Reasoning about Knowledge and Strategies under Hierarchical Information

Bastien Maubert and Aniello Murano
Università degli Studi di Napoli "Federico II"

Abstract

Two distinct semantics have been considered for knowledge in the context of strategic reasoning, depending on whether players know each other's strategy or not. In the former case, that we call the *informed* semantics, distributed synthesis for epistemic temporal specifications is undecidable, already on systems with hierarchical information. However, for the other, *uninformed* semantics, the problem is decidable on such systems. In this work we generalise this result by introducing an epistemic extension of Strategy Logic with imperfect information. The semantics of knowledge operators is uninformed, and captures agents that can change observation power when they change strategies. We solve the model-checking problem on a class of "hierarchical instances", which provides a solution to a vast class of strategic problems with epistemic temporal specifications, such as distributed or rational synthesis, on hierarchical systems.

Introduction

Logics of programs, many of which are based on classic temporal logics such as LTL or CTL, are meant to specify desirable properties of algorithms. When considering distributed systems, an important aspect is that each processor has a partial, local view of the whole system, and can only base its actions on the available information. Many authors have argued that this imperfect information calls for logical formalisms that would allow the modelling and reasoning about what different processes know of the system, and of other processes' state of knowledge. For instance, Halpern and Moses wrote in (Halpern and Moses 1990):

"[...] explicitly reasoning about the states of knowledge of the components of a distributed system provides a more general and uniform setting that offers insight into the basic structure and limitations of protocols in a given system."

To reason about knowledge and time, temporal logics have been extended with the knowledge operator K_a from epistemic logic, giving rise to a family of temporal epistemic logics (Fagin et al. 1995), which have been applied to, e.g., information flow and cryptographic protocols (van der Meyden and Su 2004; Halpern and O'Neill 2005), coordination problems in distributed systems (Neiger and Bazzi 1992) and motion planning in robotics (Brafman et al. 1997).

Distributed systems are often open systems, i.e., they interact with an environment and must react appropriately to actions taken by this environment. As a result, if we take the analogy where processors are players of a game, and processes are strategies for the processors, the task of synthesising distributed protocols can be seen as synthesising winning strategies in multi-player games with imperfect information. This analogy between the two settings is well known, and Ladner and Reif already wrote in (Ladner and Reif 1986) that "Distributed protocols are equivalent to (i.e., can be formally modelled as) games".

To reason about a certain type of game-related properties in distributed systems, Alternating-time Temporal Logic (ATL) was introduced (Alur, Henzinger, and Kupferman 2002). It can express the existence of strategies for coalitions of players in multi-player games, but cannot express some important game-theoretic concepts, such as the existence of Nash equilibria. To remedy this, Strategy Logic (SL) (Chatterjee, Henzinger, and Piterman 2010; Mogavero et al. 2014) was defined. Treating strategies as explicit first-order objects makes it very expressive, and it can for instance talk about Nash equilibria in a very natural way. These logics have been studied both for players with perfect information and players with imperfect information, and in the latter case either with the assumption that agents have no memory, or that they remember everything they observe. This last assumption, called *perfect recall*, is the one usually considered in distributed synthesis (Pnueli and Rosner 1990) and games with imperfect information (Reif 1984), and it is also central in logics of knowledge and time (Fagin et al. 1995). It is the one we consider in this work.

In order to reason about knowledge and strategic abilities in distributed systems, epistemic temporal logics and strategic logics have been combined. In particular, both ATL and SL have been extended with knowledge operators (van der Hoek and Wooldridge 2003; Jamroga and van der Hoek 2004; Belardinelli 2015; Dima, Enea, and Guelev 2010; Belardinelli et al. 2017a; 2017b). However, few decidable cases are known for the model checking of these logics with imperfect information and perfect recall. This is not surprising since strategic logics typically can express the existence of distributed strategies, a problem known to be undecidable for perfect recall, already for purely temporal specifications (Peterson and Reif 1979; Pnueli and Rosner 1990).

Semantics of knowledge with strategies. Mixing knowledge and strategies raises intricate questions of semantics. As a matter of fact, we find in the litterature two distinct semantics for epistemic operators in strategic contexts, one in works on distributed synthesis from epistemic temporal specifications, and another one in epistemic strategic logics. To explain in what they differ, let us first recall the semantics of the knowledge operator in epistemic temporal logics: a formula $K_a\varphi$ holds in a history h (finite sequence of states) of a system if φ holds in all histories h' that agent a cannot distinguish from h. In other words, agent a knows that φ holds if it holds in all histories that may be the current one according to what she observed. Now consider that the system is a multi-player game, and fix a strategy σ_b for some player b. Two semantics are possible for K_a: one could say that $K_a\varphi$ holds if φ holds in all possible histories that are indistinguishable to the current one, as in epistemic temporal logics, or one could restrict attention to those in which player b follows σ_b, discarding indistinguishable histories that are not consistent with σ_b. In the latter case, one may say that player a's knowledge is *refined* by the knowledge of σ_b, i.e., she knows that player b is using strategy σ_b, and she has the ability to refine her knowledge with this information, eliminating inconsistent possible histories. In the following this is what we will be referring to when saying that an agent *knows* some other agent's strategy. We shall also refer to the semantics where a knows σ_b as the *informed semantics*, and that in which she ignores it as the *uninformed semantics*.

The two semantics are relevant as they model different reasonable scenarios. For instance if players collaborate and have the ability to communicate, they may share their strategies with each other. But in many cases, components of a distributed system each receive only their own strategy, and thus are ignorant of other components' strategies.

All epistemic extensions of ATL and SL we know of consider the uninformed semantics. In contrast, works on distributed synthesis from epistemic temporal specifications use the informed semantics (van der Meyden and Vardi 1998; van der Meyden and Wilke 2005), even though it is not so obvious that they do. Indeed these works consider specifications in classic epistemic temporal logic, without strategic operators. But they ask for the existence of distributed strategies so that such specifications hold in the system restricted to the outcomes of these strategies. This corresponds to what we call the informed semantics, as the semantics of knowledge operators is, in effect, restricted to outcomes of the strategies that are being synthesised.

We only know of two works that discuss these two semantics. In (Bozzelli, Maubert, and Pinchinat 2015) two knowledge-like operators are studied, one for each semantics, but distributed synthesis is not considered. More interestingly, Puchala already observes in (Puchala 2010) that the distributed synthesis problem studied in (van der Meyden and Vardi 1998; van der Meyden and Wilke 2005) considers the informed semantics. While the problem is undecidable for this semantics even on hierarchical systems (van der Meyden and Wilke 2005), Puchala sketches a proof that the problem becomes decidable on the class of hierarchical systems when the uninformed semantics is used.

Contributions. We introduce a logic for reasoning about knowledge and strategies. Our Epistemic Strategy Logic (ESL) is based on Strategy Logic, and besides boolean and temporal operators, it contains the imperfect-information strategy quantifier $\langle\langle x\rangle\rangle^o$ from $\mathsf{SL}_{\mathrm{ii}}$ (Berthon et al. 2017), which reads as "there exists a strategy x with observation o", and epistemic operators K_a for each agent a. Our logic allows reasoning about agents whose means of observing the system changes over time, as agents may successively use strategies associated with different observations. This can model, for instance, an agent that is granted higher security clearance, giving her access to previously hidden information. The semantics of our epistemic operators takes into account agents' changes of observational power. ESL also contains the outcome quantifier $\mathbf{A}$ from Branching-time Strategy Logic (BSL) (Knight and Maubert 2015), which quantifies on outcomes of strategies currently used by the agents, and the unbinding operator $(a, ?)$, which frees an agent from her current strategy. The latter was introduced in (Laroussinie and Markey 2015) for ATL with strategy context and is also present in BSL. The outcome quantifier together with the unbinding operator allow us to express branching-time temporal properties without resorting to artificial strategy quantifications which may either affect the semantics of agents' knowledge or break hierarchicality.

We solve the model-checking problem for *hierarchical instances* of ESL. As in $\mathsf{SL}_{\mathrm{ii}}$, hierarchical instances are formula/model pairs such that, as one goes down the syntactic tree of the formula, observations annotating strategy quantifiers $\langle\langle x\rangle\rangle^o$ can only become finer. In addition, in ESL we require that knowledge operators do not refer to (outcomes of) strategies quantified higher in the formula.

Any problem which can be expressed as hierarchical instances of our logic is thus decidable, and since ESL is very expressive such problems are many. A first corollary is an alternative proof that distributed synthesis from epistemic temporal specifications with uninformed semantics is decidable on hierarchical systems. Puchala announced this result in (Puchala 2010), but we provide a stronger result by going from linear-time to branching-time epistemic specifications. We also allow for nesting of strategic operators in epistemic ones, as long as hierarchicality is preserved and epistemic formulas do not refer to previously quantified strategies. Another corollary is that rational synthesis (Kupferman, Perelli, and Vardi 2016; Condurache et al. 2016) with imperfect information is decidable on hierarchical systems for epistemic temporal objectives with uninformed semantics.

