Strategy Logic with Imperfect Information

Raphaël Berthon
École Normale Supérieure de Rennes
Rennes, France
raphael.berthon@ens-rennes.fr

Bastien Maubert
University of Naples "Federico II"
Naples, Italy
bastien.maubert@gmail.com

Aniello Murano
University of Naples "Federico II"
Naples, Italy
murano@na.infn.it

Sasha Rubin
University of Naples "Federico II"
Naples, Italy
sasha.rubin@unina.it

Moshe Vardi
Rice University
Houston, Texas, U.S.A.
vardi@cs.rice.edu

We introduce an extension of Strategy logic for the imperfect-information setting, called $\mathsf{SL}_{\mathrm{ii}}$, and study its model-checking problem. As this logic naturally captures multi-player games with imperfect information, the problem turns out to be undecidable. We introduce a syntactical class of "hierarchical instances" for which, intuitively, as one goes down the syntactic tree of the formula, strategy quantifications are concerned with finer observations of the model. We prove that model-checking $\mathsf{SL}_{\mathrm{ii}}$ restricted to hierarchical instances is decidable. This result, because it allows for complex patterns of existential and universal quantification on strategies, greatly generalises previous ones, such as decidability of multi-player games with imperfect information and hierarchical observations, and decidability of distributed synthesis for hierarchical systems.

1 Introduction

Logics for strategic reasoning are a powerful tool for expressing correctness properties of multi-player graph-games, which in turn are natural models for reactive systems and discrete event systems. In particular, ATL^* and its related logics were introduced to capture the realisability/synthesis problem for open systems with multiple components. These logics were designed as extensions of branching-time logics such as CTL^* that allow one to write alternating properties directly in the syntax, i.e., statements of the form "there exist strategies, one for each player in A, such that for all strategies of the remaining players, the resulting play satisfies φ". Strategy logic (SL) [26] generalises these by treating strategies as first-order objects x that can be quantified $\langle\langle x\rangle\rangle$ (read "there exists a strategy x") and bound to players (a,x) (read "player a uses strategy x"). This syntax has flexibility very similar to first-order logic, and thus allows one to directly express many solution concepts from game-theory, e.g., the SL formula $\langle\langle x_1\rangle\rangle\langle\langle x_2\rangle\rangle(a_1,x_1)(a_2,x_2)\wedge_{i=1,2}[\langle\langle y_i\rangle\rangle(a_i,y_i)goal_i]\to goal_i$ expresses the existence of a Nash equilibrium in a two-player game (with individual Boolean objectives).

An essential property of realistic multi-player games is that players only have a limited view of the state of the system. This is captured by introducing partial-observability into the models, i.e., equivalence-relations o (called *observations*) over the state space that specify indistinguishable states. In the formal-methods literature it is typical to associate observations to players. In this paper, instead, we associate observations to strategies. Concretely, we introduce an extension $\mathsf{SL}_{\mathrm{ii}}$ of SL that annotates the strategy quantifier $\langle\langle x\rangle\rangle$ by an observation o, written $\langle\langle x\rangle\rangle^o$. Thus, both the model and the formulas mention observations o. This novelty allows one to express, in the logic, that a player's observation changes over time.

Submitted to:
SR 2018

Our logic $\mathsf{SL}_{\mathrm{ii}}$ is very powerful: it extends SL and the imperfect-information strategic logics $\mathsf{ATL}^*_{\mathrm{i,R}}$ [5] and $\mathsf{ATL}^*_{\mathrm{sc,i}}$ [23]. A canonical specification in multi-player games of partial observation is that the players, say $a_1,\dots,a_n$, can form a coalition and beat the environment player, say b. This can be expressed in $\mathsf{SL}_{\mathrm{ii}}$ as $\Phi_{\mathrm{SYNTH}} := \langle\langle x_1\rangle\rangle^{o_1}\dots\langle\langle x_n\rangle\rangle^{o_n}[[y]]^{o}(a_1,x_1)\dots(a_n,x_n)(b,y)\varphi$, where φ is quantifier- and binding-free. Also, $\mathsf{SL}_{\mathrm{ii}}$ can express more complicated specifications by alternating quantifiers, binding the same strategy to different agents and rebinding (these are inherited from SL), as well as changing observations.

The complexity of $\mathsf{SL}_{\mathrm{ii}}$ is also visible from an algorithmic point of view. Its satisfiability problem is undecidable (this is already true of SL), and its model-checking problem is undecidable (this is already true of $\mathsf{ATL}^*_{\mathrm{i,R}}$, even for the single formula $\langle\{a,b\}\rangle\mathbf{F}p$ [8]). In fact, similar undecidability occurs in foundational work in multi-player games of partial observation, and in distributed synthesis [31, 29]. Since then, the formal-methods community has spent much effort finding restrictions and variations that ensure decidability [17, 28, 12, 30, 32, 7, 3, 2, 4]. The common thread in these approaches is that the players' observations (or what they can deduce from their observations) are hierarchical.

Motivated by the problem of finding decidable extensions of strategy logic in the imperfect-information setting, we introduce a syntactic class of "hierarchical instances" of $\mathsf{SL}_{\mathrm{ii}}$, i.e., formula/model pairs, and prove that the model-checking problem on this class of instances is decidable. Intuitively, an instance of $\mathsf{SL}_{\mathrm{ii}}$ is hierarchical if, as one goes down the syntactic tree of the formula, the observations annotating strategy quantifications can only become finer. Although the class of hierarchical instances refers not only to the syntax of the logic but also to the model, the class is syntactical in the sense that it depends only on the structure of the formula and the observations in the model.

The class of hierarchical instances generalises some existing approaches and supplies new classes of systems and properties that can be model-checked. For instance, suppose that there is a total order $\preceq$ among the players such that $a \preceq b$ implies player b's observation is finer than player a's observation — such games are said to yield "hierarchical observation" in [4]. In such games it is known that synthesis for ω-regular specifications is decidable [28, 4]). This corresponds to hierarchical instances of $\mathsf{SL}_{\mathrm{ii}}$ in which the observations form a total order in the model and the formula is of the form Φ_{SYNTH} above. On the other hand, in hierarchical instances of $\mathsf{SL}_{\mathrm{ii}}$, the ordering on observations can be a pre partial-order, and one can arbitrarily alternate quantifiers in the formulas. For instance, hierarchical instances allow one to decide if a game that yields hierarchical information has a Nash equilibrium.

As a tool to study $\mathsf{SL}_{\mathrm{ii}}$ we introduce $\mathsf{QCTL}^*_{\mathrm{ii}}$, an extension to the imperfect-information setting of QCTL^* [21], itself an extension of CTL^* by second-order quantifiers over atoms. This is a low-level logic that does not mention strategies and into which one can effectively compile instances of $\mathsf{SL}_{\mathrm{ii}}$. States of the models of the logic $\mathsf{QCTL}^*_{\mathrm{ii}}$ have internal structure, much like the multi-player game structures from [27] and distributed systems [16]. Model-checking $\mathsf{QCTL}^*_{\mathrm{ii}}$ is also undecidable (indeed, we show how to reduce from the MSO-theory of the binary tree extended with the equal-length predicate, known to be undecidable [24]). We introduce the syntactical class $\mathsf{QCTL}^*_{\mathrm{i},\subseteq}$ of hierarchical formulas as those in which innermost quantifiers observe more than outermost quantifiers, and prove that model-checking is decidable using an extension of the automata-theoretic approach for branching-time logics (our decision to base models of $\mathsf{QCTL}^*_{\mathrm{ii}}$ on local states greatly eases the use of automata). Moreover, the compilation of $\mathsf{SL}_{\mathrm{ii}}$ into $\mathsf{QCTL}^*_{\mathrm{ii}}$ preserves being hierarchical, thus establishing our main contribution, i.e., that model checking the hierarchical instances of $\mathsf{SL}_{\mathrm{ii}}$ is decidable.

