A Process Algebraic Framework for Multi-Agent Dynamic Epistemic Systems

Alessandro Aldini[1,*]

[1]University of Urbino Carlo Bo, Piazza della Repubblica 13, 61029, Urbino, Italy

Abstract

This paper combines the classical model of labeled transition systems with the epistemic model for reasoning about knowledge. The result is a unifying framework for modeling and analyzing multi-agent, knowledge-based, dynamic systems. On the modeling side, we propose a process algebraic, agent-oriented specification language that makes such a framework easy to use for practical purposes. On the verification side, we define a modal logic encompassing temporal and epistemic operators.

Keywords

Labeled transition system, Kripke model, epistemic model, modal logic, process algebra

1. Introduction

The formal modeling of agent-based systems and the knowledge transfer enabled by the related interactions is a research field common to several areas, ranging from concurrency theory to epistemic logic.

In the former setting, two basic models are mainly adopted to describe the dynamics of systems: (i) *Kripke structures* are graphs where the nodes are annotated with atomic propositions stating what is true in the system state associated with the node, and (ii) *labeled transition systems* (LTSs) are graphs where the arcs are annotated with actions representing the events causing a change of system state. Both paradigms are equipped with temporal logics for the description of properties, like, e.g., Computation Tree Logic (CTL) for state-based structures [1] and Hennessy-Milner Logic (HML) for action-based systems [2].

In the latter setting, the focus is on reasoning about knowledge from the viewpoint of non-omniscient agents in terms of their capability of distinguishing different scenarios [3]. The standard way to model epistemic notions is through a state-based epistemic model called *Kripke model*. Every state (called *possible world*) is characterized by the propositional statements that hold in it, as in Kripke structures. At the same time, an *accessibility* relation determines, from the viewpoint of the agent under consideration, which worlds are compatible (*indistinguishable*) with her knowledge in the current world. In this setting, epistemic logic introduces a knowledge modality for reasoning about what agents know or can deduce from the information at their disposal and, possibly, for tracking the information flow among agents.

ICTCS'24: Italian Conference on Theoretical Computer Science, September 11–13, 2024, Torino, Italy

*Corresponding author.

alessandro.aldini@uniurb.it (A. Aldini)

https://www.uniurb.it/persone/alessandro-aldini (A. Aldini)

0000-0002-7250-5011 (A. Aldini)

The connections between the two strands of research are evident and, in some cases, the mutual foundational influence between traditional concurrency models and epistemic models is investigated (see, e.g., [4, 5, 6, 7]). Specific examples of cross-fertilization can be found in the formal analysis of security protocols; see [8] for a survey and, in particular, [9, 10, 11, 12], where logical formalizations of knowledge are integrated into modeling frameworks based on pi-calculus in order to characterize the intruder's capability of breaking security properties.

The main goal of the paper is to combine the advantages of the two approaches by merging in a novel framework the capability of the LTS-based semantics of modeling dynamic, temporal behaviors with the capability of the epistemic models of representing what agents know or do not know. The rough idea behind the combination is associating a Kripke model with each state of an LTS. As additional contributions, this novel framework is enriched with a logic including dynamic and epistemic modalities and a high-level, process-algebraic specification language.

In the following, we introduce the model of Kripke labeled transition systems (Section 2) as a combination of epistemic models and LTSs. We define a logic for describing properties for such a model and establish the equivalence relation that is characterized by the logic. Then, we propose a process algebraic language for modeling agent-oriented concurrent systems with semantics based on Kripke labeled transition systems (Section 3). To emphasize the usability of this language, we describe a case study based on a popular, classical board game (Section 4). Finally, we discuss related work and potential future directions (Section 5).

2. Kripke labeled transition systems

Let $\mathcal{A}$ be a set of agents (ranging over $i, j, \ldots$), Act a set of actions (ranging over $\pi, \pi', \ldots$), and At a set of atomic propositions (ranging over $p, q, \ldots$); we will use $X, Y, \ldots$ to denote subsets of At. First of all, we recall the definitions of labeled transition system and multi-agent epistemic model.

Definition 1. *A labeled transition system (LTS) is a tuple (S, T, s_0) where S is a non-empty set of states (with s_0 the initial state) and $T \subseteq S \times Act \times S$ is the action-labeled transition relation.*

In the setting of computation modeling, LTSs describe the evolving behavior of discrete systems, where the actions labeling the transitions represent events leading from one configuration of the system to another.

Definition 2. *A multi-agent epistemic model (called Kripke model) is a tuple $(S, \{R_i \mid i \in \mathcal{A}\}, v)$, where S is a non-empty set of states; for every $i \in \mathcal{A}$, $R_i \in 2^{S \times S}$ is a binary (accessibility) relation over S; $v : S \to 2^{At}$ is a valuation function assigning to each state the set of propositions that hold in the state.*

A pointed (resp., rooted) Kripke model is a pair $((S, \{R_i \mid i \in \mathcal{A}\}, v), s)$, where $s \in S$ is the current (resp., initial) state. Kripke models serve as the basis of the semantics for various modal logics and, in the case of epistemic languages, allow us to reason about knowledge in terms of information accessibility.

For our purposes, combining the dynamic action-based nature of LTSs with the possible worlds description of Kripke models results in action-based systems, the states of which are associated with accessibility relations and valuations.

