An Epistemic Strategy Logic (Extended Abstract)

Xiaowei Huang
The University of New South Wales

Ron van der Meyden
The University of New South Wales

Introduction

There are many subtle issues concerning agent knowledge in settings where multiple agents act strategically. In the process of understanding these issues, there has been a proliferation of modal logics dealing with epistemic reasoning in strategic settings, e.g., [7, 8, 4]. The trend has been for these logics to contain large numbers of operators each of which combines several different concerns, such as the existence of strategies, and knowledge that groups of agents may have about these strategies. We argued in a previous work [3] that epistemic temporal logic already has the expressiveness required for many applications of epistemic strategy logics, provided that one works in a semantic framework in which strategies are explicitly rather than (as in most alternating temporal epistemic logics) implicitly represented, and makes the minor innovation of including new agents whose local states correspond to the strategies being used by other agents. This gives a more compositional basis for epistemic strategic logic. In the case of imperfect recall strategies and knowledge operators, and a CTL* temporal basis, this leads to a temporal epistemic strategy logic with a PSPACE complete model checking problem.

However, some of our applications in [3] required a restriction to cases not involving a common knowledge operator. In the present paper, we develop a remedy for this weakness. We propose an epistemic strategy logic which, like [1, 6], supports explicit naming and quantification over strategies. However we achieve this in a slightly more general way: we first generalize temporal epistemic logic to include operators for quantification over global states and reference to their components, and then apply this generalization to a system that includes strategies encoded in the global states and references these using the ”strategic” agents of [3] . The resulting framework can express many of the subtly different notions that have been the subject of proposals for alternating temporal epistemic logics. In particular, it generalizes the expressiveness of the logic in [3] but is able to also deal with the common knowledge issues that restricted the scope of that work. Moreover, the extra expressiveness comes without extra cost: model checking remains PSPACE-complete.[1]

An extended temporal epistemic logic

The following definitions are used in the standard semantics for temporal epistemic logic. Consider a system for a set of agents *Ags*. Let *Prop* be a set of atomic propositions. A *global state* is an element of the set $\mathcal{G} = L_e \times \Pi_{i \in Ags} L_i$, where L_e is a state of the environment and each L_i is a *local state* for agent i. A *run* is a mapping $r : \mathbb{N} \to \mathcal{G}$ giving a global state at each moment of time. A *point* is a pair (r,m) consisting of a run r and a time m. An *interpreted system* is a pair $\mathcal{I} = (\mathcal{R}, \pi)$, where $\mathcal{R}$ is a set of runs and π is an *interpretation*, mapping each point (r,m) with $r \in \mathcal{R}$ to a subset of *Prop*. For $n \leq m$, write $r[n \dots m]$ for the sequence $r(n)r(n+1)\dots r(m)$. For each agent $i \in Ags \cup \{e\}$, we write $r_i(m)$ for the component of

[1] Our workshop presentation would have some more detail on the applications treated in [3] than given in this abstract, where we focus on formulating the extended version of our logic. We intend to reformulate these applications in the framework of the present paper for purposes of an archival publication.

Submitted to:
SR2014

$r(m)$ in L_i, and then define an equivalence relation on points by $(r,m) \sim_i (r',m')$ if $r_i(m) = r'_i(m')$. We also define $\sim^D_G \equiv \cap_{i\in G} \sim_i$, and $\sim^E_G \equiv \cup_{i\in G} \sim_i$, and $\sim^C_G \equiv (\cup_{i\in G} \sim_i)^*$ for $G \subseteq Ags$. We take $\sim^D_\emptyset$ to be the universal relation on points.