Related work. Formal methods for reasoning about reactive systems with multiple components have been studied mainly in two theoretical frameworks: a) multi-player graph-games of partial-observation [29, 28, 4] and b) synthesis in distributed architectures [31, 17, 12, 32, 15] (the relationship between these two frameworks is discussed in [4]). All of these works consider the problem of synthesis, which (for objectives in temporal logics) can be expressed in $\mathsf{SL}_{\mathrm{ii}}$ using the formula Φ_{SYNTH} mentioned above. Limited alternation was studied in [7] that, in the language of SL_{ii}, considers the model-checking problem of formulas of the form $\langle\langle x_1\rangle\rangle^{o_1}[\![x_2]\!]^{o_2}\langle\langle x_3\rangle\rangle^{o_3}(a_1,x_1)(a_2,x_2)(a_3,x_3)\varphi$, where φ is an ω-regular objective. They prove that this is decidable in case player 3 has perfect observation and player 2 observes at least as much as player 1.

In contrast to all these works, formulas of SL_{ii} can express much more complex specifications by alternating quantifiers, sharing strategies, rebinding, and changing observations.

We are aware of two papers that (like we do) give simultaneous structural constraints on both the formula and the model that result in decidability: in the context of synthesis in distributed architecture with process delays, [15] considers CTL^* specifications that constrain external variables by the input variables that may effect them in the architecture; and in the context of asynchronous perfect-recall, [30] considers a syntactical restriction on instances for Quantified μ-Calculus with partial observation (in contrast, we consider the case of synchronous perfect recall).

The work closest to ours is [13] which introduces a decidable logic CL in which one can encode many distributed synthesis problems. However, CL is close in spirit to our $\mathsf{QCTL}^*_{\text{i},\subseteq}$, and is more appropriate as a tool than as a high-level specification logic like SL_{ii}. Furthermore, by means of a natural translation we derive that CL is strictly included in the hierarchical instances of SL_{ii} (Section 2.5). In particular, we find that hierarchical instances of SL_{ii} can express non-observable goals, while CL does not. Non-observable goals arise naturally in problems in distributed synthesis [31].

Finally, our logic SL_{ii} is the first generalisation of SL to include strategies with partial observation and, unlike CL, to generalise previous logics with partial-observation strategies, i.e., $\mathsf{ATL}^*_{\text{i,R}}$ [5] and $\mathsf{ATL}^*_{\text{sc,i}}$ [23]. A comparison of SL_{ii} to SL, $\mathsf{ATL}^*_{\text{i,R}}$, $\mathsf{ATL}^*_{\text{sc,i}}$ and CL is given in Section 2.5.

Outline. The definition of SL_{ii} and of hierarchical instances, and the discussion about Nash equilibria, are in Section 2. The definition of $\mathsf{QCTL}^*_{\text{ii}}$ and its hierarchical fragment $\mathsf{QCTL}^*_{\text{i},\subseteq}$ are in Section 3, together with the decidability result for model checking $\mathsf{QCTL}^*_{\text{i},\subseteq}$. The translation of SL_{ii} into $\mathsf{QCTL}^*_{\text{ii}}$, and the fact that this preserves hierarchy, are in Section 4.

2 SL with imperfect information

In this section we introduce SL_{ii}, an extension of SL [26] to the imperfect-information setting with synchronous perfect-recall. Our logic presents two original features: first, observations are not bound to players (as is done in imperfect information extensions of ATL [1] or logics for reasoning about knowledge [11]), and second, we have syntactic observations in the language, which need to be interpreted.

2.1 Notation

Let Σ be an alphabet. A *finite* (resp. *infinite*) *word* over Σ is an element of Σ^* (resp. Σ^ω). Words are written $w = w_0w_1w_2\ldots$, i.e., indexing begins with 0. The *length* of a finite word $w = w_0w_1\ldots w_n$ is $|w| := n+1$, and $\text{last}(w) := w_n$ is its last letter. 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 $[n] := \{i \in \mathbb{N} : 1 \leq i \leq n\}$. The literature sometimes refers to "imperfect information" and sometimes to "partial observation"; we will use the terms interchangeably.

2.2 Syntax

The syntax of $\mathsf{SL}_{\mathrm{ii}}$ is similar to that of strategy logic SL as defined in [26]: the only difference is that we annotate strategy quantifiers $\langle\langle x\rangle\rangle$ by *observation symbols* o. For the rest of the paper, for convenience we fix 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*. When we consider model-checking problems, these data are implicitly part of the input.

Definition 1 ($\mathsf{SL}_{\mathrm{ii}}$ Syntax)**.** *The syntax of* $\mathsf{SL}_{\mathrm{ii}}$ *is defined by the following grammar:*

$$\varphi := p \mid \neg\varphi \mid \varphi\vee\varphi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \langle\langle x\rangle\rangle^{o}\varphi \mid (a,x)\varphi$$

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

We use abbreviations $\top := p\vee\neg p$, $\bot := \neg\top$, $\varphi\to\varphi' := \neg\varphi\vee\varphi'$, $\varphi\leftrightarrow\varphi' := \varphi\to\varphi'\wedge\varphi'\to\varphi$ for boolean connectives, $\mathbf{F}\varphi := \top\mathbf{U}\varphi$, $\mathbf{G}\varphi := \neg\mathbf{F}\neg\varphi$ for temporal operators, and finally $[\![x]\!]^{o}\varphi := \neg\langle\langle x\rangle\rangle^{o}\neg\varphi$.

The notion of free variables and sentences are defined as for SL: A variable x appears *free* in a formula φ if it appears out of the scope of a strategy quantifier, and a player a appears free in φ if a temporal operator (either $\mathbf{X}$ or $\mathbf{U}$) appears in φ out of the scope of any binding for player a. We let *free*(φ) be the set of variables and players that appear free in φ. If *free*(φ) is empty, φ is a *sentence*.

2.3 Semantics

The models of $\mathsf{SL}_{\mathrm{ii}}$ are like those of SL, i.e., concurrent game structures, but extended by a finite set of observations Obs and, for each $o \in \mathrm{Obs}$, by an equivalence-relation $O(o)$ over positions that represents what a player using a strategy with that observation can see. That is, $O(o)$-equivalent positions are indistinguishable to a player using a strategy associated with observation o.

Definition 2 ($\mathrm{CGS}_{\mathrm{ii}}$)**.** *A* concurrent game structure with imperfect information *(or CGS*$_{\mathrm{ii}}$ *for short) is a structure* $\mathcal{G} = (\mathrm{Ac}, V, E, \ell, v_\iota, O)$ *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*,* $v_\iota \in V$ *is an* initial position*, and* $O : \mathrm{Obs} \to 2^{V\times V}$ *is an* observation interpretation *mapping observations to equivalence relations on positions.*

We may write $\sim_o$ for $O(o)$, and $v \in \mathcal{G}$ for $v \in V$.

Joint actions. 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 $\boldsymbol{c} \in \mathrm{Ac}^{\mathrm{Ag}}$ is a *joint action* $(c_a)_{a\in\mathrm{Ag}}$. Given a joint action $\boldsymbol{c} = (c_a)_{a\in\mathrm{Ag}}$ and $a \in \mathrm{Ag}$, we let $\boldsymbol{c}_a$ denote c_a. For each position $v \in V$, $\ell(v)$ is the set of atomic propositions that hold in v.

Plays and strategies. A *finite* (resp. *infinite*) *play* is a finite (resp. infinite) word $\rho = v_0\ldots v_n$ (resp. $\pi = v_0v_1\ldots$) such that $v_0 = v_\iota$ and for all i with $0 \le i < |\rho|-1$ (resp. $i \ge 0$), 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 the set of all strategies is denoted *Str*.

Assignments. An *assignment* is a function $\chi : \mathrm{Ag}\cup\mathrm{Var} \to \mathit{Str}$, assigning a strategy to each player and variable. For an assignment χ, player a and strategy σ, $\chi[a\mapsto\sigma]$ is the assignment 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.

Outcomes. For an assignment χ and a finite play ρ, we let $\mathrm{out}(\chi,\rho)$ be the only infinite play that starts with ρ and is then extended by letting players follow the strategies assigned by χ. Formally, $\mathrm{out}(\chi,\rho) := \rho\cdot v_1v_2\ldots$ where, for all $i \ge 0$, $v_{i+1} = E(v_i,\boldsymbol{c})$, where $v_0 = \mathrm{last}(\rho)$ and $\boldsymbol{c} = (\chi(a)(\rho\cdot v_1\ldots v_i))_{a\in\mathrm{Ag}}$.