Definition 3. *A Kripke labeled transition system (KLTS) is a tuple* $(S, T, \{r_i \mid i \in \mathcal{A}\}, v)$*, where* S *is a non-empty set of states;* $T \subseteq S \times Act \times S$ *is a transition relation; for every* $i \in \mathcal{A}$*,* $r_i : S \to 2^{2^{At} \times 2^{At}}$ *is a function mapping each state to a binary (accessibility) relation over* 2^{At}*;* $v : S \to 2^{At}$ *is a valuation function.*

Pointed and rooted KLTSs are defined as expected. Firstly, states should not be considered dependent on atomic propositions. They are primitive semantic objects so that the set of propositions satisfied by a state does not uniquely identify the state. Secondly, each accessibility relation $r_i(s)$ relates elements of 2^{At} and expresses the actual observational power of agent i in state s with respect to the truth of the propositions in At. In other words, $r_i(s)$ describes the distinguishing power of agent i in s, intended as her capability of distinguishing the possible worlds identified by the values of the propositions. Under the indistinguishability interpretation of epistemic logic, $r_i(s)$ expresses informational indistinguishability between possible worlds. More precisely, $(X, Y) \in r_i(s)$ means that in s the agent i has insufficient information to establish whether we are in a state in which all and only the propositions of X hold or in a state in which all and only the propositions of Y hold. Hence, both X and Y are compatible with the knowledge of the agent i in s. By virtue of this interpretation, in the following we assume that the accessibility relations are equivalence relations. Thirdly, the transition relation T and the valuation function v are interpreted as usual.

Example 1. *If* $(\{p\} \cup X, \{p\} \cup Y)$ *belongs to* $r_i(s)$ *for any choice of* $X, Y \in 2^{At}$*, then, in* s*, all the possible worlds in which* p *holds are mutually indistinguishable from the viewpoint of agent* i*. If we also have that* $(\{p\} \cup X, Y) \notin r_i(s)$ *whenever* $p \notin Y$*, we conclude that agent* i *distinguishes all and only the pairs of worlds differing for the valuation of* p*. Later, we will realize that this means that, in* s*, agent* i *knows the truth value of* p *and is ignorant of any other proposition.*

Remark 1. *From a rooted KLTS* $((S, T, \_, \_), s_0)$ *the accessibility relations and the valuation function, we obtain an LTS. Moreover, the KLTS* $(2^{At}, \emptyset, \{r_i \mid i \in \mathcal{A}\}, id)$ *– where each state represents a subset of* At*, id is the identity function, and* $r_i(s) = r_i(s')$ *for all* $i \in \mathcal{A}$ *and for any* $s, s' \in 2^{At}$ *– is a Kripke model.*

LTSs and Kripke models provide the semantics for interpreting properties expressed in various modal logics. Inspired by temporal logics and epistemic logics, we propose a modal logic that naturally combines temporal and epistemic ingredients, called Kripke Temporal (KT) logic.

Definition 4 (KT Logic). *The language* $\mathcal{L}_{KT}$ *of the KT logic is defined by the following two-layers grammar:*

$$\begin{array}{lcl} \phi & \to & \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \langle\pi\rangle\phi \mid \psi \\ \psi & \to & \top \mid p \mid \neg\psi \mid \psi \wedge \psi \mid K_i\psi \end{array}$$

The ψ formulas are called epistemic formulas. Note that the KT logic results from the combination and encompasses both HML [2] and Epistemic Logic [3].

Definition 5. *Given a KLTS* $M := (S, T, \{r_i \mid i \in \mathcal{A}\}, v)$ *and denoted* $M_s := (2^{At}, \{r_i(s) \mid i \in \mathcal{A}\}, id)$*, with* $s \in S$*, the truth of* $\varphi \in \mathcal{L}_{KT}$ *at* $s \in S$*, written* $M, s \models \varphi$*, is defined as follows:*

1. $M, s \models \top$
2. $M, s \models p$ *iff* $p \in v(s)$
3. $M, s \models \neg\varphi$ *iff* $M, s \not\models \varphi$
4. $M, s \models \varphi_1 \wedge \varphi_2$ *iff* $M, s \models \varphi_1$ *and* $M, s \models \varphi_2$
5. $M, s \models \langle\pi\rangle\phi$ *iff* $\exists s'.\, (s, \pi, s') \in T$ *and* $M, s' \models \phi$
6. $M, s \models K_i\psi$ *iff* $M_s, v(s) \models_{\mathrm{K}} K_i\psi$*, where the relation* $\models_{\mathrm{K}}$ *is defined as:*
   (a) $M_s, X \models_{\mathrm{K}} \top$
   (b) $M_s, X \models_{\mathrm{K}} p$ *iff* $p \in X$
   (c) $M_s, X \models_{\mathrm{K}} \neg\psi$ *iff* $M_s, X \not\models_{\mathrm{K}} \psi$
   (d) $M_s, X \models_{\mathrm{K}} \psi_1 \wedge \psi_2$ *iff* $M_s, X \models_{\mathrm{K}} \psi_1$ *and* $M_s, X \models_{\mathrm{K}} \psi_2$
   (e) $M_s, X \models_{\mathrm{K}} K_i\psi$ *iff* $\forall Y.\, (X, Y) \in r_i(s) : M_s, Y \models_{\mathrm{K}} \psi$

Note that the formula $K_i\psi$ is evaluated in $s \in S$ with respect to the accessibility relations associated with s, thus emphasizing the view of the KLTS as an LTS where each state s is equipped with a Kripke model $M_s := (2^{At}, \{r_i(s) \mid i \in \mathcal{A}\}, id)$ [1]. Hence, the semantics of an epistemic formula evaluated in s depends on such a Kripke model ($\models_{\mathrm{K}}$ is the classical satisfiability relation for Kripke models). By virtue of the indistinguishability interpretation we adopted, since we are assuming to work with accessibility relations that are reflexive, symmetric, and transitive, the reference system for the knowledge modality is S5 [13].

Based on the semantics above, two states s and s' are modal equivalent, written $s \equiv s'$, if and only if they satisfy the same formulas. The KT logic characterizes the following notion of behavioral equivalence.

