

Reasoning about Knowledge and Strategies: Epistemic Strategy Logic

Francesco Belardinelli

Laboratoire IBISC – Université d’Evry

belardinelli@ibisc.fr

In this paper we introduce Epistemic Strategy Logic (ESL), an extension of Strategy Logic with modal operators for individual knowledge. This enhanced framework allows us to represent explicitly and to reason about the knowledge agents have of their own and other agents’ strategies. We provide a semantics to ESL in terms of epistemic concurrent game models, and consider the corresponding model checking problem. We show that the complexity of model checking ESL is not worse than (non-epistemic) Strategy Logic.

1 Introduction

Formal languages to represent and reason about strategies and coalitions are a thriving area of research in Artificial Intelligence and multi-agent system [4, 8, 19]. Recently, a wealth of multi-modal logics have appeared, which allow to formalise complex strategic abilities and behaviours of individual agents and groups [2, 5]. In parallel to these developments, in knowledge representation there is a well-established tradition of extending logics for reactive systems with epistemic operators to reason about the knowledge agents have of systems evolution. These investigations began in the ’80s with contributions on combinations of linear- and branching-time temporal logics with multi-agent epistemic languages [9, 10, 6]. Along this line of research, [11] introduced alternating-time temporal epistemic logic (ATEL), an extension of ATL with modalities for individual knowledge. The various flavours of logics of time and knowledge have been successfully applied to the specification of distributed and multi-agent systems in domains as diverse as security protocols, UAVs, web services, and e-commerce, as well as to verification by model checking [7, 16].

In this paper we take inspiration from the works above and pursue further this line of research by introducing Epistemic Strategy Logic, an extension of Strategy Logic (SL) [5, 17] that allows agents to reason about their strategic abilities. The extension here proposed is naive in the sense that it suffers many of the shortcomings of its relative ATEL [12]. Nonetheless, we reckon that it constitutes an excellent starting point to analyse the interaction of knowledge and strategic abilities in a language, such as SL, that explicitly allow for quantification on strategies.

Related Work. This paper builds on previous contributions on Strategy Logic. SL has been introduced in [5] for two-player concurrent game structures (CGS). In [17] the semantics has been extended to a multi-player setting. Also, [17] introduced bind operators for strategies in the syntax. In the present contribution we consider multi-agent CGS in line with [17]. However, we adopt an agent-based perspective and consider agents with possibly different actions and protocols [6]. Also, our language do not include bind operators to avoid the formal machinery associated with these operators. We leave such an extension for future and more comprehensive work. Finally, the model checking results in Section 4 are inspired by and use techniques from [17].

Even though to our knowledge no epistemic extension of SL has been proposed yet, the interaction between knowledge and strategic reasoning has been studied extensively, especially in the context of alternating-time temporal logic. An extension of ATL with knowledge operators, called ATEL, was put forward in [11], and immediately imperfect information variants of this logic were considered in [14], which introduces alternating-time temporal observational logic (ATOL) and ATEL-R*, as well as uniform strategies. Notice that [14] also analyses the distinction between *de re* and *de dicto* knowledge of strategies; this distinction will also be considered later on in the context of Epistemic Strategy Logic. Further, [13] enriches ATL with a constructive notion of knowledge. As regards (non-epistemic) ATL, more elaborate notions of strategy have been considered. In [1] commitment in strategies has been analysed; while [15] introduced a notion of “feasible” strategy. In future work it might be worth exploring to what extent the theoretical results available for the various flavours of ATEL transfer to ESL.

Scheme of the paper. In Section 2 we introduce the epistemic concurrent game models (ECGM), which are used in Section 3 to provide a semantics to Epistemic Strategy Logic (ESL). In Section 4 we consider the model checking problem for this setting and state the corresponding complexity results. Finally, in Section 5 we discuss the results and point to future research. For reasons of space, all proofs are omitted. An extended version of this paper with complete proofs is available [3].