Synchronous perfect recall. In this work we consider players with *synchronous perfect recall*, meaning that each player remembers the whole history of a play, a classic assumption in games with imperfect information and logics of knowledge and time. Each observation relation is thus extended to finite plays as follows: $\rho \sim_o \rho'$ if $|\rho| = |\rho'|$ and $\rho_i \sim_o \rho'_i$ for every $i \in \{0,\dots,|\rho|-1\}$. For $o \in \text{Obs}$, an *o-strategy* is a strategy $\sigma : V^+ \to \text{Ac}$ such that $\sigma(\rho) = \sigma(\rho')$ whenever $\rho \sim_o \rho'$. The latter constraint captures the essence of imperfect information, which is that players can base their strategic choices only on the information available to them. For $o \in \text{Obs}$ we let Str_o be the set of all o-strategies.

Definition 3 (SL_{ii} Semantics)**.** *The semantics $\mathcal{G},\chi,\rho \models \varphi$ is defined inductively, where φ is an SL_{ii}-formula, $\mathcal{G} = (\text{Ac}, V, E, \ell, v_\iota, O)$ is a CGS$_{\text{ii}}$, ρ is a finite play, and χ is an assignment:*

$$
\begin{array}{lll}
\mathcal{G},\chi,\rho \models p & \textit{if} & p \in \ell(\mathit{last}(\rho)) \\
\mathcal{G},\chi,\rho \models \neg\varphi & \textit{if} & \mathcal{G},\chi,\rho \not\models \varphi \\
\mathcal{G},\chi,\rho \models \varphi \vee \varphi' & \textit{if} & \mathcal{G},\chi,\rho \models \varphi \textit{ or } \mathcal{G},\chi,\rho \models \varphi' \\
\mathcal{G},\chi,\rho \models \langle\!\langle x \rangle\!\rangle^{o}\varphi & \textit{if} & \exists \sigma \in \text{Str}_o \textit{ s.t. } \mathcal{G},\chi[x \mapsto \sigma],\rho \models \varphi \\
\mathcal{G},\chi,\rho \models (a,x)\varphi & \textit{if} & \mathcal{G},\chi[a \mapsto \chi(x)],\rho \models \varphi
\end{array}
$$

and, writing $\pi = \text{out}(\chi,\rho)$:

$$
\begin{array}{lll}
\mathcal{G},\chi,\rho \models \mathbf{X}\varphi & \textit{if} & \mathcal{G},\chi,\pi_{\leq|\rho|} \models \varphi \\
\mathcal{G},\chi,\rho \models \varphi\mathbf{U}\varphi' & \textit{if} & \exists i \geq 0 \textit{ s.t. } \mathcal{G},\chi,\pi_{\leq|\rho|+i-1} \models \varphi' \\
 & & \textit{and } \forall j \textit{ s.t. } 0 \leq j < i,\ \mathcal{G},\chi,\pi_{\leq|\rho|+j-1} \models \varphi
\end{array}
$$

Clearly, the satisfaction of a sentence is independent of the assignment. For an SL_{ii} sentence φ we thus let $\mathcal{G},\rho \models \varphi$ if $\mathcal{G},\chi,\rho \models \varphi$ for some assignment χ, and we write $\mathcal{G} \models \varphi$ if $\mathcal{G},v_\iota \models \varphi$.

2.4 Model checking and hierarchical instances

Model Checking. An SL_{ii}-*instance* is a formula/model pair $(\Phi,\mathcal{G})$ where $\Phi \in \mathsf{SL}_{\text{ii}}$ and $\mathcal{G}$ is a CGS$_{\text{ii}}$. The *model-checking problem* for SL_{ii} is the decision problem that, given an SL_{ii}-instance $(\Phi,\mathcal{G})$, returns 'yes' if $\mathcal{G} \models \Phi$, and 'no' otherwise.

It is well known that deciding the existence of winning strategies in multi-player games with imperfect information is undecidable for reachability objectives [27]. Since this problem is easily reduced to the model-checking problem for SL_{ii}, we get the following result:

Theorem 1. *The model-checking problem for* SL_{ii} *is undecidable.*

Hierarchical instances. We isolate a sub-problem obtained by restricting attention to *hierarchical instances*. Intuitively, an SL_{ii}-instance $(\Phi,\mathcal{G})$ is hierarchical if, as one goes down a path in the syntactic tree of Φ, the observations tied to quantifications become finer.

Definition 4. *An* SL_{ii}*-instance* $(\Phi,\mathcal{G})$ *is* hierarchical *if for all subformulas φ_1,φ_2 of Φ of the form $\varphi_2 = \langle\!\langle x \rangle\!\rangle^{o_2}\varphi'_2$ and $\varphi_1 = \langle\!\langle y \rangle\!\rangle^{o_1}\varphi'_1$ where φ_1 is a subformula of φ'_2, it holds that $O(o_1) \subseteq O(o_2)$.*

If $O(o_1) \subseteq O(o_2)$ we say that o_1 is *finer* than o_2 in $\mathcal{G}$, and that o_2 is *coarser* than o_1 in $\mathcal{G}$. Intuitively, this means that a player with observation o_1 observes game $\mathcal{G}$ no worse than, i.e., is not less informed, a player with observation o_2.

Example 1 (Security levels)**.** *We illustrate hierarchical instances in a "security levels" scenario, where higher levels have access to more data. Assume that the CGS$_{\text{ii}}$ $\mathcal{G}$ has $O(o_3) \subseteq O(o_2) \subseteq O(o_1)$ (level 3 is the highest security clearance, level 1 is the lowest). Let $\varphi = (a_1,x_1)(a_2,x_2)(a_3,x_3)\mathbf{G}p$. The SL_{ii} formula $\Phi := \langle\!\langle x_1 \rangle\!\rangle^{o_1}[\![x_2]\!]^{o_2}\langle\!\langle x_3 \rangle\!\rangle^{o_3}\varphi$ and $\mathcal{G}$ together form a hierarchical instance. It expresses that player a_1 (with lowest clearance) can collude with player a_3 (with highest clearance) to ensure a safety property p, even in the presence of an adversary a_2 (with intermediate clearance), as long as the strategy* *used by a_3 can depend on the strategy used by a_2. On the other hand, formula $\langle\langle x_1\rangle\rangle^{o_1}\langle\langle x_3\rangle\rangle^{o_3}[\![x_2]\!]^{o_2}\varphi$, which is similar to Φ except that the strategy used by a_3 cannot depend on the adversarial strategy used by a_2, does not form a hierarchical instance with $\mathscr{G}$.*

Here is the main contribution of this paper:

Theorem 2. *The model-checking problem for $\mathsf{SL}_{\mathrm{ii}}$ on the class of hierarchical instances is decidable.*

This is proved in Section 4 by reducing it to the model-checking problem of the hierarchical fragment of a logic called $\mathsf{QCTL}^*_{\mathrm{ii}}$, which we introduce, and prove decidable, in Section 3. We now give an important corollary of Theorem 2.

A Nash equilibrium in a game is a tuple of strategies such that no player has the incentive to deviate. Assuming that $\mathrm{Ag} = \{a_i : i \in [n]\}$ and goals are written in $\mathsf{SL}_{\mathrm{ii}}$, say $goal_i$ for $i \in [n]$, the following formula of $\mathsf{SL}_{\mathrm{ii}}$ expresses the existence of a Nash equilibrium:

$$\Phi_{\mathrm{NE}} := \langle\langle x_1\rangle\rangle^{o_1} \ldots \langle\langle x_n\rangle\rangle^{o_n}(a_1,x_1)\ldots(a_n,x_n)\Psi_{\mathrm{NE}}$$

where $\Psi_{\mathrm{NE}} := \bigwedge_{i\in[n]}[(\langle\langle y_i\rangle\rangle^{o_i}(a_i,y_i)goal_i) \to goal_i]$.

