Lecture Notes on Modal Logic in Computer Science

R. Ramanujam
The Institute of Mathematical Sciences
C.I.T. Campus
Chennai 600 113
jam@imsc.ernet.in

School on Logic in Computer Science
Indian Statistical Institute
Calcutta
December 23, 1996 to January 10, 1997

Plan of lectures

1. Introduction to Modal Logic
2. Basic Modal Logic: Correspondence Theory
3. Basic Modal Logic: Completeness and Decidability
4. Propositional Dynamic Logic
5. Hintikka's Logic of Knowledge
6. The problem of logical omniscience
7. Knowledge and time
8. Common Knowledge
9. Probabilistic Knowledge
10. Nonmonotonic reasoning

1 Introduction to Modal Logic

In this lecture, we first recall the basics of classical propositional logic and then introduce the basic modal logic of necessity.

1.1 Propositional Logic

The propositional calculus studies the logical structure of arguments. Assuming the truth of *atomic* propositional statements, the calculus involves the use of *logical connectives* to derive the truth or falsity of compound statements.

Formally, we assume a countable set of propositional letters $P = \{p_0, p_1, \ldots\}$, and the syntax of formulas is given by: $\alpha, \beta, \ldots \in PL ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta$ where $\neg\alpha$ is read 'not α' and $\alpha \vee \beta$ is read 'α or β' referring, respectively, to negation and disjunction. Conjunction is then defined by $\alpha \wedge \beta \stackrel{\text{def}}{=} \neg(\neg\alpha \vee \neg\beta)$, implication by $\alpha \supset \beta \stackrel{\text{def}}{=} \neg\alpha \vee \beta$, and equivalence by $\alpha \equiv \beta \stackrel{\text{def}}{=} (\alpha \supset \beta) \wedge (\beta \supset \alpha)$.

A model is defined to be a set $V \subseteq P$. The idea is that every $p \in V$ is true in the model V and every $q \notin V$ is false in V. We can then define the notion 'α holds in the model V' (also 'V satisfies α') for every formula α, denoted $V \models \alpha$, inductively as follows:

- $V \models p$ iff $p \in V$.
- $V \models \neg\alpha$ iff $V \not\models \alpha$.
- $V \models \alpha \vee \beta$ iff $V \models \alpha$ or $V \models \beta$.

It is easy to check that

- $V \models \alpha \wedge \beta$ iff $V \models \alpha$ and $V \models \beta$.
- $V \models \alpha \supset \beta$ iff it is not the case that $V \models \alpha$ and $V \not\models \beta$.

We say that a formula $\alpha \in PL$ is **satisfiable** iff there exists a model V such that V satisfies α. α is said to be **valid** iff every model V satisfies α, and is denoted $\models \alpha$. It is easy to see that α is valid iff $\neg\alpha$ is not satisfiable.

The *satisfiability problem* asks, given a formula α, whether it is satisfiable. The *model checking problem* asks, given a model V and formula α whether V satisfies α. For propositional logic, model checking takes (and requires) $O(|V| + |\alpha|)$ time. For satisfiability, we have the following famous theorem.

Theorem 1.1 (Cook): *The satisfiability problem for PL is NP-complete.*

A nondeterministic Turing machine needs only to guess a subset of the propositions mentioned in the given formula and run the model checking algorithm on it. NP-hardness is proved by coding up Turing machine configurations, and the proof can be found in standard computability texts (see, for instance [HU79]).

There are many axiom systems for deriving the valid formulas (tautologies) of propositional logic – natural deduction, Gentzen's sequent calculus, tableau procedures, Hilbert-style schemes and so on. In these lectures, we will follow Hilbert-style presentations, where we present *axiom schemes* rather than individual axioms – each scheme stands for infinitely many formulas obtained by substitution in the scheme, and similarly for inference rules. Consider the following system PC.

$(A0)$	$\alpha \supset (\beta \supset \alpha)$
$(A1)$	$(\alpha \supset (\beta \supset \gamma)) \supset ((\alpha \supset \beta) \supset (\alpha \supset \gamma))$
$(A2)$	$(\neg\beta \supset \neg\alpha) \supset ((\neg\beta \supset \alpha) \supset \beta)$

$$(MP)\ \ \frac{\alpha, \quad \alpha \supset \beta}{\beta}$$

A **thesis** of the system is a formula which is either an instance of one of the axiom schemes, or obtained from the axioms by finitely many applications of the inference rules. $\vdash \alpha$ denotes that α is a thesis. We say that α is **consistent** iff $\neg\alpha$ is not a thesis of the system, that is, $\not\vdash \neg\alpha$. A finite set of formulas $\{\alpha_1, \dots, \alpha_k\}$ is consistent iff their conjunction $\alpha_1 \wedge \dots\ land\alpha_k$ is consistent, and a set of formulas is consistent iff every finite subset is.

Theorem 1.2 Completeness: *For every formula α of PL, $\vdash_{PC} \alpha$ iff $\models \alpha$.*

The forward direction, called *soundness*, consists of showing that the axioms are valid and that the inference rules preserve validity. The other way, called *completeness* is equivalent to showing that every consistent formula is satisfiable (check !). We will see many examples of such *Henkin-style* completeness arguments later on. These proofs give insights into the model construction process, of particular interest in computer science.

We defined the notion of a formula being satisfiable. We say that a set Γ of formulas is satisfiable iff there exists a model V such that $\forall\alpha \in \Gamma$: $V \models \alpha$.

Theorem 1.3 Compactness: *Let Γ be any set of PL formulas. Γ is satisfiable iff every finite subset is.*

1.2 Necessary Truth

The statements of propositional logic can be seen as unqualified in the following sense: contrast the following two statements — "The bird is seen only in the mornings" and "I see the bird now". Assigning a truth value to the latter requires specification of who the speaker is, how far her visibility extends, as well as the time and place of utterance. Such statements of *qualified truth* depend on *contexts*: at different contexts, they assume different truth values. Again, the statements "$(x = 5) \supset (x > 2)$", "there exists a prime which divides 20" and "$\forall x, y, z, n.(n > 2) \supset (x^n + y^n \neq z^n)$" can be assigned truth values independent of any context, whereas statements like "$x > 2$", "eventually the system will deadlock", "the computer knows that the player will be stuck in 2 more moves, and that she does not know this" are all context-dependent *modal* statements.

Modal logic is very much about qualified truth. The qualifiers are called **modalities** and the basic modality studied is: 'α holds `necessarily`', and its dual, 'α `possibly` holds'. Here we may be referring to physical necessity (e.g. the laws of gravity) or logical necessity (e.g. that the primes are infinitely many), or something diffferent depending on the situation being modelled. For example, consider the use of modalities in the sentence: "it is necessary that a Congress Prime Minister is also the head of the Congress Parliamentary Party at that time, though it is possible that the Congress President is a different person (from the Congress PM)." In computer science, we tend to study qualifiers like *necessary, obligatory, true after an action* (such as running a computer program), *eventually, infinitely often, so far, from now forever, somewhere, known, knowable, provable, believed* and so on (see [GG84] for a range of modalities).

Studying these qualifiers formally and understanding their properties is interesting as well as challenging, since everyday intuition is not much help. For instance if α holds necessarily, is it a necessary truth that it holds necessarily ? On the other hand, can we have a situation where α does not necessarily hold, but it is necessary that it does not necessarily hold ? If we think of tense as a modality, should we consider time to be discrete or continuous ? Can properties depend on explicit time instants (as for instance in the statement 'the bell will ring at 3 o' clock') ? Should the future be fixed, or not ? The past determined, or not ?

The attraction of modal logic for computer scientists is not only mathematical, but also because the contexts mentioned above can be thought as *machine states.* The aim of modal logic being the study of how truth varies across contexts, it seems a suitable vehicle to study state changes.

1.3 Syntax and Semantics

Formally, propositional modal logic is an enrichment of classical propositional logic. With $P = \{p_0, p_1, \ldots\}$, the syntax of formulas is given by: $\alpha, \beta, \ldots \in ML ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid \Box\alpha$ where $\Box\alpha$ is read 'Box α' and denotes necessity of α. The dual modality $\Diamond\alpha$ (read 'Diamond α') denoting possibility of α is given by: $\Diamond\alpha \stackrel{\text{def}}{=} \neg\Box\neg\alpha$.

Though the mathematics of propositional logic has been studied since the early years of this century, an intuitively appealing semantics for modal logic took a long time to formulate. An algebraic semantics was proposed in the 1940's but it was not until the advent of relational semantics in the 1960's that the study of modal logics became popular. Though Kanger and Hintikka came up with similar ideas (independently), relational semantics is mainly associated with the name of Saul Kripke who called it *possible worlds semantics* [Kr59], and this is the approach we will follow in these lectures. (The phrase originates from Leibniz, who defined necessary truth to be truth in all possible worlds.)

Definition 1.4 *A* **frame** *is a pair $F = (W, R)$, where W is a nonempty set and $R \subseteq W \times W$ is a binary relation. A* model *is a pair $M = (F, V)$, where $F = (W, R)$ is a frame and $V : W \to 2^P$ is the* valuation function *in M.*

Informally, we refer to W as the set of possible worlds in M and say w' is *accessible* from w when $(w, w') \in R$. When $p \in V(w)$, we say p holds in w in M, and otherwise, we consider p to be false in w in M.

The notion 'α holds in the world w in the model M', denoted $M, w \models \alpha$, is defined inductively as follows:

- $M, w \models p$ iff $p \in V(w)$.
- $M, w \models \neg\alpha$ iff $M, w \not\models \alpha$.
- $M, w \models \alpha \vee \beta$ iff $M, w \models \alpha$ or $M, w \models \beta$.
- $M, w \models \Box\alpha$ iff $\forall w'$ such that $(w, w') \in R$: $M, w' \models \alpha$.

As one would expect, a formula α is satisfiable if there exists a model $M = ((W, R), V)$ and some $w \in W$ such that $M, w \models \alpha$, and α is defined to be valid iff its negation is not satisfiable.

It is easy to check that $M, w \models \Diamond\alpha$ iff there exists w' such that $(w, w') \in R$ and $M, w' \models \alpha$. Thus, $\Diamond$ is definite and $\Box$ indefinite: the latter formula is always satisfied in states w where $\{w' \mid (w, w') \in R\}$ is empty.

To consider examples, note that $\Box(p \vee \neg p)$ is valid but $\Box p \vee \Box\neg p$ is not. To see the latter, consider the frame $(\{w_0, w_1, w_2\}, \{(w_0, w_1), (w_0, w_2)\})$ with the valuation V given by: $V(w_1) = \{p\}$ and $V(w_0) = V(w_2) = \emptyset$. It is easy to see that the formula does not hold at w_0 in this model. On the other hand, to see that $\Box(p \vee \neg p)$ is valid, consider any model $M = ((W, R), V)$ and any $w, w' \in W$ such that $(w, w') \in R$. Clearly, $M, w' \models p \vee \neg p$. This is true for arbitrary such w', hence $M, w \models \Box(p \vee \neg p)$. The following exercise should serve to get some practice in analyzing modalities.