2 Epistemic Concurrent Game Models

In this section we present the epistemic concurrent game models (ECGM), an extension of concurrent game structures [2, 11], starting with the notion of *agent*.

Definition 1 (Agent) *An agent is a tuple $i = \langle L_i, Act_i, Pr_i \rangle$ such that (i) L_i is the set of local states l_i, l'_i, \dots ; (ii) Act_i is the finite set of actions $\sigma_i, \sigma'_i, \dots$; and (iii) $Pr_i : L_i \mapsto 2^{Act_i}$ is the protocol function.*

Intuitively, each agent i is situated in some local state $l_i \in L_i$, representing her local information, and performs the actions in Act_i according to the protocol function Pr_i [6]. Differently from [17], we assume that agents have possibly different actions and protocols. To formally describe the interactions between agents, we introduce their synchronous composition. Given a set AP of atomic propositions and a set $Ag = \{i_0, \dots, i_n\}$ of agents, we define the set L of global states s, s', \dots (resp. the set Act of joint actions σ, σ', \dots) as the cartesian product $L_0 \times \dots \times L_n$ (resp. $Act_0 \times \dots \times Act_n$). In what follows we denote the j th component of a tuple t as t_j or, equivalently, as $t(j)$.

Definition 2 (ECGM) *Given a set $Ag = \{i_0, \dots, i_n\}$ of agents $i = \langle L_i, Act_i, Pr_i \rangle$, an epistemic concurrent game model is a tuple $\mathcal{P} = \langle Ag, s_0, \tau, \pi \rangle$ such that (i) $s_0 \in L$ is the initial global state; (ii) $\tau : L \times Act \mapsto L$ is the global transition function, where $\tau(s, \sigma)$ is defined iff $\sigma_i \in Pr_i(l_i)$ for every $i \in Ag$; and (iii) $\pi : AP \mapsto 2^L$ is the interpretation function for atomic propositions in AP .*

The transition function τ describes the evolution of the ECGM from the initial state s_0 . We now introduce some notation that will be used in the rest of the paper. The *transition relation* \rightarrow on global states is defined as $s \rightarrow s'$ iff there exists $\sigma \in Act$ s.t. $\tau(s, \sigma) = s'$. A *run* λ from a state s , or *s-run*, is an infinite sequence $s^0 \rightarrow s^1 \rightarrow \dots$, where $s^0 = s$. For $n, m \in \mathbb{N}$, with $n \leq m$, we define $\lambda(n) = s^n$ and $\lambda[n, m] = s^n, s^{n+1}, \dots, s^m$. A state s' is *reachable from* s if there exists an *s-run* λ s.t. $\lambda(i) = s'$ for some $i \geq 0$. We define S as the set of states reachable from the initial state s_0 . Further, let \sharp be a placeholder for arbitrary individual actions. Given a subset $A \subseteq Ag$ of agents, an *A-action* σ_A is an $|Ag|$ -tuple s.t. (i) $\sigma_A(i) \in Act_i$ for $i \in A$, and (ii) $\sigma_A(j) = \sharp$ for $j \notin A$. Then, Act_A is the set of all *A-actions* and $D_A(s) = \{\sigma_A \in Act_A \mid \text{for every } i \in A, \sigma_i \in Pr_i(l_i)\}$ is the set of all *A-actions* enabled at $s = \langle l_0, \dots, l_n \rangle$. A joint action σ *extends* an *A-action* σ_A , or $\sigma_A \sqsubseteq \sigma$, iff $\sigma_A(i) = \sigma(i)$ for all $i \in A$. The *outcome* $out(s, \sigma_A)$

of action σ_A at state s is the set of all states s' s.t. there exists a joint action $\sigma \sqsupseteq \sigma_A$ and $\tau(s, \sigma) = s'$. Finally, two global states $s = \langle l_0, \dots, l_n \rangle$ and $s' = \langle l'_0, \dots, l'_n \rangle$ are *indistinguishable* for agent i , or $s \sim_i s'$, iff $l_i = l'_i$ [6].

