = \pi_a(q')$$

where we write $\pi_a(q'') = \pi(q'') \cap \Pi_a$ for any $q'' \in Q$. $\Box$

$\pi_a(q)$ is the set of true formulae in q observable by a.

A *strategy* for a player a in a concurrent game structure with incomplete information is a strategy with the following restriction:

$$f_a(q_1 \cdots q_m) = f_a(q'_1 \cdots q'_m)$$

for any $m \geq 0$ and $q_i \in Q$ having

$$\pi_a(q_i) = \pi_a(q'_i)$$

for all $1 \leq i \leq m$. These definitions are very general adaptions of the turn-based case. Thus, for satisfiability with respect to a concurrent game structure with incomplete information only these restricted strategies will be considered.

It should be noted that strategies under incomplete information is a subtle issue. Advantages and problems with the properties expressed here in addition to other possibilities are discussed in depth in [20].

2. Epistemic States and Mechanisms

An *epistemic state* is a finite, but otherwise arbitrary, set of formulae in an *object language*. Here, we assume that the object language is propositional logic. Generally, given a set Θ', $prop(\Theta')$ is the least set which contains Θ' and is closed under the connectives $\wedge$ and $\neg$.

The object language OL is assumed to be parameterized by a set of primitive propositions Θ, and an epistemic state s is a finite subset of OL:

$$OL = prop(\Theta) \qquad\qquad s \in \wp^{fin}(OL)$$

An epistemic state is a model of an agent at a point in time, while a *mechanism* models the possible future behaviour of the agent. In other words, a mechanism describes how the epistemic state of the agents can change over time. We assume that each agent at each time step sends a finite (possibly empty) set of formulae to all of the agents in the next time step including to himself. The latter is used to model reasoning. This action can be described by a tuple

$$(s'_1, \ldots, s'_n)$$

with one finite set s'_j of object formulae for each agent j. We also assume that the agent can have several different *choices* of tuples to use. Since the agent cannot discern between situations in which he has the same epistemic state, the possible choices must be a function of the epistemic state. Thus, we model a mechanism for agent i as mapping an epistemic set s_i to a set $R_i(s_i)$ of tuples.

Definition 3 (Joint Mechanism) A *joint mechanism* (for n agents) is a tuple

$$R = (R_1, \ldots, R_n)$$

where

$$R_i \subseteq \wp^{fin}(OL) \times (\wp^{fin}(OL))^n$$

and

$$R_i(s) = \{g : (s, g) \in R_i\} \neq \emptyset$$

for every $s \in \wp^{fin}(OL)$. $\Box$

Note the different status of the members of $(s'_1, \ldots, s'_n) \in R_i(s_i)$: s'_i is the set i wants himself to know (maybe just remember from last time step, if $s'_i \subseteq s_i$) in the next time step, while s'_j $(j \neq i)$ is the information i sends to another agent j. $s'_j = \emptyset$ means "do not send anything to j", while $s'_i = \emptyset$ means "forget everything and do not make any new inferences". It is required that every agent can choose at least one tuple in any state.

A *global state* in the multi agent system consists of one epistemic state for each agent. The transition between global states is defined as follows: each agent selects a tuple, and an agent's next epistemic state is the union of the sets sent to that agent from all agents (including from himself). At first glance, this definition where an agent can "insert" formulae directly into another agent's epistemic state, may seem unrealistic. However, there are several reasons for it. First, there are many applications with secure communication where an agent *will* know something just because another agent said it. Second, the language presented in the next section can express restrictions on the form of the formulae an agent can "insert" into another agent. For example, we could imagine Bob only being able to alter Mary's epistemic state with formulae on the form "Bob said that ...". Third, non-secure communication can be modeled by an "environment" agent. Semantics will be formally defined in Sec. 4, after the logical language is presented.

3. Language

The logical language is the language of propositional logic extended with three new fragments. First, *epistemic* *operators* are used in statements about the agents' current knowledge, i.e. about their epistemic state. Second, *rule operators* and *rules* are used in statements about the agents' ability to reason and communicate, i.e. about their mechanisms. Third, *temporal operators* are used to express statements about strategic cooperation.

The temporal operators in the language are those of ATL. The epistemic and rule operators are now introduced, before the language is formally defined.

3.1. Knowledge

The fact that agent i *knows* a finite set X of OL formulae is expressed by the formula

$$\triangle_i X$$

The reason for this non-standard notation, compared to the more commonly used $K_i\phi$ where ϕ is a single formula, is that we also define a dual operator shown in the following formula:

$$\triangledown_i X$$

The latter formula means that agent i knows *at most* the formulae X; he does not necessarily know *all* of them but he does not know anything *not* in X. The former formula can thus be read as "agent i knows *at least* X", and their conjunction, written $\Diamond_i X \equiv \triangle_i X \wedge \triangledown_i X$, as "agent i knows *exactly* X". Semantically, $\triangle_i X$, $\triangledown_i X$ and $\Diamond_i X$ will respectively be true if i's epistemic state is included in, includes and coincides with X.