Exercise 1.5 *Prove or disprove the validity of the following formulas:*

1. $\Box\neg\alpha \supset \neg\Box\alpha$.
2. $\Box(\alpha \wedge \beta) \supset (\Box\alpha \wedge \Box\beta)$.
3. $(\Box\alpha \wedge \Box\beta) \supset \Box(\alpha \wedge \beta)$.
4. $\Box(\alpha \vee \beta) \supset (\Box\alpha \vee \Box\beta)$.

5. $(\Box\alpha \vee \Box\beta) \supset \Box(\alpha \vee \beta)$.

6. $\Diamond\neg\alpha \supset \neg\Diamond\alpha$.

7. $\Diamond(\alpha \wedge \beta) \supset (\Diamond\alpha \wedge \Diamond\beta)$.

8. $(\Diamond\alpha \wedge \Diamond\beta) \supset \Diamond(\alpha \wedge \beta)$.

9. $\Diamond(\alpha \vee \beta) \supset (\Diamond\alpha \vee \Diamond\beta)$.

10. $(\Diamond\alpha \vee \Diamond\beta) \supset \Diamond(\alpha \vee \beta)$.

While the exercise above studies how modalities interact with propositional connectives, more interesting is the situation with repeated and alternating modalities. Consider, for instance, the formulas: $\Box\Box\alpha$, $\Box\Diamond\alpha$, $\Diamond\Box\alpha$ and $\Diamond\Diamond\alpha$. While the formulas in the exercise above can be still read and understood in terms of necessity and possibility, it gets harder now. When we consider formulas like $\Box\Diamond p \wedge \Diamond\Box\neg q$ we are talking a different language altogether.

In fact, this is a language we are very familiar with: it is the one in which we are used to talking about state changes.

1.4 Transition Systems

In computer science, *behaviour* of systems (programs, circuits, protocols, ...) is usually given by state-transition systems of the form $(S, \rightarrow, s_0)$ where S is a set of states, $\rightarrow \subseteq (S \times \Sigma \times S)$ is a transition relation where Σ is a finite alphabet of system actions, and $s_0 \in S$ is the initial state of the system described. States typically refer to values of system variables, and actions update these values. $\rightarrow$ being a relation, these are typically nondeterministic systems. This is how operational semantics of programs is usually given. Computations are associated with these systems by considering the unfolding of the system into a tree rooted at s_0 wth tree edges corresponding to transitions. Depending on the properties studied, we either look at paths in the tree (the runs of the system) or the tree itself with the branching information therein.

What has this to do with modal logic ? Operational descriptions of systems are often far too detailed, and in the context of system *specification* we often wish to abtract out detail, and this is where logic proves useful. We may not be interested in the actual value of a variable x but in the truth value of a proposition like $x > 5$. Thus states can be simply associated with the set of propositions that are true in them. Of course different states can have the same propositions true in them (states with $x = 12$ and $x = 15$ both make the above proposition true), and thus we have our 'possible worlds' W and $V : W \rightarrow 2^P$. It is now trivial to see the use of modalities: $\Diamond$ specifies branching, and $\Box$ specifies what all children at a node must see.

The abstraction thus provided enables us to identify all trees which satisfy the same formula. For instance the formula $(x > 5) \wedge \Box(x > 7) \wedge \Diamond(y < 2) \wedge \Diamond(y > 10)$ specifies infinitely many state-transition trees of depth 1 with 3 nodes, where states specify numerical values for x and y and transitions update these values, but in all of them, if the value of x was 6 or 7 at the root, it gets suitably incremented. We can thus see modal logic as providing a logical language for describing behaviours of systems.

Exercise 1.6 *Show that the tree associated with any finite acyclic transition system can be described by a unique ML formula.*

Of course, ML is not much good as a specification language. We want lots more, we would like to talk about which states are reachable, how a choice determines future states, whether some actions can be performed concurrently, whether a run is terminating, whether a particular property is observable to the computing agent, and so on. And indeed, this is how different modalities are defined and studied in computer science, and we will see a few in the course of these lectures.

1.5 Interesting, but ...

The literature on modal logic is extensive, and in the last 30 years, a vast body of intellectual work has emerged. Here is a sampler of topics which we will not touch on in these lectures, but are nevertheless both interesting and important in the study of modal logics.

1.5.1 Quantified modal logics

We have only introduced propositional modal logic, but surely we can extend the language of predicate logic with modalities. However, finding appropriate semantics for such first-order modal logics is difficult: semantics for first order formulas is given over structures of the form $(D, R_1, \ldots, R_n)$, where D is a set whose elements give meaning to the first order variables and atomic formulas get associated with the specified relations on D. When we consider possible worlds semantics, we can think of each world as such a structure. Should the set D be different for each world ? Can a constant symbol be associated with different elements in different worlds ? What about elements which simply do not exist in some worlds ? These and other questions have led to intense debate and the mathematical implications of one or other answer in each case have been studied. What is clear now is that there is no really canonical semantics for predicate modal logic, but that appropriate semantics should be chosen depending on application (see [HC68], [Fi93]). For instance, in first order dynamic logic the question of whether constants can denote different elements in different worlds is answered yes if the class of programs considered includes creation and death of processes, and no, otherwise.

1.5.2 Modal algebras

The mathematics of propositional logic is studied via boolean algebras, and there are beautiful theorems relating algebraic structure in the logic to topology. What do modalities correspond to algebraically ?

Definition 1.7 *A* normal modal algebra *is a tuple* $(A, \cup, \cap, -, L)$, *where* $(A, \cup, \cap, -)$ *is a boolean algebra, and* $L : A \to A$ *is a map such that* $L(a \cap b) = La \cap Lb$ *and* $L1 = 1$, *where* 1 *is the unit of the boolean algebra.*

We can then define a valuation of an ML formula α on a normal modal algebra inductively, sending propositions to elements of A, such that $v(\Box\alpha) = L(v(\alpha))$. It can then be checked that every valid formula maps to the unit of every normal modal algebra under every valuation. Significantly, the converse also holds: that is, if a formula is not valid, then there is a normal modal algebra and a valuation which maps it to an element different from the unit of that algebra. This is called Lindenbaum's construction, and is of prime imporatnce in the study of modal algebras. The algebraic and topological structures associated with different modalities have been studied extensively, particularly by Helena Rasiowa and her co-authors [Ras74].

1.5.3 Proof theory

While we will discuss Hilbert-style axiom systems, these will be more in the nature of demonstrations that the set of valid formulas admit a finite axiomatization, rather then establishing proof systems from which theorems can be derived in practice. The structure of proofs is the subject of proof theory, and in the case of modal logics, sequent calculi, tableau systems and resolution methods exist for such a study (see [Fi83]). This is an important area which we will skip entirely.

These notes provide only a detailed motivation for studying modal logic and logics of knowledge. It is hoped that the reader will be inspired to follow up the references (for instance, [BS84], [Che80], [Fi 93], [HC68], [HC 84] and [Seg68]) and read further.

2 Basic correspondence theory

The semantics of modal necessity is given in terms of accessibility relations. Invariably, in studied systems, these relations are not some arbitrary binary relations, but satisfy some conditions: for instance it may be an ordering relation. Can we determine the properties of the accessibility relation by appropriate modal formulas ? This is the subject matter of correspondence theory. More specifically, when we consider a formula which is not valid, we can ask: what is the subclass of models over which it is valid ? If we can in fact find a subclass of *frames* (rather than models) over which the formula is valid, then the formula can be seen as determining the accessibility relation in those frames. (See [Ben84] for a detailed treatment.)

We defined a valid formula to be one whose negation is not satisfiable. This ensures that α is valid iff for every model $M = ((W, R), V)$ and for every $w \in W$, $M, w \models \alpha$. More restricted notions of validity are also of interest: Given a model $M = ((W, R), V)$, we call a formula α *M-valid* iff for every $w \in W$, $M, w \models \alpha$. Given a frame $F = (W, R)$, the formula α is said to be *F-valid* iff for every model $M = (F, V)$, α is M-valid. The latter definition leads us to the notion of characterizing formulas.

Let $\mathcal{F}$ denote the class of all frames, that is, sets with binary relations on them. We will denote subclasses of $\mathcal{F}$ by, $\mathcal{C}, \mathcal{C}'$ etc.

Definition 2.1 *We say that a modal formula α* **characterizes** *a class $\mathcal{C}$ of frames iff $\mathcal{C} = \{F \mid \alpha$ is F-valid $\}$.*

Proposition 2.2 *Let $\mathcal{C}'$ be the class of* reflexive *frames, i.e., $\mathcal{C}' = \{(W, R) \mid R$ is reflexive $\}$. Let $p \in P$. The formula $\Box p \supset p$ characterizes $\mathcal{C}'$.*

Proof: ($LHS \subseteq RHS$): Suppose $F \in \mathcal{C}'$, that is, $F = (W, R)$, where R is reflexive. We need to show that the given formula is F-valid. Suppose not. Let $M = (F, V)$ be a model such that for some $w \in W$: $M, w \not\models \Box p \supset p$. Therefore, $M, w \models \Box p$ and $M, w \not\models p$. But R being reflexive, $(w, w) \in R$, and hence when $M, w \models \Box p$, we must also have $M, w \models p$, a contradiction.

($RHS \subseteq LHS$): Suppose F is a frame such that the formula $\Box p \supset p$ is F-valid, but $F \notin \mathcal{C}'$. Hence $F = (W, R)$, and for some $w \in W$, $(w, w) \notin R$. Now consider the model $M = (F, V)$, where $V(w') = \{p\}$ for every w' such that $(w, w') \in R$ and $V(w'') = \emptyset$ for all other $w'' \in W$. Clearly, $M, w \models \Box p$ and $M, w \not\models p$, a contradiction.

$\Box$

Of course, there can be many different formulas characterizing the same class of frames. For instance, the dual of the formula above, $p \supset \Diamond p$ also characterizes the class of reflexive frames.