We extend temporal epistemic logic with a set of variables $SVar$, an operator $\langle\langle x\rangle\rangle$ and a construct (i,x), where x is a variable and $\langle\langle x\rangle\rangle\phi$ says, intuitively, that there exists in the system a global state x such that ϕ holds, and (i,x) says that agent i has the same local state at the current point and at the global state x. Formally, the language ETLK($Ags, Prop, SVar$) has syntax given by the grammar:

$$\phi \equiv p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid A\phi \mid \bigcirc\phi \mid \phi_1 U \phi_2 \mid \langle\langle x\rangle\rangle\phi \mid (i,x) \mid D_G\phi \mid C_G\phi$$

where $p \in Prop$, $x \in SVar$, $i \in Ags$, and $G \subseteq Ags$. The construct $D_G\phi$ expresses that agents in G have distributed knowledge of ϕ, i.e., could deduce ϕ if they pooled their information, and $C_G\phi$ says that ϕ is common knowledge to group G. The temporal formulas $\bigcirc\phi$, $\phi_1 U \phi_2$, $A\phi$ have the standard meanings from CTL^*. Other operators can be obtained in the usual way, e.g., $\phi_1 \wedge \phi_2 = \neg(\neg\phi_1 \vee \neg\phi_2)$, $\Diamond\phi = (true U \phi)$, $\Box\phi = \neg\Diamond\neg\phi$, etc. The universal form $[[x]]\phi = \neg\langle\langle x\rangle\rangle\neg\phi$ expresses that ϕ holds for all global states x. For an agent $i \in Ags$, we write $K_i\phi$ for $D_{\{i\}}\phi$; this expresses that agent i knows the fact ϕ. The notion of everyone in group G knowing ϕ can then be expressed as $E_G\phi = \bigwedge_{i\in G} K_i\phi$.

A *context* for an interpreted system $\mathcal{I}$ is a mapping Γ from $SVar$ to global states occurring in $\mathcal{I}$. We write $\Gamma[g/x]$ for the result of changing Γ by assigning g to variable x. The semantics of the language ETLK is given by a relation $\Gamma, \mathcal{I}, (r,m) \models \phi$, representing that formula ϕ holds at point (r,m) of the interpreted system $\mathcal{I}$, relative to context Γ. This is defined inductively on the structure of the formula ϕ, as follows:

- $\Gamma, \mathcal{I}, (r,m) \models p$ if $p \in \pi(r,m)$;
- $\Gamma, \mathcal{I}, (r,m) \models \neg\phi$ if not $\Gamma, \mathcal{I}, (r,m) \models \phi$;
- $\Gamma, \mathcal{I}, (r,m) \models \phi \wedge \psi$ if $\Gamma, \mathcal{I}, (r,m) \models \phi$ and $\Gamma, \mathcal{I}, (r,m) \models \psi$;
- $\Gamma, \mathcal{I}, (r,m) \models A\phi$ if $\Gamma, \mathcal{I}, (r',m) \models \phi$ for all $r' \in \mathcal{R}$ with $r[0\ldots m] = r'[0\ldots m]$;
- $\Gamma, \mathcal{I}, (r,m) \models \bigcirc\phi$ if $\Gamma, \mathcal{I}, (r,m+1) \models \phi$;
- $\Gamma, \mathcal{I}, (r,m) \models \phi U \psi$ if there exists $m' \geq m$ such that $\Gamma, \mathcal{I}, (r,m') \models \psi$ and $\Gamma, \mathcal{I}, (r,k) \models \phi$ for all k with $m \leq k < m'$;
- $\Gamma, \mathcal{I}, (r,m) \models \langle\langle x\rangle\rangle\phi$ if $\Gamma[r'(m')/x], \mathcal{I}, (r,m) \models \phi$ for some point (r',m') of $\mathcal{I}$;
- $\Gamma, \mathcal{I}, (r,m) \models (i,x)$ if $r_i(m) = \Gamma(x)_i$;
- $\Gamma, \mathcal{I}, (r,m) \models D_G\phi$ if $\Gamma, \mathcal{I}, (r',m') \models \phi$ for all (r',m') such that $(r',m') \sim_G (r,m)$;
- $\Gamma, \mathcal{I}, (r,m) \models C_G\phi$ if $\Gamma, \mathcal{I}, (r',m') \models \phi$ for all (r',m') such that $(r',m') \sim^C_G (r,m)$.

