A Logic of Probabilistic Knowledge and Strategy *

Xiaowei Huang and Cheng Luo
School of Computer Science and Engineering
University of New South Wales, Australia
{xiaoweih,luoc}@cse.unsw.edu.au

ABSTRACT

The ability of reasoning about knowledge and strategy is key to the autonomy of an intelligent system of multiple players. In this paper, we study the logic of knowledge and strategy in stochastic multiagent systems, where the system's behaviour is determined by both the behaviour of the players and by some random elements. Players have incomplete information about the system and do not have memory. A logic PATEL*, whose semantics is based on partially-observed concurrent game structures, is proposed. The computational complexities of model checking the logic and its sublogics are solved.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

Keywords

Logic of Knowledge, Strategy Logic, Probabilistic Reasoning, Multiagent Systems, Computational Complexity

1. INTRODUCTION

Model checking [6] is an approach to the verification of system designs. Taking as input a system model and a logic formula, it is to automatically determine whether the formula holds on the model by algorithms. Starting from temporal logics, model checking has been extensively studied in various logics, e.g., probabilistic temporal logics, strategic logics, and temporal epistemic logics, etc.

A stochastic multiagent system consists of a set of players (or agents, processes, etc) operating concurrently in a stochastic environment. The system's behaviour is determined both by the behaviour of the players and by some random elements, e.g., the inaccuracy of the sensors, etc. In such a system, each player is assumed to have its own local state and a system state consists of a local state for each player. The system is incomplete information if players can not directly observe the entire system state, especially other players' local states. At each system state, every player will take a local action and the next system state is determined by taking into consideration players' local actions and the random elements.

Notable examples of stochastic multiagent systems include various card games, security protocols, and military operations.

In an intelligent system of multiple players, it's key to the autonomy of the players that they have the ability of reasoning about knowledge and strategy. Every player has its own goals and may achieve its goals by cooperating with its alliance and competing with its adversaries. In a play, the players may gain knowledge and utilise the obtained knowledge to construct an optimal strategy. Reasoning about knowledge and strategy has applications in e.g., distributed systems, artificial intelligence, game theory, and computer security, etc. Logics have been the fundamental instruments when reasoning about knowledge and strategy. Among all relevant logics, the logic of knowledge [8] and alternating time temporal logic [1] are two prominent ones that have been widely used in specifying the *possibility* of a property in a multi-player system from the players' *subjective* point of views. E.g., "the player i can eventually know the fact ϕ" and "the set A of players has a strategy to enforce the goal ϕ, no matter what their opponent's strategy is".

In stochastic multiagent systems, we are interested in not only the possibility but also the *probability* of a property. E.g., "the player i can eventually know the fact ϕ with a probability more than p_1" and "the set A of players have a strategy to achieve the goal ϕ with a probability no less than p_2". These probabilistic properties may provide more insightful information into the system.

We make the following contributions in the paper. First, we study a logic, named probabilistic alternating-time temporal epistemic logic PATEL*, that combines knowledge, strategy and temporal modalities with probabilistic measures to specify the properties of a stochastic multiagent system. The logic has two probabilistic operators: probabilistic knowledge operator and probabilistic strategy operator. The semantics of the logic is based on partially-observed probabilistic concurrent game structures. We assume that players do not have memory to remember their observation histories (or *observational semantics*). While it may be arguable that assuming players have unbounded memory to remember all its past observations and local actions (or perfect recall semantics) is meaningful in reasoning about critical systems, as it imposes the optimal assumption over the adversaries, the important problem of model checking is highly intractable. In non-probabilistic systems, [26] proved that $LTLK_n$ is undecidable, when common knowledge operator is included, and is non-elementary, otherwise. It is widely believed [1, 23] that ATL is undecidable if assuming perfect recall. In stochastic multiagent systems, it is worse, as [17, 14] shows that the undecidability holds even for the single-player fragment of the PATL logic and the probabilistic temporal epistemic logic without common knowledge operator.

Second, the computational complexities of model checking PATEL* and its sublogics are studied. Generally, model checking

Logic	Combined Complexity	Model Complexity	Formula Complexity
PATEL*	2-EXP-compl	Σ_2^P-compl	2-EXP-compl
PATEL	Δ_3^P-compl	Σ_2^P-compl	P
PCTL*K$_n$	2-EXP-compl	P-compl	2-EXP-compl
PCTLK$_n$	P-compl	P-compl	P

Table 1: Model Checking Complexities

PATEL* is 2-EXPTIME-complete. More subtleties are discovered when we look into the model complexity and formula complexity, by assuming that the formula or the model is fixed. The model complexity of PATEL* is Σ_2^P-complete and the formula complexity is 2-EXPTIME-complete. These complexities may be lowered if we impose constraints on the syntax of the logic. Table 1 collects the complexity results of PATEL* and its sublogics.

On one hand, in the logic PCTL*K$_n$, we can only reason about those strategies where all players are in a group, by writing formulas like $\langle\!\langle Agt\rangle\!\rangle^{\bowtie d}\phi$. In this logic, the combined complexity and the formula complexity remain at 2-EXPTIME-complete, while the model complexity falls to P-complete.

On the other hand, in the logics PATEL and PCTLK$_n$, we impose, on the logics PATEL* and PCTL*K$_n$, a constraint that every temporal operator is immediately preceded by a probabilistic strategy operator. In these two logics, the model complexity remains, while the combined complexity and the formula complexity are lowered. For PATEL logic, the combined complexity is Δ_3^P-complete and the formula complexity is in P. For the PCTLK$_n$ logic, the combined complexity is P-complete and the formula complexity is in P.

2. A LOGIC OF PROBABILISTIC KNOWLEDGE AND STRATEGY

Suppose that we are working with a stochastic multiagent system with a finite set Agt of players. Let $Prop$ be a set of propositions. To specify the properties of the system, we present a logic PATEL* that combines the temporal operators, the knowledge operator, the strategy operator and probability measures. Its syntax is given by

$$\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid X\phi \mid \phi_1 U\phi_2 \mid \mathrm{Pr}_i^{\bowtie d}\phi \mid \langle\!\langle A\rangle\!\rangle^{\bowtie d}\phi$$

where $p \in Prop$, $i \in Agt$, $A \subseteq Agt$, d is a rational constant in $[0, 1]$, and $\bowtie$ is a relation symbol in the set $\{\leq, <, >, \geq\}$. Intuitively, formula $X\phi$ expresses that ϕ holds at the next time, $\phi_1 U\phi_2$ expresses that ϕ_1 holds until ϕ_2 becomes true, $\mathrm{Pr}_i^{\bowtie d}\phi$ expresses that player i knows the fact ϕ with a probability in relation $\bowtie$ with constant d, and $\langle\!\langle A\rangle\!\rangle^{\bowtie d}\phi$ expresses that players in A can collaboratively enforce the fact ϕ with a probability in relation $\bowtie$ with constant d. Other operators can be obtained in the usual way, e.g., $F\phi \equiv True U\phi$, $G\phi \equiv \neg F\neg\phi$, $\phi_1 R\phi_2 \equiv \neg(\neg\phi_1 U\neg\phi_2)$, etc.

From PATEL* logic, we may have several variants by reducing its expressiveness.

- PATEL logic, in which every temporal operator has to be immediately preceded by a probabilistic strategy operator. Like other branching time logics CTL and ATL, we need both $\langle\!\langle A\rangle\!\rangle^{\bowtie d}\phi_1 U\phi_2$ and $\langle\!\langle A\rangle\!\rangle^{\bowtie d}\phi_1 R\phi_2$.