3.2. Reasoning and Communication Rules

Rules are defined over two sets of *variables*: $V_F = \{a, b, c, \ldots\}$ and $V_T = \{t, u, v, \ldots\}$, used as place-holders for formulae and finite sets of formulae respectively. A rule consists of an *antecedent* and a *consequent*; both representing finite sets of formulae — possibly containing variables. An example of a rule is:

$$R_1 = \frac{t \sqcup \{a \to b, a\}}{t \sqcup \{b\}}$$

The intended meaning of "knowing a (reasoning) rule" is that if the current epistemic state matches the antecedent, then it can be changed to an epistemic state matching the consequent.

Rules are used in the meta-language by introducing new epistemic operators for rules.

$$\overset{\leadsto}{\triangle}_i\{R_1\}$$

denotes the fact that agent i knows at least the rule R_1 (but he may know more rules), meaning that i must be able to use the rule in all possible ways in the current epistemic state.

$$\overset{\leadsto}{\triangledown}_i\{R_1, R_2, R_3\}$$

denotes the fact that at most rules R_1, R_2 and R_3 are known by i (but it may be the case that i does not know all of them), meaning that everything his mechanism can do in the current epistemic state must be explained by one of the rules.

Even though $\triangle_i$ and $\overset{\leadsto}{\triangle}_i$ ($\triangledown_i$ and $\overset{\leadsto}{\triangledown}_i$) are similar in appearance and have similar meanings, *rules are not formulae* — they have a different ontological status. Hence, it is not possible to write e.g. $\triangle_i\{p, R_1\}$ to denote the fact that agent i knows both the proposition p and the rule R_1 — this fact should be written $\triangle_i\{p\} \wedge \overset{\leadsto}{\triangle}_i\{R_1\}$. Furthermore, unlike formulae rules are not closed under propositional connectives; $\overset{\leadsto}{\triangle}_i\{R_1 \wedge R_2\}$ is not a well formed formula. Rules *only* appear as arguments to the operators $\overset{\leadsto}{\triangle}_i, \overset{\leadsto}{\triangledown}_i$.

The ability to communicate can be expressed in a very similar manner to reasoning: agent i can in a certain epistemic state perform a communication action resulting in a change in the epistemic state of agent j. In this view, communication can be seen as a generalization of reasoning. To be able to express communication, the $\overset{\leadsto}{\triangle}$-operator just introduced is generalized to taking *two* subscripts:

$$\overset{\leadsto}{\triangle}_{ij}\{R_1\}$$

means that agent i knows at least the rule R_1 for communication to agent j.

$$\overset{\leadsto}{\triangledown}_{ij}\{R_1\}$$

means that agent i knows at most the rule R_1 for communication to agent j. Note that $\overset{\leadsto}{\triangledown}_{ij}\{R_1\}$ does not say anything about agent i's ability to communicate with agent k when $k \neq j$. The first version, denoting reasoning, is defined as a short-hand notation for the second; $\overset{\leadsto}{\triangle}_i \equiv \overset{\leadsto}{\triangle}_{ii}$ and similarly for $\overset{\leadsto}{\triangledown}_i$. Although e.g. $\overset{\leadsto}{\triangle}_{ii}$ and $\overset{\leadsto}{\triangle}_{ij}$ ($i \neq j$) are syntactically and semantically similar, they are *used* in very different ways. For example, a $\overset{\leadsto}{\triangle}_{ij}\{\frac{t \sqcup \{a\}}{\{a\}}\}$ says that i can communicate anything he knows to j, while rule R_1 above would seldom be used as a communication rule.

The exact meaning of the rule operators will become clear in Sec. 4.

3.3. Formal Definitions

As discussed above, the language will involve statements about sets of object language formulae and of rules. Formally, sets will be represented as terms. The following definition constructs the set of terms for a given language L

and a set of atomic terms S (the latter can e.g. be variables standing for terms, as will be the case when we define rules shortly).

Definition 4 ($TL(L,S)$) Given a set of formulae L and a set of (atomic) terms S, the *term language* $TL(L,S)$ is the least set such that

- $S \subseteq TL(L,S)$
- If $\alpha_1,\ldots,\alpha_k \in L$ then $\underline{\{}\alpha_1\underline{,}\ldots\underline{,}\alpha_k\underline{\}} \in TL(L,S)$
- If $T,U \in TL(L,S)$ then $\left.\begin{array}{c}(T \sqcup U)\\(T \sqcap U)\end{array}\right\} \in TL(L,S)$ □

Given a term $T \in TL(L,\emptyset)$ in a term language without atomic terms, the *set interpretation* $[T] \subseteq L$ of the term is the finite set of L-formulae intuitively represented by the term:

$$[\underline{\{}\alpha_1\underline{,}\ldots\underline{,}\alpha_k\underline{\}}] = \{\alpha_1,\ldots,\alpha_n\}$$
$$[T \sqcup U] = [T] \cup [U] \qquad [T \sqcap U] = [T] \cap [U]$$

Henceforth, we drop the underlined notation for terms and write them in the usual set notation.