Since interpreted systems are infinite objects, for purposes of model checking, we work with a finite state object from which an interpreted system is generated. Define an *epistemic transition system* for a set of agents Ags to be a tuple $\mathcal{E} = \langle S, I, \rightarrow, \{O_i\}_{i\in Ags}, \pi\rangle$, where S is a set of states, $I \subseteq S$ is the set of initial states, $\rightarrow \subseteq S \times S$ is a state transition relation, for each $i \in Ags$, component $O_i : S \rightarrow L_i$ is a function giving an observation in some set L_i for the agent i at each state, and $\pi : S \rightarrow \mathcal{P}(Prop)$ is a propositional assignment. A *run* of $\mathcal{E}$ is a sequence $r : \mathbb{N} \rightarrow S$ such that $r(0) \in I$ and $r(k) \rightarrow r(k+1)$ for all $k \in \mathbb{N}$. To ensure that every partial run can be completed to a run, we assume that the transition relation is *serial*, i.e., that for all states s there exists a state t such that $s \rightarrow t$.

Given an epistemic transition system $\mathcal{E}$, we obtain an interpreted system $\mathcal{I}(\mathcal{E}) = (\mathcal{R}, \pi')$ as follows. For a run $r : \mathbb{N} \rightarrow S$ of $\mathcal{E}$, define the lifted run $\hat{r} : \mathbb{N} \rightarrow S \times \Pi_{i\in Ags} L_i$ (here $L_e = S$), by $\hat{r}_e(m) = r(m)$ and $\hat{r}_i(m) = O_i(r(m))$ for $i \in Ags$. Then we take $\mathcal{R}$ to be the set of lifted runs $\hat{r}$ with r a run of $\mathcal{E}$. The assignment π' is given by $\pi'(r,m) = \pi(r(m))$.

Strategic Environments

In order to deal with agents that operate in an environment by strategically choosing their actions, we introduce a richer type of transition system that models the available actions and their effects on the state. An *environment* for agents Ags is a tuple $E = \langle S, I, Acts, \rightarrow, \{O_i\}_{i \in Ags}, \pi \rangle$, where S is a set of states, I is a subset of S, representing the initial states, $Acts = \Pi_{i \in Ags} Acts_i$ is a set of joint actions, where each $Acts_i$ is a nonempty set of actions that may be performed by agent i, component $\rightarrow \subseteq S \times Acts \times S$ is a transition relation, $O_i : S \rightarrow L_i$ is an observation function, and $\pi : S \rightarrow \mathcal{P}(Prop)$ is a propositional assignment. An environment is said to be finite if all its components, i.e., $S, Ags, Acts_i, L_i$ and $Prop$ are finite. Intuitively, a joint action $a \in Acts$ represents a choice of action $a_i \in Acts_i$ for each agent $i \in Ags$, performed simultaneously, and the transition relation resolves this into an effect on the state. We assume that $\rightarrow$ is serial in the sense that for all $s \in S$ and $a \in Acts$ there exists $t \in S$ such that $(s, a, t) \in \rightarrow$.

A *strategy* for agent $i \in Ags$ in such an environment E is a function $\alpha : S \rightarrow \mathcal{P}(Acts_i) \setminus \{\emptyset\}$, selecting a set of actions of the agent at each state. [2] We call these the actions *enabled* at the state. A strategy is *deterministic* if $\alpha(s)$ is a singleton for all s. A group strategy is *deterministic* if $\alpha_i(s)$ is a singleton for all states s and all $i \in G$. A strategy α_i for agent i is *uniform* if for all states s, t, if $O_i(s) = O_i(t)$, then $\alpha_i(s) = \alpha_i(t)$. A strategy $\alpha_G = \langle \alpha_i \rangle_{i \in G}$ for a group G is *locally uniform (deterministic)* if α_i is uniform (respectively, deterministic) for each agent $i \in G$. Given an environment E, we write $\Sigma^{det}(E)$ for the set of deterministic strategies, $\Sigma^{unif}(E)$ for the set of all locally uniform joint strategies, and $\Sigma^{unif,det}(E)$ for the set of all deterministic locally uniform joint strategies.