Definition 6. *Let* $(S, T, \{r_i \mid i \in \mathcal{A}\}, v)$ *be a KLTS. A binary equivalence relation* B *on* S *is a bisimulation iff whenever* $(s, t) \in B$ *then:*

1. $v(s) = v(t)$*;*
2. *if* $(s, a, s') \in T$ *then* $\exists t'.\, (t, a, t') \in T$ *and* $(s', t') \in B$*;*
3. *there exists a binary equivalence relation* $\mathcal{B}_{st}$ *between the worlds of the Kripke models* $(2^{At}, \{r_i(s) \mid i \in \mathcal{A}\}, id)$ *pointed at* $v(s)$ *and* $(2^{At}, \{r_i(t) \mid i \in \mathcal{A}\}, id)$ *pointed at* $v(t)$*, such that* $(v(s), v(t)) \in \mathcal{B}_{st}$ *and for any* $X, Y \in 2^{At}$*, whenever* $(X, Y) \in \mathcal{B}_{st}$ *then:*
   - $X = Y$*;*
   - *if* $(X, X') \in r_i(s)$ *for* $i \in \mathcal{A}$*, then* $\exists Y'.\, (Y, Y') \in r_i(t)$ *and* $(X', Y') \in \mathcal{B}_{st}$*.*

Note that conditions 1. and 3. resemble the definition of modal bisimulation for Kripke models [13], while condition 2. characterizes the strong bisimulation for LTSs [2]. Two states s and s' are bisimilar, written $s \sim s'$, if and only if there exists a bisimulation B such that $(s, s') \in B$. The correspondence theorem relates bisimilar states and equivalent states whenever the KLTS is image-finite, i.e., for all states and actions, the image of s (under any accessibility relation) and the image of s, π (under the transition relation) are finite.

[1] Each state of M_s represents a subset of At denoting the propositions that are true in the state (note that the valuation function is the identity function), while the accessibility relations of M_s are those associated with s.

Theorem 1. *For any image-finite KLTS, $\sim$ coincides with $\equiv$.*

As a consequence of the grammar structure and the semantics of the KT logic, decidability and verification algorithms are inherited from the results related to epistemic logic and HML.

3. A language for Kripke labeled transition systems

In this section, we define a process-algebraic, agent-oriented language with value passing, the semantics of which is given in terms of KLTSs. We start by defining a basic calculus (see, e.g., [14, 15]) with value passing (see, e.g., [16, 17]) for the description of sequential process terms. Let A be a set of action names (ranging over $a, b, \ldots$) including the special action names τ and *set*. To model value passing, we will use variables ($x, y, \ldots, f, g, \ldots$), values ($v, v', \ldots$) from fixed domains, and expressions ($e, e', \ldots$) that usually represent simple values.

Definition 7. *The set $\mathcal{L}$ of process terms of the calculus for sequential processes is generated through the following syntax:*

$$P \to \underline{0} \mid \textstyle\sum_{k \in I} \pi_k \,.\, P_k \mid C(e_1, \ldots, e_n)$$
$$\pi \to b \mid a(y, f) \mid \bar{a}(i, \psi) \mid set(p, w)$$

where $b \in A \backslash \{set\}$, $a \in A \backslash \{\tau, set\}$, I is any finite indexing set, w is a boolean value, C is a constant name with the natural $n \geq 0$ being the arity of C.

The constant $\underline{0}$ stands for the inactive, halted process. The summation operator represents a nondeterministic choice enacting one of the guarded process terms $\pi_k.P_k$, which executes action π_k and then behaves as process term P_k (we will use E to denote a non-empty summation). The constant C is used to express recursive processes with $n \geq 0$ parameters, and must be associated with a defining equation of the form $C(x_1, \ldots, x_n) := P$. The notation π stands for any action, which can be an internal action b (including the unobservable action τ), an input action $a(y, f)$, an output action $\bar{a}(i, \psi)$, or an assignment action $set(p, w)$.

An assignment action has the effect of setting the proposition p to the boolean value w. An output action communicates an epistemic formula ψ to the agent i, while an input action receives a formula assigned to the variable f from an agent assigned to the variable y.

As usual in calculi with value passing, each occurrence of any variable in a process term P is bound by either an input action or a constant definition. For instance, x is bound in $C(x) := \bar{a}(x, p \wedge q) \,.\, C(x + 1)$ and in $a(x, f) \,.\, \bar{b}(x, \top) \,.\, \underline{0}$, but not in $\bar{a}(x, p \wedge q) \,.\, \underline{0}$. Moreover, we write i/x and ψ/f for substitutions of values for variables, and denote by $P[i/x, \psi/f]$ the result of substituting i (resp., ψ) for all free (not bound) occurrences of x (resp., f) in P.

Formally, the behavior of a process term Q is described in structural operational semantics style as the LTS rooted at Q and defined by the transition relation $T \subseteq \mathcal{L} \times Act \times \mathcal{L}$ that is the least transition relation generated by the axioms and the rules in Table 1. All the pre- and post-conditions associated with knowledge-based behaviors (i.e., communications and assignments) will be defined when introducing the parallel composition of process terms and the knowledge structures.

Table 1
Semantics rules for sequential processes

$$(prefix) \qquad b\,.\,P \xrightarrow{\ b\ } P \qquad set(p,w)\,.\,P \xrightarrow{\ set(p,w)\ } P \qquad \bar{a}(i,\psi)\,.\,P \xrightarrow{\ \bar{a}(i,\psi)\ } P$$

$$(input) \qquad a(y,f)\,.\,P \xrightarrow{\ a(i,\psi)\ } P[i/y,\psi/f] \quad \text{for any } i \in \mathcal{A} \text{ and epistemic formula } \psi$$

$$(sum) \qquad \frac{\pi\,.\,P \xrightarrow{\ \pi\ } P}{\pi\,.\,P + E \xrightarrow{\ \pi\ } P}$$

$$(recursion) \qquad \frac{P[v_1/x_1,\ldots,v_n/x_n] \xrightarrow{\ \pi\ } P'}{C(e_1,\ldots,e_n) \xrightarrow{\ \pi\ } P'} \qquad C(x_1,\ldots,x_n) := P \text{ and each } e_i \text{ evaluates to } v_i$$

Example 2. *The process term* $Agent := receive(y,f).\overline{send}(y+1,f).Agent$ *represents an agent without parameters that is available to receive as input a formula from an agent* y*, and then forwards such a formula as an output to agent* $y+1$ *(here, we assume that agent identities are naturals).*

3.1. Agents and pool of agents

Process terms represent behavioral patterns of agents, while an agent is an instance of a process term with a unique identity. Several agents may communicate with each other to form a network of agents. Hence, we need to formalize the notion of agent and how agents interact in a so-called pool of agents. A dynamic knowledge structure will be added to regulate such interactions.