The logical, or meta, language is called TEL. Several other formal languages are involved in its definition below, particularly in the definition of rules. The language of rules is called RL and involves an extension OL_V of the object language with the formula variables V_F. Antecedents/consequents of rules are terms in the term language of OL_V with term (set) variables V_T as atomic terms. $TL(OL,\emptyset)$ and $TL(RL,\emptyset)$ represent sets of object formulae and rules, respectively.

The language is defined for a number of agents n, primitive propositions Θ, formula variables V_F and term (set) variables V_T.

Definition 5 (OL_V,RL) RL is the least set such that:

- $OL_V = \mathit{prop}(\Theta \cup V_F)$
- If $T,U \in TL(OL_V, V_T)$ then $\frac{T}{U} \in RL$ □

Definition 6 (TEL) TEL is the least set such that:

- $\Theta \subseteq TEL$
- If $T,U \in TL(OL,\emptyset)$ then $T \doteq U \in TEL$
- If $T \in TL(OL,\emptyset)$ then
  - $\triangle_i T \in TEL$
  - $\triangledown_i T \in TEL$
- If $T^R \in TL(RL,\emptyset)$ then
  - $\overset{\leadsto}{\triangle}_i T^R \in TEL$
  - $\overset{\leadsto}{\triangledown}_i T^R \in TEL$
- If $\alpha,\beta \in TEL$ then
  - $(\alpha \rightarrow \beta) \in TEL$
  - $\neg\alpha \in TEL$
- If $\alpha,\beta \in TEL$ and $G \in \wp(\{1,\ldots,n\})$ then
  - $\langle\langle G\rangle\rangle \bigcirc \alpha \in TEL$
  - $\langle\langle G\rangle\rangle \Box \alpha \in TEL$
  - $\langle\langle G\rangle\rangle \alpha \mathcal{U} \beta \in TEL$ □

In addition, the following short hand notation is used for TEL formulae in addition to the usual derived propositional connectives: $T \preceq U$ for $T \sqcup U \doteq U$, $\Diamond_i T$ for $\triangle_i T \wedge \triangledown_i T$, $\overset{\leadsto}{\Diamond}_{ij} T^R$ for $\overset{\leadsto}{\triangle}_{ij} T^R \wedge \overset{\leadsto}{\triangledown}_{ij} T^R$, $\langle\langle G\rangle\rangle \mathcal{F}\phi$ for $\langle\langle G\rangle\rangle \top \mathcal{U} \phi$ and $[\![G]\!] A\phi$ for $\neg\langle\langle G\rangle\rangle A \neg\phi$ where $A \in \{\bigcirc, \Box, \mathcal{F}\}$. The following notation will be used for (meta) variables: $\phi,\psi,\ldots$ for elements in TEL, $\alpha,\beta,\ldots$ for elements in OL, $T,U,\ldots$ for terms (sets) over OL, $R_1,R_2,\ldots$ for elements in RL, $T_1^R,T_2^R,\ldots$ for terms (sets) over RL, $a,b,\ldots$ for elements in V_F and $t,u,\ldots$ for elements in V_T.

The subset of TEL with no occurrences of rule operators or temporal operators is called the *epistemic fragment*. Formulae starting with rule operators are called *rule formulae*.

3.3.1. Substitutions A *substitution* Ω is a pair of functions

$$\Omega_F : V_F \rightarrow OL$$
$$\Omega_T : V_T \rightarrow TL(OL,\emptyset)$$

mapping each formula-variable $a \in V_F$ to an object formula and each term variable $t \in V_T$ to a term standing for a set of object formulae. The set of all substitutions is called Subst. We abuse notation and write only Ω for both Ω_F and Ω_T.

Substitutions will be used to define *instances* of a rule $R \in RL$ — i.e. to identify agent formulae *matching* the antecedent and consequent of the rule.

Definition 7 (T_Ω) Let $\Omega \in \mathit{Subst}$ and $T \in TL(OL_V, V_T)$. T_Ω is the result of replacing every variable $x \in V_F \cup V_T$ occurring in T with $\Omega(x)$. Clearly, $T_\Omega \in TL(OL,\emptyset)$. □

4. Semantics

We formally define the semantics of TEL. First, rules are interpreted as relations over epistemic states.

4.1. Interpretation of Rules

The semantics of a rule term is defined by combining all substitutions with all rules in the set interpretation of the rule term.