Exercise 2.3 *Show that, for* any *formula α, $\Box\alpha \supset \alpha$ characterizes the class of reflexive frames.*

Proposition 2.4 *Let $p \in P$.*

1. *$\Box p \supset \Box\Box p$ characterizes the class of* transitive *frames.*
2. *$p \supset \Box\Diamond p$ characterizes the class of* symmetric *frames.*
3. *$\Box p \supset \Diamond p$ characterizes the class of* serial *frames. (A binary relation R on W is said to be serial iff for all $w \in W$, there exists $w' \in W$ such that $(w, w') \in R$).*

Exercise 2.5 1. *Prove the proposition above.*

2. *We saw in the last section that the formula $\Box p \vee \Box\neg p$ is not a valid formula. Which is the subclass of $\mathcal{F}$ characterized by this formula ?*

3. *Call a binary relation R on W* euclidean *if it has the following property: for all $w_0, w_1 \in W$ such that $(w_0, w_1) \in R$, if there exists w_2 such that $(w_0, w_2) \in R$, then $(w_1, w_2) \in R$. Find a formula that characterizes euclidean frames.*

Thus we have formulas of modal logic characterizing pre-orders (reflexive and transitive frames) and equivalence relations. Aren't there some subclasses of frames that *cannot* be characterized by modal formulas ? There are, and in plenty. The simplest ones to consider are simply the negated variants of the conditions mentioned above.

Theorem 2.6 *There is no modal formula that characterizes the class of* irreflexive *frames.*

Proof: It suffices to prove that for any formula α_0, α_0 is satisfiable iff it is satisfiable in a model based on an irreflexive frame. (If we have this, any characterizing formula would be valid over all frames, and clearly $\mathcal{F}$ is a proper superset of the class of irreflexive frames.) One direction is trivial, so let us assume that α_0 is satisfiable, and show that it is indeed satisfiable in an irreflexive frame.

Let $M = ((W, R), V)$ be the given model such that for some $w_0 \in W$, we have $M, w_0 \models \alpha_0$. Define the new model $M' = ((W', R'), V')$ by: $W' \stackrel{\text{def}}{=} W \times \{0, 1\}$. $(w, b)R'(w', b')$ iff wRw' and $((w \neq w')$ or $(w = w'$ and $b \neq b'))$. $V'(w, b) \stackrel{\text{def}}{=} V(w)$, for all $w \in W$, $b \in \{0, 1\}$. Clearly, (W', R') is an irreflexive frame. We can now show the following: $\forall\alpha, \forall(w, b) \in W' : M, w \models \alpha$ iff $M', (w, b) \models \alpha$. This is proved by induction on the structure of α. The base case is true by the definition of V', and the cases when α is of the form $\neg\beta$ or $\beta_1 \vee \beta_2$ follow from the induction hypothesis. Now suppose α is of the form $\Box\beta$. Let $(w, b) \in W'$ such that $M, w \models \Box\beta$, and suppose $(w, b)R'(w', b')$. But then wRw' and by our assumption, $M, w' \models \beta$. By induction hypothesis, we get $M', (w', b') \models \beta$, as required. On the other hand, suppose $M', (w, b) \models \Box\beta$ and $M, w \not\models \Box\beta$. then for some w' such that $(w, w') \in R$, $M, w' \not\models \beta$. Suppose $w \neq w'$. We have $(w, b)R'(w', b)$, and by induction hypothesis, $M', (w', b) \not\models \beta$, contradicting the fact that $M', (w, b) \models \Box\beta$. If $w = w'$, we apply a similar argument using the fact that $(w, b)R'(w', 1 - b)$ and appealing to the induction hypothesis on the latter element of W'. This completes the inductive proof of the statement above. Now, $M', (w_0, 0) \models \alpha_0$, and hence we have shown that α_0 is satisfied in a model based on an irreflexive frame.

$\Box$

Exercise 2.7 *Show that there is no modal formula that characterizes the class of intransitive frames.*

There are a number of similar results in correspondence theory. Another way of understanding such expressiveness or inexpressiveness results is by studying the embedding of modal logic into first order logic. Consider the map τ from ML formulas and a variable to first order formulas in which that variable is free: (here we assume that we have a binary relation symbol R as an atomic one in the language, and a unary predicate letter corresponding to each propositional letter in ML.) $\tau(p, x) = p(x)$; $\tau(\neg\alpha, x) = \neg\tau(\alpha, x)$; $\tau(\alpha \vee \beta, x) = \tau(\alpha, x) \vee \tau(\beta, x)$; $\tau(\Box\alpha, x) = (\forall x')(R(x, x') \supset \tau(\alpha, x'))$. It is easily seen that a formula α of ML is valid iff $(\forall x)\tau(\alpha, x)$ is a valid first order sentence. Thus what we have on hand is a restricted fragment of first order logic.

In a different direction, there are theorems which show formal relationships between the expressiveness of modal logic and that of nonclassical propositional logics. For instance there is a bijective correspondence between modal formulas valid over reflexive transitive frames and those of propositional *intuitionistic* logic. (See [Fi83], [AB75], [Seg68] and [Ras74].)

We conclude our mini-foray into correspondence theory with just one more theorem, similar in spirit to that of the theorem above. Before that, an exercise:

Exercise 2.8 *Show that every satisfiable formula of ML is actually satisfiable in a countable model. (Start with the given model, and cut out worlds from it, keeping only witnesses for $\Diamond$ formulas, and there can be only countably many of these.)*

Theorem 2.9 *Modal logic cannot distinguish between pre-orders and partial orders.*

Proof: (Sketch) We need to show that every formula that is satisfiable in a model based on a pre-ordered frame is also satisfiable in a partially ordered frame. Let $M = ((W, R), V)$ be a model where R is a pre-order, and for some $\widehat{w} \in W$ let $M, \widehat{w} \models \alpha_0$. By the exercise above, we can assume W to be at most countable. Let $\{w_0, w_1, \ldots\}$ be an enumeration of W. We say $w_i \sqsubseteq w_j$ iff $i \leq j$. Define the new model $M' = ((W', R'), V')$ by: $W' \stackrel{\text{def}}{=} W \times \mathcal{N}$, where $\mathcal{N}$ is the set of all natural numbers; $(w_1, m_1)R'(w_2, m_2)$ iff $w_1 R w_2$ and either $(w_2, w_1) \notin R$, or $((w_2, w_1) \in R$ and $(m_1 < m_2$ or $(m_1 = m_2$ and $w_1 \sqsubseteq w_2)))$. $V'((w, m)) \stackrel{\text{def}}{=} V(w)$ for all m. It can be checked that (W', R') is a partial order. We can now show by induction that for every formula α, for all $(w, m) \in W'$, $M, w \models \alpha$ iff $M', (w, m) \models \alpha$. We now have $M', (\widehat{w}, 0) \models \alpha_0$, and we are done.

$\square$

Exercise 2.10 *Fill out the details in the proof of the theorem above.*

Exercise 2.11 *Show that modal logic cannot distinguish between pre-orders and trees.*

3 Completeness and Decidability

Once we have defined the syntax and semantics of any logic, natural questions that arise are: is the set of valid formulas axiomatizable ? Is validity recursive ? What is the complexity of checking whether a formula is true in a structure ? and so on. These are the questions addressed in this section.

3.1 System K

Below we present a complete axiom system for basic modal logic. This is customarily referred to as System K (for Kripke).

Axiom Schemes:

$(A0)$ *Axioms of PC*

$(A1)$ $\Box(\alpha \supset \beta) \supset (\Box\alpha \supset \Box\beta)$

Inference Rules:

$$(MP)\ \frac{\alpha, \quad \alpha \supset \beta}{\beta} \qquad (Nec)\ \frac{\alpha}{\Box\alpha}$$

The soundness of the system is easy to see: if $\alpha \supset \beta$ holds in every accessible world and α holds in each of them, surely β holds in each of them as well. Similarly the necessitation rule Nec is obvious: if α holds in every world in every model, it also holds in every accessible world in every model. Note that the rule is very different from the (non-valid) formula $\alpha \supset \Box\alpha$!

It is trivial to note that axiom $A1$ and the rule Nec give the following derived inference rule:

$$(DR1)\ \frac{\alpha \supset \beta}{\Box\alpha \supset \Box\beta}$$

Exercise 3.1 *1. Show that the following formulas are theses of System K.*

(a) $(\Box\alpha \wedge \Diamond\beta) \supset \Diamond(\alpha \wedge \beta)$.

(b) $\Box(\alpha \wedge \beta) \equiv (\Box\alpha \wedge \Box\beta)$.

(c) $(\Box\alpha \vee \Box\beta) \supset \Box(\alpha \vee \beta)$.

(d) $\Diamond(\alpha \vee \beta) \equiv (\Diamond\alpha \vee \Diamond\beta)$.

(e) $\Diamond(\alpha \wedge \beta) \supset (\Diamond\alpha \wedge \diamond\beta)$.

2. Derive the following inference rule :

$$(DR2)\quad \frac{\alpha \supset \beta}{\Diamond\alpha \supset \Diamond\beta}$$

There is another form in which the rule Nec can be usefully understood. From Nec we get that if $\vdash \neg\alpha$ then $\vdash \Box\neg\alpha$. Hence, contrapositively, we get $\not\vdash \Box\neg\alpha$ then $\not\vdash \neg\alpha$. This is another way of saying that whenever $\Diamond\alpha$ is consistent, then so also is α.

We can now embark on a proof of the completeness of System K. The technique is to show that every consistent formula is satisfiable. Fix a consistent formula α_0. Below we construct a **canonical** model for α_0, in the sense that it is determined syntactically.

Firstly, a consistent set of formulas A is said to be *maximal* iff for every formula $\alpha \notin A$, $A \cup \{\alpha\}$ is inconsistent. We will refer to maximal consistent sets as MCS's and denote them by $A, B, \ldots$ etc. Clearly, if X is a consistent set of formulas, there exists an MCS A such that $X \subseteq A$. (Prove this by considering an enumeration of all formulas and adding them one at a time to X checking for consistency at each stage.)

We can define a binary relation on MCS's as follows: $A \sqsubseteq B$ iff $\{\alpha | \Box\alpha \in A\} \subseteq B$. It is easy to check that $A \sqsubseteq B$ iff $\{\Diamond\alpha | \alpha \in B\} \subseteq A$. Now consider the frame $F = (W, \sqsubseteq)$, where W is the set of all MCS's, and the model $M = (F, V)$, where $V(A) \stackrel{\text{def}}{=} A \cap P$. Note that α_0 being consistent, there exists an MCS $A_0 \in W$ such that $\alpha_0 \in A_0$, so to show satisfiability of α_0, it suffices to prove the following: $\forall\alpha :\ \forall A \in W :\ \alpha \in A$ iff $M, A \models \alpha$.

This is proved by induction on the structure of α, and the proof reduces to showing the following: for every MCS A, if $\Diamond\alpha \in A$, then there exists an MCS B such that $A \sqsubseteq B$ and $\alpha \in B$. To verify this, suppose $\Diamond\alpha \in A$, and define $\Gamma \stackrel{\text{def}}{=} \{\beta | \Box\beta \in A\} \cup \{\alpha\}$. It suffices to prove that Γ is consistent; if so, there exists an MCS B such that $\Gamma \subseteq B$, and from the definition of $\sqsubseteq$ we find that $A \sqsubseteq B$ and $\alpha \in \Gamma \subseteq B$, as required.