Agents are described by tuples of the form $\langle i \in \mathcal{A}, P \in \mathcal{L}\rangle$ and are ranged over by $\mathcal{I}, \mathcal{J}, \ldots$. The semantics of $\langle i, P\rangle$ is given by the LTS expressing the behavior of P, up to the renaming of the actions as defined by the semantic rule:

$$(agent) \qquad \frac{P \xrightarrow{\ \pi\ } P'}{\langle i,P\rangle \xrightarrow{\ i.\pi\ } \langle i,P'\rangle}$$

So far, we abstracted from the interaction among agents and the underlying knowledge base. Now, we combine the behavior of several agents by integrating the notion of knowledge, which will allow us to specify how they can interact.

Definition 8. *A pool of agents is a tuple* $(\cup_i \mathcal{I}_i, \cup_i R_i, X)$*, where, for* i *ranging over* $\mathcal{A}$*:*

- $\cup_i \mathcal{I}_i$ *denotes a finite set of agents;*
- $\cup_i R_i$ *denotes a finite set of binary accessibility relations over* 2^{At}*;*
- $X \subseteq At$ *is the set of true propositions.*

The behavior of the set $\cup_i \mathcal{I}_i$ depends on the behavior of each $\mathcal{I}_i$ and is defined as an element of the cartesian product $(\mathcal{A} \times \mathcal{L})^n$, where n denotes the cardinality (i.e., the number of agents) of the pool. Then, for each agent i, the accessibility relation R_i expresses the capability of i to

Table 2
Semantics rules for a pool of agents

$$(pool) \quad \frac{\mathcal{J} \xrightarrow{j.b} \mathcal{J}'}{(\cup_{i\neq j}\mathcal{I}_i \cup \mathcal{J}, R, X) \xrightarrow{j.b} (\cup_{i\neq j}\mathcal{I}_i \cup \mathcal{J}', R, X)}$$

$$(set) \quad \frac{\mathcal{J} \xrightarrow{j.set(p,w)} \mathcal{J}'}{(\cup_{i\neq j}\mathcal{I}_i \cup \mathcal{J}, \cup_{i\neq j}R_i \cup R_j, X) \xrightarrow{\tau} (\cup_{i\neq j}\mathcal{I}_i \cup \mathcal{J}', \cup_{i\neq j}R'_i \cup R'_j, X')}$$

where $X' = \begin{cases} X\backslash\{p\} & \text{if } w = 0 \\ X \cup \{p\} & \text{if } w = 1 \end{cases}$ and, for $N := (2^{At}, \cup_{i\neq j}R_i \cup R_j, id)$:

$- R'_j = R_j\backslash\{(Y, Y') \mid diff(N, Y, Y', p)\}$
$- R'_i = closure(R_i \cup \{(\{p\} \cup Y, Y) \mid p \notin Y\} \cup \{(Y, \{p\} \cup Y) \mid p \notin Y\})$

$$(com) \quad \frac{(\cup_{k\neq i,j}\mathcal{I}_k \cup \mathcal{I} \cup \mathcal{J}, \cup_{k\neq j}R_k \cup R_j, X) \models K_i\psi \quad \mathcal{I} \xrightarrow{i.\bar{a}(j,\psi)} \mathcal{I}' \quad \mathcal{J} \xrightarrow{j.a(i,\psi)} \mathcal{J}' \quad i \neq j}{(\cup_{k\neq i,j}\mathcal{I}_k \cup \mathcal{I} \cup \mathcal{J}, \cup_{k\neq j}R_k \cup R_j, X) \xrightarrow{\tau} (\cup_{k\neq i,j}\mathcal{I}_k \cup \mathcal{I}' \cup \mathcal{J}', \cup_{k\neq j}R_k \cup R'_j, X)}$$

where, for $N := (2^{At}, \cup_{k\neq j}R_k \cup R_j, id) : R'_j = R_j\backslash\{(Y, Y') \mid diff(N, Y, Y', \psi)\}$

$$diff(N, X, Y, \psi) := (N, X \models_{\mathrm{K}} \psi \wedge N, Y \not\models_{\mathrm{K}} \psi) \vee (N, X \not\models_{\mathrm{K}} \psi \wedge N, Y \models_{\mathrm{K}} \psi)$$

$$closure(R_k) := R_k \cup \{(X, Y) \mid \exists Z.\, (X, Z) \in R_k \wedge (Z, Y) \in R_k\}$$

distinguish the possible worlds based on the values that can be attributed to the propositions of At. Finally, set X denotes the current truth assignment for the propositions of At.

The agents of a pool can perform actions, either synchronously or autonomously, thus making the system dynamic. On the one hand, the internal actions that are not related to knowledge and the assignment actions represent the autonomous actions of agents. On the other hand, input and output actions represent synchronous communications that express knowledge transfer between agents.

Formally, such a joint knowledge-based and action-based behavior is represented by a KLTS describing the evolution of the pool of agents.

Definition 9. *Let $\mathcal{P} := (\cup_i\mathcal{I}_i, \cup_i R_i, X)$ be a pool of agents of cardinality n. The semantics of $\mathcal{P}$ is given by the KLTS $((S, T, \cup_i r_i, v), \mathcal{P})$ rooted at $\mathcal{P}$, which is built as follows:*

- *the states in S are pool tuples, where $\mathcal{P} \in S$ is the initial state;*
- *T is the least transition relation generated by the rules of Table 2;*
- *for each $s \in S$ of the form $(\_, \cup_i R_i, X)$, it holds that $r_i(s) = R_i$ for each $i \in \mathcal{A}$ and $v(s) = X$.*

We now illustrate the rules of Table 2. The rule ($pool$) describes the asynchronous execution of autonomous actions of the form $b \in A \backslash \{set\}$ by any agent of the pool. Note that such actions do not change the knowledge structure, which is modeled by the set R of accessibility relations and by the truth assignment X.