Definition 8 ($[\![T^R]\!]$) Let $T^R \in TL(RL, \emptyset)$ be a rule term. $[\![T^R]\!]$ is the following relation:

$$[\![T^R]\!] = \{([T^A_\Omega], [T^C_\Omega]) : \frac{T^A}{T^C} \in [T^R], \Omega \in Subst\}$$

We will write $[\![T^R]\!](s)$ for $\{s' : (s, s') \in [\![T^R]\!]\}$. □

Examples:

$$[\![\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{b\}}]\!](s) = \{s' \cup \{\beta\} : s = s' \cup \{\alpha, \alpha \to \beta\}\}$$

$$[\![\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}]\!](s) = \{s \cup \{\beta\} : \alpha, \alpha \to \beta \in s\}$$

Note an important property of the $\sqcup$ operator: it denotes not-necessarily disjoint union. Thus, the variable t in the first example rule above *can* denote a set which includes the premises denoted by the expressions a and $a \to b$ in the premise.

4.2. Induced Concurrent Game Structures and Satisfaction

A joint mechanism describes a concurrent game structure as follows (explanation follows).

Definition 9 (Ind. Conc. Game Str. with Incompl. Inf.) Given a mechanism for n agents, $R = (R_1, \ldots, R_n)$, and a truth assignment $\pi : \Theta \to \{\mathbf{true}, \mathbf{false}\}$ of Θ, the *induced concurrent game structure with incomplete information* is the concurrent game structure

$$\overrightarrow{R}_\pi = (n, Q, \Pi, \pi', ACT, d, \delta)$$

together with sets Π_a of observable propositions for each agent $1 \le a \le n$ where

- $Q = (\wp^{fin}(OL))^n$
- $\Pi = \Theta$
  $\cup \{T \doteq U : T, U \in TL(OL, \emptyset)\}$
  $\cup \{\triangle_i T, \triangledown_i T : T \in TL(OL, \emptyset), 1 \le i \le n\}$
  $\cup \{\overset{\leadsto}{\triangle}_i T^R, \overset{\leadsto}{\triangledown}_i T^R : T^R \in TL(RL, \emptyset), 1 \le i \le n\}$
- Let $q = (s_1, \ldots, s_n) \in Q$. Let $\phi \in \Pi$.
  - $\phi = p \in \Theta$: $\phi \in \pi'(q)$ iff $\pi(p) = \mathbf{true}$.
  - $\phi = T \doteq U$: $\phi \in \pi'(q)$ iff $[T] = [U]$.
  - $\phi = \triangle_i T$: $\phi \in \pi'(q)$ iff $[T] \subseteq s_i$.
  - $\phi = \triangledown_i T$: $\phi \in \pi'(q)$ iff $s_i \subseteq [T]$.
  - $\phi = \overset{\leadsto}{\triangle}_{ij} T^R$: $\phi \in \pi'(q)$ iff
  $$\begin{gathered} s' \in [\![T^R]\!](s_i) \text{ and } (s'_1, \ldots, s'_n) \in R_i(s_i) \\ \Downarrow \\ (s''_1, \ldots, s''_n) \in R_i(s_i) \\ \text{where } s''_k = \begin{cases} s' & k = j \\ s'_k & \text{otherwise} \end{cases} \end{gathered}$$
  - $\phi = \overset{\leadsto}{\triangledown}_{ij} T^R$: $\phi \in \pi'(q)$ iff $(s'_1, \ldots, s'_n) \in R_i(s_i) \Rightarrow s'_j \in [\![T^R]\!](s_i)$
- $ACT = (\wp^{fin}(OL))^n$
- Let $q = (s_1, \ldots, s_n) \in Q$ and $1 \le a \le n$. $d_a(q) = R_a(s_a) \subseteq ACT$.
- Let $q = (s_1, \ldots, s_n) \in Q$ and $v = (g_1, \ldots, g_n) \in D(q) = R_1(s_1) \times \cdots \times R_n(s_n)$ where $g_i = (s^i_1, \ldots, s^i_n)$. Then,
  $$\delta(q, v) = (\bigcup_{i=1}^n s^i_1, \ldots, \bigcup_{i=1}^n s^i_n)$$
- Let $i \in [1, n]$. $\Pi_i = \{\triangle_i T, \triangledown_i T : T \in TL(OL, \emptyset)\}$ □

The definition will be discussed in detail shortly. Truth of a formula is defined relative to a joint mechanism, an epistemic state for each agent, and a truth assignment to the primitive propositions.