- PCTL*K$_n$ logic, in which the probabilistic strategy operator is restricted by only allowing to write $\langle\!\langle Agt\rangle\!\rangle^{\bowtie d}\phi$.

- PCTLK$_n$ logic, in which both the restrictions are imposed.

Moreover, by removing the probabilistic knowledge operator from PCTL*K$_n$ and PCTLK$_n$ logic, we may obtain the same expressiveness as that of PCTL* and PCTL logics [11].

3. PROBABILISTIC CONCURRENT GAME STRUCTURE

Let $Agt = \{1, ..., n\}$ be a set of players. A finite partially-observed probabilistic concurrent game structure (PO-PCGS) is a tuple $M = (S, s_{init}, \{Act_x\}_{x\in Agt\cup\{e\}}, \{N_x\}_{x\in Agt\cup\{e\}}, \{PO_i\}_{i\in Agt}, PT, \pi)$, where S is a finite set of states, $s_{init} \in S$ is the initial state, Act_x is the set of local actions of the environment e or the players in Agt, $N_x : S \to \mathcal{P}(Act_x)$ indicates the set of legal actions that are available to the environment or the player $x \in Agt \cup \{e\}$ at a specific state. Let $Act = Act_e \times Act_1 \times ... \times Act_n$ be the set of global actions. Function $PT : S \times Act \times S \to [0, 1]$ represents a probability transition matrix such that $\sum_{s'\in S} PT(s, a, s') \in \{0, 1\}$, for all $s \in S$ and $a \in Act$, and $\exists a \in Act \exists s' \in S : PT(s, a, s') > 0$, for all $s \in S$. For each player $i \in Agt$, we have an observation function $PO_i : S \times O \to [0, 1]$, such that

1. $\forall s \in S : PO_i(s, o_1) \neq 0 \wedge PO_i(s, o_2) \neq 0 \Rightarrow o_1 = o_2$,

2. $\forall o \in O : (\sum_{s\in S} PO_i(s, o)) = 0 \vee (\sum_{s\in S} PO_i(s, o)) = 1$, and

3. $PO_i(s_{init}, o) = 1$ for all $i \in Agt$ and some $o \in O$.

Finally, the labelling function $\pi : S \to \mathcal{P}(Prop)$ is an interpretation of the atomic propositions $Prop$ at the states. We assume that all states in S are reachable from s_{init}. Moreover, we use $O_i(s)$ to denote the observation $o \in O$, if $PO_i(s, o) > 0$, and write $PO_i(s)$ for $PO_i(s, O_i(s))$. Here we make an assumption that, a PO-PCGS M has a single initial state s_{init} and every player can distinguish it from other states. The constraints are simply for the ease of the notations and can be relaxed without affecting the conclusions.

We introduce some preliminary notions for probabilistic systems. A *probability space* is a triple (W, F, μ) such that W is a set, called the *carrier*, $F \subseteq \mathcal{P}(W)$ is a set of *measurable* sets in $\mathcal{P}(W)$, closed under countable union and complementation, and $\mu : F \to [0, 1]$ is a *probability measure*, such that $\mu(W) = 1$ and $\mu(U \cup V) = \mu(U) + \mu(V)$ if $U \cap V = \emptyset$. As usual, we define the conditional probability $\mu(U|V) = \mu(U \cap V)/\mu(V)$ when $\mu(V) \neq 0$.

We assume that for all states $s_1, s_2 \in S$ and $i \in Agt$, $O_i(s_1) = O_i(s_2)$ implies $N_i(s_1) = N_i(s_2)$. Intuitively, a player can distinguish two states if they have different legal actions. We suppose that the environment reacts *deterministically*, and thus all nondeterminism in the system comes from the actions of the players. Let $k_i(s) = \{s' \in S \mid O_i(s) = O_i(s')\}$ be the set of states in which the player i can not distinguish from state s.

Let $s, s' \in S$ and $a \in Act$. A *path* ρ from a state s is a finite or infinite sequence of states and actions $s_0 a_0 s_1 a_1 \ldots$ such that $s_0 = s$ and $PT(s_k, a_k, s_{k+1}) > 0$ for all k such that $k < |\rho| - 1$, where $|\rho|$ is the total number of states on ρ. Given a path ρ, we use $s(\rho, m)$ to denote its $(m+1)$-th state, $a(\rho, m)$ to denote its m-th action, in which $a_x(\rho, m)$ is its m-th local action of the environment or the player $x \in Agt \cup \{e\}$. Moreover, we use $s(\rho, 0..m)$ to denote the sequence of states $s(\rho, 0)...s(\rho, m)$, $a(\rho, 1..m)$ to denote the sequence of global actions $a(\rho, 1)...a(\rho, m)$ and $a_i(\rho, 1..m)$ the sequence of local actions $a_i(\rho, 1)...a_i(\rho, m)$ of player i. A *fullpath* from a state s is an infinite path from s. Let $\rho[m]$ be the suffix of the path ρ from state $s(\rho, m)$. More specifically, $\rho[0] = \rho$.

A strategy σ_i of a player i is a function that maps each finite path $\rho = s_0 a_0 s_1 a_1 \ldots s_n$ to an action in $N_i(s_n)$. A (finite or infinite) path ρ is compatible with σ_i if $a_i(\rho, k) = \sigma_i(s_0 a_0 \ldots s_{k-1})$ for all $k \leq |\rho|$. A *memoryless* strategy σ_i of a player i is a function that maps each

state $s \in S$ to an action in $N_i(s)$, i.e., $\sigma_i(s) \in N_i(s)$. A (finite or infinite) path ρ is compatible with σ_i if $a_i(\rho, k) = \sigma_i(s(\rho, k-1))$ for all $k \geq 1$. Given a PO-PCGS M and a player i's strategy σ_i, we write $Path(M, \sigma_i)$ for the set of fullpaths in M that are compatible with σ_i. In the following, strategies are always memoryless.

A strategy σ_i is *uniform* if for all paths $\rho, \rho' \in Path(M, \sigma_i)$ and $m, m' \in \mathbb{N}$, we have $s_i(\rho, m) = s_i(\rho', m')$ implies $a_i(\rho, m) = a_i(\rho', m')$, i.e., i's reactions following σ_i are a function of its local states. The local state $s_i(\rho, m)$ depends on the view $\mathcal{V}$ that the player i has on the observations. For perfect recall view, we have that $s_i(\rho, m) = O_i(s(\rho, 0))a_i(\rho, 1)...O_i(s(\rho, m))$, representing that player i remembers all its past observations and local actions. For observational view, we have that $s_i(\rho, m) = O_i(s(\rho, m))$, representing that player i can make a current observation. In an incomplete information system, a memoryless strategy usually corresponds with observational semantics.

Let A be a set of players. A coalition strategy σ_A fixes a strategy σ_i for each player $i \in A$. We call σ_A a complete coalition strategy if $A = Agt$, or an incomplete coalition strategy if $A \subset Agt$. We assume the following definitions on a set of players when writing formulas like $\langle\langle A \rangle\rangle^{\bowtie d} \phi$. Let $k_A(s) = \bigcap_{i \in A} k_i(s)$ and

$$PO_A(s) = \frac{\Pi_{i \in A} PO_i(s)}{\sum_{s' \in k_A(s)} \Pi_{i \in A} PO_i(s')},$$

For example, given two players i and j and two states s and t, if $PO_i(s) = 1/4$, $PO_i(t) = 3/4$, $PO_j(s) = 1/3$, and $PO_j(t) = 2/3$, then $PO_{\{i,j\}}(s) = \frac{1/4 \times 1/3}{1/4 \times 1/3 + 3/4 \times 2/3} = 1/7$ and $PO_{\{i,j\}}(t) = \frac{3/4 \times 2/3}{1/4 \times 1/3 + 3/4 \times 2/3} = 6/7$, which, comparing with the observation functions PO_i and PO_j, suggest that the addition of observation functions between two players strengthens a consensus view between them that state s is less possible than state t. This definition can be seen as a probabilistic variant of the distributed knowledge [8]. Similarly, we can define O_A.