To prove consistency of Γ, consider an arbitrary finite subset $\{\beta_1, \ldots, \beta_k, \alpha\}$. It suffices to show that this set is consistent. Note that $\{\Box\beta_1, \ldots, \Box\beta_k, \Diamond\alpha\} \subseteq A$. Now, in any MCS, if it has formulas γ_1 and γ_2, it also has $\gamma_1 \wedge \gamma_2$. Thus $\Box\beta_1 \wedge \ldots \wedge \Box\beta_k \wedge \Diamond\alpha$ is in A. By the thesis *b* in the exercise above, we find that $\Box(\beta_1 \wedge \ldots \wedge \beta_k) \wedge \Diamond\alpha \in A$. By thesis *a* in the same exercise, we get $\Diamond(\beta_1 \wedge \ldots \wedge \beta_k \wedge \alpha) \in A$. Then this formula is consistent, and by the observation above regarding the rule Nec, the formula $\beta_1 \wedge \ldots \wedge \beta_k \wedge \alpha$ is consistent, as required.

We have thus established the following theorem:

Theorem 3.2 $\vdash_K \alpha$ *if and only if* $\models \alpha$.

3.2 Other systems

The canonical model construction is quite general, and provides a methodology for obtaining complete axiomatizations of various subclasses of frames: simply add the characteristic formula and show that the canonical model satisfies the appropriate frame condition.

Let System T = System K + the following scheme $A2$.

$(A2)\quad \Box\alpha \supset \alpha$

Consider any maximal consistent set A, where consistency is with respect to System T. If $\Box\alpha \in A$ then $\alpha \in A$ by axiom $A2$ and hence $A \sqsubseteq A$, thus making the accessibility relation reflexive. Note that the rest of the proof is unchanged, and we get:

Theorem 3.3 *System T provides a complete axiomatization of formulas valid over reflexive frames.*

Similarly we have the axiom scheme $A3$ below, which added to System K gives the resulting system K4, complete for formulas valid over transitive frames. Adding the scheme to T gives system S4 which is complete for pre-orders and hence for partial orders. Adding $A4$ further to S4 gives the system S5 which is complete for equivalence relations. (The names given to these systems, like S4 etc, are for historical reasons and have no other significance.)

$(A3)\quad \Box\alpha \supset \Box\Box\alpha$
$(A4)\quad \alpha \supset \Box\Diamond\alpha$

Exercise 3.4 *Obtain a complete axiomatization of the set of formulas valid over* total orders*, which are pre-orders where any two elements are ordered. Use the following scheme:*

$$(\Diamond\alpha \wedge \Diamond\beta) \supset (\Diamond(\alpha \wedge \Diamond\beta) \vee \Diamond(\beta \wedge \Diamond\alpha))$$

3.3 Decidability

Having a complete axiomatization ensures that the set of valid formulas is recursively enumerable. Therefore if we can show that the set of satisfiable formulas is also recursively enumerable, then both of these sets must be recursive, and hence the logic would be decidable. For this, it suffices to prove the *finite model property*: that is, a formula is satisfiable if and only if it is satisfiable in a finite model. (The set of finite models is recursively enumerable, and checking truth in a structure is recursive.)

We will do better. We will show a **small model property**: a formula α is satisfiable iff it is satisfiable in a model whose size is bounded by some (at most) exponential function of $|\alpha|$. This implies the following decidability result.

Theorem 3.5 *Satisfiability of an ML formula α can be checked in time $2^{O(|\alpha|)}$.*

The small model is constructed by a technique known as *filtration*. This requires looking at which subformulas need to be satisfied in order to satisfy a formula, and collapsing the rest of the model to retain only an appropriate substructure. We first need the notion of *subformula closure*. For any formula α, $CL'(\alpha)$ is defined to be the smallest set of formulas containing α and satisfying the following conditions:

- if $\neg\beta \in CL'(\alpha)$ then $\beta \in CL'(\alpha)$.
- if $\beta_1 \vee \beta_2 \in CL'(\alpha)$ then $\{\beta_1, \beta_2\} \subseteq CL'(\alpha)$.
- if $\Box\beta \in CL'(\alpha)$ then $\beta \in CL'(\alpha)$.

$CL(\alpha) \stackrel{\text{def}}{=} CL'(\alpha) \cup \{\neg\beta | \beta \in CL'(\alpha)\}$. CL is closed under negation, where $\neg\neg\beta$ is considered as the same as β. The size of $CL(\alpha)$ is linear in the length of α.

Now consider a satisfiable formula α_0, and let $M = ((W, R), V)$ be the model such that for some w_0, $M, w_0 \models \alpha_0$. We will call $CL(\alpha_0)$ simply CL from now. Consider the following equivalence relation on W: $w \sim w'$ iff $\forall\gamma \in CL$: $M, w \models \gamma$ iff $M, w' \models \gamma$. Let $[w]$ denote the equivalence class of

w under $\sim$. Define the small model $M' = ((W', R'), V')$ by: $W' \stackrel{\text{def}}{=} \{[w] | w \in W\}$. $[w]R'[w']$ iff there exist $w_1 \in [w], w_2 \in [w']$ such that $w_1 R w_2$. $V'([w]) \stackrel{\text{def}}{=} V(w) \cap CL$. Note that V' is well-defined, since all worlds in $[w]$ satisfy the same propositions in CL.

Note that $[w_0] \in W'$, so it suffices to prove the following: $\forall \gamma \in CL$: $\forall w \in W$: $M, w \models \gamma$ iff $M', [w] \models \gamma$. As usual, the proof is by induction on γ. The base case and the boolean cases are routine. Suppose γ is of the form $\Box\alpha$. Let $M', [w] \models \Box\alpha$. To prove that $M, w \models \Box\alpha$, consider w' such that wRw' holds. But then $[w]R'[w']$ holds, and by assumption, $M, [w'] \models \alpha$. Since $\alpha \in CL$, the induction hypothesis applies, so $M, w' \models \alpha$, as required.

On the other hand, suppose $M, w \models \Box\alpha$, and $[w]R'[w']$. We need to prove that $M', [w'] \models \alpha$. Let $w_1 \in [w]$, $w_2 \in [w']$ such that $w_1 R w_2$. Since $w \sim w_1$ and $\Box\alpha \in CL$, we have that $M, w_1 \models \Box\alpha$ and hence $M, w_2 \models \alpha$. Again since $w' \sim w_2$, we get that $M, w' \models \alpha$. But then by induction hypothesis on α, $M', [w'] \models \alpha$, as required. This completes the inductive argument, and we have a model for α_0 whose size is at most exponential in the size of α_0 (as W' consists of subsets of CL).

In fact, we can show that the decidability problem is PSPACE-complete, but this is a tedious Turing machine encoding argument and hence omitted here. The reader is referred to [Lad77].

Exercise 3.6 *Show that ML is compact, that is, a set of ML formulas is satisfiable iff every finite subset is.*

Exercise 3.7 *Modify the filtration procedure above to get decidability results over reflexive frames, pre-orders etc. (Note that if we do filtration as above, the small structure need not be a transitive frame even when the given structure is.)*

3.4 Checking truth

Often we are not as much interested in knowing whether a formula is satisfiable, but whether it is satisfied in a given finite structure at a specific world ('the initial state'). An algorithm for doing this is easy to construct: enumerate the subformulas of the given formula in increasing order of complexity. Assume that the given structure is decorated with the propositions true at each world. In one pass, determine negations: $\neg\beta$ is added to a world only if β is not already present. Then a pass for disjunctions. When it comes to $\Box$ formulas, the subformulas have already been decided, so only the successor worlds need to be checked that they have the appropriate subformula; similarly for a $\Diamond$ formula, one of the successor worlds should have the subformula. If at the end of this procedure, the designated world has the given formula, we say 'yes'.

Theorem 3.8 *Checking truth of a formula of size m in a structure of size k can be done in time $O(km)$.*

Exercise 3.9 *Do we need to make any changes in the truth checking procedure when the frames are restricted to be transitive frames, pre-orders etc ?*

4 Dynamic logic

Earlier, we described modal logic as a language for talking about state changes. During the late seventies and early eighties, there was extensive research in the area of *logics of programs*, where the central idea was to study behaviours of programs using formal logics. An important development in this regard was the programming methodology of *predicate transformers* by Dijkstra and its subsequent formalization in a logic of *partial correctness assertions* by Hoare (subsequently these logics were called Hoare logics). Vaughan Pratt [Pra79] suggested that the view of programs as state transformers is best seen as a modal logic, where a *dynamic* modality specifies what happens after the execution of a program. The technical literature on dynamic logic rapidly encompassed and subsumed the considerations of other program logics and provided a foundation for most subsequent studies on system verification, notably in temporal logics ([Gol92], [MP 92]), the propositional μ-calculus ([Ko83]) and logics of knowledge and belief. In this section, we study the basic concepts of dynamic logic.

4.1 Syntax and semantics

While it is customary in philosophical logic to first define the modalities of interest motivated by philosophical intuition, propose suitable axioms and then look for models, the approach is different when we look for modal logics to describe computations. Here we begin with a class of intended structures guided by computational considerations, and look for modalities which adequately describe these structures. In this tradition, automata or action-labelled transition systems are the beginning point for the understanding of dynamic logics.

To recapitulate, a transition system is a tuple $TS = (S, \rightarrow, s_0)$, where S is a set of states, $\rightarrow \subseteq (S \times \Sigma \times S)$ where Σ is a finite alphabet of actions, and $s_0 \in S$ is the initial state. Without loss of generality, we can assume that every state in S is reachable from s_0. Now suppose that S is finite. How would we describe such a transition system in a modal logic ?

An obvious suggestion is to think of one proposition for each state. Thus, let $S \subseteq P$. We can then write down the following formulas to describe TS:

1. $\delta_0 \stackrel{\text{def}}{=} s_0$.
2. $\delta_1 \stackrel{\text{def}}{=} \bigvee_{s \in S} s$.
3. $\delta_2 \stackrel{\text{def}}{=} \bigwedge_{s \neq s'} (s \supset \neg s')$.
4. $\delta_3 \stackrel{\text{def}}{=} \bigwedge_{s \in S} (s \supset (\mathit{in\ every\ a - successor\ state}) \bigvee_{s' : s \xrightarrow{a} s'} s')$.
5. $\delta_4 \stackrel{\text{def}}{=} \bigwedge_{s \in S} (s \supset \bigwedge_{s' : s \xrightarrow{a} s'} (\mathit{in\ some\ a - successor\ state}) s')$.

We could use action-indexed versions of the familiar $\Box$ and $\Diamond$ modalities respectively for the ones required in the formulas δ_3 and δ_4 respectively. However, the conjunction $\bigwedge_{i \in \{0,\ldots,4\}} \delta_i$ does *not* express the transition system TS, but only the state s_0 and its successors in TS. What we need is another modality: *'in every reachable state'*, (with dual 'in some reachable state'), and with such a modality, the formula: $s_0 \wedge \mathit{in\ every\ reachable\ state} \bigwedge_{i \in \{1,\ldots,4\}} \delta_i$ does describe the transition system TS precisely.