3 Epistemic Strategy Logic

We now introduce Epistemic Strategy Logic as a specification language for ECGM. Hereafter we consider a set Var_i of strategy variables x_i, x'_i, \dots , for every agent $i \in Ag$.

Definition 3 (ESL) For $p \in AP$, $i \in Ag$ and $x_i \in Var_i$, the ESL formulas ϕ are defined in BNF as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \rightarrow \phi \mid X\phi \mid \phi U \phi \mid K_i\phi \mid \exists x_i\phi$$

The language ESL is an extension of the Strategy Logic in [5] to a multi-agent setting, including an epistemic operator K_i for each $i \in Ag$. Alternatively, ESL can be seen as the epistemic extension of the Strategy Logic in [17], minus the bind operator. We do not consider bind operators in ESL for ease of presentation. The ESL formula $\exists x_i\phi$ is read as “agent i has some strategy to achieve ϕ ”. The interpretation of LTL operators X and U is standard. The epistemic formula $K_i\phi$ intuitively means that “agent i knows ϕ ”. The other propositional connectives and LTL operators, as well as the strategy operator \forall , can be defined as standard. Also, notice that we can introduce the *nested-goal* fragment ESL[NG], the *boolean-goal* fragment ESL[BG], and the *one-goal* fragment ESL[1G] in analogy to SL [17]. Further, the *free* variables $fr(\phi) \subseteq Ag$ of an ESL formula ϕ are inductively defined as follows:

$$\begin{aligned} fr(p) &= \emptyset \\ fr(\neg\phi) = fr(K_i\phi) &= fr(\phi) \\ fr(\phi \rightarrow \phi') &= fr(\phi) \cup fr(\phi') \\ fr(X\phi) = fr(\phi U \phi') &= Ag \\ fr(\exists x_i\phi) &= fr(\phi) \setminus \{i\} \end{aligned}$$

A *sentence* is a formula ϕ with $fr(\phi) = \emptyset$, and the set $bnd(\phi)$ of bound variables is defined as $Ag \setminus fr(\phi)$.

To provide a semantics to ESL formulas in terms of ECGM, we introduce the notion of strategy.

Definition 4 (Strategy) Let γ be an ordinal s.t. $1 \leq \gamma \leq \omega$ and $A \subseteq Ag$ a set of agents. A γ -recall A -strategy is a function $F_A[\gamma] : \bigcup_{1 \leq n < 1+\gamma} S^n \mapsto \bigcup_{s \in S} D_A(s)$ s.t. $F_A[\gamma](\kappa) \in D_A(last(\kappa))$ for every $\kappa \in \bigcup_{1 \leq n < 1+\gamma} S^n$, where $1 + \gamma = \gamma$ for $\gamma = \omega$ and $last(\kappa)$ is the last element of κ .

Hence, a γ -recall A -strategy returns an enabled A -action for every sequence $\kappa \in \bigcup_{1 \leq n < 1+\gamma} S^n$ of states of length at most γ . Notice that for $A = \{i\}$, $F_A[\gamma]$ can be seen as a function from $\bigcup_{1 \leq n < 1+\gamma} S^n$ to Act_i s.t. $F_A[\gamma](\kappa) \in Pr_i(last(\kappa))$ for $\kappa \in \bigcup_{1 \leq n < 1+\gamma} S^n$. In what follows we write $F_i[\gamma]$ for $F_{\{i\}}[\gamma]$. Then, for $A = \{i_0, \dots, i_m\} \subseteq Ag$, $F_A[\gamma]$ is equal to $F_{i_0}[\gamma] \times \dots \times F_{i_m}[\gamma]$, where for every $\kappa \in \bigcup_{1 \leq n < 1+\gamma} S^n$, $(F_{i_0}[\gamma] \times \dots \times F_{i_m}[\gamma])(\kappa)$ is defined as the set of actions $\sigma \in \bigcup_{s \in S} D_A(s)$ s.t. $\sigma_i = F_i[\gamma](\kappa)$ if $i \in A$, $\sigma_i = \#$ otherwise. Therefore, a group strategy is the composition of its members' strategies. Further, the *outcome* of strategy $F_A[\gamma]$ at state s , or $out(s, F_A[\gamma])$, is the set of all s -runs λ s.t. $\lambda(i+1) \in out(\lambda(i), F[\gamma](\lambda[j, i]))$ for all $i \geq 0$ and $j = \max(i - \gamma + 1, 0)$. Depending on γ we can define positional strategies, strategies with perfect recall, etc. [8]. However, these different choices do not affect the following results, so we assume that γ is fixed and omit it. Moreover, by Def. 4 it is apparent that agents have perfect information, as their strategies are determined by global states [4]; we leave contexts of imperfect information for future research.