The idea of defining a semantics for the PATEL* logic is based on breaking a PO-PCGS into a set of partially-observed discrete-time Markov chains (PO-DTMCs). A PO-DTMC C of a PO-PCGS M is a tuple $(S, s_{init}, \{Act_x\}_{x \in Agt \cup \{e\}}, \{\sigma_i\}_{i \in Agt}, \{PO_x\}_{x \in Agt}, PT_C, \pi)$ where $S, s_{init}, Act_x, PO_x, \pi$ are defined the same as in M, σ_i is a strategy of player i, and PT_C is a probability transition matrix such that

$$PT_C(s, a, s') = \begin{cases} PT(s, a, s') & \text{if } \sigma_{Agt}(s) = a, \\ 0 & \text{otherwise.} \end{cases}$$

A PO-DTMC C of M is uniform for player i if $\sigma_i(s) = \sigma_i(s')$ for all $s' \in k_i(s)$.

Let $\mathcal{R}_{C,s}$ be the set of fullpaths in a PO-DTMC C that starts from state s. We now define a probability space on $\mathcal{R}_{C,s}$, using a well-known construction (e.g., that of [27]). Given a finite path ρ of $m+1$ states and m actions such that $\rho(0) = s$, write $\mathcal{R}_{C,s}(\rho) = \{\rho' \in \mathcal{R}_{C,s} \mid s(\rho', 0..m) = s(\rho, 0..m), a(\rho', 1..m) = a(\rho, 1..m)\}$ for the set of fullpaths with prefix ρ. (One may view this as a *cone* of fullpaths sharing the same prefix ρ.) Let $F_{C,s}$ be the minimal algebra with basis the sets $W_{C,s} = \bigcup\{\mathcal{R}_{C,s}(\rho) \mid \rho$ prefixes some $r \in \mathcal{R}_{C,s}\}$, i.e., $F_{C,s}$ is the set of all sets of fullpaths that can be constructed from the basis by using countable union and complement. We define measure $\mu_{C,s}$ on the basis sets by

$$\mu_{C,s}(\mathcal{R}_{C,s}(\rho)) = \prod_{i=0}^{m-1} PT(s(\rho, i), a(\rho, i+1), s(\rho, i+1)).$$

Intuitively, $\mu_{C,s}(\mathcal{R}_{C,s}(\rho))$ denotes the probability of those infinite paths in C that having ρ as prefix and s as starting state. There is a unique extension of $\mu_{C,s}$ that satisfies the constraints on proba-

bility measures (i.e., countable additivity and universality), and we also denote this extension by $\mu_{C,s}$.

PROPOSITION 1. *Given a PO-DTMC C and a state s, the triple $(W_{C,s}, F_{C,s}, \mu_{C,s})$ defines a probability space.*

Now we lift the probability space $(W_{C,s}, F_{C,s}, \mu_{C,s})$ on a PO-DTMC C and a state s to the space $(W_{C,A,s}, F_{C,A,s}, \mu_{C,A,s})$ by considering the observation function of players A. More specifically, we let $W_{C,A,s} = \bigcup_{s' \in k_A(s)} W_{C,s'}$, $F_{C,A,s} = \bigcup_{s' \in k_A(s)} F_{C,s'}$, and

$$\mu_{C,A,s}(\mathcal{R}_{C,s'}(\rho)) = \begin{cases} PO_A(s') \times \mu_{C,s'}(\mathcal{R}_{C,s'}(\rho)) & \text{if } O_A(s) = O_A(s'), \\ 0 & \text{otherwise.} \end{cases}$$

where $O_A(s) = O_A(s')$ if and only if for all $i \in A$, $O_i(s) = O_i(s')$. Intuitively, $\mu_{C,A,s}(\mathcal{R}_{C,s'}(\rho))$ normalises those probabilities $\mu_{C,s'}(\mathcal{R}_{C,s'}(\rho))$ for s' by the observation function PO_A.

PROPOSITION 2. *Given a PO-DTMC C, a set A of players and a state s, the triple $(W_{C,A,s}, F_{C,A,s}, \mu_{C,A,s})$ defines a probability space.*

Given a PO-PCGS M and a complete uniform coalition strategy $\sigma_{Agt} = \{\sigma_i \mid \sigma_i$ is uniform, $i \in Agt\}$, we can obtain a uniform PO-DTMC $C = M(\sigma_{Agt})$. Two PO-DTMCs C_1 and C_2 are strategic equivalent with respect to a coalition strategy σ_A if players in A follow the same strategies, more specifically, if $C_1 = M(\sigma_{Agt}^1)$ and $C_2 = M(\sigma_{Agt}^2)$ then $\sigma_i^1 = \sigma_i^2 = \sigma_i$ for all $i \in A$. Let $C[M]$ be the set of PO-DTMCs that can be obtained from M and $C[M, \sigma_A]$ be the subset of $C[M]$ in which all PO-DTMCs are strategic equivalent with respect to coalition strategy σ_A.

Given a PO-PCGS M, a complete coalition strategy σ_{Agt}, a state s and a formula ϕ, we write

$$\mathcal{R}(M(\sigma_{Agt}), s, \phi) = \{\rho \in Path(M, \sigma_{Agt}) \mid \rho(0) = s, \ M, \sigma_{Agt}, \rho \models \phi\}$$

to be the set of fullpaths that start from the state s and satisfy the formula ϕ, and

$$\mathcal{R}(M(\sigma_{Agt}), A, s, \phi) = \bigcup_{s' \in k_A(s)} \mathcal{R}(M(\sigma_{Agt}), s', \phi)$$

to be the set of fullpaths that start from a state $s' \in k_A(s)$ and satisfy the formula ϕ. Based on them, we define a probabilistic notation $Pr(M, \sigma_{Agt}, A, s, \phi)$ as follows.

$$\begin{aligned} &Pr(M, \sigma_{Agt}, A, s, \phi) \\ =\ &\mu_{M(\sigma_{Agt}),A,s}(\mathcal{R}(M(\sigma_{Agt}), A, s, \phi) \mid \mathcal{R}(M(\sigma_{Agt}), A, s, \mathbb{T})) \quad (1) \\ =\ &\mu_{M(\sigma_{Agt}),A,s}(\mathcal{R}(M(\sigma_{Agt}), A, s, \phi)) \end{aligned}$$

It is a conditional probability of the set of fullpaths which satisfy the formula ϕ, given the set of paths that start from those states that are indistinguishable for the players in A from the current state s. The second equation holds because we have that

$$\begin{aligned} &\mu_{M(\sigma_{Agt}),A,s}(\mathcal{R}(M(\sigma_{Agt}), A, s, \mathbb{T})) \\ =\ &\mu_{M(\sigma_{Agt}),A,s}(\bigcup_{s' \in k_A(s)} \mathcal{R}(M(\sigma_{Agt}), s', \mathbb{T})) \\ =\ &\sum_{s' \in k_A(s)} PO(s') \times \mu_{M(\sigma_{Agt}),s'}(\mathcal{R}(M(\sigma_{Agt}), s', \mathbb{T})) \\ =\ &\sum_{s' \in k_A(s)} PO(s') \\ =\ &1 \end{aligned}$$

The relation $M, \sigma_{Agt}, \rho \models \phi$ is defined inductively as follows.