A $\mathrm{CGS}_{\mathrm{ii}}$ $\mathscr{G}$ is said to *yield hierarchical observation* [4] if the “finer-than” relation is a total ordering, i.e., if for all $o,o' \in \mathrm{Obs}$, either $O(o) \subseteq O(o')$ or $O(o') \subseteq O(o)$. Note that even if $\mathscr{G}$ yields hierarchical information, the instance $(\Phi_{\mathrm{NE}},\mathscr{G})$ is *not* hierarchical (unless $O(o_i) = O(o_j)$ for all $i,j \in [n]$). Nonetheless, we can decide if a game that yields hierarchical observation has a Nash equilibrium:

Corollary 3. *Given a $\mathrm{CGS}_{\mathrm{ii}}$ that yields hierarchical observation, whether $\mathscr{G} \models \Phi_{\mathrm{NE}}$ is decidable.*

Proof. The idea is that in a one-player game of partial-observation (such a game occurs when all but one player have fixed their strategies, as in the definition of Nash equilibrium), the player has a strategy enforcing some goal iff the player has a uniform strategy enforcing that goal. Here are the details. Let $\mathscr{G} = (\mathrm{Ac},V,E,\ell,v_\iota,O)$ be a $\mathrm{CGS}_{\mathrm{ii}}$ that yields hierarchical observation. Suppose the observation set is Obs. To decide if $\mathscr{G} \models \Phi_{\mathrm{NE}}$ first build a new $\mathrm{CGS}_{\mathrm{ii}}$ $\mathscr{G}' = (\mathrm{Ac},V,E,\ell,v_\iota,O')$ over observations $\mathrm{Obs}' := \mathrm{Obs} \cup \{o_p\}$ such that $O'(o) = O(o)$ and $O'(o_p) = \{(v,v) : v \in V\}$, and consider the sentence

$$\Phi' := \langle\langle x_1\rangle\rangle^{o_1} \ldots \langle\langle x_n\rangle\rangle^{o_n}(a_1,x_1)\ldots(a_n,x_n)\Psi'$$

where $\Psi' := \bigwedge_{i\in[n]}[(\langle\langle y_i\rangle\rangle^{o_p}(a_i,y_i)goal_i) \to goal_i]$.

Then $(\Phi',\mathscr{G}')$ is a hierarchical instance, and by Theorem 2 we can decide $\mathscr{G}' \models \Phi'$. We claim that $\mathscr{G}' \models \Phi'$ iff $\mathscr{G} \models \Phi_{\mathrm{NE}}$. To see this, it is enough to establish that:

$$\mathscr{G}',\chi,v_\iota \models \langle\langle y_i\rangle\rangle^{o_p}(a_i,y_i)goal_i \leftrightarrow \langle\langle y_i\rangle\rangle^{o_i}(a_i,y_i)goal_i,$$

for every $i \in [n]$ and every assignment χ such that $\chi(x_i) = \chi(a_i)$ is an o_i-uniform strategy.

To this end, fix i and χ. The right-to-left implication is immediate (since o_p is finer than o_i). For the converse, let σ be a p-uniform strategy (i.e., perfect-information) such that $\mathscr{G}',\chi[y_i \mapsto \sigma, a_i \mapsto \sigma],v_\iota \models goal_i$. Let $\pi := \mathrm{out}(\chi[y_i \mapsto \sigma, a_i \mapsto \sigma],v_\iota)$. Construct an o_i-uniform strategy σ' that agrees with σ on prefixes of π. This can be done as follows: if $\rho \sim_{o_i} \pi_{\leq j}$ for some j then define $\sigma'(\rho) = \sigma(\pi_{\leq j})$ (note that this is well-defined since if there is some such j then it is unique), and otherwise define $\sigma'(\rho) = a$ for some fixed action $a \in \mathrm{Ac}$. $\square$

2.5 Comparison with other logics

The main difference between SL and ATL-like strategic logics is that in the latter a strategy is always bound to some player, while in the former bindings and quantifications are separated. This separation adds expressive power, e.g., one can bind the same strategy to different players. Extending ATL with imperfect-information is done by giving each player an indistinguishability relation that its strategies must respect [5]. Our extension of SL by imperfect information, instead, assigns each strategy x an indistinguishability relation o when it is quantified $\langle\langle x\rangle\rangle^o$. Thus $\langle\langle x\rangle\rangle^o\varphi$ means "there exists a strategy with observation o such that φ holds". Associating observations in this way, i.e., to strategies rather than players has two consequences. First, it is a clean generalisation of SL in the perfect information setting [26]. Define the *perfect-information fragment of* SL_{ii} to be the logic SL_{ii} assuming that $\text{Obs} = \{o\}$ and $O(o) = \{(v,v) : v \in \mathscr{G}\}$ for every CGS_{ii} $\mathscr{G}$; also let us assimilate such structures with classic perfect-information concurrent game structures (CGS), which are the models of SL. Finally, let $\text{tr}_1 : \mathsf{SL} \to \mathsf{SL}_{\text{ii}}$ be the trivial translation that annotates each strategy quantifier $\langle\langle x\rangle\rangle$ with observation o. The next proposition says that the perfect-information fragment of SL_{ii} is a notational variant of SL.

Proposition 4. *For every* SL *sentence* φ *and every CGS* $\mathscr{G}$*, it holds that* $\mathscr{G} \models \varphi$ *iff* $\mathscr{G} \models \text{tr}_1(\varphi)$.

Second, SL_{ii} subsumes imperfect-information extensions of ATL^* that associate observations to players. We recall that an $\mathsf{ATL}^*_{\text{i,R}}$ formula[1] $\langle A\rangle\psi$ reads as "there are strategies for players in A such that ψ holds whatever players in $\text{Ag}\setminus A$ do". Consider the translation $\text{tr}_2 : \mathsf{ATL}^*_{\text{i,R}} \to \mathsf{SL}_{\text{ii}}$ that replaces each subformula of the form $\langle A\rangle\psi$, where $A = \{a_1,\ldots,a_k\} \subset \text{Ag}$ is a coalition of players and $\text{Ag}\setminus A = \{a_{k+1},\ldots,a_n\}$, with formula $\langle\langle x_1\rangle\rangle^{o_1}\ldots\langle\langle x_k\rangle\rangle^{o_k}[\![x_{k+1}]\!]^{o_p}\ldots[\![x_n]\!]^{o_p}(a_1,x_1)\ldots(a_n,x_n)\psi'$, where $\psi' = \text{tr}_2(\psi)$. Also, for every CGS_{ii} as considered in the semantics of ATL_{i}, i.e., where each agent is assigned an equivalence relation on positions (let us refer to such structures as ATL-CGS_{ii}), define the CGS_{ii} $\mathscr{G}'$ by interpreting each o_i as the equivalence relation for agent a_i in $\mathscr{G}$, and interpreting o_p as the identity relation.

Proposition 5. *For every* $\mathsf{ATL}^*_{\text{i,R}}$ *formula* φ *and ATL-CGS*$_{\text{ii}}$ $\mathscr{G}$*, it holds that* $\mathscr{G} \models \varphi$ *iff* $\mathscr{G}' \models \text{tr}_2(\varphi)$.

Third, SL_{ii} also subsumes the imperfect-information extension of ATL^* with strategy context (see [23] for the definition of $\mathsf{ATL}^*_{\text{sc}}$ with partial observation, which we refer to as $\mathsf{ATL}^*_{\text{sc,i}}$). The only difference between $\mathsf{ATL}^*_{\text{sc,i}}$ and $\mathsf{ATL}^*_{\text{i,R}}$ is the following: in $\mathsf{ATL}^*_{\text{i,R}}$, when a subformula of the form $\langle A\rangle\psi$ is met, we quantify existentially on strategies for players in A, and then we consider all possible outcomes obtained by letting other players behave however they want. Therefore, if any player in $\text{Ag}\setminus A$ had previously been assigned a strategy, it is forgotten. In $\mathsf{ATL}^*_{\text{sc,i}}$ on the other hand, these strategies are stored in a *strategy context*, which is a *partial* assignment χ, defined for the subset of players currently bound to a strategy. A strategy context allows one to quantify universally only on strategies of players who are not in A and who are not already bound to a strategy. It is then easy to define a translation $\text{tr}_3 : \mathsf{ATL}^*_{\text{sc,i}} \to \mathsf{SL}_{\text{ii}}$ by adapting translation tr_2 from Proposition 5, with the strategy context as parameter. For an ATL-CGS_{ii} $\mathscr{G}$, the CGS_{ii} $\mathscr{G}'$ is defined as for Proposition 5.