Definition 10 (Satisfiability of *TEL*) A formula $\phi \in TEL$ is *satisfied* by a joint mechanism R, epistemic states $s_i \in \wp^{fin}(OL)$, $1 \le i \le n$, and a truth assignment $\pi : \Theta \to \{\mathbf{true}, \mathbf{false}\}$, written

$$(R, s_1, \ldots, s_n, \pi) \models \phi$$

iff

$$\overrightarrow{R}_\pi, (s_1, \ldots, s_n) \models \phi$$ □

Validity of a formula ϕ in a joint mechanism R, written $R \models \phi$, is defined as truth in that mechanism and all epistemic states and all truth assignments; validity of ϕ, written $\models \phi$, is defined as validity in all mechanisms.

It is easy to see that the observable propositions are exactly those which describe the agent's epistemic states, i.e. that $\pi_a(s_1, \ldots, s_n) = \pi_a(s'_1, \ldots, s'_n) \Leftrightarrow s_a = s'_a$, and that Def. 9 indeed defines a concurrent game structure with incomplete information.

We want to model agents with incomplete information who must base their decisions on their (possibly historical) knowledge, and in Def. 10 the notion of a strategy is implicitly restricted as described in Sec. 1.1.1 since the induced concurrent game structure is defined with incomplete information.

(Global) states are tuples of epistemic states, and an action corresponds to selecting a tuple $(s_1, \ldots, s_n)$ for communication to all agents. The set of actions available in a state q, $d_a(q)$, is the set described by a's mechanism. δ maps an action $(s^i_1, \ldots, s^i_n)$ for each agent i to a new tuple of epistemic states $(\cup_{i=1}^n s^i_1, \ldots, \cup_{i=1}^n s^i_n)$. In other words, that agent i uses action $(s^i_1, \ldots, s^i_n)$ means that he "sends" s^i_j to each agent j – including himself – and thereby forces the new epistemic state of j to be at or above s^i_j.

π' extends the truth assignment of Θ to a truth assignment for the epistemic fragment and the rule formulae, relative to a state. $\triangle_i T$, $\triangledown_i T$ and $\Diamond_i T$ are true, respectively, when i's epistemic state is s_i if $[T] \subseteq s_i$, $s_i \subseteq [T]$ and $s_i = [T]$.

$\overset{\rightsquigarrow}{\triangle}_{ij} T^R$, i knowing "at least" a rule term T^R for communication to j, when i's state and mechanism are s_i and R_i holds if we can substitute (finite) sets of formulae for the term (set) variables $t, u, \ldots$ and singular formulae for the formula variables $a, b, \ldots$ such that the antecedent of a rule in T^R is equal to s_i and there is a $(s'_1, \ldots, s'_n) \in R_i(s_i)$ such that the consequent of the rule is equal to s'_j[2]. In other words, i must be able to send every set in $[\![T^R]\!](s_i)$ to j. Agent i may be able to do *more* with his mechanism, in state s_i, than described by the rule; he may be able to send sets of formulae to j which are not in $[\![T^R]\!](s_i)$. If he knows "at most" T^R for communication to j in s_i, $\overset{\rightsquigarrow}{\triangledown}_{ij} T^R$, however, every set he can send to j must be in $[\![T^R]\!](s_i)$: if s_i and R_i are i's epistemic state and mechanism, this formula holds if for every $(s'_1, \ldots, s'_n) \in R_i(s_i)$, s'_i is equal to the consequent of one of the rules with substitutions such that the antecedent is equal to s_i. But by the fact that he knows "at most" T^R alone it is not required that he actually *can* send every set in $[\![T^R]\!](s_i)$ to j.

Examples follow below.

Note that the derived $\overset{\rightsquigarrow}{\Diamond}_{ij} T^R \equiv \overset{\rightsquigarrow}{\triangle}_{ij} T^R \wedge \overset{\rightsquigarrow}{\triangledown}_{ij} T^R$ can be used to specify a complete joint mechanism[3]:

$$\begin{gathered}(R, s_1, \ldots, s_n, \pi) \models \bigwedge_{j\in[1,n]} \overset{\rightsquigarrow}{\Diamond}_{ij} T_j^R \\ \Updownarrow \\ R_i(s_i) = [\![T_1^R]\!](s_i) \times \cdots \times [\![T_n^R]\!](s_i)\end{gathered}$$

5. Examples and Properties

We focus here on properties of the rule operators and how they can be used together with the temporal operators to express statements about how knowledge evolves; see [4] for a discussion about the epistemic operators (and $\triangledown_i$ in particular).

As an example of a rule for reasoning (rather than communication), "agent i knows at least modus ponens" can be expressed by the formula

$$\phi_1 = \overset{\rightsquigarrow}{\triangle}_{ii}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{a, a \to b, b\}}\}$$

If this formula is true when s_i and R_i are i's epistemic state and mechanism, and $p, \to\ q \in s_i$, then there is a $(s'_1, \ldots, s'_n) \in R_i(s_i)$ such that $s'_i = s_i \cup \{q\}$.