- $M, \sigma_{Agt}, \rho \models p$ if $p \in \pi(s(\rho, 0))$.

- $M, \sigma_{Agt}, \rho \models \neg\phi$ if not $M, \sigma_{Agt}, \rho \models \phi$.

- $M, \sigma_{Agt}, \rho \models \phi \wedge \psi$ if $M, \sigma_{Agt}, \rho \models \phi$ and $M, \sigma_{Agt}, \rho \models \psi$.

- $M, \sigma_{Agt}, \rho \models X\phi$ if $M, \sigma_{Agt}, \rho[1] \models \phi$.

- $M, \sigma_{Agt}, \rho \models \phi U \psi$ if there exists a time $m' \geq m$ such that $M, \sigma_{Agt}, \rho[m'] \models \psi$ and $M, \sigma_{Agt}, \rho[m''] \models \phi$ for all m'' with $m \leq m'' < m'$.

- $M, \sigma_{Agt}, \rho \models \mathrm{Pr}_i^{\bowtie d} \phi$ if we have $Pr(M, \sigma'_{Agt}, \{i\}, \rho(0), \phi) \bowtie d$ for all possible complete strategies σ'_{Agt}. Intuitively, to enable a probabilistic knowledge, the player i's subjective probabilistic measurement on the formula ϕ has to hold in all possible PO-DTMCs. Note that, all strategies include not only uniform strategies but also non-uniform strategies.

- $M, \sigma_{Agt}, \rho \models \langle\!\langle A \rangle\!\rangle^{\bowtie d} \phi$ if there exists a strategic equivalence class $C[M, \sigma'_A]$ for some uniform strategy σ'_A of players in A, such that for all PO-DTMC $C = M(\sigma'_{Agt}) \in C[M, \sigma'_A]$ with σ'_{Agt} a uniform strategy of all players Agt, we have that $Pr(M, \sigma'_{Agt}, A, \rho(0), \phi) \bowtie d$. Intuitively σ'_A represents a joint winning strategy of A such that for all joint opponent strategies, σ'_A enforces a win on every compatible state.

A formula ϕ is said to hold in M, written $M \models \phi$, if $M, \sigma_{Agt}, \rho \models \phi$ for all paths ρ such that $s(\rho, 0) = s_{init}$ and all complete coalition strategy σ_{Agt}. The model checking problem is then to determine, given a PO-PCGS M and a formula ϕ, whether $M \models \phi$.

The *combined complexity* is the complexity of determine $M \models \phi$, given a PO-PCGS M and a formula ϕ. The *model complexity* of a fixed formula ϕ is the complexity of determine $M \models \phi$, given a PO-PCGS M. This gives a measure of the complexity of model checking as a function of the size of the model. The *formula complexity* of a fixed model M is the complexity of determine $M \models \phi$, given a formula ϕ. This captures the contribution to the complexity of model checking that derives from the formula.

3.1 Relation with Perfect Recall

Before proceeding, let's compare the above definitions for observational semantics with those of perfect recall semantics. In [15], a semantics on probabilistic knowledge based on [10] is presented to work with fully-probabilistic systems. A fully-probabilistic system is a PO-DTMC and therefore contains no nondeterminism. In [17], the probabilistic strategy is defined in a logic PATL*, which is shown to be undecidable in model checking even for the single-player fragment.

Both the definitions of probabilistic knowledge operator and probabilistic strategy operator in the papers are based on a conditional probability

$$\mu_{r,m,i}^c(U) = \mu_c(\mathcal{R}(U) \mid \mathcal{R}(\mathcal{K}_i^c(r,m)))$$

with $U = \{(r', m') \mid (r', m') \in \mathcal{K}_i^c(r,m), \ \mathcal{I}, (r', m') \models \phi\}$. Interested readers can refer to the papers for formal treatments on the semantics. Roughly speaking, μ_c is a probability measure over a measurable set c of fullpaths, the set $\mathcal{R}(U)$ is the set of fullpaths that are indistinguishable to the player i and satisfy the formula ϕ, and the set $\mathcal{R}(\mathcal{K}_i^c(r,m))$ contains all fullpaths that are indistinguishable to the player i. Therefore, $\mu_{r,m,i}^c(U)$ is the probability of satisfying the formula ϕ under the condition of indistinguishability.

Our definition of $Pr(M, \sigma_{Agt}, A, s, \phi)$ in the equation (1) resembles this idea. The measure $\mu_{M(\sigma_{Agt}), A, s}$ is a probability measure over a measurable set $W_{M(\sigma_{Agt}), A, s}$ of fullpaths, the set $\mathcal{R}(M(\sigma_{Agt}), A, s, \phi)$ is the set of fullpaths whose starting states are indistinguishable to the state s and satisfy the formula ϕ, and the set $\mathcal{R}(M(\sigma_{Agt}), A, s, \mathbb{T})$ contains all fullpaths whose starting states are indistinguishable to the state s. Therefore, $Pr(M, \sigma_{Agt}, A, s, \phi)$ is also the conditional probability of satisfying the formula ϕ under the condition of indistinguishability.

Although similar ideas of defining the operators by conditional probability, their differences are significant if we look into the com-

ponents of the conditional probabilities. In perfect recall semantics, the indistinguishable set $\mathcal{K}_i^c(r,m)$ changes over the time, which makes the sets $\mathcal{R}(U)$ and $\mathcal{R}(\mathcal{K}_i^c(r,m))$, and therefore the conditional probability $\mu_{r,m,i}^c(U)$, also change over the time. However in observational semantics, $\mathcal{R}(M(\sigma_{Agt}), A, s, \mathbb{T})$ and $\mathcal{R}(M(\sigma_{Agt}), A, s, \phi)$, and therefore $Pr(M, \sigma_{Agt}, A, s, \phi)$, do not change with the time.

The differences can also be witnessed in the definitions of PO-PCGSs and the ways how to compute the two conditional probabilities. In perfect recall semantics, because the conditional probability changes over the time, a recursive procedure is developed to compute it from the conditional probability of previous round, as is shown in the Theorem 1 of [17] that

$$\mu_{r,m+1,i}^c(U) = \mu_{r,m,i}^c(U_{r,m,i} \mid \mathcal{K}_i^c(r, m+1)),$$

where $U_{r,m,i} = \{(r', m') \in \mathcal{K}_i^c(r,m) \mid \exists m'' : (r', m'') \in U\}$. Therefore, in a PO-PCGS, only the initial distribution PI such that $\sum_{s \in S} PI(s) = 1$ and the non-probabilistic observation function O_i are needed to define the measure $\mu_{r,0,i}^c$. In observational semantics, as the conditional probability does not change over the time, in a PO-PCGS M, a probabilistic observation function PO_i is needed for every player $i \in Agt$. Intuitively, the observation function PO_i assumes a prior probability distribution over those states indistinguishable for player i.

The above differences demonstrate the potentially different applications of the observational semantics and the perfect recall semantics. In a system having both nondeterminism and probability, a nondeterministic choice may represent an unspecified probabilistic distribution that one of the players does not know when the game starts. By taking perfect recall semantics, the player can gradually discover it by reasoning about its past observations and local actions, as the evidence theory discussed in [21]. On the other hand, if the nondeterministic choices are inherent, i.e., there should not be a single distribution on it, then we may use observational semantics to design a strategy based on it. In this case, the nondeterminism is resolved by the strategy of the players.