Propositional dynamic logic is a propositional modal logic with two modalities, one referring to successor states (worlds) and the second to worlds reachable by paths 'stepping' through the other. Since we wish to remain close to the intuition above, we will choose not one next state modality, but finitely many such modalities, indexed by actions. Thus the logic is parametrized by a finite alphabet Σ of actions.

As usual let $P = \{p_0, p_1, \ldots\}$ and $\Sigma = \{a_1, \ldots, a_n\}$. The syntax of formulas is given by: $DL_\Sigma ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid [a]\alpha,\ a \in \Sigma \mid \Box\alpha$. The dual modalities are: $\langle a \rangle \alpha \stackrel{\text{def}}{=} \neg[a]\neg\alpha$ and $\Diamond\alpha \stackrel{\text{def}}{=} \neg\Box\neg\alpha$. We also have derived modalities $\bigcirc\alpha \stackrel{\text{def}}{=} \bigvee_{a \in \Sigma} \langle a \rangle \alpha$ and its dual $\odot\, \alpha \stackrel{\text{def}}{=} \neg\bigcirc\neg\alpha$.

To consider an example of how these modalities are used, suppose we are describing systems where the set of states is partitioned into two regions: *good* and *bad*. then the formula $\Box((good \supset \bigcirc bad) \wedge (bad \supset \Diamond good))$ specifies that from every state in the good region it is possible to exit the region in a single step, whereas a return may need several steps, though always possible.

Frames are transition systems $TS = (S, \rightarrow)$ over Σ. We dispense with the explicit initial state, as formulas are asserted at a state and that is implicitly the initial state. Models are pairs $M = (TS, V)$, where $V : S \rightarrow 2^P$. Let $R(s)$ denote the set of all *reachable* states from s in TS: a state s' is said to be reachable from s if there exists a sequence $s_0, s_1, \ldots, s_k$ of states in S such that $s_0 = s$, $s_k = s'$ and for all $0 \leq i < k$, $s_i \xrightarrow{a_i} s_{i+1}$ in TS for suitable $a_0, a_1, \ldots, a_{k-1}$.

We are now ready to define the notion of when a formula holds at a state in a model.

- $M, s \models p$ iff $p \in V(s)$.
- $M, s \models \neg\alpha$ iff $M, s \not\models \alpha$.
- $M, s \models \alpha \vee \beta$ iff $M, s \models \alpha$ or $M, s \models \beta$.
- $M, s \models [a]\alpha$ iff $\forall s'$ such that $s \xrightarrow{a} s'$: $M, s' \models \alpha$.
- $M, s \models \Box\alpha$ iff $\forall s' \in R(s)$, $M, s' \models \alpha$.

The notions of satisfiability, validity etc are as before. An immediate consequence of the semantic definition is that the logic is *not compact*. Consider the set of formulas $\{\Diamond p, \neg p, \odot\neg p, \odot\odot\neg p, \ldots\}$. Every finite subset is satisfiable, but the set itself is not. This is a feature of all program logics, which complicates technical study.

4.2 Axiomatization and decidability

Below we present a Hilbert-style axiom system for dynamic logic. Instead of showing completeness and decidability of the logic separately, we will do it together in one shot, a style due to Kozen and Parikh [KP81]. The strategy is as follows: a standard Henkin-style completeness proof shows that every consistent formula is satisfiable. Instead we show that every consistent formula is satisfiable in a model whose size is bounded by 2^{cm} where m is the length of the formula, and c is some constant. Thus we have:

- By soundness, every satisfiable formula is consistent.
- By this completeness proof, every consistent formula is satisfiable in a small model.
- Every formula satisfiable in a small model is, trivially, satisfiable.

Thus satisfiability coincides with consistency (completeness) and a formula is satisfiable iff it is satisfiable in a small model (decidability).

Axiom Schemes:

$(A0)$	$Axioms\ of\ PC$
$(A1)$	$\Box(\alpha \supset \beta) \supset (\Box\alpha \supset \Box\beta)$
$(A2)$	$[a](\alpha \supset \beta) \supset ([a]\alpha \supset [a]\beta), \quad a \in \Sigma$
$(A3)$	$\Box\alpha \supset (\alpha \wedge \odot\Box\alpha)$

Inference Rules:

$$(MP)\ \frac{\alpha,\quad \alpha \supset \beta}{\beta} \qquad (Nec)\ \frac{\alpha}{\Box\alpha}$$

$$(Ind)\ \frac{\alpha \supset \bigodot \alpha}{\alpha \supset \Box\alpha}$$

The axioms are quite straightforward: we have two modalities now, so we need two Kripke axioms, and these are given as $A1$ and $A2$. The scheme $A3$ relates the two modalities. The new kind of reasoning in dynamic logic is typified only in the new rule (Ind) which is an induction principle for the logic: to infer that a formula α holds in all reachable states, it must hold at the asserted state, and whenever it holds in any reachable state, it must hold in every one-step successor of that state. Every multi-modal logic, where one modality describes stepping through another requires an induction principle of this type.

(The $\Box$ modality being transitive, it might be surprising not to find the characteristic formula for transitivity in the axiom system. The following exercise shows that it is derivable.)

Exercise 4.1 *1. Derive the following theses:*

(a) $\Box\alpha \supset \bigodot \alpha$.

(b) $\Box\alpha \supset \Box\Box\alpha$.

(c) $\Box(\alpha \supset \bigodot \alpha) \supset (\alpha \supset \Box\alpha)$.

2. Derive the rule: from $\vdash \alpha$, *infer* $\vdash [a]\alpha$.

3. Derive the rule: from $\vdash \alpha \supset \beta$, *infer* $\vdash \langle a\rangle\alpha \supset \langle a\rangle\beta$.

We now embark on the completeness proof. As before, we first define the notion of subformula closure, we only need the following additional closure condition: if $\Box\alpha \in CL'$, then $\{\alpha, [a]\Box\alpha\} \subseteq CL'$, $a \in \Sigma$. Note that the size of $CL(\alpha)$ still remains linear in the size of α. Fix a consistent formula α_0, and as before, we will refer to $CL(\alpha_0)$ simply as CL.

Instead of maximal consistent sets of formulas, we will consider only maximal consistent subsets of CL. Let AT denote the set of all such MCSubsets of CL. We refer to them as *atoms* and use w, w' etc to range over AT. Since α_0 was assumed to be consistent, there exists $w_0 \in AT$ such that $\alpha_0 \in w_0$. Each $w \in AT$ is a finite set of formulas, so we can speak of the conjunction of all formulas in w; we denote this conjunction by $\widehat{w}$. If $\alpha \in w$ then $\vdash \widehat{w} \supset \alpha$. When we have a nonempty subset X of AT, we use the notation $\widetilde{X}$ to denote the disjunction of all $\widehat{w}$, $w \in X$. If $w \in X$, then $\vdash \widehat{w} \supset \widetilde{X}$.

Exercise 4.2 *1. Show that* $\vdash \widetilde{AT}$.

2. Let $X, X' \subseteq AT$ *such that* $X \cup X' = AT$ *and* $X \cap X' = \emptyset$. *Show that* $\vdash \widetilde{X} \equiv \neg\widetilde{X'}$.

Define a transition relation on AT as follows: $w \overset{a}{\Rightarrow} w'$ iff $\widehat{w} \wedge \langle a\rangle\widehat{w'}$ is consistent. Now define the model $M = ((AT, \Rightarrow), V)$, where $V(w) \overset{\text{def}}{=} w \cap P$. We will show that $M, w_0 \models \alpha_0$.

Suppose $w \overset{a}{\Rightarrow} w'$. Note that if $[a]\alpha \in w$, then $\alpha \in w'$. (Proof: $\widehat{w} \wedge \langle a\rangle\widehat{w'}$ is consistent, hence, omitting some conjuncts, $[a]\alpha \wedge \langle a\rangle\widehat{w'}$ is consistent. Therefore, $\langle a\rangle(\alpha \wedge \widehat{w'})$ is consistent, and so, (by derived rule) $\alpha \wedge \widehat{w'}$ is consistent. By maximality of w', $\alpha \in w'$.) Further, if $\Box\alpha \in w$ then $\{\alpha, \Box\alpha\} \subseteq w'$. (Proof: If $\Box\alpha \in w$, by axiom $A2$ and the definition of CL, $[a]\Box\alpha \in w$, and by the remark above $\Box\alpha \in w'$. Again applying axiom $A2$, we find that $\alpha \in w'$ as well.) This ensures that whenever we have a path $w \overset{a_1}{\Rightarrow} w_1 \overset{a_2}{\Rightarrow} \ldots \overset{a_k}{\Rightarrow} w_k$ and $\Box\alpha \in w$, we have $\{\alpha, \Box\alpha\} \subseteq w_k$.

Now suppose $\langle a \rangle \alpha \in w$. As before, define $\Gamma \stackrel{\text{def}}{=} \{\beta | [a]\beta \in w\} \cup \{\alpha\}$, and show that $\widehat{w} \wedge \langle a \rangle \widehat{\Gamma}$ is consistent. This also means that Γ is consistent, and hence there is at least one atom containing Γ. Let X be the set of atoms extending Γ. $\vdash \widehat{\Gamma} \supset \widetilde{X}$ and hence $\widehat{w} \wedge \langle a \rangle \widetilde{X}$ is consistent. Therefore, there exists $w' \in X$ such that $w \stackrel{a}{\Rightarrow} w'$ and $\alpha \in \Gamma \subseteq w'$.

Now suppose $\Diamond \alpha \in w$. We would like to show that there is a path $w \stackrel{a_1}{\Rightarrow} \ldots \stackrel{a_k}{\Rightarrow} w_k$ in $(AT, \Rightarrow)$ such that $\alpha \in w_k$. Towards this, consider the least set R containing w and closed under the following condition: if $w_1 \in R$ and $w_1 \stackrel{a}{\Rightarrow} w_2$ for some a, then $w_2 \in R$. Clearly, R is a finite nonempty subset of AT, so $\widetilde{R}$ is defined. If there exists any atom in $w' \in R$ such that $\alpha \in w'$, we are done, as every atom in R is reachable by a finite path from w.

Suppose not. Then no atom in R has α, and we have $\vdash \widetilde{R} \supset \neg\alpha$, and hence $\vdash \Box\widetilde{R} \supset \Box\neg\alpha$.

Claim: $\vdash \widetilde{R} \supset \odot \widetilde{R}$.