Our approach to solve the model-checking problem for our logic extends that followed in (Laroussinie and Markey 2015; Berthon et al. 2017), which consists in "compiling" the strategic logic under study into an opportune variant of Quantified CTL^*, or QCTL^* for short (Laroussinie and Markey 2014). This is an extension of CTL^* with second-order quantification on propositions which serves as an intermediary, low-level logic between strategic logics and tree automata. In (Laroussinie and Markey 2015), model checking ATL^* with strategy context is proved decidable by reduction to QCTL^*. In (Berthon et al. 2017), model checking $\mathsf{SL}_{\mathrm{ii}}$ is proved decidable for a class of hierarchical instances by reduction to the hierarchical fragment of an imperfect information extension of QCTL^*, called $\mathsf{QCTL}^*_{\mathrm{ii}}$. In this work we define $\mathsf{EQCTL}^*_{\mathrm{ii}}$, which extends further $\mathsf{QCTL}^*_{\mathrm{ii}}$ with epistemic operators and an operator of observation change introduced recently in (Barrière et al. 2018) in the context of epistemic temporal logics. We define the hierarchical fragment of $\mathsf{EQCTL}^*_{\mathrm{ii}}$, which strictly contains that of $\mathsf{QCTL}^*_{\mathrm{ii}}$, and solve its model-checking problem for this fragment.

Related work. We know of five other logics called Epistemic Strategy Logic. In (Huang and Van Der Meyden 2014), epistemic temporal logic is extended with a first-order quantification $\exists x$ on points in runs of the system and an operator $\boldsymbol{e}_i(x)$ that compares local state i at x and at the current point. When interpreted on systems where strategies are encoded in local states, this logic can express existence of strategies and what agents know about it. However it only concerns memoryless strategies. Strategy Logic is extended with epistemic operators in (Cermák et al. 2014), but they also consider memoryless agents. (Belardinelli 2015) extends a fragment of SL with epistemic operators, and considers perfect-recall strategies, but model checking is not studied. The latter logic is extended in (Belardinelli et al. 2017a), in which its model-checking problem is solved on the class of broadcast systems. In (Knight and Maubert 2015) SL is also extended with epistemic operators and perfect-recall agents. Their logic does not require strategies to be uniform, but this requirement can be expressed in the language. However no decidability result is provided. The result we present here is the first for an epistemic strategic logic with perfect recall on hierarchical systems. In addition, ours is the first epistemic strategic logic to allow for changes of observational power.

Plan. We first define ESL and hierarchical instances, and announce our main result. Next we introduce $\mathsf{EQCTL}^*_{\mathrm{ii}}$ and solve the model-checking problem for its hierarchical fragment. We then establish our main result by reducing model checking hierarchical instances of ESL to model checking hierarchical $\mathsf{EQCTL}^*_{\mathrm{ii}}$. We finally present two corollaries, exemplifying what our logic can express, and we finish with a discussion on the semantics of knowledge and strategies.

Notations

Let Σ be an alphabet. A *finite* (resp. *infinite*) *word* over Σ is an element of Σ^* (resp. Σ^ω). The *length* of a finite word $w = w_0 w_1 \dots w_n$ is $|w| := n+1$, $\mathrm{last}(w) := w_n$ is its last letter, and we note ϵ for the empty word. Given a finite (resp. infinite) word w and $0 \leq i \leq |w|$ (resp. $i \in \mathbb{N}$), we let w_i be the letter at position i in w, $w_{\leq i}$ is the prefix of w that ends at position i and $w_{\geq i}$ is the suffix of w that starts at position i. We write $w \preccurlyeq w'$ if w is a prefix of w', and $w^{\preccurlyeq}$ is the set of finite prefixes of word w. Finally, the domain of a mapping f is written $dom(f)$, and for $n \in \mathbb{N}$ we let $[n] := \{i \in \mathbb{N} : 1 \leq i \leq n\}$.

We fix for the rest of the paper a number of parameters for our logics and models: AP is a finite set of *atomic propositions*, Ag is a finite set of *agents* or *players*, Var is a finite set of *variables* and Obs is a finite set of *observation symbols*. These data are implicitly part of the input for the model-checking problems we consider.

Epistemic Strategy Logic

In this section we introduce our epistemic extension of Strategy Logic with imperfect information.

Models

The models of ESL are essentially the same as those of $\mathsf{SL}_{\mathrm{ii}}$, i.e., concurrent game structures extended by an observation interpretation $\mathcal{O}$, that maps each observation symbol $o \in \mathrm{Obs}$ to an equivalence relation $\mathcal{O}(o)$ over positions of the game structure. However models in ESL contain, in addition, an initial observation for each player. This initial observation may change if the player receives a strategy corresponding to a different observation.

Definition 1 ($\mathrm{CGS}_{\mathrm{ii}}$). *A* concurrent game structure with imperfect information *(or CGS_{ii} for short) is a structure $\mathcal{G} = (\mathrm{Ac}, V, E, \ell, \mathcal{O}, v^\iota, \boldsymbol{o}^\iota)$ where*

- Ac *is a finite non-empty set of* actions*,*
- V *is a finite non-empty set of* positions*,*
- $E : V \times \mathrm{Ac}^{\mathrm{Ag}} \to V$ *is a* transition function*,*
- $\ell : V \to 2^{\mathrm{AP}}$ *is a* labelling function*,*
- $\mathcal{O} : \mathrm{Obs} \to V \times V$ *is an* observation interpretation*, and for each $o \in \mathrm{Obs}$, $\mathcal{O}(o)$ is an equivalence relation,*
- $v^\iota \in V$ *is an* initial position*, and*
- $\boldsymbol{o}^\iota \in \mathrm{Obs}^{\mathrm{Ag}}$ *is a tuple of* initial observations*.*

Two positions being equivalent for relation $\mathcal{O}(o)$ means that a player using a strategy with observation o cannot distinguish them. In the following we may write $\sim_o$ for $\mathcal{O}(o)$ and $v \in \mathcal{G}$ for $v \in V$.

Joint actions. When in a position $v \in V$, each player a chooses an action $c_a \in \mathrm{Ac}$ and the game proceeds to position $E(v, \boldsymbol{c})$, where $\boldsymbol{c} \in \mathrm{Ac}^{\mathrm{Ag}}$ stands for the *joint action* $(c_a)_{a \in \mathrm{Ag}}$. If $\boldsymbol{c} = (c_a)_{a \in \mathrm{Ag}}$, we let $\boldsymbol{c}_a$ denote c_a for $a \in \mathrm{Ag}$.

Plays and strategies. A *finite* (resp. *infinite*) *play* is a finite (resp. infinite) word $\rho = v_0 \dots v_n$ (resp. $\pi = v_0 v_1 \dots$) such that $v_0 = v_\iota$ and for all i with $0 \leq i < |\rho| - 1$ (resp. $i \geq 0$), there exists a joint action $\boldsymbol{c}$ such that $E(v_i, \boldsymbol{c}) = v_{i+1}$. We let Plays be the set of finite plays. A *strategy* is a function $\sigma : \mathrm{Plays} \to \mathrm{Ac}$, and we let *Str* be the set of all strategies.

Assignments. An *assignment* $\chi : \mathrm{Ag} \cup \mathrm{Var} \rightharpoonup Str$ is a partial function assigning to each player and variable in its domain a strategy. For an assignment χ, a player a and a strategy σ, $\chi[a \mapsto \sigma]$ is the assignment of domain $dom(\chi) \cup \{a\}$ that maps a to σ and is equal to χ on the rest of its domain, and $\chi[x \mapsto \sigma]$ is defined similarly, where x is a variable; also, $\chi[a \mapsto ?]$ is the assignment of domain $dom(\chi) \setminus \{a\}$, on which it is equal to χ.

Outcomes. For an assignment χ and a finite play ρ, we let $\mathrm{out}(\chi, \rho)$ be the set of infinite plays that start with ρ and are then extended by letting players follow the strategies assigned by χ. Formally, $\mathrm{out}(\chi, \rho)$ is the set of infinite plays of the form $\rho \cdot v_1 v_2 \dots$ such that for all $i \geq 0$, there exists $\boldsymbol{c}$

such that for all $a \in dom(\chi) \cap \mathrm{Ag}$, $\boldsymbol{c}_a \in \chi(a)(\rho \cdot v_1 \dots v_i)$ and $v_{i+1} = E(v_i, \boldsymbol{c})$, with $v_0 = \mathrm{last}(\rho)$.

Synchronous perfect recall. Players with perfect recall remember the whole history of a play. Each observation relation is thus extended to finite plays as follows: $\rho \approx_o \rho'$ if $|\rho| = |\rho'|$ and $\rho_i \sim_o \rho'_i$ for every $i \in \{0, \dots, |\rho| - 1\}$.

Uniform strategies. For $o \in \mathrm{Obs}$, an *o-strategy* is a strategy $\sigma : V^+ \to \mathrm{Ac}$ such that $\sigma(\rho) = \sigma(\rho')$ whenever $\rho \approx_o \rho'$. For $o \in \mathrm{Obs}$ we let Str_o be the set of all o-strategies.