The rule (set) describes the asynchronous execution of autonomous actions of the form $set(p, w)$ by any agent j, whose side effect is that the truth assignment X associated with the current tuple is updated according to the assignment $p = w$ (see the definition of X'). The accessibility relations are also updated accordingly. On the one hand, the agent j performing the assignment acquires knowledge (if not yet possessed) of p. Hence, in R_j, all the possible worlds differing for the valuation of p (see function *diff*) cannot be mutually accessible anymore, as they are distinguishable by the value of p. Note that, as we will show, such suppression of connections ensures that the accessibility relation remains an equivalence. On the other hand, all the other agents $i \neq j$ lose knowledge (if previously possessed) of p, as the assignment is not considered public (as emphasized by the fact that the resulting action is a silent action τ). Therefore, in each accessibility relation of those agents, all the possible worlds differing only for the valuation of p must become mutually accessible, as they cannot be distinguished anymore. Note that such addition of connections considers the symmetric pairs and, through the *closure* operation, the transitive relations, thus ensuring, as we will show, that the accessibility relation remains an equivalence.

The most interesting rule is (com), which expresses a communication from an output to a corresponding input (the two actions refer to the same action name a). The agent i performing the output and the agent j performing the corresponding input synchronize, i.e., they both advance simultaneously. However, the resulting synchronization is enabled only if the epistemic formula ψ communicated from i to j is known by i. If this is the case, j acquires knowledge of ψ, and the accessibility relation R_j is updated accordingly. In fact, agent j becomes able to distinguish those possible worlds that differ from each other for the evaluation of ψ. The communication is private (the synchronization result is a silent action τ), i.e., the knowledge transfer involves only the agent j and no one else.

Lemma 1. *The KLTS modeling the behavior of a pool of agents is image-finite.*

This result immediately derives by Definition 9 and the semantics of Table 2. As anticipated, another important result is that the semantics of Table 2 preserves the indistinguishability interpretation of the accessibility relations.

Theorem 2. *Let $\mathcal{P} := (\cup_i \mathcal{I}_i, \cup_i R_i, X)$ be a pool of agents such that each R_i, with $i \in \mathcal{A}$, is a* P*-relation (for* P *in {reflexive, symmetric, transitive}) and $((S, T, \cup_i r_i, v), \mathcal{P})$ be the semantics of $\mathcal{P}$. Then, for each $i \in \mathcal{A}$ and for each $s \in S$, it holds that $r_i(s)$ is a* P*-relation.*

4. Use case: playing Cluedo

The present case study is designed to highlight the modeling features and analysis opportunities of our framework. Despite its simplicity, this use case encompasses many of the features of real-world applications, including strategic thinking, private and public communications, and

knowledge transfer. For the sake of brevity, instead of the full Cluedo game[2] we model a simplified version. Let us consider a game set with 3 players, a dealer, and 8 cards, numbered from 1 to 8. At the beginning of the game, the dealer samples secretly and puts aside two cards, shuffles the remaining cards together, making sure none of the cards are seen by any of the players, and then deals two cards per player. Then, the game starts and proceeds by sequential turns. On her turn, each player makes publicly a suggestion of the form: *I suggest that the two secret cards of the dealer are i and j*. There are no constraints about the specific choice of i and j. Then, if the player on the right of the one making the suggestion has at least one of the cards mentioned, she must show one of these cards secretly to her. Then, the inquiry passes to the player on the left, with the same rule. At the end of her turn, the player wins the game if she has learned and can correctly declare what the dealer's cards are. Otherwise, the game proceeds with the following turns until one of the players wins.

Formally, we model the game set through the propositions p_i^j and q_i, for $0 \leq j \leq 2$ and $1 \leq i \leq 8$, where p_i^j means that player j has card i and q_i means that card i is one of the two secret cards of the dealer. The pool includes one dealer and three players and, initially, is defined as the tuple: $(\{\langle \text{Mr. Black}, \mathit{Dealer}\rangle, \langle 0, \mathit{Player}(0)\rangle, \langle 1, \mathit{Player}(1)\rangle, \langle 2, \mathit{Player}(2)\rangle\}, R, X)$. The three players have the same behavioral pattern, given by the process term Player, which is fed with a parameter representing the player identity. Set X is empty (the cards have yet to be shuffled by the dealer Mr. Black - hence all the propositions are set to 0). The accessibility relation of the dealer, $R_{\mathit{Mr.Black}}$, contains only the reflexive pairs, i.e., each possible world is a singleton. In fact, by assumption, the dealer is like an oracle and can distinguish any possible scenario. As we will see, $R_{\mathit{Mr.Black}}$ is immutable. The accessibility relation for each player j, denoted R_j, is such that two possible worlds are related if and only if they coincide for the values of the propositions p_i^j, $1 \leq i \leq 8$. The intuition is that, at least, a player is able to distinguish two possible worlds differing in the values of the cards she receives. All such accessibility relations are equivalence relations but are not immutable, as the knowledge of the players will change as the game proceeds.

Initially, the dealer shuffles the cards and chooses nondeterministically the two secret cards and the assignments for the players (see actions *set*):

$$\begin{aligned}
&\mathit{Dealer} := \textstyle\sum_{k_1,k_2} \mathit{set}(q_{k_1},1).\mathit{set}(q_{k_2},1).\mathit{Deal}(k_1,k_2)\\
&\mathit{Deal}(x,y) := \textstyle\sum_{i_1,i_2\notin\{x,y\}} \mathit{set}(p^0_{i_1},1).\mathit{set}(p^0_{i_2},1).\overline{\mathit{deal}}(0,p^0_{i_1}\wedge p^0_{i_2}).(\\
&\qquad\qquad \textstyle\sum_{i_3,i_4\notin\{i_1,i_2,x,y\}} \mathit{set}(p^1_{i_3},1).\mathit{set}(p^1_{i_4},1).\overline{\mathit{deal}}(1,p^1_{i_3}\wedge p^1_{i_4}).(\\
&\qquad\qquad \textstyle\sum_{i_5,i_6\notin\{i_1,\ldots,i_4,x,y\}} \mathit{set}(p^2_{i_5},1).\mathit{set}(p^2_{i_6},1).\overline{\mathit{deal}}(2,p^2_{i_5}\wedge p^2_{i_6}).\mathit{Play}(0)))
\end{aligned}$$