Now let χ be an assignment that maps each agent $i \in Ag$ to an i -strategy F_i . For $Ag = \{i_0, \dots, i_n\}$, we denote $\chi(i_0) \times \dots \times \chi(i_n)$ as F^χ , that is, the Ag -strategy s.t. for every $\kappa \in \bigcup_{1 \leq n < 1+\gamma} S^n$, $F^\chi(\kappa) = \sigma \in$

$\bigcup_{s \in S} D_{Ag}(s)$ iff $\sigma_i = \chi(i)(\kappa)$ for every $i \in Ag$. Since $|out(s, F^\chi)| = 1$, we simply write $\lambda = out(s, F^\chi)$. Also, $\chi_{F_i}^i$ denotes the assignment s.t. (i) for all agents j different from i , $\chi_{F_i}^i(j) = \chi(j)$, and (ii) $\chi_{F_i}^i(i) = F_i$.

Definition 5 (Semantics of ESL) We define whether an ECGM \mathcal{P} satisfies a formula ϕ at state s according to assignment χ , or $(\mathcal{P}, s, \chi) \models \phi$, as follows (clauses for propositional connectives are straightforward and thus omitted):

$$\begin{aligned} (\mathcal{P}, s, \chi) &\models p && \text{iff } s \in \pi(p) \\ (\mathcal{P}, s, \chi) &\models X\psi && \text{iff for } \lambda = out(s, F^\chi), (\mathcal{P}, \lambda(1), \chi) \models \psi \\ (\mathcal{P}, s, \chi) &\models \psi U \psi' && \text{iff for } \lambda = out(s, F^\chi) \text{ there is } k \geq 0 \text{ s.t. } (\mathcal{P}, \lambda(k), \chi) \models \psi' \text{ and } 0 \leq j < k \text{ implies } (\mathcal{P}, \lambda(j), \chi) \models \psi \\ (\mathcal{P}, s, \chi) &\models K_i \psi && \text{iff for all } s \in S, s \sim_i s' \text{ implies } (\mathcal{P}, s', \chi) \models \psi \\ (\mathcal{P}, s, \chi) &\models \exists x_i \psi && \text{iff there exists an } i\text{-strategy } F_i \text{ s.t. } (\mathcal{P}, s, \chi_{F_i}^i) \models \psi \end{aligned}$$

An ESL formula ϕ is *satisfied* at state s , or $(\mathcal{P}, s) \models \phi$, if $(\mathcal{P}, s, \chi) \models \phi$ for all assignments χ ; ϕ is *true* in \mathcal{P} , or $\mathcal{P} \models \phi$, if $(\mathcal{P}, s_0) \models \phi$. The satisfaction of formulas is independent from bound variables, that is, $\chi(fr(\phi)) = \chi'(fr(\phi))$ implies that $(\mathcal{P}, s, \chi) \models \phi$ iff $(\mathcal{P}, s, \chi') \models \phi$. In particular, the satisfaction of sentences is independent from assignments.

We can now state the model checking problem for ESL.