Syntax

ESL extends $\mathsf{SL}_{\mathrm{ii}}$ with knowledge operators K_a for each agent $a \in \mathrm{Ag}$, and the *outcome quantifier* from Branching-time Strategy Logic, introduced in (Knight and Maubert 2015), which quantifies on outcomes of the currently fixed strategies. While in SL temporal operators could only be evaluated in contexts where all agents were assigned to a strategy, this outcome quantifier allows for evaluation of (branching-time) temporal properties on partial assignments of strategies to agents. This outcome quantifier can be simulated in usual, linear-time variants of Strategy Logic, by quantifying on strategies for agents who do not currently have one. But in the context of imperfect information, where strategy quantifiers are parameterised by an observation, this may cause to either break the hierarchy or artificially modify an agent's observation, which affects his knowledge.

Definition 2 (ESL Syntax)**.** *The syntax of* ESL *is defined by the following grammar:*

$$\varphi : p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle x \rangle\!\rangle^o \varphi \mid (a, x)\varphi \mid (a, ?)\varphi \mid K_a\varphi \mid \mathbf{A}\psi$$
$$\psi : \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi,$$

where $p \in \mathrm{AP}$*,* $x \in \mathrm{Var}$*,* $o \in \mathrm{Obs}$ *and* $a \in \mathrm{Ag}$*.*

Formulas of type φ are called *history formulas*, those of type ψ are *path formulas*. We may use usual abbreviations $\top := p \vee \neg p$, $\bot := \neg\top$, $\varphi \to \varphi' := \neg\varphi \vee \varphi'$, $\mathbf{F}\varphi := \top\mathbf{U}\varphi$, $\mathbf{G}\varphi := \neg\mathbf{F}\neg\varphi$, $[\![x]\!]^o\varphi := \neg\langle\!\langle x \rangle\!\rangle^o\neg\varphi$ and $\mathbf{E}\psi := \neg\mathbf{A}\neg\psi$.

A variable x appears *free* in a formula φ if it appears out of the scope of a strategy quantifier. We let *free*(φ) be the set of variables that appear free in φ. An assignment χ is *variable-complete* for a formula φ if its domain contains all free variables in φ. Finally, a *sentence* is a history formula φ such that *free*$(\varphi) = \emptyset$.

Remark 1. *Without loss of generality we can assume that each strategy variable x is quantified at most once in an* ESL *formula. Thus, each variable x that appears in a sentence is uniquely associated to a strategy quantification $\langle\!\langle x \rangle\!\rangle^o$, and we let $o_x = o$.*

Discussion on the syntax. In $\mathsf{SL}_{\mathrm{ii}}$ as well as in ESL, the observation used by a strategy is specified at the level of strategy quantification: $\langle\!\langle x \rangle\!\rangle^o\varphi$ reads as "there exists a strategy x with observation o such that φ holds". When a strategy with observation o is assigned to some agent a via a binding (a, x), it seems natural to consider that agent a starts observing the system with observation o. As a result agents can change observation when they change strategy, and thus they can observe the game with different observation powers along a same play. This contrasts with most of the literature on epistemic temporal logics, where each agent's observation power is usually fixed in the model.

Epistemic relations with changing observations

Dynamic observation change has been studied recently in the context of epistemic temporal logics in (Barrière et al. 2018), from which come the following definitions.

First, dealing with the possibility to dynamically change observation requires to remember which observation each agent had at each point in time.

Observation records. An *observation record* r_a for agent $a \in \mathrm{Ag}$ is a finite word over $\mathbb{N} \times \mathrm{Obs}$, i.e., $r_a \in (\mathbb{N} \times \mathrm{Obs})^*$.

If at time n, an agent a with current observation record r_a receives a strategy with observation o, her new observation record is $r_a \cdot (o, n)$. The observation record $r_a[n]$ is the projection of r_a on $\{n\} \times \mathrm{Obs}$, and represents the sequence of observation changes that occurred at time n.

Given an observation record for each agent $\boldsymbol{r} = (r_a)_{a \in \mathrm{Ag}}$, we note $\boldsymbol{r}_a$ for r_a. We say that an observation record $\boldsymbol{r}$ *stops at time* n if $r_a[m]$ is empty for all $m > n$, and $\boldsymbol{r}$ *stops at a finite play* ρ if it stops at time $|\rho| - 1$. If $\boldsymbol{r}$ stops at time n, we let $\boldsymbol{r} \cdot (n, o)_a$ be the observation record $\boldsymbol{r}$ where $\boldsymbol{r}_a$ is replaced with $\boldsymbol{r}_a \cdot (n, o)$.

At each step of a play, each agent observes the new position with her current observation power. Then, if an agent changes strategy, she observes the same position with the observation of the new strategy, which may be different from the previous one. Also, due to the syntax of ESL, an agent may change observation several times before the next step. Therefore, the *observation sequence* $os_a(\boldsymbol{r}, n)$ with which agent a observes the game at time n consists of the observation she had when the n-th move is taken, plus those corresponding to strategy changes that occur before the next step. It is defined by induction on n:

$$os_a(\boldsymbol{r}, 0) = o_1 \cdot \ldots \cdot o_k, \quad \text{if } \boldsymbol{r}_a[0] = (0, o_1) \cdot \ldots \cdot (0, o_k), \text{ and}$$
$$os_a(\boldsymbol{r}, n+1) = \mathrm{last}(os_a(\boldsymbol{r}, n)) \cdot o_1 \cdot \ldots \cdot o_k, \quad \text{if } \boldsymbol{r}_a[n+1] = (n+1, o_1) \cdot \ldots \cdot (n+1, o_k).$$

If at time n agent a does not receive a new strategy, $os_a(\boldsymbol{r}, n)$ contains only one observation, which will be either that of the last strategy taken by the agent or the agent's initial observation, given by the $\mathrm{CGS}_{\mathrm{ii}}$.

The indistinguishability relation for synchronous perfect-recall with observation change is defined as follows.

Definition 3. *For ρ and ρ' two finite plays and $\boldsymbol{r}$ an observation record, ρ and ρ' are observationally equivalent to agent a, written $\rho \approx^{\boldsymbol{r}}_a \rho'$, if $|\rho| = |\rho'|$ and, for every $i \in \{0, \dots, |\rho| - 1\}$, for every $o \in os_a(\boldsymbol{r}, i)$, $\rho_i \sim_o \rho'_i$.*

Remark 2. *Observe that, at a given point in time, the order in which an agent observes the game with different observation does not matter. Intuitively, all that matters is the total information gathered before the next step. Also, in the case of an empty observation record, the above definition corresponds to blind agents, for which all finite plays of same* *length are indistinguishable. However in the following observation records will never be empty, but will always be initialised with the initial observations given by the model.*

Semantics

We now define the semantics of ESL.

Definition 4 (ESL Semantics)**.** *The semantics of a history formula is defined on a game $\mathcal{G}$ (omitted below), an assignment χ variable-complete for φ, and a finite play ρ. For a path formula ψ, the finite play is replaced with an infinite play π and an index $i \in \mathbb{N}$. The definition is as follows:*

$$
\begin{array}{lll}
\chi, \boldsymbol{r}, \rho \models p & \textit{if} & p \in \ell(\textit{last}(\rho)) \\
\chi, \boldsymbol{r}, \rho \models \neg\varphi & \textit{if} & \chi, \boldsymbol{r}, \rho \not\models \varphi \\
\chi, \boldsymbol{r}, \rho \models \varphi \vee \varphi' & \textit{if} & \chi, \boldsymbol{r}, \rho \models \varphi \ \textit{or} \ \chi, \boldsymbol{r}, \rho \models \varphi' \\
\chi, \boldsymbol{r}, \rho \models \langle\!\langle x \rangle\!\rangle^{o}\varphi & \textit{if} & \exists \sigma \in \mathrm{Str}_o.\ \chi[x \mapsto \sigma], \boldsymbol{r}, \rho \models \varphi \\
\chi, \boldsymbol{r}, \rho \models (a, x)\varphi & \textit{if} & \chi[a \mapsto \chi(x)], \boldsymbol{r}, \rho \models \varphi \\
\chi, \boldsymbol{r}, \rho \models (a, ?)\varphi & \textit{if} & \chi[a \mapsto ?], \boldsymbol{r}, \rho \models \varphi \\
\chi, \boldsymbol{r}, \rho \models K_a\varphi & \textit{if} & \forall \rho' \in \textit{Plays s.t. } \rho' \approx_a^{\boldsymbol{r}} \rho, \\
 & & \chi, \boldsymbol{r}, \rho' \models \varphi \\
\chi, \boldsymbol{r}, \rho \models \mathbf{A}\psi & \textit{if} & \forall \pi \in \mathrm{out}(\chi, \rho), \\
 & & \chi, \boldsymbol{r}, \pi, |\rho| - 1 \models \psi \\
\chi, \boldsymbol{r}, \pi, i \models \varphi & \textit{if} & \chi, \boldsymbol{r}, \pi_{\leq i} \models \varphi \\
\chi, \boldsymbol{r}, \pi, i \models \neg\psi & \textit{if} & \chi, \boldsymbol{r}, \pi, i \not\models \psi \\
\chi, \boldsymbol{r}, \pi, i \models \psi \vee \psi' & \textit{if} & \chi, \boldsymbol{r}, \pi, i \models \psi \ \textit{or} \ \chi, \boldsymbol{r}, \pi, i \models \psi' \\
\chi, \boldsymbol{r}, \pi, i \models \mathbf{X}\psi & \textit{if} & \chi, \boldsymbol{r}, \pi, i + 1 \models \psi \\
\chi, \boldsymbol{r}, \pi, i \models \psi\mathbf{U}\psi' & \textit{if} & \exists\, j \geq i \textit{ s.t. } \chi, \boldsymbol{r}, \pi, j \models \psi' \textit{ and} \\
 & & \forall\, k \in [i, j[,\ \chi, \boldsymbol{r}, \pi, k \models \psi
\end{array}
$$

The satisfaction of a sentence is independent of the assignment; for an ESL sentence φ we thus let $\mathcal{G}, \boldsymbol{r}, \rho \models \varphi$ if $\mathcal{G}, \chi, \boldsymbol{r}, \rho \models \varphi$ for some assignment χ. We also write $\mathcal{G} \models \varphi$ if $\mathcal{G}, \boldsymbol{r}^{\iota}, v^{\iota} \models \varphi$, where $\boldsymbol{r}^{\iota} = (0, \boldsymbol{o}_a^{\iota})_{a \in \mathrm{Ag}}$.