Suppose the claim is true. By the induction rule, we get $\vdash \widetilde{R} \supset \Box\widetilde{R}$ and from our assumption, $\vdash \widetilde{R} \supset \Box\neg\alpha$. But $w \in R$ and hence $\vdash \widehat{w} \supset \widetilde{R}$, and we get $\vdash \widehat{w} \supset \Box\neg\alpha$, contradicting our assumption that $\Diamond\alpha \in w$. Thus we only need to prove the claim.

Proof of Claim: Suppose the claim is false. Then $\widetilde{R} \wedge \bigcirc\neg\widetilde{R}$ is consistent. Hence for some $a \in \Sigma$, $\widetilde{R} \wedge \langle a \rangle \neg\widetilde{R}$ is consistent. Now let $R' = AT - R$. If $R' = \emptyset$, then $AT = R$, and hence by exercise above $\vdash \widetilde{R}$; by necessitation, $\vdash \odot \widetilde{R}$ and hence the claim cannot be false as assumed. Therefore, $R' \neq \emptyset$. But then, $\neg\widetilde{R} \equiv \widetilde{R'}$. Thus $\widetilde{R} \wedge \langle a \rangle \widetilde{R'}$. Hence for some $w_1 \in R$ and $w_2 \in R'$, $\widehat{w_1} \wedge \langle a \rangle \widehat{w_2}$ is consistent. But this means that $w_1 \stackrel{a}{\Rightarrow} w_2$ and by closure condition on R, $w_2 \in R$, contradicting the fact that $w_2 \in (AT - R)$.

Thus we have established the following:

- Whenever $[a]\alpha \in w$ and $w \stackrel{a}{\Rightarrow} w'$, $\alpha \in w'$.
- Whenever $\Box\alpha \in w$ and $w' \in R(w)$, $\alpha \in w'$.
- Whenever $\langle a \rangle \alpha \in w$, there exists w' such that $w \stackrel{a}{\Rightarrow} w'$ and $\alpha \in w'$.
- Whenever $\Diamond\alpha \in w$, there exists $w' \in R(w)$ such that $\alpha \in w'$.

From these, it is a routine inductive argument to show that: $\forall \gamma \in CL$: $\forall w \in W : \gamma \in w$ iff $M, w \models \gamma$. This ensures that $M, w_0 \models \alpha_0$, and we have a model for α_0 whose size is bounded by $2^{c \cdot |\alpha_0|}$, for some constant c. Thus we have:

Theorem 4.3 *The axiom system above for DL is complete. Satisfiability is decidable in nondeterministic exponential time.*

We can modify the procedure by simply considering syntactic subsets of CL which are propositionally consistent, and pruning the atom graph checking for violations, thus obtaining a deterministic exponential time decision procedure. See [Har84] for details. Pratt has shown that deterministic exponential time is in fact a lower bound.

Exercise 4.4 *Give an algorithm for determining truth of a DL formula in a given finite transition system. What is the complexity of this problem ?*

Exercise 4.5 *Check that a completeness proof for DL along the lines of the proof in the last section (using MCSs) does not work. What goes wrong ?*

4.3 Dynamic logic and programs

To set the record straight, dynamic logic is not defined with the syntax above, but over a class of programs. PDL, the propositional dynamic logic of *regular* programs [FL79] is defined as follows: Let Reg_Σ denote the set of regular expressions over Σ given by the following syntax: $\pi \in Reg_\Sigma ::= a \in \Sigma \mid \pi_1 + \pi_2 \mid \pi_1 ; \pi_2 \mid \pi^*$. The syntax of formulas is then given by: $PDL_\Sigma ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid [\pi]\, \alpha, \pi \in Reg_\Sigma$.

How is the semantics given ? For any regular expression r, let $L(r)$ denote the regular language associated with it. Further, given a transition system $TS = (S, \rightarrow)$ over Σ, and $\rho \in \Sigma^*$, we can define the extended transition relation $\xrightarrow{\rho}$. Thus, for any regular expression π, we define $s \overset{\pi}{\Rightarrow} s'$ iff for some $\rho \in L(\pi), s \xrightarrow{\rho} s'$. We can now define the semantics of the modality: $M, s \models [\pi]\alpha$ iff for every s' such that $s \overset{\pi}{\Rightarrow} s'$, $M, s' \models \alpha$.

What are the axioms for PDL ? We need new axioms to describe how regular expressions are built up. The obvious ones are:

- $\langle \pi_1 + \pi_2 \rangle \alpha \equiv (\langle \pi_1 \rangle \alpha \vee \langle \pi_2 \rangle \alpha)$.
- $\langle \pi_1 ; \pi_2 \rangle \alpha \equiv \langle \pi_1 \rangle \langle \pi_2 \rangle \alpha$.
- $\langle \pi^* \rangle \alpha \equiv (\alpha \vee \langle \pi \rangle \langle \pi^* \rangle \alpha)$.

What is the new form of the induction rule ? It is worth finding out.

Exercise 4.6 *Modify the induction rule appropriately for PDL and re-work the completeness -cum-decidability proof.*

In PDL, apart from regular programs, it is also usual to consider **test** programs: formulas and programs are defined by mutual recursion, and if α is a formula then $\alpha?$ is a program. The idea is that a state has an $s \overset{\alpha?}{\Rightarrow} s'$ transition iff $s = s'$ and $M, s \models \alpha$. Thus, it is a test of whether α holds, and if not, the program aborts. It is easy to see that $\models \langle \alpha? \rangle \beta \equiv (\alpha \wedge \beta)$ and in fact, adding this as an axiom gives completeness as well.

PDL has been enriched in many directions: first-order dynamic logic, the dynamic logic of non-regular programs, the logic of programs with inverses, and so on. Typically these tend to be undecidable or of high complexity. See [Har84] for an excellent survey of the subject.

5 Logic of Knowledge

It is easy to see that truth of a proposition is quite different from knowledge that it is true. Consider a proposition like "The sun rises in the east". We not only claim it to be true, but also easily claim knowledge of it, based on our unvarying experience that the sun, indeed, rises in the east. On the other hand, consider the proposition that it is the earth that goes around the sun and not the other way around. Again, we claim that this is true, but claim knowledge much less easily. In fact, defending the assertion is quite nontrivial.

In this discussion, there is a tacit assumption that knowledge has to do with the ability to defend the claim to knowledge. This has crucial importance in computing systems as well: a database system may answer "I don't know" and we take it to mean that it has not enough evidence to justify knowledge of whatever that's queried. In multi-agent systems, this kind of reasoning is carried further: A says that something is true, knows that B has no way of knowing it (maybe because some crucial bit of evidence is available only to A and not B), and capitalises on this knowledge of B's ignorance. Game theoretic situations in economics are typical examples of this kind. In distributed protocols, we can easily envisage a situation where a process knows that another process does not know the latest information about something and hence proceeds to inform the latter about it (for instance, that a third process is blocked indefinitely).

These examples suggest that reasoning about knowledge is at once natural, useful and tricky. Epistemology is a very old branch of philosophy, but with the advent of modal logics, there have been attempts to study knowledge-like notions mathematically. Recently, in the last ten years, there has been great interest in this area among computer scientists interested in distributed protocols or in building knowledge bases (for building intelligent systems), among mathematical economists studying information in the context of bargains, and among philosophers attempting to characterize the dynamics of epistemic revision and update (see [FHMV95] for references).

In this section, we present the basic issues in studying logics of knowledge, offering just a flavour of the subject.

5.1 Hintikka's Logic

Jakko Hintikka [Hin62] was the first to suggest studying knowledge as a modality with possible worlds semantics. The syntax is simply that of modal logic, with a knowledge modality rather than necessity: $LK ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid \mathbf{K}\alpha$. The modality is read "$\alpha$ is known". The dual modality, defined by $\mathbf{L}\alpha \stackrel{\text{def}}{=} \neg\mathbf{K}\neg\alpha$ is read as "α is allowed".

Hintikka's definition of knowledge is as follows: when a reasoning agent is unsure about the truth or falsity of a proposition p, she considers both worlds possible, one in which p is true, and the other in which p is false. She 'allows' for both possibilities. Stating the same thing conversely, when an agent is certain about the truth of p, or when she **knows** that p holds, p must be true in all the worlds that the agent considers possible at that state. Thus, at each state, an agent considers a set of worlds possible, or cannot distinguish between them to determine which is the actual state of the world, and knows (at that state) whatever is invariant in all these worlds. This notion captures the sense in which knowledge is talked about, at least in some of the application areas mentioned above.

Formally a frame is a pair $F = (W, \sim)$ where $\sim$ is an *indistinguishability* relation for the agent. If $w \sim w'$, then the agent cannot tell, in w, whether the actual state of the world is w or w'. A model is a pair $M = (F, V)$, where V is a valuation function, and the notion of truth is defined inductively as usual. The definition for the modality is given as follows: $M, w \models \mathbf{K}\alpha$ iff $\forall w' \sim w, M, w' \models \alpha$.

Technically, we have met this logic before. This is simply the modal logic of equivalence frames, and we studied the System $S5$ which axiomatizes these frames. Below we present the characteristic axioms, which specify respectively reflexivity, transitivity and euclideanness (from which symmetry can be derived).

- $\mathbf{K}(\alpha \supset \beta) \supset (\mathbf{K}\alpha \supset \mathbf{K}\beta)$.
- $\mathbf{K}\alpha \supset \alpha$.
- $\mathbf{K}\alpha \supset \mathbf{K}\mathbf{K}\alpha$.
- $\neg\mathbf{K}\alpha \supset \mathbf{K}\neg\mathbf{K}\alpha$.

The first of these asserts that knowledge is closed under logical consequence. The next says that whatever is known should be true. The third and fourth are respectively known as positive and negative introspection axioms. They say that we know about our knowledge as well as about our ignorance. These are quite debatable, and much debated. For the present, let us simply examine the technical questions related to the logic. That these axioms, along with PC and the necessitation rule, give completeness is known to us. We also know that the logic is decidable, and that determining that a formula of length m is true in a structure of size k is possible in time $O(mk)$. A closer look shows that the decision procedure we studied earlier can be improved.

Theorem 5.1 *The satisfiability problem for LK is NP-complete.*

It is, of course, NP-hard, as the logic includes propositional logic. To obtain an NP algorithm, it suffices to show that every satisfiable formula of length m is satisfiable in a model whose size is linear in m. For this, consider a given model and let w be the world at which the given formula is satisfied. First, we need to keep only the $\sim$-equivalence class of w and throw away the rest. Secondly, among elements in $[w]$, we need to keep exactly as many worlds as needed as witnesses for $\mathbf{L}$-formulas (note that all worlds in an equivalence class satisfy the same modal formulas). We need to consider only those which are subformulas of the given formula, and hence their number is linear in m.

Exercise 5.2 *Fill out the details in the proof above.*

5.2 Systems of many reasoners

The main interest in reasoning about knowledge is when we have systems of many reasoners. Fix $n > 0$ and we will speak of n-agent systems. For this we enrich the logic to have n knowledge modalities $\mathbf{K}_i$, $i \in \{1, \ldots, n\}$. Correspondingly, the frames are given by tuples $F = (S, \sim_1, \ldots, \sim_n)$, and the semantics of the modalities is given in the obvious way. Call this logic LK_n.