A reasoning property involving the "at most" rule operator is the following, where $t, u \in V_T$:

$$\phi_2 = \overset{\rightsquigarrow}{\triangledown}_{ii}\{\frac{t}{t \sqcup u}\} \tag{1}$$

In general there is no requirement about *monotonicity* of the reasoning part of mechanisms; agents may forget. ϕ_2 is an axiomatization of monotone mechanisms. If ϕ_2 is true in a state then every possible set of formulae i can "send" to his own next epistemic state is an instance of the consequent of the rule at the same time that the (whole) current epistemic state is an instance of the antecedent of the rule. Since the antecedent of the rule is a variable representing a set, it can only match the current epistemic state in one way – as the whole epistemic state. Thus, the consequent can only extend the epistemic state (or leave it unchanged) [4].

An example where the "at most" operator is used to express a property of communication is the following, where $t \in V_T$ and $a \in V_F$:

$$\phi_3 = \overset{\rightsquigarrow}{\triangledown}_{ij}\{\frac{t \sqcup \{a\}}{\{a\}}\} \tag{2}$$

ϕ_3 expresses the fact that everything agent i can communicate to agent j must be known by agent i , and that i can send at most one formula to j at each time step[5].

An example involving several of the operators is the following formula, where $p, q \in \Theta$, $t \in V_T$ and $a, b \in V_F$:

$$\begin{aligned}&(\triangle_1\{p\} \wedge \triangle_2\{p \to q\} \wedge \langle\langle\emptyset\rangle\rangle\Box \\ &\Big(\overset{\rightsquigarrow}{\triangle}_{12}\{\frac{t \sqcup \{a\}}{\{a\}}\} \wedge \overset{\rightsquigarrow}{\triangle}_{22}\{\frac{t \sqcup \{a, a \to b\}}{t \sqcup \{b\}}\} \wedge \overset{\rightsquigarrow}{\triangledown}_{22}\{\frac{t}{t \sqcup u}\}\Big)) \\ &\qquad \to \langle\langle\{1,2\}\rangle\rangle \mathcal{F} \triangle_2 \{q\}\end{aligned} \tag{3}$$

The intended meaning of this formula is that if agent 2 knows the rule modus ponens[6] and the formula $p \to q$ and reasons monotonically, and agent 1 knows the formula p and can communicate any known formula to 2, then agents 1 and 2 can cooperate to make 2 know the formula q in the future. Note that the formula $\overset{\rightsquigarrow}{\triangle}_{ij} T^R$ means only that agent i knows the rules T^R *now*, i.e. that his mechanism

2 In addition agent i can send the set s'_j to j *independently* of what he sends to the other agents. Unfortunately, there is not space here to elaborate on this point, but see footnote 3.

3 Due to the way $\overset{\rightsquigarrow}{\triangle}_{ij}$ is defined as forming a cartesian product.

4 Technically, if ϕ_2 is true when the epistemic state and mechanism of i are s_i and R_i, and $(s'_1, \ldots, s'_n) \in R_i(s_i)$, then $s_i \subseteq s'_i$.

5 Technically, if ϕ_2 is true when the epistemic state and mechanism of i are s_i and R_i, and $(s'_1, \ldots, s'_n) \in R_i(s_i)$, then it must be the case that $s'_j = \{\alpha\}$ for some $\alpha \in s_i$.

6 Note that although the version of modus ponens used here does not explicitly say that the formulae used in the antecedent must be remembered, they will be because of the monotony clause. If the other version of modus ponens from the formula ϕ_1 was used instead, the formula would be equivalent.

can do the things described by T^R at the current epistemic state – but not necessarily in the future when he may be in other epistemic states (much like the formula $\triangle_i\{p\}$ means that i knows p *now* but not necessarily in the future). The fact that ϕ is true now and always in the future can be expressed in ATL by the formula $\langle\langle\emptyset\rangle\rangle\Box\phi$, and this construction is used in the formula (3) to express the fact that the requirements on the mechanisms hold globally. The formula (3) is valid.

Some other syntactic properties of the logic are shown in the following Lemma.

Lemma 11 The following are valid formulae:

1. $\Diamond_i T \to$
$$\left(\bigwedge_{1\leq j\leq n} \overset{\leadsto}{\Diamond}_{ij} T_j^R \leftrightarrow \langle\langle\emptyset\rangle\rangle\Box(\Diamond_i T \to \bigwedge_{1\leq j\leq n} \overset{\leadsto}{\Diamond}_{ij} T_j^R)\right)$$
2. When $i \notin \Gamma$:
$$\langle\langle\Gamma\rangle\rangle \bigcirc \triangle_j T \to [\![\{i\}]\!] \bigcirc \triangle_j T$$
3. When $\Gamma \cap \Gamma' = \emptyset$:
$$(\langle\langle\Gamma\rangle\rangle \bigcirc \triangle_j T \wedge \langle\langle\Gamma'\rangle\rangle \bigcirc \triangle_j T') \to \langle\langle\Gamma \cup \Gamma'\rangle\rangle \bigcirc \triangle_j (T \sqcup T') \quad \Box$$

Lemma 11.1 says that an agent knows exactly the same rules in identical epistemic states. It is a statement about incomplete information, and expresses a condition similar to cond. 1 in Sec. 1.1.1.