In a practical scenario, they may be used in two phases of reasoning: the perfect recall semantics can be used to reason about some information and then these information are used as a prior for the observational semantics to design a strategy. For example, in a pursuit-evasion game that will be described in the next section, a subgame of perfect recall can be used to track the appearance probability of the evader in specific area (e.g., 70% in area 1 and 30% in area 2) and then a subgame of observational semantics is used to see if we can have a successful strategy based on this prior (e.g., in case the appearance probability of the evader is higher in area 1, we should deploy more pursuers in that area).

4. APPLICATIONS

In the following, we present some examples to demonstrate the applications of the PATEL* logic of observational semantics.

4.1 A Simple Example

We take the simple example from [17] as our first example. The system $M = (S, s_{init}, Act_i, N_i, PO_i, PT, \pi)$ is a system of a single player i, where $S = \{s_{init}, s_1, s_2, s_3, s_4, s_5\}$, $Act_i = \{h, t\}$, and $N_i(s) = \{h, t\}$ for all states s. s_4 is the only state in which proposition p holds. The transition matrix PT is shown in Figure 1, where states s_i and s_j are connected by an arrow labeled with (act, p_k) if $PT(s_i, act, s_j) = p_k \neq 0$. The observation function PO_i are defined as $PO_i(s_2) = 1/3$ and $PO_i(s_3) = 2/3$, and $PO_i(s_i) = 1$ for $s_i \in S \setminus \{s_2, s_3\}$. Intuitively, player i can distinguish every two states except for s_2 and s_3.
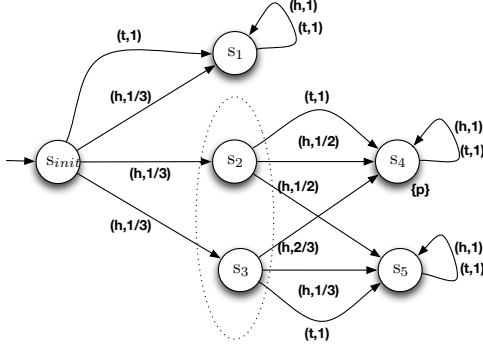
Figure 1: A Simple Example

We discuss three strategies σ_1, σ_2 and σ_3 by omitting irrelevant choices at the terminal states: (1) $\sigma_1(s_{init}) = t$; (2) $\sigma_2(s_{init}) = h$, $\sigma_2(s_2) = \sigma_2(s_3) = t$; (3) $\sigma_3(s_{init}) = h$, $\sigma_3(s_2) = \sigma_3(s_3) = h$. Note that $\sigma(s_2) = \sigma(s_3)$ for all uniform strategies σ.

Initially, player $\mathtt{i}$ does not have a strategy to eventually reach state s_4 in a probability more than $1/2$.

$$M \models \neg \langle\!\langle \mathtt{i} \rangle\!\rangle^{>1/2}(Fp) \qquad (2)$$

The probabilities of satisfying Fp by taking the three strategies are 0, $1/3$ and $7/18$, respectively. For example, if player $\mathtt{i}$ takes strategy σ_2 that decides a unique PO-DTMC C_1, then we have that $Pr(M, \sigma_2, \{\mathtt{i}\}, s_{init}, Fp) = \mu_{C_1, \{\mathtt{i}\}, s_{init}}(\mathcal{R}(C_1, \{\mathtt{i}\}, s_{init}, Fp)) = \mu_{C_1, s_{init}}(\mathcal{R}(C_1, s_{init}, Fp)) = 1/3$.

On the other hand, the player has a strategy to reach in a probability more than $1/2$ those next states, from which it has a strategy to reach state s_4 in a probability more than $1/2$, as expressed in the following expression.

$$M \models \langle\!\langle \mathtt{i} \rangle\!\rangle^{>1/2} X \langle\!\langle \mathtt{i} \rangle\!\rangle^{>1/2} Fp \qquad (3)$$

One may find that the first $\langle\!\langle \mathtt{i} \rangle\!\rangle$ can be enforced by either σ_2 or σ_3, and the second is enforced by σ_3. The first step is to reach from state s_{init} to states s_2 or s_3, which can be done in the probability $2/3$ by taking the action h. To show the second, we show that $M, s_2 \models \langle\!\langle \mathtt{i} \rangle\!\rangle^{\geq 1/2} Fp$ (and therefore $M, s_3 \models \langle\!\langle \mathtt{i} \rangle\!\rangle^{\geq 1/2} Fp$ since $O_{\mathtt{i}}(s_2) = O_{\mathtt{i}}(s_3)$). By fixing the strategy σ_3 that decides a unique PO-DTMC C_2, we have that $Pr(M, \sigma_3, \{\mathtt{i}\}, s_2, Fp) = \mu_{C_2, \{\mathtt{i}\}, s_2}(\mathcal{R}(C_2, \{\mathtt{i}\}, s_2, Fp)) = 1/3 \times \mu_{C_2, s_2}(\mathcal{R}(C_2, s_2, Fp)) + 2/3 \times \mu_{C_2, s_3}(\mathcal{R}(C_2, s_3, Fp)) = 1/3 * 1/2 + 2/3 * 2/3 = 11/18 \geq 1/2$.

4.2 Pursuit Evasion Games

Pursuit-Evasion games are a type of multi-player games in which one or more pursuers have the objective of identifying the presence of one or more evaders, and of capturing them. They are recently approached by the model checking of temporal epistemic logics [16, 18], which do not take probabilistic information into consideration and do not explicitly work with strategy operator. Here we describe a simple scenario with probabilistic information and characterise properties with PATEL* formulas.

Let $G = (V, E)$ be a discrete graph consisting of a set V of positions and a set E of edges connecting positions. Assume that there are several players, a set D of pursuers and an evader $\mathtt{ak}$. Every player i stays at some position pos_i. The evader is captured by the pursuers if it is in the same position with one of the pursuers, i.e.,

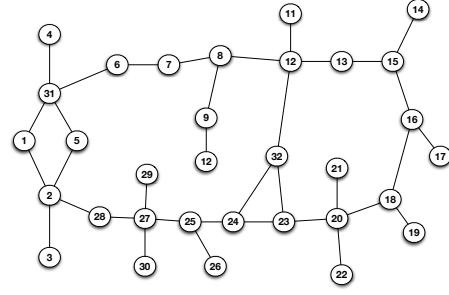$$capture \equiv \bigvee_{i \in D} pos_{\mathtt{ak}} = pos_i.$$

Figure 2: A Graph for Pursuit Evasion Games

Let v_0 and v_0' be two positions that do not in the graph G, that is, $\{v_0, v_0'\} \cap V = \emptyset$. Let $\bot \notin \Phi$ denote the empty strategy. A game state is a tuple $s = (\{pos_i\}_{i \in D}, \{\tau_i\}_{i \in D}, pos_{\mathtt{ak}})$ such that $pos_i \in V \cup \{v_0\}$ and $\tau_i \in \Phi \cup \{\bot\}$ for $i \in D$, and $pos_{\mathtt{ak}} \in V \cup \{v_0'\}$. Initially, all pursuers appear at position v_0, all pursuers' strategies are empty strategy, and the evader appears at position v_0'. Formally, $s_{init} = (\{v_0\}_{i \in D}, \{\bot\}_{i \in Agt}, v_0')$.