Discussion on the semantics. First, the semantics of the knowledge operator corresponds, as announced, to what we called *uninformed semantics* in the introduction. Indeed it is not restricted to outcomes of strategies followed by the players: $K_a\varphi$ holds in a finite play ρ if φ holds in *all finite plays in the game* that are indistinguishable to ρ for agent a.

Also, note that the relation for perfect recall with observation change, and thus also observation records, are only used in the semantics of the knowledge operators. As usual, a strategy with observation o has to be uniform with regards to the classic perfect-recall relation $\sim_o$ for static observations, even if it is assigned to an agent who previously had a different observation. The reasons to do so are twofold.

First, we do not see any other natural definition. One may think of parameterising strategy quantifiers with observation records instead of mere observations, but this would require to know at the level of quantification at which points in time the strategy will be given to an agent, and what previous observations this agent would have had, which is not realistic.

More importantly, when one asks for the existence of a uniform strategy after some finite play ρ, it only matters how the strategy is defined on suffixes of ρ, and thus the uniformity constraint also is relevant only on such plays. But for such plays, the "fixed observation" indistinguishability relation is the same as the "dynamic observation" one. More precisely, if agent a receives observation o at the end of ρ, i.e., $\mathrm{last}(\boldsymbol{r}_a) = (|\rho| - 1, o)$, then for all finite plays ρ', ρ'' that are suffixes of ρ, we have $\rho' \approx_o \rho''$ if, and only if, $\rho' \approx_a^{\boldsymbol{r}} \rho''$. Indeed, since we use the S5 semantics of knowledge, i.e., indistinguishability relations are equivalence relations, the prefix ρ is always related to itself, be it for $\approx_a^{\boldsymbol{r}}$ or $\approx_o$, and after ρ both relations only consider observation o.

Model checking and hierarchical instances

We now introduce the decision problem studied in this paper, i.e., the model-checking problem for ESL.

Model checking. An *instance* is a pair $(\Phi, \mathcal{G})$ where Φ is a sentence of ESL and $\mathcal{G}$ is a $\mathrm{CGS_{ii}}$. The *model-checking problem* for ESL is the decision problem that, given an instance $(\Phi, \mathcal{G})$, returns 'yes' if $\mathcal{G} \models \Phi$, and 'no' otherwise.

$\mathsf{SL_{ii}}$ can be translated into ESL, by adding outcome quantifiers before temporal operators. Since model checking $\mathsf{SL_{ii}}$ is undecidable (Berthon et al. 2017), we get the following result:

Theorem 1. *Model checking* ESL *is undecidable.*

Hierarchical instances. We now isolate a sub-problem obtained by restricting attention to *hierarchical instances*. Intuitively, an ESL-instance $(\Phi, \mathcal{G})$ is hierarchical if, as one goes down a path in the syntactic tree of Φ, the observations parameterising strategy quantifiers become finer. In addition, epistemic formulas must not talk about currently defined strategies.

Given an ESL sentence Φ and a syntactic subformula φ of Φ, by parsing Φ's syntactic tree one can define the set $\mathrm{Ag}(\varphi)$ of agents who are bound to a strategy at the level of φ, as well as where in Φ these strategies are quantified upon.

Definition 5. *Let Φ be an* ESL *sentence. A subformula $K_a\varphi$ is* free *if for every subformula $\mathbf{A}\psi$ of φ, the current strategy of each agent in $\mathrm{Ag}(\mathbf{A}\psi)$ is quantified within φ.*

In other words, an epistemic subformula $K_a\varphi$ is free if it does not talk about strategies that are quantified before it.

Example 1. *If $\Phi = \langle\!\langle x \rangle\!\rangle^{o}(a, x)K_a\mathbf{AX}p$, then $K_a\mathbf{AX}p$ is not free in Φ, because at the level of $\mathbf{A}$ agent a is bound to strategy x which is quantified "outside" of $K_a\mathbf{AX}p$. But if $\Phi = \langle\!\langle x \rangle\!\rangle^{o}(a, x)K_a(a, ?)\mathbf{AX}p$, then $K_a(a, ?)\mathbf{AX}p$ is free in Φ, because at the level of $\mathbf{A}$ no agent is bound to a strategy. Also if $\Phi = \langle\!\langle x \rangle\!\rangle^{o}(a, x)K_a\langle\!\langle y \rangle\!\rangle^{o'}(a, y)\mathbf{AX}p$, then $K_a\langle\!\langle y \rangle\!\rangle^{o'}(a, y)\mathbf{AX}p$ is free in Φ, because at the level of $\mathbf{A}$ the only agent bound to a strategy is a, and her strategy is quantified upon after the knowledge operator.*

We can now define the hierarchical fragment for which we establish decidability of the model-checking problem.

Definition 6 (Hierarchical instances)**.** *An* ESL*-instance $(\Phi, \mathcal{G})$ is* hierarchical *if all epistemic subformulas of Φ are free in Φ and, for all subformulas of the form $\varphi_1 = \langle\!\langle x \rangle\!\rangle^{o_1}\varphi_1'$ and $\varphi_2 = \langle\!\langle x \rangle\!\rangle^{o_2}\varphi_2'$ where φ_2 is a subformula of φ_1', it holds that $\mathcal{O}(o_2) \subseteq \mathcal{O}(o_1)$.*

In other words, an instance is hierarchical if innermost strategy quantifiers observe at least as much as outermost ones, and epistemic formulas do not talk about current strategies. Here is the main contribution of this work:

Theorem 2. *The model-checking problem for* ESL *restricted to the class of hierarchical instances is decidable.*

We prove this result by reducing it to the model-checking problem for the hierarchical fragment of an extension of QCTL^* with imperfect information, knowledge and observation change, which we now introduce and study in order to use it as an intermediate, "low-level" logic between tree automata and ESL.

$\mathsf{QCTL}^*_{\text{ii}}$ with knowledge and observation change

QCTL^* extends CTL^* with second order quantification on atomic propositions (Emerson and Sistla 1984; Kupferman 1995; Kupferman et al. 2000; French 2001; Laroussinie and Markey 2014). It was recently extended to model imperfect-information aspects, resulting in the logic called $\mathsf{QCTL}^*_{\text{ii}}$ (Berthon et al. 2017). In this section we first define an epistemic extension of $\mathsf{QCTL}^*_{\text{ii}}$ with operators for knowledge and dynamic observation change, that we call $\mathsf{EQCTL}^*_{\text{ii}}$. Then we define the syntactic class of *hierarchical formulas* and prove that model checking this class of formulas is decidable.

Models

The models of $\mathsf{EQCTL}^*_{\text{ii}}$, as those of $\mathsf{QCTL}^*_{\text{ii}}$, are structures in which states are tuples of local states. Fix $n \in \mathbb{N}$.

Local states. Let $\{L_i\}_{i\in[n]}$ denote n disjoint finite sets of *local states*. For $I \subseteq [n]$, we let $L_I := \prod_{i\in I} L_i$ if $I \neq \emptyset$, and $L_\emptyset := \{\mathbf{0}\}$ where $\mathbf{0}$ is a special symbol.

Concrete observations. A set $\mathrm{o} \subseteq [n]$ is a *concrete observation* (to distinguish from observation symbols o of ESL).

Fix $\mathrm{o} \subseteq [n]$ and $I \subseteq [n]$. Two tuples $x, x' \in L_I$ are o-*indistinguishable*, written $x \sim_{\mathrm{o}} x'$, if for each $i \in I \cap \mathrm{o}$, $x_i = x'_i$. Two words $u = u_0 \dots u_i$ and $u' = u'_0 \dots u'_j$ over alphabet L_I are o-*indistinguishable*, written $u \approx_{\mathrm{o}} u'$, if $i = j$ and for all $k \in \{0, \dots, i\}$ we have $u_k \sim_{\mathrm{o}} u'_k$.