Proposition 6. *For every* $\mathsf{ATL}^*_{\text{sc,i}}$ *formula* φ *and ATL-CGS*$_{\text{ii}}$ $\mathscr{G}$*, it holds that* $\mathscr{G} \models \varphi$ *iff* $\mathscr{G}' \models \text{tr}_3(\varphi)$.

Fourth, there is a natural and simple translation of instances of the model-checking problem of CL [13] into the hierarchical instances of SL_{ii}. Moreover, the image of this translation consists of instances of SL_{ii} with a very restricted form, i.e., atoms mentioned in the SL_{ii}-formula are *observable* for all observations of the CGS_{ii}, i.e., players know the truth value of all atoms in all positions, for any observation they are assigned.

[1] See [5] for the definition of $\mathsf{ATL}^*_{\text{i,R}}$, where subscript i refers to "imperfect information" and subscript R to "perfect recall". Also, we consider the so-called *objective semantics* for $\mathsf{ATL}^*_{\text{i,R}}$.

Proposition 7. *There is an effective translation that, given a* CL*-instance* (S,φ) *produces a hierarchical* $\mathsf{SL}_{\mathrm{ii}}$*-instance* $(\mathcal{G},\Phi)$ *such that* $S\models\varphi$ *iff* $\mathcal{G}\models\Phi$*, and for all atoms* p *in* Φ*, observations* $o\in\mathrm{Obs}$ *and positions* $v,v'\in\mathcal{G}$*, if* $v\sim_o v'$ *then* $p\in\ell(v)\leftrightarrow p\in\ell(v')$*.*

To do this, one first translates CL into (hierarchical) $\mathsf{QCTL}^*_{\mathrm{ii}}$, the latter is defined in the next section. This step is a simple reflection of the semantics of CL in that of $\mathsf{QCTL}^*_{\mathrm{ii}}$. Then one translates $\mathsf{QCTL}^*_{\mathrm{ii}}$ into $\mathsf{SL}_{\mathrm{ii}}$ by a simple adaptation of the translation of QCTL^* into $\mathsf{ATL}^*_{\mathrm{sc}}$ [22].

3 QCTL^* with imperfect information

In this section we introduce an imperfect-information extension of QCTL^* [33, 10, 18, 19, 14, 21]. In order to introduce imperfect information, instead of considering equivalence relations between states as in concurrent game structures, we will enrich Kripke structures by giving internal structure to their states, i.e., we see states as n-tuples of local states. This way of modelling imperfect information is inspired from Reif's multi-player game structures [27] and distributed systems [16], and we find it very suitable to application of automata techniques, as discussed in Section 3.3.

The syntax of $\mathsf{QCTL}^*_{\mathrm{ii}}$ is similar to that of QCTL^*, except that we annotate second-order quantifiers by subsets $\mathbf{o}\subseteq[n]$. The idea is that quantifiers annotated by $\mathbf{o}$ can only "observe" the local states indexed by $i\in\mathbf{o}$. We define the tree-semantics of $\mathsf{QCTL}^*_{\mathrm{ii}}$: this means that we interpret formulas on trees that are the unfoldings of Kripke structures, which is necessary to capture synchronous perfect recall.

We then define the syntactic class of *hierarchical formulas* and state that model checking this class of formulas is decidable.

3.1 $\mathsf{QCTL}^*_{\mathrm{ii}}$ Syntax

The syntax of $\mathsf{QCTL}^*_{\mathrm{ii}}$ is very similar to that of QCTL^*: the only difference is that we annotate quantifiers by a set of indices that defines the "observation" of that quantifier.

Definition 5 ($\mathsf{QCTL}^*_{\mathrm{ii}}$ Syntax)**.** *Fix* $n\in\mathbb{N}$*. The syntax of* $\mathsf{QCTL}^*_{\mathrm{ii}}$ *is defined by the following grammar:*

$$\varphi := p \mid \neg\varphi \mid \varphi\vee\varphi \mid \mathbf{E}\psi \mid \exists^{\mathbf{o}}p.\,\varphi$$
$$\psi := \varphi \mid \neg\psi \mid \psi\vee\psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi$$

where $p\in\mathrm{AP}$ *and* $\mathbf{o}\subseteq[n]$*.*

Formulas of type φ are called *state formulas*, those of type ψ are called *path formulas*, and $\mathsf{QCTL}^*_{\mathrm{ii}}$ consists of all the state formulas defined by the grammar. We use standard abbreviation $\mathbf{A}\psi := \neg\mathbf{E}\neg\psi$. We also use $\exists p.\,\varphi$ as a shorthand for $\exists^{[n]}p.\,\varphi$, and we let $\forall p.\,\varphi := \neg\exists p.\,\neg\varphi$.

Given a $\mathsf{QCTL}^*_{\mathrm{ii}}$ formula φ, we define the set of *quantified propositions* $\mathrm{AP}_\exists(\varphi)\subseteq\mathrm{AP}$ as the set of atomic propositions p such that φ has a subformula of the form $\exists^{\mathbf{o}}p.\,\varphi$. We also define the set of *free propositions* $\mathrm{AP}_f(\varphi)\subseteq\mathrm{AP}$ as the set of atomic propositions that appear out of the scope of any quantifier of the form $\exists^{\mathbf{o}}p$. Observe that $\mathrm{AP}_\exists(\varphi)\cap\mathrm{AP}_f(\varphi)$ may not be empty in general, i.e., a proposition may appear both free and quantified in (different places of) a formula.

3.2 $\mathsf{QCTL}^*_{\mathrm{ii}}$ tree-semantics

We define the semantics on structures whose states are tuples of local states.

Local states. Let $\{L_i\}_{i\in[n]}$ denote $n \in \mathbb{N}$ 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 $\mathbf{o} \subseteq [n]$ is called a *concrete observation* (to distinguish it from observations o in the definitions of $\mathsf{SL}_{\mathrm{ii}}$).

Compound Kripke structures. These are Kripke structures where states are from $L_{[n]}$. A *compound Kripke structure*, or CKS, over AP, is a tuple $\mathcal{S} = (S, R, s_\iota, \ell)$ where $S \subseteq L_{[n]}$ is a set of *states*, $R \subseteq S \times S$ is a left-total[2] *transition relation*, $s_\iota \in S$ is an *initial state*, and $\ell : S \to 2^{\mathrm{AP}}$ is a *labelling function*.

A *path* in $\mathcal{S}$ is an infinite sequence of states $\lambda = s_0 s_1 \ldots$ such that for all $i \in \mathbb{N}$, $(s_i, s_{i+1}) \in R$. For $s \in S$, we let $\mathrm{Paths}(s)$ be the set of all paths that start in s. A *finite path* is a finite non-empty prefix of a path. We may write $s \in \mathcal{S}$ for $s \in S$. Since we will interpret $\mathsf{QCTL}^*_{\mathrm{ii}}$ on unfoldings of CKS, we now define infinite 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 $r \in X$, called the *root* of τ, such that each $u \in \tau$ starts with r ($r \preccurlyeq u$);
- if $u \cdot x \in \tau$ and $u \neq \varepsilon$, 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 $\lambda = u_0 u_1 \ldots$ such that for all $i \in \mathbb{N}$, u_{i+1} is a child of u_i, and $Paths(u)$ 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 $\ell : \tau \to 2^{\mathrm{AP}}$ is a *labelling*. For a labelled tree $t = (\tau, \ell)$ and an atomic proposition $p \in \mathrm{AP}$, we define the *p-projection of t* as the labelled tree $t\Downarrow_p := (\tau, \ell\Downarrow_p)$, where for each $u \in \tau$, $\ell\Downarrow_p(u) := \ell(u) \setminus \{p\}$. Finally, 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 $\mathsf{QCTL}^*_{\mathrm{ii}}$ the intuitive meaning of $\exists^{\mathbf{o}} p.\, \varphi$ in a tree t is that there is some equivalent tree t' modulo p such that t' is $\mathbf{o}$-uniform in p and satisfies φ. Intuitively, a tree is $\mathbf{o}$-uniform in p if it is uniformly labelled by p, i.e., if every two nodes that are indistinguishable when projected onto the local states indexed by $\mathbf{o} \subseteq [n]$ agree on their labelling of p.