From the initial state s_{init}, the pursuers move to a specific position $v_1 \in V$ and the evader moves to any other position $v_1' \in V$ such that $v_1' \neq v_1$. In the mean time, every pursuer chooses a strategy from $\Phi = \{\tau_1, ..., \tau_m\}$ by a prior distribution μ_Φ such that $\sum_{\tau \in \Phi} \mu_\Phi(\tau) = 1$. From a game state $s \neq s_{init}$, a player can either moves to one of its neighbouring positions or stays still. The pursuers' strategies do not change. All strategies τ in Φ are deterministic strategies, i.e., for any position $v \in V$, $\tau(v) \in \{v\} \cup \{v' \mid (v, v') \in E\}$. Intuitively, the probabilistic transitions occur at the beginning of the game, when every pursuer is assigned with a strategy to follow by a prior distribution μ_Φ on a set of deterministic strategies. After that, every pursuer follows the assigned strategy.

As discussed at the end of Section 3.1, the prior distribution μ_Φ may be obtained from a pre-game, where the pursuers discover (by perfect recall semantics) the appearance probability of the evader in several areas. A deterministic strategy that patrols an area where the evader has higher appearance probability in the pre-game should be given higher prior probability in the game.

The observation functions are defined as follows. For the pursuer $i \in D$, we let $O_i((\{pos_i\}_{i \in D}, \{\tau_i\}_{i \in D}, pos_{\mathtt{ak}})) = (pos_i, \tau_i)$. For the evader $\mathtt{ak}$, we let $O_{\mathtt{ak}}((\{pos_i\}_{i \in D}, \{\tau_i\}_{i \in D}, pos_{\mathtt{ak}})) = (\{pos_i'\}_{i \in Agt}, pos_{\mathtt{ak}})$ such that $pos_i' = pos_i$ if $(pos_i, pos_{\mathtt{ak}}) \in E$ and $pos_i' = \bot$, otherwise. Intuitively, the pursuers can only observe their current positions and their strategies, and the evader $\mathtt{ak}$ can observe the neighboring nodes, but is not able to observe the strategies of the players in D.

Now consider an area as described in Figure 2 [18]. An example strategy τ is denoted as a patrolling path $\rho_\tau = 1 \to 2 \to 28 \to 27 \to 25 \to 24 \to 23 \to 20 \to 18 \to 16 \to 15 \to 13 \to 12 \to 8 \to 7 \to 6 \to 31 \to 1$ such that $\tau(s) = t$ if $(s \to t) \in \rho_\tau$.

In such a game, the objective of the evader $\mathtt{ak}$ is to avoid being captured by the pursuers D. An interesting specification is

$$\langle\!\langle \{\mathtt{ak}\} \rangle\!\rangle^{>0.8} G(\neg capture) \qquad (4)$$

which expresses that the attacker $\mathtt{ak}$ has a strategy that in a probability of more than 80% to avoid being captured. Another specification can be

$$G \neg \mathtt{Pr}_{\mathtt{ak}}^{\geq 0.9} F(\neg capture) \qquad (5)$$

which says that the players $\mathtt{ak}$ can never have a knowledge that he will win the game in a probability no less than 90%.

5. MODEL CHECKING COMPLEXITY

In this section, we will solve the computational complexities of model checking PATEL* logic and its sublogics.

5.1 Probabilistic Knowledge

First, we will consider the impact of adding probabilistic knowledge operator into the logics PCTL and PCTL*. Consider the relation $M, \sigma_{Agt}, \rho \models \text{Pr}_i^{\leq d} \phi$. By definition, we need to compute

$$
\begin{aligned}
& \max_{\sigma_{Agt}} Pr(M, \sigma_{Agt}, \{i\}, \rho(0), \phi) \\
\equiv\ & \max_{\sigma_{Agt}} \mu_{M(\sigma_{Agt}),\{i\},\rho(0)}(\mathcal{R}(M(\sigma_{Agt}), \{i\}, \rho(0), \phi)) \\
\equiv\ & \max_{\sigma_{Agt}} \mu_{M(\sigma_{Agt}),\{i\},\rho(0)}(\bigcup_{s \in k_i(\rho(0))} \mathcal{R}(M(\sigma_{Agt}), s, \phi)) \\
\equiv\ & \max_{\sigma_{Agt}} \sum_{s \in k_i(\rho(0))}(PO_i(s) \times \mu_{M(\sigma_{Agt}),s}(\mathcal{R}(M(\sigma_{Agt}), s, \phi)))
\end{aligned}
\tag{6}
$$

If working with a formula of nested probabilistic knowledge, for example $\text{Pr}_i^{\leq d_1}(\phi_1 \wedge F\ \text{Pr}_j^{\leq d_2}\phi_2)$, then the model checking starts from the innermost subformulas (i.e., the most nested ones) and then goes to the outer ones. Therefore, the complexity of model checking a nested formula is polynomial with respect to both the size of formula and the complexity of model checking a subformula without nested probabilistic knowledge operators.

THEOREM 1. *Model checking PCTL*K_n in PO-PCGSs is 2-EXPTIME-complete for combined complexity, P-complete for model complexity, and 2-EXPTIME-complete for formula complexity.*

Proof: (sketch) By [22], an LTL formula ϕ can be converted into a deterministic Rabin automata $\mathcal{A}_\phi$ by expanding its size by doubly exponential. Then the size of product automata $M \times \mathcal{A}_\phi$ is polynomial with respect to the model M and doubly exponential with respect to the size of formula ϕ. Let $\mathcal{A} = (V, \mu_V)$ be the product automata. An end component is a set of states $T \subseteq V$ such that 1) for all $s \in T$ and $a \in Act$, if $\mu_V(s, a, t) > 0$ then $t \in T$, and 2) the underlying graph of (T, μ_V) is strongly connected. A maximal end-component is an end-component that is maximal under set inclusion. The computation of maximal end components can be done in polynomial time by a variant of Tarjan's algorithm.

The component $\mu_{M(\sigma_{Agt}),s}(\mathcal{R}(M(\sigma_{Agt}), s, \phi)$ can be rewritten into the reachability probability of the set of maximal end components of the automata $M \times \mathcal{A}_\phi$. A reachability probability can be translated into the solution of a set of linear equations of polynomial size [2]. Therefore, the computation of equation (6) on a PO-PCGS M is then reduced to the solution of the maximal value of a linear combination of reachability probabilities. As the complexity of linear programming is polynomial, we have that model checking PCTL*K_n can be solved in 2-EXPTIME. Note that in equation (6), σ_{Agt} ranges over all memoryless strategies, including non-uniform ones. A polynomial reduction to linear programming problem does not exist if σ_{Agt} ranges over all uniform strategies, since the complexity of finding a uniform strategy is NP-complete [24].

The lower bound is matched by the LTL model checking over MDP [7], which is 2-EXPTIME-complete for combined complexity.

Model Complexity. If we fix the formula ϕ, the product automata $M \times \mathcal{A}_\phi$ is linear with respect to the model M and therefore the model complexity is in P.