Compound Kripke structures. These are like Kripke structures except that the states are elements of $L_{[n]}$. A *compound Kripke structure*, or CKS, over AP, is a tuple $\mathcal{S} = (S, R, \ell, s^\iota, \mathbf{o}^\iota)$ where $S \subseteq L_{[n]}$ is a set of *states*, $R \subseteq S \times S$ is a left-total[1] *transition relation*, $\ell : S \to 2^{\text{AP}}$ is a *labelling function*, $s^\iota \in S$ is an *initial state*, and $\mathbf{o}^\iota = (\mathrm{o}^\iota_a)_{a\in\text{Ag}}$ is an *initial concrete observation* for each agent.

A *path* in $\mathcal{S}$ is an infinite sequence of states $\pi = s_0 s_1 \dots$ such that for all $i \in \mathbb{N}$, $(s_i, s_{i+1}) \in R$, and a *finite path* $\rho = s_0 s_1 \dots s_n$ is a finite prefix of a path. For $s \in S$, we let $\text{Paths}^\omega(s)$ be the set of all paths that start in s, and $\text{Paths}^*(s)$ is the set of finite paths that start in s.

Syntax of $\mathsf{EQCTL}^*_{\text{ii}}$

The syntax of $\mathsf{EQCTL}^*_{\text{ii}}$ extends that of $\mathsf{QCTL}^*_{\text{ii}}$ with epistemic operators K_a and observation-change operators Δ^{o}_a, which were recently introduced and studied in (Barrière et al. 2018) in an epistemic temporal logic without second-order quantification.

[1] i.e., for all $s \in S$, there exists s' such that $(s, s') \in R$.

Definition 7 ($\mathsf{EQCTL}^*_{\text{ii}}$ Syntax)**.** *The syntax of* $\mathsf{EQCTL}^*_{\text{ii}}$ *is defined by the following grammar:*

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{A}\psi \mid \exists^{\mathrm{o}} p.\, \varphi \mid K_a \varphi \mid \Delta^{\mathrm{o}}_a \varphi$$
$$\psi := \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi \mathbf{U} \psi$$

where $p \in \text{AP}$, $a \in \text{Ag}$ *and* $\mathrm{o} \subseteq [n]$.

Formulas of type φ are called *state formulas*, those of type ψ are called *path formulas*, and $\mathsf{EQCTL}^*_{\text{ii}}$ consists of all the state formulas. $\mathbf{A}$ is the classic path quantifier from branching-time temporal logics. $\exists^{\mathrm{o}}$ is the second-order quantifier with imperfect information from $\mathsf{QCTL}^*_{\text{ii}}$ (Berthon et al. 2017). $\exists^{\mathrm{o}} p.\, \varphi$ holds in a tree if there is way to choose a labelling for p such that φ holds, with the constraint that o-equivalent nodes of the tree must be labelled identically. $K_a\varphi$ means "agent a knows that φ holds", where the knowledge depends on the sequence of observations agent a has had; finally, $\Delta^{\mathrm{o}}_a\varphi$ means that after agent a switches to observation o, φ holds.

Given an $\mathsf{EQCTL}^*_{\text{ii}}$ formula φ, we define the set of *quantified propositions* $\text{AP}_\exists(\varphi) \subseteq \text{AP}$ as the set of atomic propositions p such that φ has a subformula of the form $\exists^{\mathrm{o}} p.\, \varphi$. We also define the set of *free propositions* $\text{AP}_f(\varphi) \subseteq \text{AP}$ as the set of atomic propositions p that appear out of the scope of any quantifier of the form $\exists^{\mathrm{o}} p$.

Semantics of $\mathsf{EQCTL}^*_{\text{ii}}$

Before defining the semantics of the logic we first recall some definitions for trees.

Trees. Let X be a finite set (typically a set of states). An X*-tree* τ is a nonempty set of words $\tau \subseteq X^+$ such that:

- there exists $x^\iota \in X$, called the *root* of τ, such that each $u \in \tau$ starts with x^ι (i.e., $x^\iota \preccurlyeq u$);
- if $u \cdot x \in \tau$ and $u \neq \epsilon$, then $u \in \tau$, and
- if $u \in \tau$ then there exists $x \in X$ such that $u \cdot x \in \tau$.

The elements of a tree τ are called *nodes*. If $u \cdot x \in \tau$, we say that $u \cdot x$ is a *child* of u. A *path* in τ is an infinite sequence of nodes $\lambda = u_0 u_1 \dots$ such that for all $i \in \mathbb{N}$, u_{i+1} is a child of u_i, and $\text{Paths}^\omega(u)$ is the set of paths that start in node u. An AP*-labelled* X*-tree*, or (AP, X)*-tree* for short, is a pair $t = (\tau, \ell)$, where τ is an X-tree called the *domain* of t and $\ell : \tau \to 2^{\text{AP}}$ is a *labelling*. For a labelled tree $t = (\tau, \ell)$ and an atomic proposition $p \in \text{AP}$, we define the p*-projection of* t as the labelled tree $t \Downarrow_p := (\tau, \ell \Downarrow_p)$, where for each $u \in \tau$, $\ell \Downarrow_p(u) := \ell(u) \setminus \{p\}$. Two labelled trees $t = (\tau, \ell)$ and $t' = (\tau', \ell')$ are *equivalent modulo* p, written $t \equiv_p t'$, if $t \Downarrow_p = t' \Downarrow_p$ (in particular, $\tau = \tau'$).

Quantification and uniformity. In $\mathsf{EQCTL}^*_{\text{ii}}$, as in $\mathsf{QCTL}^*_{\text{ii}}$, $\exists^{\mathrm{o}} p.\, \varphi$ holds in a tree t if there is some o-uniform p-labelling of t such that t with this p-labelling satisfies φ. A p-labelling of a tree is o-uniform if every two nodes that are indistinguishable for observation o agree on their p-labelling.

Definition 8 (o-uniformity)**.** *A labelled tree* $t = (\tau, \ell)$ *is* o-uniform in p *if for every pair of nodes* $u, u' \in \tau$ *such that* $u \approx_{\mathrm{o}} u'$, *we have* $p \in \ell(u)$ *iff* $p \in \ell(u')$.

Changing observations. To capture how the observation-change operator affects the semantics of the knowledge operator, we use again observation records $\boldsymbol{r}$ and the associated notion of observation sequence $os_a(\boldsymbol{r}, n)$. They are defined as for ESL except that observation symbols o are replaced with concrete observations o. For $u = u_0 \dots u_i$ and $u' = u'_0 \dots u'_j$ over alphabet L_I, and an observation record $\boldsymbol{r}$, we say that u and u' are observationally equivalent to agent a, written $u \approx_a^{\boldsymbol{r}} u'$, if $i = j$ and, for every $k \in \{0, \dots, i\}$ and every $\mathrm{o} \in os_a(\boldsymbol{r}, k)$, $u_k \sim_{\mathrm{o}} u'_k$.

Finally, we inductively define the satisfaction relation $\models$. Let $t = (\tau, \ell)$ be a 2^{AP}-labelled L_I-tree, u a node and $\boldsymbol{r}$ an observation record that stops at u:

$t, \boldsymbol{r}, u \models p$	if	$p \in \ell(u)$
$t, \boldsymbol{r}, u \models \neg\varphi$	if	$t, \boldsymbol{r}, u \not\models \varphi$
$t, \boldsymbol{r}, u \models \varphi \vee \varphi'$	if	$t, \boldsymbol{r}, u \models \varphi$ or $t, \boldsymbol{r}, u \models \varphi'$
$t, \boldsymbol{r}, u \models \mathbf{E}\psi$	if	$\exists\, \lambda \in \text{Paths}^{\omega}(u)$ s.t. $t, \boldsymbol{r}, \lambda \models \psi$
$t, \boldsymbol{r}, u \models \exists^{\mathrm{o}} p.\, \varphi$	if	$\exists\, t' \equiv_p t$ s.t. t' is o-uniform in p and $t', \boldsymbol{r}, u \models \varphi$
$t, \boldsymbol{r}, u \models K_a\varphi$	if	$\forall u' \in t$ s.t. $u \approx_a^{\boldsymbol{r}} u'$, $t, \boldsymbol{r}, u' \models \varphi$

And if λ is a path in τ and $\boldsymbol{r}$ stops at λ_0:

$t, \boldsymbol{r}, \lambda \models \varphi$	if	$t, \boldsymbol{r}, \lambda_0 \models \varphi$
$t, \boldsymbol{r}, \lambda \models \neg\psi$	if	$t, \boldsymbol{r}, \lambda \not\models \psi$
$t, \boldsymbol{r}, \lambda \models \psi \vee \psi'$	if	$t, \boldsymbol{r}, \lambda \models \psi$ or $t, \boldsymbol{r}, \lambda \models \psi'$
$t, \boldsymbol{r}, \lambda \models \mathbf{X}\psi$	if	$t, \boldsymbol{r}, \lambda_{\geq 1} \models \psi$
$t, \boldsymbol{r}, \lambda \models \psi \mathbf{U} \psi'$	if	$\exists\, i \geq 0$ s.t. $t, \boldsymbol{r}, \lambda_{\geq i} \models \psi'$ and $\forall j$ s.t. $0 \leq j < i$, $t, \boldsymbol{r}, \lambda_{\geq j} \models \psi$

We let $t, \boldsymbol{r} \models \varphi$ denote $t, \boldsymbol{r}, x^{\iota} \models \varphi$, where x^{ι} is t's root.