We now define an interpreted system that contains all the possible runs generated when agents Ags behave by choosing a strategy from some set Σ of joint strategies in the context of an environment E. This interpreted system is obtained as the system generated from an epistemic transition system that we now define. One innovation, introduced in [3], is that the construction of this epistemic transition system introduces new agents $\sigma(i)$, for each $i \in Ags$. The observation of $\sigma(i)$ is the strategy currently being used by agent i. Agent $\sigma(i)$ is not associated with any actions, and is primarily for use in epistemic operators to allow reference to what can be deduced were agents to reason using information about each other's strategies. For $G \subseteq Ags$, we write $\sigma(G)$ for the set $\{\sigma(i) \mid i \in G\}$. Additionally, we include an agent e for representing the state of the environment.

Given an environment $E = \langle S, I, Acts, \rightarrow, \{O_i\}_{i \in Ags}, \pi \rangle$ for agents Ags, and a set Σ of strategies for the group Ags, we define the *strategy space* epistemic transition system $\mathcal{E}(E, \Sigma) = \langle S^*, I^*, \rightarrow^*, \{O_i^*\}_{i \in Ags \cup \sigma(Ags) \cup \{e\}}, \pi^* \rangle$ for agents $Ags \cup \sigma(Ags) \cup \{e\}$ as follows. The state space is defined by $S^* = S \times \Sigma$, i.e., a state $(s, \alpha) \in S^*$ consists of a state s of E together with a joint strategy α for the set of all agents in E. The transition relation is given by $(s, \alpha) \rightarrow (t, \alpha')$ if $\alpha = \alpha'$ and there exists a joint action a such that $(s, a, t) \in \rightarrow$ and $a_i \in \alpha_i(s)$ for all agents $i \in Ags$. Intuitively, in a transition, each agent in Ags first selects one of the actions enabled by its strategy, and we then make a transition in E using the resulting joint action. There are no changes to the agents' strategies resulting from the transition. The initial states are given by $I^* = I \times \Sigma$, i.e., an initial state consists just of an initial state in E and a choice of joint strategy in Σ. For the observation functions, the definition of O_j^* at a state $(s, \alpha) \in S^*$ depends on the type of j. We define $O_j^*(s, \alpha) = O_j(s)$, for $j \in Ags$. For $j = \sigma(i)$, with $i \in Ags$, we define $O_j^*(s, \alpha) = \alpha_i$. For $j = e$, we define $O_j^*(s, \alpha) = s$. Finally, $\pi^*(s, \alpha) = \pi(s)$ for all states $(s, \alpha) \in S^*$.

Our epistemic strategy logic is now just an instantiation of the extended temporal epistemic logic in the strategy space generated by an environment. That is, we start with an environment E and an associated set of strategies Σ, and then work with the language $\mathrm{ETLK}(Ags \cup \sigma(Ags) \cup \{e\}, Prop, SVar)$

[2] More generally, a strategy could be a function of the history, but we focus here on strategies that depend only on the final state.

in the interpreted system $\mathcal{I}(\mathcal{E}(E,\Sigma))$. We call this instance of the language ETLK($Ags, Prop, SVar$), or just ETLK when the parameters are implicit. For brevity, given a set $G \subseteq Ags$ of agents, we write (G,x) for $\bigwedge_{i\in G}(\sigma(i),x)$. This says that at the current point, the agents in G are running the same strategies as captured by the global state named by variable x.

Connections to Other Logics

In [3], we proposed a logic CTL*K($Ags \cup \sigma(Ags), Prop$) extending temporal epistemic logic with strategy agents to allow the reasoning about knowledge and strategy by standard epistemic operators. The language introduced above is a generalization of the definitions in [3], to which we have added the constructs $\langle\langle x\rangle\rangle\phi$ and (i,x). For formulas without these constructs, the semantics ignores the context Γ, so this component of the triple $\Gamma, \mathcal{I}(\mathcal{E}(E,\Sigma)),(r,m)$ can be removed from the definition, and it collapses to the definitions for CTL*K($Ags \cup \sigma(Ags), Prop$) in [3].