The valid formulas of LK_n are axiomatized by a straightforward generalization of the axiomatization of LK: simply replace $\mathbf{K}$ by $\mathbf{K}_i$. However, the decidability question is harder now.

Theorem 5.3 *The satisfiability problem for LK_n is PSPACE-complete.*

That is because the logic includes modalities like $K_i K_j$ etc. and these do not behave like LK modalities at all. In fact, we can now consider quite nontrivial modalities like 'group knowledge'.

Let $G \subseteq \{1, 2, \ldots, n\}$. Consider the following modalities: $E_G\alpha$ which stands for: "everyone in the group G knows that α holds"; $I_G\alpha$ for: "it is implicitly known among the members of the group G that α holds"; $C_G\alpha$ for: "it is common knowledge among members of G that α holds". The first of these is easiest to define:

$M, w \models E_G\alpha$ iff $\forall i \in G, M, w \models K_i\alpha$.

What about implicit or 'distributed' knowledge ? Such a situation arises, when for instance, we have $K_1\alpha$ and $K_2(\alpha \supset \beta)$, but neither of the two know β. Still the information that β holds is implicitly available to them and can be discovered through communication. How can this be formalized ? A little bit of thought shows that combining the knowledge of several agents amounts to eliminating any world any agent in G would consider impossible. That is, we need to *intersect* the worlds considered possible by agents in G.

$M, w \models I_G\alpha$ iff $\forall w'$: if $(\forall i \in G w \sim_i w')$ then $M, w' \models \alpha$.

How do we define common knowledge ? We say that $C_G\alpha$ is true, if everyone in G knows α, everyone in G knows that everyone in G knows α etc. Let $E_G^0\alpha \stackrel{\text{def}}{=} \alpha$ and $E_G^{k+1}\alpha \stackrel{\text{def}}{=} E_G E_G^k\alpha$. C_G stands for the infinite conjunction of E_G^k.

$M, w \models C_G\alpha$ iff $\forall k \geq 0, M, w \models E_G^k\alpha$.

An interesting implication of the definition above is graph-theoretic: given a frame $(W, \sim_1, \ldots, \sim_n)$, define a graph $(W, \rightarrow)$ as follows, where $\rightarrow \subseteq (W \times \{1, \ldots, n\} \times W)$. $w \xrightarrow{i} w'$ iff $w \sim_i w'$. A G-path from w to w' is a path in this graph from w to w' where all the edges are labelled by names from G. We then say w' is reachable from w.

Proposition 5.4

1. *$M, w \models E_G^k\alpha$ iff for every w' which is reachable from w by a path of length k, $M, w' \models \alpha$.*
2. *$M, w \models C_G\alpha$ iff for all w' reachable from w, $M, w' \models \alpha$.*

In terms of characteristic axioms, the following ones suffice to give completeness for appropriate addition of these operators: (we need to add the deductive closure axioms for each modality.)

- $E_G\alpha \equiv \bigwedge_{i \in G} K_i\alpha$.
- $C_G\alpha \supset E_G(\alpha \wedge C_G\alpha)$.
- From $\vdash \alpha \supset E_G\alpha$, infer $\vdash \alpha \supset C_G\alpha$.
- $I_{\{i\}}\alpha \equiv K_i\alpha$.
- $I_G\alpha \supset I_{G'}\alpha$, for $G \subseteq G'$.

Exercise 5.5 *Show the completeness of axiom systems for the group knowledge operators.*

The filtration argument given earlier can be extended to show decidability for these logics as well, but the complexity is high.

Theorem 5.6 *Satisfiability of formulas in $LK_n + C_G$ is EXPTIME-complete.*

5.3 A puzzle

Following a time-honoured tradition in philosophy, we will study how reasoning about the knowledge of other agents affects the knowledge of an agent. There are two children, unimaginatively named a and b, who are playing and they have been warned by their mother not to get dirty during play. As children will, both of them get dirty. Each can see mud on the other's forehead, but cannot determine whether (s)he is dirty. In walks the logician mother, and announces: "At least one of you has a dirty forehead". (It seems strange behaviour as she is only telling them what obviously they know already, but then she is a logician.) Then she asks, "Does either of you know whether your own forehead is dirty ?". Neither children answer. Then she *asks the same question again*. This time both children answer, "I know mine is". Analyse what happened, assuming (of course !) that the children are perfect reasoners.

Since we have the logic LK_2 on hand, let's try out the analysis in the logical framework. For convenience, we will use the names a and b rather than 1 and 2 as we are used to. Let A stand for the proposition "a has a muddy forehead", and B for b's mud. Thus K_aB stands for the assertion, "a knows that B has a dirty forehead" which is true initially.

What is the initial situation ? Each child sees that the other is dirty, and knows this fact about the other too. Formally, we have: (Below we will look at everything only from the child a's point of view; the situation is symmetric for b.)
Init: $K_a(K_bA \vee K_b\neg A)$

What is the content of the mother's statement ? It ensures that both children know the (obvious) fact $A \vee B$, that is, it ensures $E_{\{a,b\}}(A \vee B)$. In fact, a little thought tells us that $A \vee B$ becomes common knowledge among the two children as a result of the mother's announcement. We don't need it here, so we pick only the weaker formula:
Mother said: $K_aK_b(A \vee B)$

After the silence that answers her mother's first question, what does a learn ? At this point, a infers that b does not know whether his own is forehead muddy or not. Thus:
After first query: $K_a\neg K_bB$

The problem now is for us to deduce from these formulas (somehow !) that at this stage a knows the answer. Therefore, we want:
Final: K_aA

We first analyse the situation semantically. We do everything from a's point of view, so the crux of the argument is how much information a has about $\sim_b$-equivalence classes of worlds at each stage. Let W be the set of all possible worlds (defined as you like). Suppose the formulas are being asserted at $w \in W$. Now consider W' such that W' is the $\sim_a$-equivalence class containing w. $\sim_b$ partitions W' further into equivalence classes among which b cannot distinguish. Let these classes be $W_1, W_2, \ldots$.

What does **Init** mean semantically ? As a knows that b knows the truth value of A, for each i, either A holds in all members of W_i or it is false in all members of W_i. Next by what **Mother said**, a knows that b knows that $A \vee B$ holds. Then looking into the a-indistinguishable cluster W', in each of the b-indistinguishable sub-clusters W_i, $A \vee B$ holds throughout W_i. Thus, combining what we said earlier, we find that for each i, either A holds throughout W_i or B holds throughout W_i.

Now, **After first query**, a knows that b knows nothing about B, and this means that we cannot have any i such that B holds throughout W_i. Therefore, A must hold throughout W_i, for each i. But then $W' = \bigcup\limits_i W_i$, and hence A holds throughout W'. This is semantically the same as saying that **Final** holds at w, which is what we want.

We also have an axiom system on hand, so it may be instructive to see how we can prove **Final** from the other formulas using the axiom system. To keep clutter down, we need an exercise.

Exercise 5.7 *Derive the following inference rules:*

1. *From* $\vdash \mathbf{K}_i\mathbf{K}_j\alpha$ *infer* $\vdash \mathbf{K}_i\alpha$.
2. *From* $\vdash \mathbf{K}_i\mathbf{K}_j(\alpha \supset \beta)$, *infer* $\vdash \mathbf{K}_i\neg\mathbf{K}_j\beta \supset \mathbf{K}_i\neg\mathbf{K}_j\alpha$.

We now present a formal proof of the argument outlined above. Note that the proof exactly mirrors the semantic argument.

Proof:

1.	$\mathbf{K}_a\mathbf{K}_b(\neg A \supset B)$	**Mother said**
2.	$\mathbf{K}_a\neg\mathbf{K}_b B \supset \mathbf{K}_a\neg\mathbf{K}_b\neg A$	Exercise above
3.	$\mathbf{K}_a\neg\mathbf{K}_b\neg A$	**After first query** and MP
4.	$\mathbf{K}_a(\neg\mathbf{K}_b\neg A \supset \mathbf{K}_b A)$	**Init**
5.	$\mathbf{K}_a\mathbf{K}_b A$	3,4, axiom A1 and MP
6.	$\mathbf{K}_a A$	Exercise above

Note that we have only used $E^2_{\{a,b\}}$ knowledge and not common knowledge. This can be generalized and we can show that if there were k children, we would need E^k knowledge and that k levels suffice.

Exercise 5.8 *Give an inductive argument, showing that the analysis works when there are* n *playing children,* k *of whom get dirty,* $k > 0$*. In this case, show that after* $k-1$ *queries for which no answer is given, all* k *children answer to the* k^{th} *query.*

Show that, no matter which state the mother begins by confirming or denying, at least one child eventually realizes her state precisely.

Show that the order in which children answer does not matter, they can answer simultaneously or in groups.

In the analysis above, we see an interesting aspect of knowledge revision: at any stage, if a reasoner i does not know α, because of negative introspection, we have $\neg\mathbf{K}_i\alpha \wedge \mathbf{K}_i\neg\mathbf{K}_i\alpha$. Now, when she gets to know α, the earlier negative knowledge gets lost. Thus, knowledge revision is inherently non-monotonic. [Pa91] studies this aspect in detail.

The puzzle also illustrates how learning occurs in dialogues. Formalizing such learning is interesting, but tricky. See [Pa92] for a study of this kind.

While the example of this puzzle does not offer any justification that the notion of knowledge presented here is 'right' in any sense, it does already portray the interdependence of knowledge, communication and action, and the possibility of studying this interaction formally. That the need for such a study exists is obvious to anyone familiar with concepts in distributed computing or artificial intelligence or trading in economics.

6 Logical omniscience

The rumblings of discontent with the logic of knowledge studied in the last section begin with the observation that according to this logic, a reasoner knows every logical consequence of any fact she knows; further she knows all the valid formulas. Human reasoners simply do not exhibit such **logical omniscience**, and hence the logic above does not reflect reasoning about human knowledge. On the other hand, if we consider the reasoner above to be a computing agent, we come up with the problem of **resource boundedness**. The agent has only limited computational resources at its disposal, and exploring all the logical consequences of an assumption is expensive business. In particular, the validity problem for the propositional calculus is already co-NP-complete, and for the reasoner to know all tautologies requires, at the least, a co-NP machine. See [FHMV 95] for a detailed treatment of this issue.

The papers of Rohit Parikh ([Pa 87a], [Pa 87b] and [Pa 94]) provide all manners of criticism which include the two observations above and much more. For instance, the logic above ignores the difference between knowledge and **information**: "it is easier to believe that he (the former U.S. President) might have had the requisite information about the Iran-Contra affair and simply failed to make the necessary deduction which could lead to *knowledge*." The logic also ignores the difference between knowledge of a proposition and knowledge of the *sentence* denoting the proposition: it is quite conceivable that an agent knows 'p' and does not know 'q' even when p and q are logically equivalent propositions. See [Pa 87a] for a series of illuminating examples describing different notions of knowledge and the need to make subtle distinctions between these notions.