The lower bound can be reduced from the monotone circuit value problem, which is P-complete (cf. [9]). It takes as input a monotone boolean circuit C and an input δ, to determine if the output value $C(\delta) = 1$. A monotone boolean circuit $C = (B, B_I, b_O, T, op)$ is a directed acyclic graph, in which B is a set of gates, $B_I \subseteq B$ is a set of input gates, b_O is a single output gate, and $T : B \to \mathcal{P}(B)$ is directed connections between gates such that

- $\forall b \in B_I : \{b' \in B \mid b \in T(b')\} = \emptyset$ (the indegree of input gates is zero),
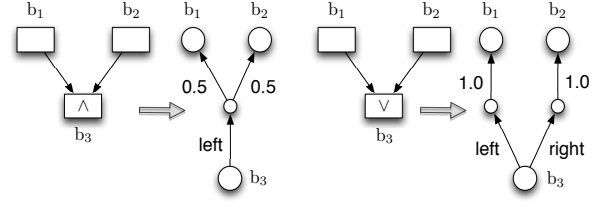
Figure 3: The transformation from boolean circuit to concurrent game structure

- $T(b_O) = \emptyset$ (the outdegree of the output gate is zero), and

- $\forall b \in B \setminus B_I : |\{b' \in B \mid b \in T(b')\}| = 2$ (the indegree of non-input gates is 2).

and $op : (B \setminus B_I) \to \{\wedge, \vee\}$ is a labelling function mapping every non-input gate to a boolean operator $\wedge$ or $\vee$. An input of the circuit C is an assignment $\delta : B_I \to \{0, 1\}$ of boolean values to input gates. The value of a non-input gate is computed as the result of boolean operation (the one on its label) on the values of its ancestors. The output value of the circuit $C(\delta)$ is the value on the output gate.

Let $M = (S, s_{init}, Act_\mathbf{i}, N_\mathbf{i}, PO_\mathbf{i}, PT, \pi)$ be the concurrent game structure of a single player $\mathbf{i}$, such that 1) $S = B$, 2) $s_{init} = b_O$, 3) $Act_\mathbf{i} = \{left, right\}$, 4) $N_\mathbf{i}(b) = \{left, right\}$, if $op(b) = \vee$, and $N_\mathbf{i}(b) = \{left\}$, otherwise, 5) $PO_\mathbf{i}(b) = 1$ for all $b \in S$, 6) $p \in \pi(b)$ iff $\delta(b) = 1$, and 7) PT is obtained by the transformations as displayed in Figure 3. Intuitively, we reverse the directions of arrows in C and turn the boolean operations into probabilistic transition relations.

Finally, the circuit valuation problem $C(\delta) = 1$ is equivalent to the model checking problem of $M \models \langle\langle\{\mathbf{i}\}\rangle\rangle^{\geq 1} F\ p$. The later is then equivalent to $M \models \langle\langle Agt\rangle\rangle^{\geq 1} F\ p$ since $Agt = \{\mathbf{i}\}$.

Formula Complexity. If we fix the model M, the product automata $M \times \mathcal{A}_\phi$ is still doubly-exponential with respect to ϕ and therefore results in the formula complexity in 2-EXPTIME. The lower bound follows from the conclusion in [7] for LTL model checking. $\square$

THEOREM 2. *Model checking PCTLK$_n$ in PO-PCGSs is P-complete for combined and model complexity, and in P for formula complexity.*

Proof: (sketch) By following a similar procedure as that of [11], the computation of the component $\mu_{M(\sigma_{Agt}),s}(\mathcal{R}(M(\sigma_{Agt}), s, \phi)$ can be reduced into the solution of a set of linear equations of polynomial size with respect to both the model M and the formula ϕ. Then by a similar procedure as the case of PCTL*K_n, the computation of equation (6) can be reduced to the solution of a linear programming problem of polynomial size with respect to both the PO-PCGS M and formula ϕ.

The lower bounds for the combined and model complexity are obtained the same as the lower bound for the model complexity of PCTL*K_n logic, by noticing that $\langle\langle Agt\rangle\rangle^{\geq 1} F\ p$ is also a PCTLK$_n$ formula. $\square$

Note that, the proof for the lower bound of the model complexity of checking PCTL*K_n logic also implies the P-hardness as the combined and model complexity of checking PCTL and PCTL* logics over MDPs, which is not unknown but does not appear in the literature. The model M is indeed an MDP and the expression $C(\delta) = 1$ is equivalent to $M \not\models \text{Pr}^{<1}(F\ p)$ in their terms.

5.2 Probabilistic Strategy

Second, we will consider the impact of adding probabilistic strategy operator into the logics PCTL*K_n and PCTLK$_n$. Similarly, the model checking is conducted from the innermost subformulas towards the outer ones.

THEOREM 3. *Model checking PATEL* in PO-PCGSs is 2-EXPTIME-complete for combined complexity, Σ_2^P-complete for model complexity, and 2-EXPTIME-complete for formula complexity.*

Proof: (sketch) Model checking can be done by induction on the structure of the formula. Given a relation $M, \sigma_{Agt}, \rho \models \langle\!\langle A \rangle\!\rangle^{\leq d} \phi$, we can decide by

1. guessing a strategy for players A, let it be σ_A, and then

2. computing $\max_{\sigma_{Agt\setminus A}} Pr(M, \sigma_{Agt}, \{i\}, \rho(0), \phi)$.

As stated in the probabilistic knowledge case, the second line can be reduced to a linear programming problem whose equations represent the model checking of the LTL formula ϕ in a MDP. From [7], model checking LTL formula on MDPs is 2-EXPTIME-complete for the formula and in P for the model. It uses an alternating turing machine of exponential space over the size of the formula to compute the result. Enhancing with the above algorithm, a further polynomial space over the model is requested to store the strategy. Put them together, the model checking can be implemented by taking an alternating turing machine of exponential space over the size of the formula and polynomial space over the size of the model, which added up to a AEXPSPACE=2-EXPTIME complexity. This complexity is hard as the LTL model checking has been 2-EXPTIME-complete.

Model Complexity. For the model complexity, we decide the relation $M, \sigma_{Agt}, \rho \models \langle\!\langle A \rangle\!\rangle^{\leq d} \phi$ by

1. guessing a strategy for players A, let it be σ_A, and

2. reversing the result of the following procedure:

   (a) guess a strategy for players $Agt \setminus A$, let it be $\sigma_{Agt\setminus A}$, and

   (b) return the result of $Pr(M, \sigma_{Agt}, \{i\}, \rho(0), \phi) > d$.

Note that the model complexity of PCTL*K_n logic is in P. Therefore, the above procedure gives a NP$^{NP} = \Sigma_2^P$ procedure.

The lower bound can be reduced from the satisfiability of quantified Boolean formulas (QBF) of 2 alternations, which is Σ_2^P-complete. Given a boolean formula f with variables partitioned into 2 sets V_1 and V_2, to determine if $\exists V_1 \forall V_2 : f(V_1 \cup V_2)$. Assume that f is in its conjunctive normal form, that is, $f = f_1 \wedge ... \wedge f_m$ such that $f_k = v_{k,1} \vee ... \vee v_{k,l}$ for $1 \leq k \leq m$, where $v_{k,j}$ is a variable in $V_1 \cup V_2$ or its negation.

The technical details of the reduction are omitted for the space limit and will appear in the longer version of the paper. The basic idea is: for each variable v, we use a player ag_v to control its truth value. The uniformity of the players' strategies are used to make sure that all appearances of a variable take the same truth value as the one decided by the players. The environment e controls the selection of clause f_k. The evaluation of a clause is simulated as a sequential evaluation of the literals such that if the literal is evaluated as false then it moves on to the next literal and if it is evaluated as true then it moves to a specific successful state $s_\top$ where the atomic proposition p holds.

Finally, the satisfiability of QBF formulas of 2 alternations is equivalent to the model checking problem of $M \models \langle\!\langle \{ag_v \mid v \in V_1\} \rangle\!\rangle^{\geq 1} F\ p$.