In the system $\mathcal{I}(\mathcal{E}(E,\Sigma))$ we may refer, using distributed knowledge operators D_G where G contains the new strategic agents $\sigma(i)$, to what agents would know, should they take into account not just their own observations, but also information about other agent's strategies. For example, the distributed knowledge $D_{\{i,\sigma(i),\sigma(j)\}}$ captures what agent i would know, taking into account its own strategy and the strategy being used by agent j. Various applications of the usefulness of these distributed knowledge operators containing strategic agents are given in [3]. For example, we describe an application in computer security in which we write formulas such as

$$\neg D_{\emptyset}\neg(\mathtt{done} \wedge \neg\mathtt{exploited} \wedge EF \bigvee_{x\in\mathtt{CCN}} D_{\{A,\sigma(A),\sigma(M)\}}(\mathtt{cc} \neq x)))$$

to state that it is possible for an attacker A on an e-commerce payment gateway to obtain information about a credit card number $\mathtt{cc}$ even after the transaction is done, provided that the attacker reasons using knowledge about their own observations, their own strategy, but also knowledge of the strategy being used by the merchant M. In further applications given in [3], we showed that CTL*K($Ags \cup \sigma(Ags), Prop$) can be used to reason about knowledge-based programs [2], and that many variants of alternating temporal epistemic logics that have been proposed in the literature can be expressed using CTL*K($Ags \cup \sigma(Ags), Prop$). We refer the reader to [3] for details.

However, we had to make a restriction for some of these expressiveness results to formulas that do not contain uses of a common knowledge operator. We now show how the extended language of the present paper can remove this restriction.

Jamroga and van der Hoek [5] formulate a construct $\langle\langle H\rangle\rangle^{\bullet}_{\mathcal{K}(G)}\phi$ that says, effectively, that there is a strategy for a group H that another group G knows (for notion of group knowledge $\mathcal{K}$, which could be E for everyone knows, D for distributed knowledge, or C for common knowledge) to achieve goal ϕ. The semantics of this construct is given with respect to an environment E and a state s, and (in outline) is given by $E,s \models \langle\langle H\rangle\rangle^{\bullet}_{\mathcal{K}(G)}\phi$ if there exists a uniform strategy α for group H such that for all states t with $s \sim^{\mathcal{K}}_{G} t$, we have that all paths ρ from t that are consistent with α satisfy ϕ. Here $\sim^{\mathcal{K}}_{G}$ is the appropriate epistemic indistinguishability relation on states of E.

In the case that $\mathcal{K}(G)$ is the common knowledge operator C_G, this definition involves the common knowledge that a group G of agents would have if they were to reason taking into consideration the strategy being used by another group H. This does not appear to be expressible using CTL*K($Ags \cup \sigma(Ags), Prop$). In particular, the formula $C_{G\cup\sigma(H)}\phi$ does not give the intended meaning. Instead, what needs to be expressed is the greatest fixpoint X of the equation $X \equiv \bigwedge_{i\in G} D_{\{i\}\cup\sigma(H)}(X \wedge \phi)$. The language

CTL*K($Ags \cup \sigma(Ags), Prop$) does not include fixpoint operators and it does not seem that the intended meaning is expressible. On the other hand, it can be expressed with ETLK($Ags, SVar, Prop$) in a natural way by a formula

$$C_G((H,x) \Rightarrow \phi)$$

which says that it is common knowledge to the group G that ϕ holds if the group H is running the strategy profile capture by the variable x. Using this idea, the construct $\langle\langle H\rangle\rangle^{\bullet}_{C(G)}\phi$ can be represented with ETLK as

$$\langle\langle x\rangle\rangle C_G((H,x) \Rightarrow \phi)\ .$$

(We remark that a carefully stated equivalence result requires an appropriate treatment of initial states in the environment E. We refer to [3] for details.) Applying similar ideas, ETLK can also be used to eliminate, from the results on reasoning about knowledge-based programs presented in [3], the restriction to knowledge-based programs not containing common knowledge operators.