Whenever clear from the context, the bounds of a summation are not specified (in general, $\sum_{i,j}$ expresses a choice over all the possible unordered pairs of different values (i,j), each one ranging from 1 to 8). Process term Dealer models the random sampling of the two secret cards, and then the invocation of process term $\mathit{Deal}(k_1,k_2)$ describes the following behavior of the dealer whenever k_1 and k_2 have been chosen. The sampling for each player is modeled analogously through a pair of subsequent actions set. The output action deal is used to communicate the assignments to the players. Then, the dealer coordinates the game rounds:

$$\mathit{Play}(x) := \overline{\mathit{start\_turn}}(x,\top).(\mathit{end\_turn}(\_,\_).\mathit{Play}((x+1)\bmod 3) + \mathit{win}(\_,\_).\underline{0})$$

[2]The reader interested in reviewing the rules of the game can refer to the official instructions by Hasbro.

by assigning each turn (through the output action $start\_turn$) to a different player, sequentially. Note that the output is sent to player x to inform that her turn is starting, without the need to communicate any other information (this justifies the choice of the truth constant $\top$). Then, the dealer waits for a response: either the player turn terminates (input action end_turn) or the player wins the game by learning the secret pair during her turn (input action *win*). For the sake of convenience, whenever unnecessary, the arguments of an input action are left unspecified (symbol _).

After receiving the cards through the input action $deal$, each player is available to start her turn (input action $start\_turn$) or to manage inputs from the other players. The process term $Player(x)$ is defined as follows:

$$
\begin{aligned}
&Player(x) := deal(y,\_).\\
&\quad (start\_turn(\_,\_).\textstyle\sum_{i_1,i_2}\overline{ask_{i_1,i_2}}((x+1)\bmod 3,\top).show(\_,\_).\\
&\qquad\qquad \overline{ask_{i_1,i_2}}((x+2)\bmod 3,\top).show(\_,\_).\\
&\qquad\qquad (\overline{end\_turn}(y,\neg\phi_x).Player(x)+\overline{win}(y,\phi_x).\underline{0}))\\
&\quad +\textstyle\sum_{i_1,i_2} ask_{i_1,i_2}(z,\_).\\
&\qquad (\overline{show}(3-x-z,p^x_{i_1}\vee p^x_{i_2}).(\overline{show}(z,p^x_{i_1}).Player(x)+\\
&\qquad\qquad \overline{show}(z,p^x_{i_2}).Player(x))+\\
&\qquad \overline{show}(3-x-z,\neg p^x_{i_1}\wedge\neg p^x_{i_2}).\overline{show}(z,\neg p^x_{i_1}\wedge\neg p^x_{i_2}).Player(x))\\
&\quad +show(\_,\_).Player(x))
\end{aligned}
$$

When initiating a new turn, the player chooses nondeterministically two cards to be asked to each other player (output action *ask*) and then waits for the related answer (input action *show*). At the end of the turn, either the player learns the secret and wins the game (output action *win*) or passes the hand (output action end_turn). The winning condition for player x determining which output is executed is given by the knowledge of the formula $\phi_x = \bigvee_{(k,k')} K_x(q_k \wedge q_{k'})$, i.e., the player knows the secret pair. Then, players respond to incoming requests through the input action *ask*. If player x receives from player z a request about cards i_1 and i_2, then we distinguish two cases. Firstly, x may have at least one of the two cards ($p^x_{i_1} \vee p^x_{i_2}$). In this case, x reveals one of the possessed cards to z, by choosing the card nondeterministically if necessary. Indirectly, even the third, silent player (identified by $3-x-z$) learns something, i.e., the fact that x has one of the two cards. We model this indirect transfer of knowledge through an explicit output directed to player $3-x-z$. Secondly, x may have none of the two cards ($\neg p^x_{i_1} \wedge \neg p^x_{i_2}$). In this case, the information is shared with both the other players. Finally, due to the outputs directed to player $3-x-z$, players must also be available to learn some information during the turns of the other players (through the input action $show$).

It is worth noting that the management of the knowledge base of the players is left to the semantics of the underlying Kripke model. At the level of the specification, only the initial setting and the communications are modeled explicitly. This is particularly significant from the viewpoint of usability, as an analogous model based on, e.g., classical Kripke structures, would be much more challenging. To appreciate this aspect, the same use case has been modeled in the software tool NuSMV [18], the specifications of which result in finite state machines that turn out to be Kripke structures.[3] Since there are 2520 ways of dealing the 8 cards to the three

[3]The specification can be found on github.

players and the dealer – the computing formula is $\binom{8}{2} \cdot \binom{6}{2} \cdot \binom{4}{2}$ – the NuSMV specification refers to one of these, chosen deterministically through external parameters that initialize the system configuration. Moreover, the NuSMV specification describes only a very simplistic version of the players' knowledge, in which each player does not deduce any information when observing the interactions between the other two players. In fact, the additional information needed to model the full deduction capabilities of the players should be represented explicitly by the designer and would make the model much more complicated and error-prone. By the way, despite these simplifications, the NuSMV specification is made out of about 200 code lines and 58 variables.[4]

To show an example of properties that can be model checked, we consider the derived *eventually* modality F, such that $M, s \models F\phi$ if and only if $M, s \models \phi$ or $\exists \pi.\, M, s \models \langle \pi \rangle F\phi$, and the derived *globally* modality G, such that $M, s \models G\phi$ if and only if $M, s \models \phi$ and $\exists \pi.\, M, s \models \langle \pi \rangle G\phi$. Then, the reachability property $F(\bigvee_x \phi_x)$ is satisfied, i.e., the *winning* state is reachable by some players. However, even the unreachability property $G(\bigwedge_x \neg\phi_x)$ holds. The reason is that the simple, nondeterministic strategy followed by the players when choosing their suggestion does not guarantee that the game can always be won.