Formula Complexity. If we fix a PO-PCGS M, the algorithm for combined complexity also gives a 2-EXPTIME procedure for the formula complexity. The hardness follows from that of LTL model checking over MDPs. □

THEOREM 4. *Model checking PATEL in PO-PCGSs is Δ_3^P-complete for combined complexity, Σ_2^P-complete for model complexity, and in P for formula complexity.*

Proof: (sketch) We reuse the algorithm for the model complexity in the proof of Theorem 3 to obtain the upper bound. First of all, because we are dealing with PCTLK$_n$ logic instead of PCTL*K_n logic, as stated in Theorem 2, the computation of the component $Pr(M, \sigma_{Agt}, \{i\}, \rho(0), \phi) > d$ takes polynomial time. Secondly, as stated in Theorem 3, the algorithm in itself is a Σ_2^P procedure with respect to the size the model M. Thirdly, for a formula of nested operators, a polynomial procedure with respect to the size of the formula is needed to query a Σ_2^P oracle implemented by the algorithm. Put them together, the combined complexity of checking PCTLK$_n$ logic is in $P^{\Sigma_2^P} = \Delta_3^P$.

The lower bound is matched by the complexity of model checking ATL logic on MDPs.

Model Complexity. A similar argument as that of PATEL* logic can be made to have the Σ_2^P-complete model complexity.

Formula Complexity. If the model M is fixed, the algorithm for combined complexity gives a polynomial procedure for checking PATEL formulas. □

The above complexity results suggest that with the addition of probabilistic strategy operator, the model complexity increases from P-complete to Σ_2^P-complete, while the formula complexity remains. Also, from PCTLK$_n$ to PATEL, the combined complexity increases from P-complete to Δ_3^P-complete.

6. RELATED WORKS

Both probabilistic verification and the logics of knowledge and strategy are active research areas. We only discuss those works that are closely related to the paper. For complete information systems, [3] presents a model checking algorithm for PCTL* and PCTL over MDP by considering the temporal operators and probability measure. The algorithm has been implemented in the tool PRISM [12]. This has been extended in [5] to PATL and PATL* to deal with strategic operator, and implemented as PRISM-games [4].

For incomplete information systems and perfect recall semantics, [17] proved that model checking PATEL* and PATEL is undecidable even for single-player fragment, [14] shows the undecidability for a probabilistic temporal epistemic logic PLTLK$_n$, and [15] gives a symbolic MTBDD-based algorithm for a small fragment of the PCTL*K_n logic in fully-probabilistic systems. For observational semantics, [25] proposed a logic that combines the knowledge operator with a probabilistic strategic operator. The semantics is substantially different with the one presented in the paper in that 1) they deal with knowledge operator instead of probabilistic knowledge operator and 2) the interpretation of probabilistic strategic operator is not related to the indistinguishable relation between states.

The incomplete information may result in several different interpretations on the statement that a set of players have a strategy, including the existence of a strategy [13, 25], the existence of a consistent strategy [19], knows the existence of a consistent strategy but don't know how to play, and knows not only the existence of a strategy but also how to play, see e.g., [20]. In the paper, we assume the last one.

7. CONCLUSIONS AND FUTURE WORK

We present a logic PATEL* to reasoning about the probabilistic knowledge and the probabilistic strategy in stochastic multiagent systems. Several examples are given to show its applications, and the model checking complexities of the logic and its sub-logics are addressed.

Several issues are left open by this work. The first one is a complete axiomatic system for the logic PATEL*. Especially, we need a set of axioms for the probabilistic variant of distributed knowledge. They are expected to be a natural generalisation of the axioms for distributed knowledge. The second one is the way how to retrieve the observation function PO_i for the agents $i \in Agt$. In the paper, they are treated as extra input. The ideal approach is, as briefly discussed in the paper, to retrieve them from a pre-computation which takes as input the stochastic multiagent system with the function O_i for $i \in Agt$. The third one is a practical model checking algorithm and its implementation. It is interesting to investigate its scalability in dealing with practical examples.

Acknowledgement

8. REFERENCES

[1] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-Time Temporal Logic. *Journal of the ACM*, 49(5):672–713, 2002.

[2] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[3] Andrea Bianco and Luca de Alfaro. Model Checking of Probabalistic and Nondeterministic Systems. In *15th Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 1026 of *Lecture Notes in Computer Science*, pages 499–513, 1995.

[4] Taolue Chen, Vojtěch Forejt, Marta Kwiatkowska, David Parker, and Aistis Simaitis. Prism-games: A model checker for stochastic multi-player games. In *19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'13)*, 2013.

[5] Taolue Chen and Jian Lu. Probabilistic Alternating-time Temporal Logic and Model Checking Algorithm. In *FSKD (2)*, pages 35–39, 2007.

[6] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. The MIT Press, 1999.

[7] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

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

[9] A. Gibbons and W. Rytter. *Efficient Parallel Algorithms*. Cambridge University Press, 1988.

[10] Joseph Y. Halpern and Mark R. Tuttle. Knowledge, probability, and adversaries. *Journal of the ACM*, 40:917–960, 1993.

[11] Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal Asp. Comput*, 6(5):512–535, 1994.

[12] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In *Proc. Conf on Tools and Algorithms for the Construction and Analysis of Systems*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.

[13] W. Hoek and M. Wooldridge. Tractable multiagent planning for epistemic goals. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, pages 1167–1174, 2002.

[14] Xiaowei Huang. Diagnosability in concurrent probabilistic systems. In *12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS2013)*, 2013.

[15] Xiaowei Huang, Cheng Luo, and Ron van der Meyden. Symbolic Model Checking of Probabilistic Knowledge. In *13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK XII)*, pages 177–186, 2011.

[16] Xiaowei Huang, Patrick Maupin, and Ron van der Meyden. Model checking knowledge in pursuit-evasion games. In *the 22nd International Joint Conference on Artificial Intelligence (IJCAI2011)*, pages 240–245, 2011.

[17] Xiaowei Huang, Kaile Su, and Chenyi Zhang. Probabilistic Alternating-time Temporal Logic of Incomplete information and Synchronous Perfect Recall. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, pages 765–771, 2012.

[18] Xiaowei Huang and Ron van der Meyden. Synthesizing Strategies for Epistemic Goals by Epistemic Model Checking: An Application to Pursuit Evasion Games. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, pages 772–778, 2012.

[19] Wojciech Jamroga. Some Remarks on Alternating Temporal Epistemic Logic. In *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, 2003.

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

[21] Riccardo Pucella Joseph Y. Halpern. A Logic for Reasoning about Evidence. *Journal of Artificial Intelligence Research*, 26:1–34, 2006.

[22] Orna Kupferman and Adin Rosenberg. The blow-up in translating ltl to deterministic automata. In Ron Meyden and Jan-Georg Smaus, editors, *Model Checking and Artificial Intelligence*, volume 6572 of *Lecture Notes in Computer Science*, pages 85–94. Springer Berlin Heidelberg, 2011.

[23] François Laroussinie, Nicolas Markey, and Ghassan Oreiby. On the Expressiveness and Complexity of ATL. *Logical Methods in Computer Science*, 4(2), 2008.

[24] Michael L. Littman. Memoryless policies: Theoretical limitations and practical results. In *the third international conference on Simulation of adaptive behavior (SAB94)*, pages 238–245, 1994.

[25] Henning Schnoor. Strategic Planning for Probabilistic Games with Incomplete Information. In *the proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 1057–1064, 2010.

[26] Ron van der Meyden and Nikolay V. Shilov. Model Checking Knowledge and Time in Systems with Perfect Recall. In *Foundations of Software Technology and Theoretical Computer Science*, pages 432–445, 1999.

[27] Moshe Y. Vardi. Automatic Verification of Probabilistic Concurrent Finite-State Programs. In *FOCS*, pages 327–338, 1985.