Tree unfoldings $t_{\mathcal{S}}$. Let $\mathcal{S} = (S, R, \ell, s^{\iota}, \mathbf{o}^{\iota})$ be a compound Kripke structure over AP. The *tree-unfolding of* $\mathcal{S}$ is the (AP, S)-tree $t_{\mathcal{S}} := (\tau, \ell')$, where τ is the set of all finite paths that start in s^{ι}, and for every $u \in \tau$, $\ell'(u) := \ell(\text{last}(u))$. Given a CKS $\mathcal{S}$ and an $\mathsf{EQCTL}^*_{\text{ii}}$ formula φ, we write $\mathcal{S} \models \varphi$ if $t_{\mathcal{S}}, \boldsymbol{r}^{\iota} \models \varphi$, where $\boldsymbol{r}^{\iota} = (0, \mathbf{o}^{\iota}_a)_{a \in \mathrm{Ag}}$.

Model-checking problem for $\mathsf{EQCTL}^*_{\text{ii}}$. The *model-checking problem for* $\mathsf{EQCTL}^*_{\text{ii}}$ is the following: given an instance $(\mathcal{S}, \varphi)$ where $\mathcal{S}$ is a CKS and φ is an $\mathsf{EQCTL}^*_{\text{ii}}$ formula, return 'Yes' if $\mathcal{S} \models \varphi$ and 'No' otherwise.

Clearly, $\mathsf{EQCTL}^*_{\text{ii}}$ subsumes $\mathsf{QCTL}^*_{\text{ii}}$. Since the latter has an undecidable model-checking problem (Berthon et al. 2017), the following is immediate:

Theorem 3. *Model checking $\mathsf{EQCTL}^*_{\text{ii}}$ is undecidable.*

We now present the syntactic fragment for which we prove that model checking is decidable. First we adapt the notion of free epistemic formula to the context of $\mathsf{EQCTL}^*_{\text{ii}}$. Intuitively, an epistemic subformula φ of a formula Φ is free if it does not contain a free occurrence of a proposition quantified in Φ. To see the connection with the corresponding notion for ESL, consider that quantification on propositions will be used to capture quantification on strategies.

Definition 9. *Let $\Phi \in \mathsf{EQCTL}^*_{\text{ii}}$, and recall that we assume $\mathrm{AP}_{\exists}(\Phi) \cap \mathrm{AP}_f(\Phi) = \emptyset$. An epistemic subformula $\varphi = K_a\varphi'$ of Φ is* free in Φ *if $\mathrm{AP}_{\exists}(\Phi) \cap \mathrm{AP}_f(\varphi) = \emptyset$.*

For instance, if $\Phi = \exists^{\mathrm{o}} p.\, (K_a p) \wedge K_a q$, then subformula $K_a q$ is free in Φ, but subformula $K_a p$ is not because p is quantified in Φ and appears free in $K_a p$.

Definition 10 (Hierarchical formulas)**.** *An $\mathsf{EQCTL}^*_{\text{ii}}$ formula Φ is* hierarchical *if all its epistemic subformulas are free in Φ, and for all subformulas φ_1, φ_2 of the form $\varphi_1 = \exists^{\mathrm{o}_1} p_1.\, \varphi'_1$ and $\varphi_2 = \exists^{\mathrm{o}_2} p_2.\, \varphi'_2$ where φ_2 is a subformula of φ'_1, we have $\mathrm{o}_1 \subseteq \mathrm{o}_2$.*

In other words, a formula is hierarchical if epistemic subformulas are free, and innermost propositional quantifiers observe at least as much as outermost ones. Note that this is very close to hierarchical formulas of ESL. We let $\mathsf{EQCTL}^*_{\text{ii},\subseteq}$ be the set of hierarchical $\mathsf{EQCTL}^*_{\text{ii}}$ formulas.

Theorem 4. *The model-checking problem for $\mathsf{EQCTL}^*_{\text{ii},\subseteq}$ is non-elementary decidable.*

Proof sketch. We build upon the tree automata construction for $\mathsf{QCTL}^*_{\text{ii}}$ presented in (Berthon et al. 2017), which we extend to take into account knowledge operators and observation change. To do so we resort to the k-trees machinery developed in (van der Meyden 1998; van der Meyden and Shilov 1999), and extended in (Barrière et al. 2018) to the case of dynamic observation change. One also needs to observe that free epistemic subformulas can be evaluated indifferently in any node of the input tree. ☐

Model-checking hierarchical ESL

In this section we prove that model checking hierarchical instances of ESL is decidable (Theorem 2), by reduction to the model-checking problem for $\mathsf{QCTL}^*_{\text{i},\subseteq}$.

Let $(\Phi, \mathcal{G})$ be a hierarchical instance of the ESL model-checking problem. The construction of the CKS is the same as in (Berthon et al. 2017), except that in addition we have to deal with initial observations.

Constructing the CKS $\mathcal{S}_{\mathcal{G}}$. Let $\mathcal{G} = (\mathrm{Ac}, V, E, \ell, \mathcal{O}, v^{\iota}, \boldsymbol{o}^{\iota})$ and $\mathrm{Obs} = \{o_1, \dots, o_n\}$. For $i \in [n]$, define the local states $L_i := \{[v]_{o_i} \mid v \in V\}$ where $[v]_o$ is the equivalence class of v for relation $\sim_o$. We also let $L_{n+1} := V$. Finally, let $\mathrm{AP}_v := \{p_v \mid v \in V\}$ be a set of fresh atomic propositions, disjoint from AP.

Define the CKS $\mathcal{S}_{\mathcal{G}} := (S, R, \ell', s^{\iota}, \mathbf{o}^{\iota})$ where

- $S := \{s_v \mid v \in V\}$, where $s_v := ([v]_{o_1}, \dots, [v]_{o_n}, v) \in \prod_{i \in [n+1]} L_i$.
- $R := \{(s_v, s_{v'}) \mid \exists \boldsymbol{c} \in \mathrm{Ac}^{\mathrm{Ag}} \text{ s.t. } E(v, \boldsymbol{c}) = v'\} \subseteq S^2$,
- $\ell'(s_v) := \ell(v) \cup \{p_v\} \subseteq \mathrm{AP} \cup \mathrm{AP}_v$,
- $s_{\iota} := s_{v_{\iota}}$,
- $\mathbf{o}^{\iota}$ is such that $\mathbf{o}^{\iota}_a = \{i\}$ if $\boldsymbol{o}^{\iota}_a = o_i$.

For every $\rho = v_0 \dots v_k$, we let $u_{\rho} := s_{v_0} \dots s_{v_k}$. The mapping $\rho \mapsto u_{\rho}$ is a bijection between finite plays in $\mathcal{G}$ and nodes in $t_{\mathcal{S}_{\mathcal{G}}}$. For $i \in [n]$ we let $\mathrm{o}_i = \{i\}$, and for an observation record $\boldsymbol{r}$ in $\mathcal{G}$ we let $\boldsymbol{r}'$ be the observation record in $\mathcal{S}_{\mathcal{G}}$ where each o_i is replaced with o_i.

Constructing the $\mathsf{EQCTL}^*_{\text{ii},\subseteq}$ formulas $(\varphi)^f$. Suppose that $\mathrm{Ac} = \{c_1, \dots, c_l\}$; let $\mathrm{AP}_c := \{p_c^x \mid c \in \mathrm{Ac} \text{ and } x \in \mathrm{Var}\}$

be a set of propositions disjoint from $\mathrm{AP} \cup \mathrm{AP}_v$. For every partial function $f : \mathrm{Ag} \rightharpoonup \mathrm{Var}$ we define $(\varphi)^f$ by induction on φ. All cases for boolean, temporal and knowledge operators are obtained by simply distributing over the operators of the logic; for instance, $(p)^f = p$ and $(K_a\varphi)^f = K_a(\varphi)^f$. We now describe the translation for the remaining cases.

$$(\langle\!\langle x\rangle\!\rangle^o\varphi)^f := \exists^{\widetilde{o}} p^x_{c_1} \dots \exists^{\widetilde{o}} p^x_{c_l}. \varphi_{\text{str}}(x) \wedge (\varphi)^f$$

where $\widetilde{o_i} = \{j \mid \mathcal{O}(o_i) \subseteq \mathcal{O}(o_j)\}$, and

$$\varphi_{\text{str}}(x) = \mathbf{AG} \bigvee_{c\in\mathrm{Ac}} (p^x_c \wedge \bigwedge_{c'\neq c} \neg p^x_{c'}).$$

Note that $\mathrm{o}_i = \{i\} \subseteq \widetilde{o_i}$. The definition of $\widetilde{o_i}$ is tailored to obtain a hierarchical $\mathsf{EQCTL}^*_{\text{ii}}$ formula. It is correct because for each additional component in $\widetilde{o_i}$ (i.e., each $j \neq i$), we have $\mathcal{O}(o_i) \subseteq \mathcal{O}(o_j)$, meaning that each such component j brings less information than component i. A strategy thus has no more information with $\widetilde{o_i}$ than it would with o_i.