Definition 6 ($\mathbf{o}$-indistinguishability and $\mathbf{o}$-uniformity in p)**.** *Fix $\mathbf{o} \subseteq [n]$ and $I \subseteq [n]$.*

- *Two tuples $x, x' \in L_I$ are* $\mathbf{o}$-indistinguishable*, written $x \approx_{\mathbf{o}} x'$, if $x\downarrow_{I\cap\mathbf{o}} = x'\downarrow_{I\cap\mathbf{o}}$.*
- *Two words $u = u_0 \ldots u_i$ and $u' = u'_0 \ldots u'_j$ over alphabet L_I are* $\mathbf{o}$-indistinguishable*, written $u \approx_{\mathbf{o}} u'$, if $i = j$ and for all $k \in \{0, \ldots, i\}$ we have $u_k \approx_{\mathbf{o}} u'_k$.*
- *A tree t is* $\mathbf{o}$-uniform in p *if for all $u, u' \in \tau$ such that $u \approx_{\mathbf{o}} u'$, we have $p \in \ell(u)$ iff $p \in \ell(u')$.*

Finally, we inductively define the satisfaction relation $\models$ for the semantics on trees, where $t = (\tau, \ell)$ is a 2^{AP}-labelled L_I-tree, u is a node and λ is a path in τ:

$$
\begin{array}{lll}
t, u \models p & \text{if} & p \in \ell(u) \\
t, u \models \neg\varphi & \text{if} & t, u \not\models \varphi \\
t, u \models \varphi \vee \varphi' & \text{if} & t, u \models \varphi \text{ or } t, u \models \varphi' \\
t, u \models \mathbf{E}\psi & \text{if} & \exists \lambda \in Paths(u) \text{ s.t. } t, \lambda \models \psi \\
t, u \models \exists^{\mathbf{o}} p.\, \varphi & \text{if} & \exists t' \equiv_p t \text{ s.t. } t' \text{ is } \mathbf{o}\text{-uniform in } p \text{ and } t', u \models \varphi.
\end{array}
$$

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

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

We write $t \models \varphi$ for $t,r \models \varphi$, where r is the root of t.

Example 2. *Consider the following* CTL *formula:*

$$\mathbf{border}(p) := \mathbf{AF}p \wedge \mathbf{AG}(p \to \mathbf{AXAG}\neg p).$$

This formula holds in a labelled tree if and only if each path contains exactly one node labelled with p. Now, consider the following $\mathsf{QCTL}^*_{\mathrm{ii}}$ *formula:*

$$\mathbf{level}(p) := \exists^{\emptyset} p.\, \mathbf{border}(p).$$

For a blind quantifier, two nodes of a tree are indistinguishable if and only if they have same depth. Therefore, this formula holds on a tree iff the p's label all and only the nodes at some fixed depth. This formula can thus be used to capture the equal level predicate on trees. Actually, just as QCTL^* *captures* MSO*, one can prove that* $\mathsf{QCTL}^*_{\mathrm{ii}}$ *with tree semantics subsumes* MSO *with equal level [9, 24, 34]. In Theorem 8 we make use of a similar observation to prove that model-checking* $\mathsf{QCTL}^*_{\mathrm{ii}}$ *is undecidable.*

Model-checking problem for $\mathsf{QCTL}^*_{\mathrm{ii}}$ under tree semantics. For the model-checking problem, we interpret $\mathsf{QCTL}^*_{\mathrm{ii}}$ on unfoldings of CKSs.

Tree unfoldings $t_S(s)$. Let $S = (S, R, s_\iota, \ell)$ be a compound Kripke structure over AP, and let $s \in S$. The *tree-unfolding of S from s* is the (AP, S)-tree $t_S(s) := (\tau, \ell')$, where τ is the set of all finite paths that start in s, and for every $u \in \tau$, $\ell'(u) := \ell(\mathrm{last}(u))$. Given a CKS S, a state $s \in S$ and a $\mathsf{QCTL}^*_{\mathrm{ii}}$ formula φ, we write $S, s \models \varphi$ if $t_S(s) \models \varphi$. Write $S \models \varphi$ if $t_S(s_\iota) \models \varphi$.

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

3.3 Discussion of the definition of $\mathsf{QCTL}^*_{\mathrm{ii}}$

Modelling of imperfect information. We model imperfect information by means of local states (rather than equivalence relations) since this greatly facilitates the use of automata techniques. More precisely, in our decision procedure for the hierarchical fragment of $\mathsf{QCTL}^*_{\mathrm{ii}}$, we make extensive use of an operation on tree automata called *narrowing*, which was introduced in [20] to deal with imperfect-information in the context of distributed synthesis for temporal specifications. Given an automaton $\mathscr{A}$ that works on $X \times Y$-trees, where X and Y are two finite sets, and assuming that we want to model an operation performed on trees while observing only the X component of each node, this narrowing operation allows one to build from $\mathscr{A}$ an automaton $\mathscr{A}'$ that works on X-trees, such that $\mathscr{A}'$ accepts an X-tree if and only if $\mathscr{A}$ accepts its widening to $X \times Y$. One can then make this automaton $\mathscr{A}'$ perform the desired operation, which will by necessity be performed uniformly with regards to the partial observation, since the Y component is absent from the input trees.

With our definition of compound Kripke structures, their unfoldings are trees over the Cartesian product $L_{[n]}$. To model a quantification $\exists^{\mathbf{o}} p$ with observation $\mathbf{o} \subseteq [n]$, we can thus use the narrowing

operation to forget about components L_i, for $i \in [n] \setminus \mathbf{o}$. We then use the classic projection of nondeterministic tree automata to perform existential quantification on atomic proposition p. Since the choice of the p-labelling is made directly on $L_{\mathbf{o}}$-trees, it is necessarily $\mathbf{o}$-uniform.
Choice of the tree semantics. QCTL^* is obtained by adding to CTL^* second-order quantification on atomic propositions. Several semantics have been considered. The two most studied ones are the *structure semantics*, in which formulas are evaluated directly on Kripke structures, and the *tree semantics*, in which Kripke structures are first unfolded into infinite trees. Tree semantics thus allows quantifiers to choose the value of a quantified atomic proposition in each *finite path* of the model, while in structure semantics the choice is only made in each state. When QCTL^* is used to express existence of strategies, existential quantification on atomic propositions labels the structure with strategic choices; in this kind of application, structure semantics reflects so-called *positional* or *memoryless* strategies, while tree semantics captures *perfect-recall* or *memoryfull* strategies. Since in this work we are interested in perfect-recall strategies, we only consider the tree semantics.

3.4 Model checking $\mathsf{QCTL}^*_{\text{ii}}$

We now prove that the model-checking problem for $\mathsf{QCTL}^*_{\text{ii}}$ under tree semantics is undecidable. This comes as no surprise since, as we will show, $\mathsf{QCTL}^*_{\text{ii}}$ can express the existence of winning strategies in imperfect-information games.

Theorem 8. *The model-checking problem for* $\mathsf{QCTL}^*_{\text{ii}}$ *under tree-semantics is undecidable.*

Proof. Let MSO_{eq} denote the extension of the logic MSO by a binary predicate symbol eq. Formulas of MSO_{eq} are interpreted on trees, and the semantics of $\text{eq}(x,y)$ is that x and y have the same depth in the tree. There is a translation of MSO-formulas to QCTL^*-formulas that preserves satisfaction [21]. This translation can be extended to map MSO_{eq}-formulas to $\mathsf{QCTL}^*_{\text{ii}}$-formulas using the formula $\mathbf{level}(\cdot)$ from Example 2 to help capture the equal-length predicate. Our result follows since the MSO_{eq}-theory of the binary tree is undecidable [24]. $\square$

The main result of this section is the identification of an important decidable fragment of $\mathsf{QCTL}^*_{\text{ii}}$.