Definition 6 (Model Checking Problem) Given an ECGM \mathcal{P} and an ESL formula ϕ , determine whether there exists an assignment χ s.t. $(\mathcal{P}, s_0, \chi) \models \phi$.

Notice that, if y_1, \dots, y_m is an enumeration of $fr(\phi)$, then the model checking problem amounts to check whether $\mathcal{P} \models \exists y_1, \dots, \exists y_m \phi$, where $\exists y_1, \dots, \exists y_m \phi$ is a sentence.

Hereafter we illustrate the formal machinery introduced thus far with a toy example.

Example. We introduce a turn-based ECGM with two agents, A and B . First, A secretly chooses between 0 and 1. Then, at the successive stage, B also chooses between 0 and 1. The game is won by agent A if the values provided by the two agents coincide, otherwise B wins. We formally describe this toy game starting with agents A and B . Specifically, A is the tuple $\langle L_A, Act_A, Pr_A \rangle$, where (i) $L_A = \{\epsilon_A, 0, 1\}$; (ii) $Act_A = \{set(0), set(1), skip\}$; and (iii) $Pr_A(\epsilon_A) = \{set(0), set(1)\}$ and $Pr_A(0) = Pr_A(1) = \{skip\}$. Further, agent B is defined as the tuple $\langle L_B, Act_B, Pr_B \rangle$, where $L_B = \{\epsilon_B, \lambda, 0, 1\}$; $Act_B = \{wait, set(0), set(1), skip\}$; $Pr_B(\epsilon_B) = \{wait\}$, $Pr_B(\lambda) = \{set(0), set(1)\}$ and $Pr_B(0) = Pr_B(1) = \{skip\}$. The intuitive meaning of local states, actions and protocol functions is clear. Also, we consider the set $AP = \{win_A, win_B\}$ of atomic propositions, which intuitively express that agent A (resp. B) has won the game. We now introduce the ECGM \mathcal{Q} , corresponding to our toy game, as the tuple $\langle Ag, s_0, \tau, \pi \rangle$, where (i) $s_0 = (\epsilon_A, \epsilon_B)$; (ii) the transition function τ is given as follows for $i, j \in \{0, 1\}$:

- $\tau((\epsilon_A, \epsilon_B), (set(i), wait)) = (i, \lambda)$
- $\tau((i, \lambda), (skip, set(j))) = (i, j)$
- $\tau((i, j), (skip, skip)) = (\epsilon_A, \epsilon_B)$

and (iii) $\pi(win_A) = \{(0, 0), (1, 1)\}$, $\pi(win_B) = \{(1, 0), (0, 1)\}$. Notice that we suppose that our toy game, represented in Fig. 1, is non-terminating.

Now, we check whether the following ESL specifications hold in the ECGM \mathcal{Q} .

$$\mathcal{Q} \models \forall x_A X K_B \exists y_B X win_B \quad (1)$$

$$\mathcal{Q} \not\models \forall x_A X \exists y_B K_B X win_B \quad (2)$$

$$\mathcal{Q} \models \forall x_A X K_B K_A \exists y_B X win_A \quad (3)$$

$$\mathcal{Q} \models \forall x_A X K_B \exists y_B K_A X win_A \quad (4)$$

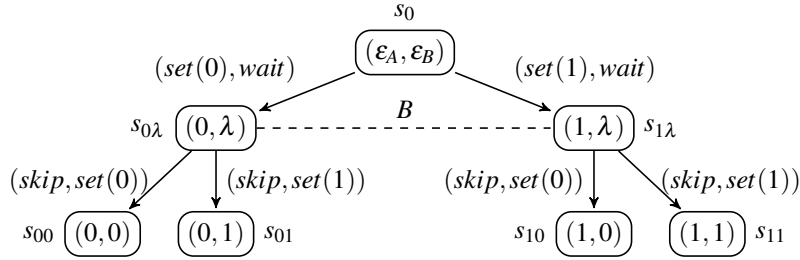


Figure 1: the ECGM \mathcal{Q} . Transitions from s_{00} , s_{01} , s_{10} , and s_{11} to s_0 are omitted.