5. Related work and conclusions

A few approaches investigate the combination of LTS-based semantics and epistemic notions, e.g., in the setting of epistemic μ-calculus [7] and of concurrent constraint programming paradigms [5, 6]. The framework proposed in [4] is the closest to our approach in principle, as it integrates LTSs with accessibility relations stating the indistinguishability between states. However, agents observe (do not control) the path of performed actions and, based on this knowledge, deduce what the actual state is. Hence, the semantics of the formulas of the underlying logic is given in terms of paths. Notably, such a logic, similarly to the KT logic, is equipped with both temporal and epistemic modalities.

An important strand of research concerns dynamic extensions of Kripke models and epistemic logic, where the dynamic dimension is related to the execution of actions over time; see, e.g., [19, 20, 21, 22, 23, 24, 25, 26, 27] and the references therein. However, all these approaches differ in the way in which we encode the dynamics of epistemic models within the LTS-style semantics. The main advantage of our encoding is that the obtained semantics facilitates the definition of a high-level process-algebraic language for the description of multi-agent systems and knowledge-based interactions. Moreover, a benefit of our LTS-based semantics is that we inherit the model-checking techniques associated with discrete-time models and temporal logics in HML style. As seen in the KT logic, these capabilities can be merged with the expressive power of Epistemic Logic.

In the field of concurrency theory, some of the ideas presented in this paper can be found in the study of temporal logics encompassing features from HML and modal μ-calculus [28]. As an example, a variant of the temporal logic CTL is defined in [29] to check properties over expressive models called $\mathrm{L}^2\mathrm{TS}$. In these models, the idea is to combine transition labels expressing the action-based dynamic behavior of a system with state-based labels expressing

[4] The underlying Kripke structure has about 2^{20} states.

the knowledge possessed in each state of the system. With respect to our proposal, no epistemic representation of derivable knowledge is given, so the study of the observational power of the agents is limited to the verification of state-based propositional logic formulas and on the model checking of temporal formulas.

Summarizing, by following suggestions deriving from works on dynamic and temporal epistemic logics [30], we embedded a structure of pointed Kripke models into a labeled transition system, the actions of which act as model-transforming operations from the viewpoint of the Kripke models. These transitions naturally model the behavior of the system and the passage of (discrete) time, while the Kripke models linked to the states visited during the temporal evolution of the system represent the way in which the knowledge of every agent evolves over time. The process algebraic language that we introduced emphasizes these effects and allows for a compact and elegant description of multi-agent systems, where the details of the knowledge evolution are left to the underlying epistemic model.

Starting from this point, several extensions can be envisioned. For instance, the semantics of our communication mechanisms assumes that only known truth can be transferred. Hence, we do not currently manage (possibly false) beliefs and the communication of information that is inconsistent with an agent's knowledge or belief. This would require the introduction of the belief modality and the treatment of contradictions resulting from the communication between agents. Moreover, this would also open to extensions in which it is possible to model the behavior of malicious agents sharing false information and, therefore, a theory of fake news [31, 32], trust, and reputation [33, 34, 35, 36]. Along the same lines, further modalities could be added to the epistemic component of our model.

Dealing with inconsistencies is a problem to face even in the present model, without bringing up the notion of belief. In particular, an unsuccessful formula is a formula that might become false as soon as it is communicated, like, e.g., in the case of $p \wedge \neg K_j p$ whenever agent i communicates it to agent j [37]. Several studies investigate the syntactic form of potential unsuccessful formulas, in particular in the setting of public announcements for multi-agent systems [38]. Obviously, even in our framework such forms can be recognized and, in particular, are limited to those cases in which a formula of the form $\neg K_j \psi$ is involved in a communication to agent j. This is because the satisfaction of $\neg K_j \psi$ before the communication could be contradicted by sharing its knowledge with the agent suffering from such a kind of ignorance. The formal investigation of these situations is left as future work.

Given the high generality of the proposed framework, it would also be interesting to investigate the relation with other abstract models, such as coalgebraic modal logics [39], to better guide the comparison with the literature. Finally, we also plan to define: (i) an axiomatization for the KT logic, (ii) quantitative extensions of the KLTS model, by adding continuous time and probabilistic choices, and (iii) additional ingredients in the process-algebraic language, by including internal actions guarded by knowledge-based requirements, *if-then-else* constructs that are based on knowledge conditions, asynchronous communication and dynamic pools of agents, broadcast communication in the style of [40].

References

[1] C. Baier, J.-P. Katoen, Principles of Model Checking (Representation and Mind Series), The MIT Press, 2008.

[2] M. Hennessy, R. Milner, On observing nondeterminism and concurrency, in: J. de Bakker, J. van Leeuwen (Eds.), Automata, Languages and Programming (ICALP 1980), volume 85 of *LNCS*, Springer, 1980, pp. 299–309.

[3] H. van Ditmarsch, J. Y. Halpern, W. van der Hoek, B. Kooi (Eds.), Handbook of Epistemic Logic, College Publications, 2015.

[4] S. Knight, R. Mardare, P. Panangaden, Combining epistemic logic and Hennessy-Milner logic, in: Logic and Program Semantics: Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday, Springer, 2012, p. 219–243.

[5] S. Knight, C. Palamidessi, P. Panangaden, F. D. Valencia, Spatial and epistemic modalities in constraint-based process calculi, in: M. Koutny, I. Ulidowski (Eds.), CONCUR 2012 - Concurrency Theory, Springer, 2012, pp. 317–332.

[6] M. Guzman, S. Haar, S. Perchy, C. Rueda, F. D. Valencia, Belief, knowledge, lies and other utterances in an algebra for space and extrusion, Journal of Logical and Algebraic Methods in Programming 86 (2017) 107–133.

[7] F. Dechesne, M. Mousavi, S. Orzan, Operational and epistemic approaches to protocol analysis: Bridging the gap, in: Logic for Programming, Artificial Intelligence, and Reasoning: 14th Int. Conf., LPAR 2007, Springer, 2007, p. 226–241.

[8] F. Dechesne, Y. Wang, To know or not to know: epistemic approaches to security protocol verification, Synthese 177 (2010) 51–76.