Definition 7 (Hierarchical formulas)**.** *A* $\mathsf{QCTL}^*_{\text{ii}}$ *formula* φ *is* hierarchical *if for all subformulas* φ_1, φ_2 *of the form* $\varphi_1 = \exists^{\mathbf{o}_1} p_1 . \varphi_1'$ *and* $\varphi_2 = \exists^{\mathbf{o}_2} p_2 . \varphi_2'$ *where* φ_2 *is a subformula of* φ_1'*, we have* $\mathbf{o}_1 \subseteq \mathbf{o}_2$.

In other words, a formula is hierarchical if innermost propositional quantifiers observe at least as much as outermost ones. We let $\mathsf{QCTL}^*_{\text{i},\subseteq}$ be the set of hierarchical $\mathsf{QCTL}^*_{\text{ii}}$ formulas.

Theorem 9. *Model checking* $\mathsf{QCTL}^*_{\text{i},\subseteq}$ *under tree semantics is non-elementary decidable.*

4 Model checking hierarchical instances of SL_{ii}

In this section we establish that the model-checking problem for SL_{ii} restricted to the class of hierarchical instances is decidable (Theorem 2). We build upon the proof in [22] that establishes the decidability of the model-checking problem for $\mathsf{ATL}^*_{\text{sc}}$ by reduction to the model-checking problem for QCTL^*. The main difference is that we use quantifiers on atomic propositions parameterised with observations that reflect the ones used in the SL_{ii} model-checking instance.

Let $(\Phi, \mathscr{G})$ be a hierarchical instance of the SL_{ii} model-checking problem, where $\mathscr{G} = (\text{Ac}, V, E, \ell, v_\iota, O)$. We will first show how to define a CKS $S_{\mathscr{G}}$ and a bijection $\rho \mapsto u_\rho$ between the set of finite plays ρ starting in a given position v and the set of nodes in $t_{S_{\mathscr{G}}}(s_v)$.

Then, for every subformula φ of Φ and partial function $f : \mathrm{Ag} \rightharpoonup \mathrm{Var}$, we will define a $\mathsf{QCTL}^*_{\mathrm{ii}}$ formula $(\varphi)^f$ (that will also depend on $\mathscr{G}$) such that the following holds:

Proposition 10. *Suppose that free*$(\varphi) \cap \mathrm{Ag} \subseteq$ *dom*(f)*, and* $f(a) = x$ *implies* $\chi(a) = \chi(x)$ *for all* $a \in$ *dom*(f)*. Then*

$$\mathscr{G}, \chi, \rho \models \varphi \quad \textit{if and only if} \quad t_{S_{\mathscr{G}}}(s_\rho) \models (\varphi)^f.$$

Applying this to the sentence Φ, any assignment χ, and the empty function $\emptyset$, we get that

$$\mathscr{G}, \chi, v_\iota \models \Phi \quad \text{if and only if} \quad t_{S_{\mathscr{G}}}(s_{v_\iota}) \models (\Phi)^{\emptyset}.$$

Constructing the CKS $S_{\mathscr{G}}$. We define $S_{\mathscr{G}}$ so that (indistinguishable) nodes in its tree-unfolding correspond to (indistinguishable) finite plays in $\mathscr{G}$. Let $\mathrm{AP}_v := \{p_v \mid v \in V\}$, that we assume to be disjoint from AP, and let $\mathrm{AP}_c := \{p^x_c \mid c \in \mathrm{Ac} \text{ and } x \in \mathrm{Var}\}$ that we assume, again, are disjoint from $\mathrm{AP} \cup \mathrm{AP}_v$.

Suppose $\mathrm{Obs} = \{o_1, \ldots, o_n\}$. 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 $\sim_o$. Since we need to know the actual position of the $\mathrm{CGS}_{\mathrm{ii}}$ to define the dynamics, we also let $L_{n+1} := V$.

Define the CKS $S_{\mathscr{G}} := (S, R, s_\iota, \ell')$ where

- $S := \{s_v \mid v \in V\}$ with $s_v := ([v]_{o_1}, \ldots, [v]_{o_n}, v) \in \prod_{i \in [n+1]} L_i$,
- $R := \{(s_v, s_{v'}) \mid \exists \boldsymbol{c} \in \mathrm{Ac}^{\mathrm{Ag}} \text{ s.t. } E(v, \boldsymbol{c}) = v'\} \subseteq S \times S$,
- $s_\iota := s_{v_\iota}$, and
- $\ell'(s_v) := \ell(v) \cup \{p_v\} \subseteq \mathrm{AP} \cup \mathrm{AP}_v$,

For every finite play $\rho = v_0 \ldots v_k$, define the node $u_\rho := s_{v_0} \ldots s_{v_k}$ in $t_{S_{\mathscr{G}}}(s_{v_0})$ (which exists, by definition of $S_{\mathscr{G}}$ and of tree unfoldings). Note that the mapping $\rho \mapsto u_\rho$ defines a bijection between the set of finite plays and the set of nodes in $t_{S_{\mathscr{G}}}(s_\iota)$.

Constructing the $\mathsf{QCTL}^*_{\mathrm{i},\subseteq}$ formulas $(\varphi)^f$. We now describe how to transform an $\mathsf{SL}_{\mathrm{ii}}$ formula φ and a partial function $f : \mathrm{Ag} \rightharpoonup \mathrm{Var}$ into a $\mathsf{QCTL}^*_{\mathrm{ii}}$ formula $(\varphi)^f$ (that will also depend on $\mathscr{G}$). Suppose that $\mathrm{Ac} = \{c_1, \ldots, c_l\}$, and define $(\varphi)^f$ by induction on φ. We begin with the simple cases: $(p)^f := p$; $(\neg\varphi)^f := \neg(\varphi)^f$; and $(\varphi_1 \vee \varphi_2)^f := (\varphi_1)^f \vee (\varphi_2)^f$.

We continue with the second-order quantifier case:

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

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

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

We describe this formula in words. For each possible action $c \in \mathrm{Ac}$, an existential quantification on the atomic proposition p^x_c "chooses" for each finite play $\rho = v_0 \ldots v_k$ of $\mathscr{G}$ (or, equivalently, for each node u_ρ of the tree $t_{S_{\mathscr{G}}}(s_{v_0})$) whether strategy x plays action c in ρ or not. Formula $\varphi_{\mathrm{str}}(x)$ ensures that in each finite play, exactly one action is chosen for strategy x, and thus that atomic propositions p^x_c indeed characterise a strategy, call it σ_x.[3]

[3] More precisely, if $\varphi_{\mathrm{str}}(x)$ holds in node u_ρ, it ensures that propositions from AP_c define a partial strategy, defined on all nodes of the subtree rooted in u_ρ. This is enough because $\mathsf{SL}_{\mathrm{ii}}$ can only talk about the future: when evaluating a formula in a finite play ρ, the definition of strategies on plays that do not start with ρ is irrelevant.

Moreover, a quantifier with concrete observation $\widetilde{o_i}$ receives information corresponding to observation o_i (observe that for all $i \in [n]$, $i \in \widetilde{o_i}$) as well as information corresponding to coarser observations. Note that including all coarser observations does not increase the information accessible to the quantifier: indeed, one can show that two nodes are $\{i\}$-indistinguishable if and only if they are $\widetilde{o_i}$-indistinguishable. However, this definition of $\widetilde{o_i}$ allows us to obtain hierarchical formulas. Since quantification on propositions p_c^x is done uniformly with regards to concrete observation $\widetilde{o_i}$, it follows that σ_x is an o_i-strategy.