Intuitively, (1) expresses the fact that at the beginning of the game, independently from agent A 's move, at the next step agent B knows that there exists a move by which she can enforce her victory. That is, if agent A chose 0 (resp. 1), then B can choose 1 (resp. 0). However, B only knows that there exists a move, but she is not able to point it out. In fact, (2) does not hold, as B does not know which specific move A chose, so she is not capable of distinguishing states $s_{0\lambda}$ and $s_{1\lambda}$. Moreover, by (3) B knows that A knows that there exists a move by which B can let A win. Also, by (4) this move is known to A , as it is the B -move matching A 's move.

Indeed, in ESL it is possible to express the difference between *de re* and *de dicto* knowledge of strategies. One of the first contributions to tackle this issue formally is [14]. Formula (1) expresses agent B 's *de dicto* knowledge of strategy y_B ; while (2) asserts *de re* knowledge of the same strategy. Similarly, in (3) agent A has *de re* knowledge of strategy y_B ; while (4) states that agent A knows the same strategy *de dicto*. The *de re/de dicto* distinction is of utmost importance as, as shown above, having a *de dicto* knowledge of a strategy does not guarantee that an agent is actually capable of performing the associated sequence of actions. Ideally, in order to have an effective strategy, agents must know it *de re*.

4 Model Checking ESL

In this section we consider the complexity of the model checking problem for ESL. In Section 4.1 and 4.2 we provide the lower and upper bound respectively. For reasons of space, we do not provide full proofs, but only give the most important partial results. We refer to [3] for detailed definitions and complete proofs.

For an ESL formula ϕ we define $alt(\phi)$ as the maximum number of alternations of quantifiers \exists and \forall in ϕ . Then, $ESL[k\text{-alt}]$ is the set of ESL formulas ϕ with $alt(\phi)$ equal to or less than k .

4.1 Lower Bound

In this section we prove that model checking ESL formulas is non-elementary-hard. Specifically, we show that for ESL formulas with maximum alternation k the model checking problem is $k\text{-EXPSPACE-hard}$. The proof strategy is similar to [17], namely, we reduce the satisfiability problem for quantified propositional temporal logic (QPTL) to ESL model checking. However, the reduction applied is different, as ESL does not contain the bind operator used in [17].

We first state that the satisfiability problem for QPTL sentences built on a finite set $AP = \{p_0, \dots, p_n\}$ of atomic propositions can be reduced to model checking ESL sentences on a ECGM \mathcal{Q} of fixed size on $|AP|$, albeit exponential.

Lemma 1 (QPTL Reduction) *Let $AP = \{p_0, \dots, p_n\}$ be a finite set of atomic propositions. There exists an ECGM \mathcal{Q} on AP s.t. for every QPTL[k -alt] sentence ϕ on AP , there exists an ESL[k -alt] sentence $\bar{\phi}$ s.t. ϕ is satisfiable iff $\mathcal{Q} \models \bar{\phi}$.*

By this result and the fact that the satisfiability problem for QPTL[k -alt] is k -EXPSPACE-hard [17], we can derive the lower bound for model checking ESL[k -alt].

Theorem 2 (Hardness) *The model checking problem for ESL[k -alt] is k -EXPSPACE-hard.*

In particular, it follows that ESL model checking is non-elementary-hard.

4.2 Upper Bound

In this section we extend to Epistemic Strategy Logic the model checking procedure for SL in [17], which is based on alternating tree automata (ATA) [18]. We state the following result, which extends Lemma 5.6 in [17].