Lemmas 11.2 and 11.3 are communication properties. The former says that an agent outside a coalition cannot prevent the coalition to ensure certain knowledge of an agent in the next state. The latter says that two disjoint coalitions can cooperate to ensure that an agent in the next state knows at least the union of the knowledge each coalition can ensure he knows individually (this is not necessarily true for overlapping coalitions).

The rule formula $\overset{\leadsto}{\triangle}_{ij} T^R$ is a statement about agent i's capability to enforce certain properties of the next state of the system — an informal description also often stated about an ATL formula such as $\langle\langle\{i\}\rangle\rangle \bigcirc \phi$. For example, $\overset{\leadsto}{\triangle}_{ij}\{\frac{p}{q}\}$ and $\triangle_i\{p\} \to \langle\langle\{i\}\rangle\rangle \bigcirc \triangle_j\{q\}$ may seem to express the same thing. However, they are not equivalent. The reason is that rule operators express something about an agent's mechanism and therefore about possible future states, while temporal operators express something about future states without regards to how these states come about. For example, $\langle\langle\emptyset\rangle\rangle \bigcirc \triangle_j\{q\}$ may be true (maybe an agent k will deterministically send q to j), which trivially implies $\langle\langle\{i\}\rangle\rangle \bigcirc \triangle_j\{q\}$, without $\overset{\leadsto}{\triangle}_{ij}\{\frac{p}{q}\}$ being true (i knows p and cannot send q to j). It turns out[7] that such local properties are difficult to express in ATL. It therefore seems that the rule operators increase the expressiveness of the language.

6. Conclusions

In this paper we have presented a logical framework for modelling how the explicit knowledge of agents who represent knowledge syntactically in a logical language can evolve as a result of reasoning and communication. No assumptions about soundness or completeness of the reasoning mechanisms were made. Semantically, each agent is modeled by an epistemic state and a mechanism specifying the possible combinations of sets of formulae the agent can send to himself and each of the other agents in a given epistemic state. The language is based on the ATL language, and extends it with epistemic operators and rule operators. The ATL operators can express statements about strategic cooperation, while the rule operators can express properties of the mechanisms. The mechanism can be seen as describing a concurrent game structure, and the language interpreted as an ATL language.

The logic is a solution to the logical omniscience problem without introducing the problem of logical ignorance. It does not follow, in general, from the fact that an agent knows something that he must know something else, but if given proper deduction rules it follows that he can get to know it if he chooses to.

Several authors have analyzed the knowledge state of an agent who knows a set of formulae [21, 27, 16, 15]. Levesque [24] introduced a logic in which *only knowing*, similar to our "knowing at most", can be expressed in the logical language. These approaches consider knowledge to be closed under consequence, and do not describe (explicit) syntactic knowledge.

Several logical frameworks for syntactically represented explicit knowledge which do not describe how knowledge *evolves* have been proposed [8, 26, 17, 11, 4], while many of the previously proposed frameworks for evolving knowledge model implicit rather than explicit knowledge [12, 7, 30]. Both Konolige's *deduction model* [22] and *active logics* [9] model agents with sets of (known) formulae and mechanisms described by rules. In these approaches, however, agents are assumed to use all their rules simultaneously in each time step, describing a linear and deterministic, instead of a branching and non-deterministic, future. There is no concept of strategic cooperation in these frameworks.

In this paper only a few of the properties of the logic are discussed. The most obvious future work would be to construct a complete axiomatization. However, it turns out

7 Condition 1. on p. 2 is inexpressible in ATEL, the epistemic extension of ATL (conjectured in [2] and later proved (manuscript submitted)). As mentioned, Lemma 11.1 *almost* captures this property.

[3, 4] that strong completeness cannot be achieved even for only the epistemic fragment[8]. A complete axiomatisation of ATL is presented in [14], and syntactic properties are also discussed in [13, 31]. Other interesting topics are: equipping the agents with sound and complete rules, using different object languages, integrate the logic with Alternating-time Temporal Epistemic Logic in order to model both implicit and explicit knowledge and maybe increase the expressiveness of ATEL, allowing rules to be nested in the object language.

8 The problem is caused by the "at most" operator $\bigtriangledown_i$. Weak completeness can be achieved [1].

References

[1] Thomas Ågotnes. *A Logic of Finite Syntactic Epistemic States*. PhD thesis, Department of Informatics, University of Bergen, 2004.