For the binding operator, agent a's observation becomes the one associated with strategy variable x (see Remark 1):

$$((a,x)\varphi)^f := \Delta^{\mathrm{o}_i}_a(\varphi)^{f[a\mapsto x]} \quad \text{if } o_x = o_i.$$

For the outcome quantifier, we let

$$(\mathbf{A}\psi)^f := \mathbf{A}(\psi_{\text{out}}(f) \to (\psi)^f), \quad \text{where}$$

$$\psi_{\text{out}}(f) = \mathbf{G} \bigwedge_{v\in V} \Big(p_v \to \bigvee_{\boldsymbol{c}\in\mathrm{Ac}^{\mathrm{Ag}}} (\bigwedge_{a\in dom(f)} p^{f(a)}_{\boldsymbol{c}_a} \wedge \mathbf{X} p_{E(v,\boldsymbol{c})})\Big)$$

The formula $\psi_{\text{out}}(f)$ selects paths in which agents who are assigned to a strategy follow it.

Proposition 5. *Suppose that $free(\varphi) \cap \mathrm{Ag} \subseteq dom(f)$, that for all $a \in dom(f)$, $f(a) = x$ iff $\chi(a) = \chi(x)$, and that $\boldsymbol{r}$ stops at ρ. Then*

$$\mathcal{G}, \chi, \boldsymbol{r}, \rho \models \varphi \quad \textit{if and only if} \quad t_{\mathcal{S}_\mathcal{G}}, \boldsymbol{r}', u_\rho \models (\varphi)^f.$$

Applying this to sentence Φ, any assignment χ, $f = \emptyset$, $\rho = v_\iota$ and initial observation records, we get that

$$\mathcal{G} \models \Phi \quad \text{if and only if} \quad t_{\mathcal{S}_\mathcal{G}} \models (\Phi)^\emptyset.$$

Preserving hierarchy. To complete the proof of Theorem 2 we show that $(\Phi)^\emptyset$ is a hierarchical $\mathsf{EQCTL}^*_{\text{ii}}$ formula.

First, observe that if $K_a\varphi$ is a free epistemic formula in Φ, then its translation is also a free epistemic formula in $(\Phi)^\emptyset$. Indeed, the only atomic propositions that are quantified in $(\Phi)^\emptyset$ are of the form p^x_c. They code for strategies, and appear only in translations of strategy quantifiers, where they are quantified upon, and outcome quantifiers. Thus they can only appear free in the translation of an epistemic formula $K_a\varphi$ if φ contains an outcome quantifier where some agent uses a strategy that is not quantified within φ. Concerning the hierarchy on observations of quantifiers, simply observe that Φ is hierarchical in $\mathcal{G}$, and for every two observations o_i and o_j in Obs such that $\mathcal{O}(o_i) \subseteq \mathcal{O}(o_j)$, by definition of $\widetilde{o_k}$ we have that $\widetilde{o_i} \subseteq \widetilde{o_j}$.

Applications

ESL being very expressive, many strategic problems with epistemic temporal specifications can be cast as model-checking problems for ESL. Our main result thus provides a decision procedure for such problems on systems with hierarchical information. We present two such applications.

Distributed synthesis

We consider the problem of distributed synthesis from epistemic temporal specifications studied in (van der Meyden and Vardi 1998; van der Meyden and Wilke 2005) and we give a precise definition to its variant with uninformed semantics of knowledge, discussed in (Puchala 2010).

Assume that $\mathrm{Ag} = \{a_1, \dots, a_n, e\}$, where e is a special player called the *environment*. Assume also that to each player a_i is assigned an observation symbol o_i. The above-mentioned works consider specifications from linear-time epistemic temporal logic LTLK, which extends LTL with knowledge operators. The semantics of knowledge operators contains an implicit universal quantification on continuations of indistinguishable finite plays. In (van der Meyden and Vardi 1998; van der Meyden and Wilke 2005), which considers the informed semantics of knowledge, i.e., where all players know each other's strategy, this quantification is restricted to continuations that follow these strategies; in (Puchala 2010), which considers the uninformed semantics, it quantifies over all possible continuations in the game.

We now prove a stronger result than the one announced in (Puchala 2010), by allowing the use of either existential or universal quantification on possible continuations after a knowledge operator. For an ESL path formula ψ, we define

$$\Phi_{\text{syn}}(\psi) := \langle\!\langle x_1\rangle\!\rangle^{o_1} \dots \langle\!\langle x_n\rangle\!\rangle^{o_n} (a_1, x_1) \dots (a_n, x_n) \mathbf{A}\psi.$$

Note that the outcome quantifier $\mathbf{A}$ quantifies on all possible behaviours of the environment.

Definition 11. *The epistemic distributed synthesis problem with uninformed semantics is the following: given a CGS_{ii} $\mathcal{G}$ and an ESL path formula ψ, decide whether $\mathcal{G} \models \Phi_{syn}(\psi)$.*

Let LTLK ($\mathsf{CTL}^*\mathsf{K}$) the set of path formulas obtained by allowing in LTL subformulas of the form $K_a\varphi$, with $\varphi \in \mathsf{CTL}^*\mathsf{K}$. The path quantifier from $\mathsf{CTL}^*\mathsf{K}$ quantifies on all possible futures, and is simulated in ESL by an unbinding for all players followed by an outcome quantifier. Therefore with specifications ψ in LTLK ($\mathsf{CTL}^*\mathsf{K}$), all epistemic subformulas are free. It follows that if the system $\mathcal{G}$ is hierarchical, and we assume without loss of generality that $\mathcal{O}(o_i) \supseteq \mathcal{O}(o_{i+1})$, then $(\mathcal{G}, \Phi_{\text{syn}}(\psi))$ is a hierarchical instance of ESL.

Theorem 6. *The epistemic distributed synthesis problem from specifications in LTLK ($\mathsf{CTL}^*\mathsf{K}$) with uninformed semantics is decidable on hierarchical systems.*

In fact we can deal with even richer specifications: as long as hierarchy is not broken and epistemic subformulas remain free, it is possible to re-quantify on agents' strategies inside an epistemic formula. Take for instance formula

$$\psi = \mathbf{F} K_{a_n}(a_1, ?) \dots (a_n, ?) \langle\!\langle x\rangle\!\rangle^{o_n} (a_n, x) \mathbf{AG} K_{a_n} p$$

It says that eventually, agent a_n knows that she can change strategy so that in all outcomes of this strategy, she will always know that p holds. If $\mathcal{G}$ is hierarchical, then $\Phi_{\text{syn}}(\psi)$ forms a hierarchical instance with $\mathcal{G}$. Consider now formula

$$\psi = \mathbf{F}K_{a_i}(a_1, ?) \dots (a_n, ?)[\![x]\!]^o(a_j, x)\mathbf{EG}\neg K_{a_j}p$$

which means that eventually agent a_i knows that for any strategy with observation o that agent a_j may take, there is an outcome in which a_j never knows that p holds. If o is finer than o_n (and thus all other o_i), for instance if o represents perfect information, then hierarchy is preserved and we can solve distributed synthesis for this specification. In addition, the semantics of our knowledge operator takes into account the fact that agent a_j changes observation power.

Rational synthesis

Consider $\text{Ag} = \{a_1, \dots, a_n\}$, each player a_i having observation symbol o_i. Given a global objective ψ_g and individual objectives ψ_i for each player a_i, define

$$\Phi_{\text{rat}} := \langle\!\langle x_1 \rangle\!\rangle^{o_1} \dots \langle\!\langle x_n \rangle\!\rangle^{o_n}(a_1, x_1) \dots (a_n, x_n)\mathbf{A}\psi_g \wedge \bigwedge_{i \in \{1,\dots,n\}} \Big([\![x_i]\!]^{o_i}\left((a_i, x_i)\,\mathbf{A}\psi_i\right) \to \mathbf{A}\psi_i \Big).$$

It is easy to see that Φ_{rat} expresses the existence of a solution to the cooperative rational synthesis problem (Kupferman, Perelli, and Vardi 2016; Condurache et al. 2016). However this formula does not form hierarchical instances, even with hierarchical systems. But the same argument used in (Berthon et al. 2017) for Nash equilibria shows that Φ_{rat} is equivalent to Φ'_{rat}, obtained from Φ_{rat} by replacing each $[\![x_i]\!]^{o_i}$ with $[\![x_i]\!]^{o_p}$, where o_p represents perfect observation.

Theorem 7. *Rational synthesis from* LTL *(*CTL*K*) specifications is decidable on hierarchical systems.*

As in the case of distributed synthesis discussed before, we can in fact handle more complex specifications, nesting knowledge and strategy quantification.

Discussion

In the uninformed semantics, players ignore each other's strategy, but they also ignore their own one, in the sense that they consider possible finite plays in which they act differently from what their strategy prescribes. This is the usual semantics in epistemic strategic logics (van der Hoek and Wooldridge 2003; Jamroga and van der Hoek 2004; Belardinelli 2015; Dima, Enea, and Guelev 2010; Belardinelli et al. 2017a; 2017b), and in some situations it may be what one wants to model. For instance, an agent may execute her strategy step by step without having access to what her strategy would have prescribed in alternative plays. In this case, it is not possible for the agent to know whether a possible play follows her strategy or not, and thus the uninformed semantics of knowledge is the right one.