Lemma 3 *Let \mathcal{P} be an ECGM and ϕ an ESL formula. Then, there exists an alternating tree automaton $\mathcal{A}_{\mathcal{P}}^{\phi}$ s.t. for every state $s \in S$ and assignment χ , we have that $(\mathcal{P}, s, \chi) \models \phi$ iff the assignment-state encoding \mathcal{T}_s^{χ} belongs to the language $\mathcal{L}(\mathcal{A}_{\mathcal{P}}^{\phi})$.*

The following result corresponds to Theorem 5.4 in [17].

Theorem 4 (ATA Direction Projection) *Let $\mathcal{A}_{\mathcal{P}}^{\phi}$ be the ATA in Lemma 3, and $s \in S$ a distinguished state. Then, there exists a non-deterministic ATA $\mathcal{N}_{\mathcal{P},s}^{\phi}$ s.t. for all $\text{Act}_{\text{fr}(\phi)}$ -labelled Δ -tree $\mathcal{T} = \langle T, V \rangle$, we have that $\mathcal{T} \in \mathcal{L}(\mathcal{N}_{\mathcal{P},s}^{\phi})$ iff $\mathcal{T}' \in \mathcal{L}(\mathcal{A}_{\mathcal{P}}^{\phi})$, where \mathcal{T}' is the $(\text{Act}_{\text{fr}(\phi)} \times S)$ -labelled Δ -tree $\langle T, V' \rangle$ s.t. $V'(x) = (V(x), \text{last}(\kappa_{s,x}))$.*

Then, by using Lemma 3 and Theorem 4 we can state the following result.

Theorem 5 *Let \mathcal{P} be an ECGM, s a state in \mathcal{P} , χ an assignment, and ϕ an ESL formula. The non-deterministic ATA $\mathcal{N}_{\mathcal{P},s}^{\phi}$ in Theorem 4 is such that $(\mathcal{P}, s, \chi) \models \phi$ iff $\mathcal{L}(\mathcal{N}_{\mathcal{P},s}^{\phi}) \neq \emptyset$.*

We can finally state the following extension to Theorem 5.8 in [17], which follows from the fact that the non-emptiness problem for alternating tree automata is non-elementary in the size of the formula.

Theorem 6 (Completeness) *The model checking problem for ESL is PTIME-complete w.r.t. the size of the model and NON-ELEMENTARYTIME w.r.t. the size of the formula.*

We remark that Theorem 6 can be used to show that the model checking problem for the nested-goal fragment ESL[NG] is PTIME-complete w.r.t. the size of the model and $(k+1)$ -EXPTIME w.r.t. the maximum alternation k of a formula. We conclude that the complexity of model checking ESL is not worse than the corresponding problem for the Strategy Logic in [17].

5 Conclusions

In this paper we introduced Epistemic Strategy Logic, an extension of Strategy Logic [17] with modalities for individual knowledge. We provided this specification language with a semantics in terms of epistemic concurrent game models (ECGM), and analysed the corresponding model checking problem. A number of developments for the proposed framework are possible. Firstly, the model checking problem for the nested-goal, boolean-goal, and one-goal fragment of SL has lower complexity. It is likely that similar results hold also for the corresponding fragments of ESL. Secondly, we can extend ESL with modalities for group knowledge, such as common and distributed knowledge. Thirdly, we can consider various assumptions on ECGM, for instance perfect recall, no learning, and synchronicity. The latter two extensions, while enhancing the expressive power of the logic, are also likely to increase the complexity of the model checking and satisfiability problems.