Model Checking

Since interpreted systems are always infinite objects, we use environments to give a finite input for the model checking problem. For an environment E, a set of strategies Σ for E, and a context Γ for $\mathcal{I}(\mathcal{E}(E,\Sigma))$, we write $\Gamma, E, \Sigma \models \phi$ if $\Gamma, \mathcal{I}(\mathcal{E}(E,\Sigma)), (r,0) \models \phi$ for all runs r of $\mathcal{I}$ of $\mathcal{I}(\mathcal{E}(E,\Sigma)$. (Often, the formula ϕ will be a sentence, i.e., will have all variables x in the scope of an operator $\langle\langle x\rangle\rangle$. In this case Γ could be omitted.) The model checking problem is to determine whether $\Gamma, E, \Sigma \models \phi$ for a finite state environment E, a set Σ of strategies and a context Γ, where ϕ is an ETLK($Ags, SVar, Prop$) formula.

For generality, we abstract Σ to a paramaterized class such that for each environment E, the set $\Sigma(E)$ is a set strategies for E. We say that the parameterized class $\Sigma(E)$ is *PTIME-presented*, if it is presented by means of an algorithm that runs in time polynomial in the size of E and verifies if a given strategy α is in $\Sigma(E)$. For example, the class $\Sigma(E)$ of all strategies for E can be PTIME-presented, as can $\Sigma^{unif}(E)$, $\Sigma^{det}(E)$ and $\Sigma^{unif,det}(E)$.

A naive model checking algorithm would construct the epistemic transition system $\mathcal{E}(E,\Sigma(E))$ and then apply model checking techniques on it. Note that a joint strategy for an environment E can be represented in space $|S| \times |Acts|$. Thus, $\mathcal{E}(E,\Sigma(E))$ generally has exponentially many states, as a function of the size of E. This means that the naive procedure would require not less than exponential time. In fact, it is possible to do better than this (assuming that PSPACE is better than EXPTIME).

Theorem 1 *Let $\Sigma(E)$ be a PTIME presented class of strategies for environments E. The complexity of deciding, given an environment E and an* ETLK *formula ϕ, whether $E, \Sigma(E) \models \phi$, is PSPACE-complete.*

That is, the additional constructs $\langle\langle x\rangle\rangle\phi$ and (i,x) that we have added to the logic of [3] to obtain ETLK do not increase the complexity of model checking.

Conclusions

Strategy Logic [1] is a (non-epistemic) generalization of ATL for perfect information strategies in which strategies may be explicitly named and quantified. Work on identification of more efficient variants of quantified strategy logic includes [6], who formulate a variant with a 2-EXPTIME-complete model checking problem. In both cases, strategies are perfect recall strategies, rather than the imperfect recall strategies that form the basis for our PSPACE-completeness result for model checking. The exploration of our logic over such a richer space of strategies is an interesting topic for future research.

References

[1] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. *Inf. Comput.*, 208(6):677–693, 2010.

[2] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge*. The MIT Press, 1995.

[3] X. Huang and R. van der Meyden. A temporal logic of strategic knowledge. Submitted to KR'14, available for download at `http://www.cse.unsw.edu.au/~meyden/research/atl_obs.pdf`.

[4] W. Jamroga and T. Ågotnes. Constructive knowledge: what agents can achieve under imperfect information. *Journal of Applied Non-Classical Logics*, 17(4):423–475, 2007.

[5] W. Jamroga and W. van der Hoek. Agents that Know How to Play . *Fundamenta Informaticae*, 62:1–35, 2004.

[6] F. Mogavero, A. Murano, and M. Y. Vardi. Reasoning about strategies. In *FSTTCS*, pages 133–144, 2010.

[7] P.-Y. Schobbens. Alternating-time logic with imperfect recall. *ENTCS*, 85(2):82–93, 2004.

[8] S. van Otterloo and G. Jonker. On Epistemic Temporal Strategic Logic. *ENTCS*, 126:77–92, 2005.