On the other hand it seems natural, especially formulated in these terms, to assume that an agent knows her own strategy. We describe how, in the case where agents do not change strategies or observation along time, this semantics can be retrieved within the uninformed semantics.

Assume that player a is assigned some observation symbol o_a. As pointed out in (Puchala 2010, p.16), in the setting of synchronous perfect recall, letting a player know her strategy is equivalent to letting her remember her own actions. To see this, assume that finite plays also contain each joint action between two positions, and let $\sim'_{o_a}$ be such that $v_\iota \boldsymbol{c}_1 v_1 \dots \boldsymbol{c}_n v_n \sim'_{o_a} v_\iota \boldsymbol{c}_1 v_1 \dots \boldsymbol{c}_n v_n$ if for all $i \in \{1, \dots, n\}$, $\boldsymbol{c}_a = \boldsymbol{c}'_a$ and $v_i \sim_{o_a} v'_i$. Then, for a strategy σ of player a and two finite plays ρ, ρ' such that $\rho \sim'_{o_a} \rho'$, it holds that ρ is consistent with a playing σ iff ρ' is consistent with a playing σ. This is because for every $i < n$ we have $\rho_{\leq i} \sim'_{o_a} \rho'_{\leq i}$ (perfect recall), the next action taken by player a is the same after $\rho'_{\leq i}$ and $\rho'_{\leq i}$ (definition of $\sim'_{o_a}$), and σ being an o_a-strategy it is defined similarly on both prefixes.

In our setting, moves are not part of finite plays. To simulate the relation $\sim'_{o_a}$ in which agent a remembers her own actions, one can put inside the positions of game structures the information of the last joint move played, possibly duplicating some positions. One then refines each observation o_a to only consider two positions equivalent if they contain the same move for player a. We then get a semantics where each agent remembers her own actions which, if agents do not change strategy or observation through time, is equivalent to knowing her own strategy. Note that doing so, a system can only be hierarchical if more informed players also observe all actions of less informed ones.

In the general case, where players can change strategies and observations, we do not know to what extent we can deal with the variant of the uninformed semantics where players know there own strategies. We leave this for future work.

Conclusion

In this paper we have discussed two possible semantics of knowledge when combined with strategies, the informed and uninformed one. Focusing on the latter, we introduced ESL, a very expressive logic to reason about knowledge and strategies in distributed systems which can handle sophisticated epistemic variants of game-theoretic notions such as Nash equilibria. In addition, it is the first logic of knowledge and strategies that permits reasoning about agents whose observation power may change. This is a very natural phenomenon: one may think of a program that receives access to previously hidden variables, or a robot that loses a sensor.

We solved the model-checking problem of our logic for hierarchical instances. To do so, we introduced an extension of QCTL* with epistemic operators and operators of observation change, and we developed an automata construction based on tree automata and k-trees. This is the first decidability result for a logic of strategies, knowledge and time with perfect recall on systems with hierarchical information. Besides, it is also the first result for epistemic strategic logics that takes into account dynamic changes of observation.

Our result implies that distributed synthesis and rational synthesis for epistemic temporal specifications and the uninformed semantics of knowledge are decidable on hierarchical systems. Similar results for other solution concepts, such as subgame-perfect equilibria or admissible strategies (Brenguier et al. 2017), could be obtained similarly.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 709188.

References

Alur, R.; Henzinger, T. A.; and Kupferman, O. 2002. Alternating-time temporal logic. *JACM* 49(5):672–713.

Barrière, A.; Maubert, B.; Murano, A.; and Rubin, S. 2018. Changing observations in epistemic temporal logic. *arXiv*.

Belardinelli, F.; Lomuscio, A.; Murano, A.; and Rubin, S. 2017a. Verification of broadcasting multi-agent systems against an epistemic strategy logic. In *IJCAI'17*, 91–97.

Belardinelli, F.; Lomuscio, A.; Murano, A.; and Rubin, S. 2017b. Verification of multi-agent systems with imperfect information and public actions. In *AAMAS'17*, 1268–1276.

Belardinelli, F. 2015. A logic of knowledge and strategies with imperfect information. In *LAMAS'15*, 1–15.

Berthon, R.; Maubert, B.; Murano, A.; Rubin, S.; and Vardi, M. Y. 2017. Strategy Logic with imperfect information. In *LICS'17*.

Bozzelli, L.; Maubert, B.; and Pinchinat, S. 2015. Uniform strategies, rational relations and jumping automata. *Inf. Comput.* 242:80–107.

Brafman, R. I.; Latombe, J.-C.; Moses, Y.; and Shoham, Y. 1997. Applications of a logic of knowledge to motion planning under uncertainty. *Journal of the ACM (JACM)* 44(5):633–668.

Brenguier, R.; Pauly, A.; Raskin, J.; and Sankur, O. 2017. Admissibility in games with imperfect information (invited talk). In *CONCUR'17*, 2:1–2:23.

Cermák, P.; Lomuscio, A.; Mogavero, F.; and Murano, A. 2014. MCMAS-SLK: A model checker for the verification of strategy logic specifications. In *CAV'14*, 525–532.

Chatterjee, K.; Henzinger, T. A.; and Piterman, N. 2010. Strategy Logic. *Inf. Comput.* 208(6):677–693.

Condurache, R.; Filiot, E.; Gentilini, R.; and Raskin, J. 2016. The complexity of rational synthesis. In *ICALP'16*, 121:1–121:15.

Dima, C.; Enea, C.; and Guelev, D. P. 2010. Model-checking an Alternating-time Temporal Logic with knowledge, imperfect information, perfect recall and communicating coalitions. *EPTCS* 25.

Emerson, E. A., and Sistla, A. P. 1984. Deciding branching time logic. In *STOC'84*, 14–24.

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1995. *Reasoning about knowledge*. MIT press Cambridge.

French, T. 2001. Decidability of quantifed propositional branching time logics. In *AJCAI'01*, 165–176.

Halpern, J. Y., and Moses, Y. 1990. Knowledge and common knowledge in a distributed environment. *Journal of the ACM (JACM)* 37(3):549–587.

Halpern, J. Y., and O'Neill, K. R. 2005. Anonymity and information hiding in multiagent systems. *Journal of Computer Security* 13(3):483–512.

Huang, X., and Van Der Meyden, R. 2014. A temporal logic of strategic knowledge. In *KR*.

Jamroga, W., and van der Hoek, W. 2004. Agents that know how to play. *Fundam. Inform.* 63(2-3):185–219.

Knight, S., and Maubert, B. 2015. Dealing with imperfect information in strategy logic. In *SR*.

Kupferman, O.; Madhusudan, P.; Thiagarajan, P. S.; and Vardi, M. Y. 2000. Open systems in reactive environments: Control and synthesis. In *CONCUR'00*, 92–107.

Kupferman, O.; Perelli, G.; and Vardi, M. Y. 2016. Synthesis with rational environments. *Ann. Math. Artif. Intell.* 78(1):3–20.

Kupferman, O. 1995. Augmenting branching temporal logics with existential quantification over atomic propositions. In *CAV'95*, 325–338.

Ladner, R. E., and Reif, J. H. 1986. The Logic of Distributed Protocols. In *TARK'86*, 207–222.

Laroussinie, F., and Markey, N. 2014. Quantified CTL: expressiveness and complexity. *LMCS* 10(4).

Laroussinie, F., and Markey, N. 2015. Augmenting ATL with strategy contexts. *IC* 245:98–123.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. Y. 2014. Reasoning about strategies: On the model-checking problem. *TOCL* 15(4):34:1–34:47.

Neiger, G., and Bazzi, R. 1992. Using knowledge to optimally achieve coordination in distributed systems. In *TARK'92*, 43–59.

Peterson, G. L., and Reif, J. H. 1979. Multiple-person alternation. In *SFCS'79*, 348–363.

Pnueli, A., and Rosner, R. 1990. Distributed reactive systems are hard to synthesize. In *FOCS'90*, 746–757.

Puchala, B. 2010. Asynchronous omega-regular games with partial information. In *MFCS'10*, 592–603.

Reif, J. H. 1984. The complexity of two-player games of incomplete information. *JCSS* 29(2):274–301.

van der Hoek, W., and Wooldridge, M. 2003. Cooperation, knowledge and time: Alternating-time Temporal Epistemic Logic and its applications. *Studia Logica* 75(1):125–157.

van der Meyden, R., and Shilov, N. V. 1999. Model checking knowledge and time in systems with perfect recall (extended abstract). In *FSTTCS*, 432–445.

van der Meyden, R., and Su, K. 2004. Symbolic model checking the knowledge of the dining cryptographers. In *CSFW'17*, 280–291.

van der Meyden, R., and Vardi, M. Y. 1998. Synthesis from knowledge-based specifications. In *CONCUR'98*. 34–49.

van der Meyden, R., and Wilke, T. 2005. Synthesis of distributed systems from knowledge-based specifications. In *CONCUR'05*, 562–576.

van der Meyden, R. 1998. Common knowledge and update in finite environments. *Inf. Comput.* 140(2):115–157.