[9] R. Chadha, S. Delaune, S. Kremer, Epistemic logic for the applied pi calculus, in: D. Lee, A. Lopes, A. Poetzsch-Heffter (Eds.), Formal Techniques for Distributed Systems, Springer, 2009, pp. 182–197.

[10] M. Balliu, M. Dam, G. Le Guernic, Epistemic temporal logic for information flow security, in: Procs. of the ACM SIGPLAN 6th Workshop on Programming Languages and Analysis for Security, PLAS'11, ACM, 2011.

[11] K. Minami, Trace equivalence and epistemic logic to express security properties, in: A. Gotsman, A. Sokolova (Eds.), Formal Techniques for Distributed Objects, Components, and Systems, Springer, 2020, pp. 115–132.

[12] K. Bavendiek, S. Schupp, A process calculus for privacy-preserving protocols in location-based service systems, Journal of Logical and Algebraic Methods in Programming 125 (2022).

[13] P. Blackburn, M. De Rijke, Y. Venema, Modal logic, volume 53 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, 2002.

[14] W. Fokkink, Introduction to Process Algebra, Springer, 2007.

[15] R. Gorrieri, C. Versari, Introduction to Concurrency Theory - Transition Systems and CCS, Springer, 2015.

[16] M. Hennessy, A proof system for communicating processes with value-passing, Formal Aspects of Computing 3 (1991) 346–366.

[17] S. Huang, Y. Cao, H. Wang, W. Qu, Value-passing CCS with noisy channels, Theoretical Computer Science 433 (2012) 43–59.

[18] A. Cimatti, E. M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, A. Tacchella, NuSMV 2: An opensource tool for symbolic model checking, in: Procs. of the 14th Int. Conf. on Computer Aided Verification, CAV'02, Springer, 2002, p. 359–364.
[19] A. Baltag, L. Moss, Logics for epistemic programs, Synthese 139 (2004) 165–224.
[20] S. van Otterloo, G. Jonker, On epistemic temporal strategic logic, Electronic Notes in Theoretical Computer Science 126 (2005) 77–92. Procs. of the 2nd Workshop on Logic and Communication in Multi-Agent Systems (2004).
[21] B. Kooi, B. Renne, Generalized arrow update logic, in: 13th Conference on Theoretical Aspects of Rationality and Knowledge, TARK XIII, ACM, 2011, pp. 205–211.
[22] J. van Benthem, J. Gerbrandy, E. Pacuit, Merging frameworks for interaction: Del and etl, in: Procs. of the 11th Conf. on Theoretical Aspects of Rationality and Knowledge, TARK'07, ACM, 2007, p. 72–81.
[23] H. van Ditmarsch, Dynamic Epistemic Logic, Springer, 2007.
[24] P. Girard, J. Seligman, F. Liu, General dynamic dynamic logic, in: T. Bolander, T. Braüner, S. Ghilardi, L. Moss (Eds.), Procs. of the 9th Conf. on Advances in Modal Logic, volume 9 of *Advances in Modal Logic*, College Publications, 2012, pp. 239–260.
[25] A. Baltag, L. S. Moss, S. Solecki, The logic of public announcements, common knowledge, and private suspicions, in: H. Arló-Costa, V. F. Hendricks, J. van Benthem (Eds.), Readings in Formal Epistemology: Sourcebook, Springer, 2016, pp. 773–812.
[26] B. Renne, J. Sack, A. Yap, Logics of temporal-epistemic actions, Synthese 193 (2016) 813–849.
[27] T. Bolander, A gentle introduction to epistemic planning: The DEL approach, Electronic Proceedings in Theoretical Computer Science 243 (2017) 1–22.
[28] R. De Nicola, F. Vaandrager, Action versus state based logics for transition systems, in: I. Guessarian (Ed.), Semantics of Systems of Concurrent Processes, Springer, 1990, pp. 407–419.
[29] M. H. ter Beek, A. Fantechi, S. Gnesi, F. Mazzanti, A state/event-based model-checking approach for the analysis of abstract system properties, Science of Computer Programming 76 (2011) 119–135.
[30] R. Parikh, R. Ramanujam, A knowledge based semantics of messages, Journal of Logic, Language and Information (2003) 453–467.
[31] M. R. Mousavi, M. Varshosaz, Telling lies in process algebra, in: 2018 Symposium on Theoretical Aspects of Software Engineering (TASE), IEEE, 2018, pp. 116–123.
[32] A. Aldini, On the modeling and verification of the spread of fake news, algebraically, Journal of Logic and Computation 32 (2022) 1272–1291.
[33] A. Aldini, M. Tagliaferri, Logics to reason formally about trust computation and manipulation, in: A. Saracino, P. Mori (Eds.), Emerging Technologies for Authorization and Authentication, volume 11967 of *LNCS*, Springer, 2020, pp. 1–15.
[34] M. Tagliaferri, A. Aldini, From belief to trust: A quantitative framework based on modal logic, Journal of Logic and Computation 32 (2022) 1017–1047.
[35] A. Aldini, G. Curzi, P. Graziani, M. Tagliaferri, Trust evidence logic, in: J. Vejnarová, N. Wilson (Eds.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 16th European Conference (ECSQARU 2021), volume 12897 of *LNAI*, Springer, 2021, pp. 575–589.

[36] A. Aldini, G. Curzi, P. Graziani, M. Tagliaferri, A probabilistic modal logic for context-aware trust based on evidence, International Journal of Approximate Reasoning 169 (2024) 109167.
[37] H. Van Ditmarsch, B. Kooi, The secret of my success, Synthese 151 (2006) 201–232.
[38] S. Saraf, S. Sourabh, Characterizing successful formulas: the multi-agent case, CoRR abs/1209.0935 (2012).
[39] C. Kupke, D. Pattinson, Coalgebraic semantics of modal logics: An overview, Theoretical Computer Science 412 (2011) 5070–5094.
[40] A. Aldini, Design and verification of trusted collective adaptive systems, Transactions on Modeling and Computer Simulation (TOMACS) 28 (2018).