[2] Thomas Ågotnes. A note on syntactic characterization of incomplete information in ATEL. In *Proceedings of the First Workshop on Knowledge and Games (KAG 2004)*, Liverpool, U.K., July 2004.

[3] Thomas Ågotnes and Michal Walicki. A logic for reasoning about explicit knowledge in finite agents. In M. Pauly, M. Ball, and Wooldridge, editors, *Abstracts from the 2002 Workshop on Logic in Games and Multiagent Systems (LoGaMAS-02)*, Liverpool, U.K., Dec 2002.

[4] Thomas Ågotnes and Michal Walicki. A logic for reasoning about agents with finite explicit knowledge. In Bjørnar Tessem, Pekka Ala-Siuru, Patrick Doherty, and Brian Mayoh, editors, *Proceedings of the 8th Scandinavian Conference on Artificial Intelligence (SCAI'03)*, Frontiers in Artificial Intelligence and Applications, pages 163–174, Bergen, Norway, Nov 2003. IOS Press.

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

[6] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. In *38th Annual Symposium on Foundations of Computer Science*, pages 100–109, Miami Beach, Florida, 20–22 October 1997. IEEE.

[7] Ho Ngoc Duc. Reasoning about rational, but not logically omniscient, agents. *Journal of Logic and Computation*, 7(5):633–648, October 1997.

[8] R. A. Eberle. A logic of believing, knowing and inferring. *Synthese*, 26:356–382, 1974.

[9] Jennifer Elgot-Drapkin, Sarit Kraus, Michael Miller, Madhura Nirkhe, and Donald Perlis. Active logics: A unified formal approach to episodic reasoning. Technical Report CS-TR-4072, 1999.

[10] Ronald Fagin and Joseph Y. Halpern. Belief, awareness and limited reasoning. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 491–501, Los Angeles, CA, 1985.

[11] Ronald Fagin and Joseph Y. Halpern. Belief, awareness and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988. A preliminary version appeared in [10].

[12] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. The MIT Press, Cambridge, Massachusetts, 1995.

[13] V. Goranko. Coalition games and alternating temporal logics. In *Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 259–272. Morgan Kaufmann, 2001.

[14] Valentin Goranko and Govert van Drimmelen. Decidability and complete axiomatization of the alternating-time temporal logic, 2003. Submitted.

[15] Joseph Y. Halpern. A theory of knowledge and ignorance for many agents. *Journal of Logic and Computation*, 7(1):79–108, February 1997.

[16] Joseph Y. Halpern and Yoram Moses. Towards a theory of knowledge and ignorance. In Krzysztof R. Apt, editor, *Logics and Models of Concurrent Systems*, pages 459–476. Springer-Verlag, Berlin, 1985.

[17] Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, July 1990.

[18] J. Hintikka. Impossible possible worlds vindicated. *Journal of Philosophical Logic*, 4:475–484, 1975.

[19] Wojciech Jamroga. Some remarks on alternating temporal epistemic logic. In *FAMAS'03 – Formal Approaches to Multi-Agent Systems, Proceedings*, pages 133–140, Warsaw, Poland, 2003.

[20] Wojciech Jamroga and Wiebe van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 62:1–35, 2004.

[21] Kurt Konolige. Circumscriptive ignorance. In David Waltz, editor, *Proceedings of the National Conference on Artificial Intelligence*, pages 202–204, Pittsburgh, PA, August 1982. AAAI Press.

[22] Kurt Konolige. *A Deduction Model of Belief and its Logics*. PhD thesis, Stanford University, 1984.

[23] Kurt Konolige. *A Deduction Model of Belief*. Morgan Kaufmann Publishers, Los Altos, California, 1986.

[24] H. J. Levesque. All I know: a study in autoepistemic logic. *Artificial Intelligence*, 42:263–309, 1990.

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

[26] R. C. Moore and G. Hendrix. Computational models of beliefs and the semantics of belief sentences. Technical Note 187, SRI International, Menlo Park, CA, 1979.

[27] Robert C. Moore. Semantical considerations on nonmonotonic logic. In Alan Bundy, editor, *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 272–279, Karlsruhe, FRG, August 1983. William Kaufmann.

[28] Antonio Moreno. Avoiding logical omniscience and perfect reasoning: a survey. *AI Communications*, 11:101–122, 1998.

[29] Kwang Mong Sim. Epistemic logic and logical omniscience: A survey. *International Journal of Intelligent Systems*, 12:57–81, 1997.

[30] Wiebe van der Hoek and Michael Wooldridge. Tractable multiagent planning for epistemic goals. In *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAAMAS-02)*, Bologna, Italy, 2002.

[31] Wiebe van der Hoek and Michael Wooldridge. Cooperation, knowledge and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75:125–157, 2003.