References

- [1] Thomas Agotnes, Valentin Goranko & Wojciech Jamroga (2007): *Alternating-time Temporal Logics with Irrevocable Strategies*. In: *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '07, ACM, New York, NY, USA, pp. 15–24, doi:10.1145/1324249.1324256.
- [2] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. *J. ACM* 49(5), pp. 672–713, doi:10.1145/585265.585270.
- [3] Francesco Belardinelli (2014): *Reasoning about Knowledge and Strategies: Epistemic Strategy Logic*. Technical Report, Universit d'Evry, Laboratoire IBISC. Available at <https://www.ibisc.univ-evry.fr/~belardinelli/Documents/sr2014.pdf>.
- [4] Nils Bulling, Jorgen Dix & Wojciech Jamroga (2010): *Model Checking Logics of Strategic Ability: Complexity**. In Mehdi Dastani, Koen V. Hindriks & John-Jules Charles Meyer, editors: *Specification and Verification of Multi-agent Systems*, Springer US, pp. 125–159, doi:10.1007/978-1-4419-6984-2.
- [5] Krishnendu Chatterjee, Thomas A. Henzinger & Nir Piterman (2010): *Strategy logic*. *Inf. Comput.* 208(6), pp. 677–693, doi:10.1016/j.ic.2009.07.004.
- [6] R. Fagin, J.Y. Halpern, Y. Moses & M.Y. Vardi (1995): *Reasoning About Knowledge*. The MIT Press.
- [7] Peter Gammie & Ron van der Meyden (2004): *MCK: Model Checking the Logic of Knowledge*. In Rajeev Alur & Doron Peled, editors: *CAV, Lecture Notes in Computer Science* 3114, Springer, pp. 479–483, doi:10.1007/978-3-540-27813-9_41.
- [8] Valentin Goranko & Wojciech Jamroga (2004): *Comparing Semantics of Logics for Multi-Agent Systems*. *Synthese* 139(2), pp. 241–280, doi:10.1023/B:SYNT.0000024915.66183.d1.
- [9] Joseph Y. Halpern & Moshe Y. Vardi (1986): *The Complexity of Reasoning about Knowledge and Time: Extended Abstract*. In Juris Hartmanis, editor: *STOC*, ACM, pp. 304–315, doi:10.1145/12130.12161.
- [10] Joseph Y. Halpern & Moshe Y. Vardi (1989): *The Complexity of Reasoning about Knowledge and Time. I. Lower Bounds*. *J. Comput. Syst. Sci.* 38(1), pp. 195–237, doi:10.1016/0022-0000(89)90039-1.
- [11] W. van der Hoek & M. Wooldridge (2003): *Cooperation, Knowledge, and Time: Alternating-time Temporal Epistemic Logic and its Applications*. *Studia Logica* 75(1), pp. 125–157, doi:10.1023/A:1026185103185.
- [12] Wojciech Jamroga (2004): *Some Remarks on Alternating Temporal Epistemic Logic*. In: *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pp. 133–140.
- [13] Wojciech Jamroga & Thomas Ågotnes (2007): *Constructive knowledge: what agents can achieve under imperfect information*. *Journal of Applied Non-Classical Logics* 17(4), pp. 423–475, doi:10.3166/jancl.17.423-475.
- [14] Wojciech Jamroga & Wiebe van der Hoek (2004): *Agents that Know How to Play*. *Fundam. Inform.* 63(2-3), pp. 185–219. Available at <http://iospress.metapress.com/content/xh738axb47d8rchf/>.
- [15] G. Jonker (2003): *Feasible strategies in Alternating-time Temporal Epistemic Logic*. Master's thesis, University of Utrecht.
- [16] A. Lomuscio, H. Qu & F. Raimondi (2009): *MCMAS: A Model Checker for the Verification of Multi-Agent Systems*. In A. Bouajjani & O. Maler, editors: *CAV, Lecture Notes in Computer Science* 5643, Springer, pp. 682–688, doi:10.1007/978-3-642-02658-4_55.
- [17] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2011): *Reasoning About Strategies: On the Model-Checking Problem*. CoRR abs/1112.6275. Available at <http://arxiv.org/abs/1112.6275>.
- [18] David E. Muller & Paul E. Schupp (1987): *Alternating Automata on Infinite Trees*. *Theor. Comput. Sci.* 54, pp. 267–276, doi:10.1016/0304-3975(87)90133-2.
- [19] Marc Pauly (2002): *A Modal Logic for Coalitional Power in Games*. *J. Log. Comput.* 12(1), pp. 149–166, doi:10.1093/logcom/12.1.149.