Here are the remaining cases:

$$\begin{aligned}((a,x)\varphi)^f &:= (\varphi)^{f[a\mapsto x]}\\ (\mathbf{X}\varphi_1)^f &:= \mathbf{A}\big(\psi_{\text{out}}(f) \to \mathbf{X}(\varphi_1)^f\big)\\ (\varphi_1\mathbf{U}\varphi_2)^f &:= \mathbf{A}\big(\psi_{\text{out}}(f) \to (\varphi_1)^f\mathbf{U}(\varphi_2)^f\big)\end{aligned}$$

where

$$\psi_{\text{out}}(f) := \mathbf{G}\left(\bigwedge_{v\in V}\bigwedge_{\boldsymbol{c}\in \text{Ac}^{\text{Ag}}}\left(p_v \wedge \bigwedge_{a\in\text{Ag}} p_{\boldsymbol{c}_a}^{f(a)} \to \mathbf{X}p_{E(v,\boldsymbol{c})}\right)\right).$$

The formula $\psi_{\text{out}}(f)$ is used to select the unique path assuming that every player, say a, follows the strategy $\sigma_{f(a)}$. This completes the justification of Proposition 10.

Preserving hierarchy. We show that $(\Phi)^{\emptyset}$ is a hierarchical $\mathsf{QCTL}^*_{\text{ii}}$ formula. This simply follows from the fact that Φ is hierarchical in $\mathscr{G}$ and that for every two observations o_i and o_j in Obs such that $O(o_i) \subseteq O(o_j)$, by definition of $\widetilde{o_k}$ we have that $\widetilde{o_i} \subseteq \widetilde{o_j}$. This completes the proof of Theorem 2.

5 Outlook

We introduced SL_{ii}, a logic for reasoning about strategic behaviour in multi-player games with imperfect information. The syntax mentions observations, and thus allows one to write formulas that talk about dynamic observations. We isolated the class of hierarchical formula/model pairs $(\Phi,\mathscr{G})$ and proved that one can decide if $\mathscr{G} \models \Phi$. The proof reduces (hierarchical) instances to (hierarchical) formulas of $\mathsf{QCTL}^*_{\text{ii}}$, a low-level logic that we introduced, and that serves as a natural bridge between SL_{ii} and automata constructions.

We believe that $\mathsf{QCTL}^*_{\text{ii}}$ is of independent interest and deserves study in its own right. Indeed, it subsumes MSO with equal-level predicate, which is undecidable and of which we know no decidable fragment that could be used to reason about imperfect information with perfect recall; yet its syntax and models make it possible to define a natural fragment (the hierarchical fragment) that has a simple definition, a decidable model-checking problem, and is suited for strategic reasoning.

Since one can alternate quantifiers in SL_{ii}, our decidability result goes beyond synthesis. As we showed, we can use it to decide if a game that yields hierarchical observation has a Nash equilibrium. A crude but easy analysis of our main decision procedure shows it is non-elementary.

This naturally leads to a number of avenues for future work: define and study the expressive power and computational complexity of fragments of SL_{ii} [25]; adapt the notion of hierarchical instances to allow for situations in which hierarchies can change infinitely often along a play [4]; and extend the logic to include epistemic operators for individual and common knowledge, as is done in [6], which are important for reasoning about distributed systems [11].

References

[1] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. *JACM* 49(5), pp. 672–713.

[2] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano & Sasha Rubin (2017): *Verification of Multi-Agent Systems with Imperfect Information and Public Actions*. In: *AAMAS'17*.

[3] Raphaël Berthon, Bastien Maubert & A. Murano (2017): *Decidability Results for ATL* with Imperfect Information and Perfect Recall*. In: *AAMAS'17*.

[4] Dietmar Berwanger, Anup Basil Mathew & Marie Van den Bogaard (2015): *Hierarchical information patterns and distributed strategy synthesis*. In: *ATVA'15*, pp. 378–393.

[5] Nils Bulling & Wojtech Jamroga (2014): *Comparing variants of strategic ability: how uncertainty and memory influence general properties of games*. In: *AAMAS'14*, 28, pp. 474–518.

[6] Petr Cermák, Alessio Lomuscio, Fabio Mogavero & Aniello Murano (2014): *MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications*. In: *CAV'14*, pp. 525–532.

[7] Krishnendu Chatterjee & Laurent Doyen (2014): *Games with a Weak Adversary*. In: *ICALP'14*, pp. 110–121.

[8] Catalin Dima & Ferucio Laurentiu Tiplea (2011): *Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable*. *CoRR* abs/1102.4225.

[9] Calvin C. Elgot & Michael. O. Rabin (1966): *Decidability and Undecidability of Extensions of Second (First) Order Theory of (Generalized) Successor*. *JSL* 31(2), pp. 169–181.

[10] E Allen Emerson & A Prasad Sistla (1984): *Deciding branching time logic*. In: *STOC'84*, pp. 14–24.

[11] Ronald Fagin, Joseph Y. Halpern, Yoram Moses & Moshe Y. Vardi (1995): *Reasoning about knowledge*. 4, MIT press Cambridge.

[12] Bernd Finkbeiner & Sven Schewe (2005): *Uniform Distributed Synthesis*. In: *LICS'05*, pp. 321–330.

[13] Bernd Finkbeiner & Sven Schewe (2010): *Coordination Logic*. In: *CSL'10*, pp. 305–319.

[14] Tim French (2001): *Decidability of quantifed propositional branching time logics*. In: *AJCAI'01*, pp. 165–176.

[15] Paul Gastin, Nathalie Sznajder & Marc Zeitoun (2009): *Distributed synthesis for well-connected architectures*. *FMSD* 34(3), pp. 215–237.

[16] Joseph Y Halpern & Moshe Y Vardi (1989): *The complexity of reasoning about knowledge and time. I. Lower bounds*. *JCSS* 38(1), pp. 195–237.

[17] Orna Kupermann & Moshe Y. Vardi (2001): *Synthesizing distributed systems*. In: *LICS'01*, pp. 389–398.

[18] Orna Kupferman (1995): *Augmenting Branching Temporal Logics with Existential Quantification over Atomic Propositions*. In: *CAV'95*, pp. 325–338.

[19] Orna Kupferman, P. Madhusudan, P. S. Thiagarajan & Moshe Y. Vardi (2000): *Open Systems in Reactive Environments: Control and Synthesis*. In: *CONCUR'00*, pp. 92–107.

[20] Orna Kupferman & Moshe Y Vardi (1999): *Church's problem revisited*. *BSL*, pp. 245–263.

[21] François Laroussinie & Nicolas Markey (2014): *Quantified CTL: Expressiveness and Complexity*. *LMCS* 10(4).

[22] François Laroussinie & Nicolas Markey (2015): *Augmenting ATL with strategy contexts*. *IC* 245, pp. 98–123.

[23] François Laroussinie, Nicolas Markey & Arnaud Sangnier (2015): *ATLsc with partial observation*. In: *GandALF'15*, pp. 43–57.

[24] Hans Läuchli & Christian Savioz (1987): *Monadic second order definable relations on the binary tree*. *JSL* 52(01), pp. 219–226.

[25] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2012): *What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic*. In: *CONCUR'12*, pp. 193–208.

[26] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2014): *Reasoning About Strategies: On the Model-Checking Problem*. *TOCL* 15(4), pp. 34:1–34:47.

[27] Gary Peterson, John Reif & Salman Azhar (2001): *Lower bounds for multiplayer noncooperative games of incomplete information*. *CAMWA* 41(7), pp. 957–992.

[28] Gary Peterson, John Reif & Salman Azhar (2002): *Decision algorithms for multiplayer noncooperative games of incomplete information*. *CAMWA* 43(1), pp. 179–206.

[29] Gary L. Peterson & John H. Reif (1979): *Multiple-Person Alternation*. In: *SFCS'79*, pp. 348–363.

[30] Sophie Pinchinat & Stéphane Riedweg (2005): *A decidable class of problems for control under partial observation*. *IPL* 95(4), pp. 454–460.

[31] Amir Pnueli & Roni Rosner (1990): *Distributed reactive systems are hard to synthesize*. In: *FOCS'90*, pp. 746–757.

[32] Sven Schewe & Bernd Finkbeiner (2007): *Distributed Synthesis for Alternating-Time Logics*. In: *ATVA'07*, pp. 268–283.

[33] A.Prasad Sistla (1983): *Theoretical Issues in the Design and Cerification of Distributed Systems.* Ph.D. thesis, Harvard University, Cambridge, MA, USA.

[34] Wolfgang Thomas (1992): *Infinite Trees and Automaton-Definable Relations over omega-Words*. *TCS* 103(1), pp. 143–159.