From the discussion in the literature (see bibliography of Chapters 9 and 10 in [FHMV 95]), we see a central thesis emerging and subsequently, two distinct problems that need to be studied. The thesis can be stated as follows:

- The notion of knowledge defined by the logic above is an **external** or **implicit** one. It is knowledge ascribed to the reasoner, rather than one possessed by the reasoner, human or computer. The latter kind of knowledge, which we may call **explicit knowledge**, can be seen as the ability to answer questions on whatever is claimed to be known.

Attempts at formalizing and studying explicit knowledge demands solution of two distinct problems:

- a *logical* problem, namely that of modal logics in which the modalities are not consequence-closed.
- a *computational* problem, as explicit knowledge depends on the algorithms used by the reasoner and the resources available to the algorithms.

Not surprisingly, much of the research on the former has been on logics with nonstandard semantics of some kind avoiding logical omniscience (Chapter 9 of [FHMV 95]) and discussion of the latter involves the invocation of an algorithm (not part of the logic) to evaluate knowledge formulas (both in [Pa 87a] and [HMV 94]).

There is *yet another* way in which explicit knowledge differs from implicit knowledge, one which is especially important in the context of distributed computing. A reasoner operating in an environment has only a limited view of the world, and her explicit knowledge is determined by the **visibility** of world states to her, whereas ascribed implicit knowledge of the agent depends on her behaviour in all possible worlds. In terms of the indistinguishability relation $\sim$ above, computing $\sim$ in itself may necessitate some effort. We can conceive of a situation where $s_0 \sim s_1 \sim s_2$, but at state s_0 only s_0 and s_1 are visible to the agent and not s_2. That is, an observer who has access to complete information about all the world states may declare that the agent would behave in the same way in all these three states, but the agent may require computation to realise this. If the agent makes the effort and computes the world state s_2, it may also realise that s_2 is in the equivalence class. However a resource limited agent might need to reuse its resources, and thus in computing the state s_2, it might 'forget' the information about s_0, for instance.

Thus, even among states within the same externally ascribed indistinguishability class, some may be 'farther' and some 'nearer' in terms of visibility to an agent. In the context of distributed systems, this happens routinely. A component of a system behaves in the same manner in all *global* system states in which its *local* state is the same. Hence all these states would be ascribed the same equivalence class for that agent. However, this component would typically communicate only with immediate neighbours in the network, and may not even be aware of the existence of many other components in the system. In this case, for the agent to compute its knowledge, the first task of computing these indistinguishable system states is well beyond its capabilities.

Moreover, as the system changes state, the view of the agent also changes, and this is also dependent on the computational resources available to the agent. For instance, a computational agent with bounded memory would forget events in the distant past. Even in static situations, computation may result in learning thereby enlarging visibility. [1] *Logically indistinguishable states may well be computationally distinguishable.* Therefore, even granting logical omniscience to a view-limited reasoner, ascribed and computed knowledge can be quite distinct. This is one aspect of the distinction between implicit and explicit knowledge which seems to have been relatively less studied in the literature.

In [R96c], we concentrate on this distinction, and to make matters simple, simply ignore the logical omniscience problem ! Our formulation of visibility-limited knowledge is inspired by the algorithm-based notion of knowledge studied in [FHMV 95] (Chapter 10) and [Pa 87a]. The latter works as follows: at any state s, when asked whether a formula ϕ holds, the reasoner evokes an algorithm available at s. If the algorithm returns 'Yes', we say that the reasoner explicitly (or algorithmically) knows ϕ. Depending on whether the algorithm gives only 'Yes/No' outputs or whether it also has the possibility of a '?' output meaning 'I cannot find out within the resources available to me', we get the notions studied in [FHMV 95] or [Pa 87a], respectively. [FHMV 95] restricts the algorithm to be local, in the sense that when $s \sim s'$, the algorithm invoked by the reasoner at s is the same as at s'.

This is a very interesting formulation of explicit knowledge and avoids many of the philosophical pitfalls which cause the criticism above. In particular, we can see the invoked algorithm as a *model checking algorithm*. This means that the reasoner explicitly knows all valid formulas (as they definitely hold at the states being checked in any model), but also knows many non-valid formulas which happen to hold at that state and cannot tell which ones among the known formulas are valid. (See [HV 91] for arguments advocating model checking as the 'right' way to think about this problem.) The framework facilitates the study of knowledge based on probabilistic algorithms ([KNP 90]) and action based on knowledge ([HF 89], [R96a]).

Unfortunately, the framework is also very general and unless we place restrictions on the class of algorithms available, the notion does not have interesting properties. For instance, if the system moves from a state s to state s', how should the knowledge algorithms available at the two states be related ? Further, the invocation of an algorithm is an extra-logical notion and as it is, we have no way of studying algorithmic knowledge *logically*.

In [R96c], we study a framework in which a form of algorithmic knowledge is studied logically,

[1] The example about learning the prime factors of 143 in [Pa 87b] is relevant here.

and which takes into account the limited visibility of agents. The idea is simple: an agent claims that ϕ holds at a state if ϕ holds in the substructure visible to the agent at that state. This corresponds to running the model checking algorithm on the substructure and the given formula. We can then say that the agent claims to know ϕ if ϕ holds in the substructure at all the visible indistinguishable states. Thus, resource bounds (which limit visibility) as well as formula structure determine the ease or difficulty of explicit knowledge of formulas.

Formally, frames are enriched with a *view function* $\nu : W \rightarrow 2^W$ (such that $w \in \nu(w)$). Explicit knowledge of α at a world w in a model M depends on α holding at w in the substructure $M\lceil\nu(w)$. Unfortunately, when $w \in \nu(w_1)$ as well as $w \in \nu(w_2)$ the truth of the same formula at w is quite distinct in the two substructures $M\lceil\nu(w_1)$ and $M\lceil\nu(w_2)$ and this makes for very high complexity. The only upper bound known for the satisfiability problem is double-exponential time, and for model checking it is singly exponential. However, if we impose a monotonicity condition on frames, which ensures that an agent's visibility about a state w is maximum at w, then the logic is easier: satisfiability is then NP-complete and model checking is only linear. See [R96c] for details.

7 Knowledge and time

Consider a distributed system of n *agents* acting autonomously. Agents communicate by sending messages to each other. Assume that every message sent is eventually delivered to the intended recipient and that messages are delivered in the order in which they were sent. Such a model is standard in the theory of distributed computing.

We can think of an agent's actions as partitioned into two sets: the *internal* actions which are part of the agent's computations, and the *communications* which enable the agent to exchange information with other agents in the system. It can be argued that the very purpose of communication is to update the state of knowledge of the agent: an agent a sends a message to b when a knows some fact and is aware that b does not know it, and when b receives the message, b gets to know the fact and that a knows the fact. Such reasoning is typical in distributed protocols.

7.1 Incorporating time

We can think of a distributed protocol as a goal-oriented activity, where we are given an initial state of knowledge of all agents in the system, and the enaction of the protocol leads to a desired state of knowledge of the agents ([PK92]). For instance, a distributed transaction commitment protocol begins with one (or some) of the agents wanting to perform a transaction and the others not knowing this, and ends in a state where it is common knowledge that all agents have committed to the transaction. (Presumably they perform the transaction afterwards.) A distributed termination protocol, if started in a state when all processes have finished their useful work but don't know it, should lead to a state when all processes know this and terminate.

A particularly interesting protocol question, much studied in the literature, involves *byzantine agreement* among processes: a group of processes, some of which may be faulty, are required to agree on the value of a bit. Whatever be the initial value that each process has for the bit, at the end, all the non-faulty ones must agree on the same value. For nontriviality (to rule out the protocol where they always decide, say, 0), if they initially had the same value, that must be the value agreed on in the end.

Analysis of protocols of this kind seems to involve the kind of knowledge-based reasoning we saw in the last section, and [FHMV95] is an excellent reference for the fair amount of work done in this area so far. In particular, knowledge-based analysis yielded a new optimal algorithm for byzantine agreement [DM90], and this is also discussed in detail in the book. Here, we only present the formal model in which such analysis is carried out.

We have defined models to be pairs $M = (F, V)$ where the frame is given by the tuple $F = (W,$

$\sim_1, \ldots, \sim_n$), and $V : W \to 2^P$. In the context of distributed systems, we are interested in system states which evolve as the computation proceeds, and hence W is a set of system runs, which are sequences of global states of the system. Formally, let L_e be the set of possible states of the environment and let L_i be the set of possible local states of agent i, $1 \leq i \leq n$. We take $G = L_e \times L_1 \times \ldots L_n$ to be the set of global states. A run over G is a map $r : \omega \to G$, which associates a global state with each discrete time instant. If $r(m) = (s_e, s_1, \ldots, s_n)$ let $r_i(m) \stackrel{\text{def}}{=} s_i$. We refer to the pair (r,m) as a *point* in the run. We are to think of a system action as something that happens between $(r, m-1)$ and (r,m), $m > 0$. It is easy now to define the indistinguishability relation for an agent i: define $(r,m) \sim_i (r',m')$ iff $r_i(m) = r'_i(m')$, that is, the agent has the same local state at both points and cannot distinguish which run it is in and at what global time instant.

There is a departure from the standard fashion when it comes to valuations. Since atomic propositions refer to properties of system states rather than time instants, a valuation is defined to be a map $V : G \to 2^P$. To sum up, we have the following definition.

Definition 7.1 *A* **distributed frame** *is a tuple $Sys = (L_e, L_1, \ldots, L_n, R)$, where $r \in R : \omega \to G$, $G = L_e \times L_1 \times \ldots \times L_n$. A model is a pair $M = (Sys, V)$, where $V : G \to 2^P$. A frame induces $\sim_i$, for $i \in \{1, \ldots, n\}$ on $R \times \omega$ by: $(r,m) \sim_i (r',m')$ iff $r_i(m) = r'_i(m')$.*

The suggestion that knowledge could be used to analyse protocols in this way came from [HM84], and concrete models were proposed by [PR85] and [CM86]. Since then many systems have been analysed using such a model. See [FHMV95] for a number of examples with detailed analysis. But even in the simplest of the examples, it becomes clear that the logic is too inexpressive to be useful. A major inadequacy of LK_n is its inability to refer to **temporal** properties. We would like to assert, in a system where messages are guaranteed to be delivered, that a sender of a message knows that the message will *eventually* be received, and this is a temporal modality here. (Actually, in the analysis of the muddy children puzzle, we glossed a bit. The formula 'After the first query' does not hold initially, but becomes true only after silence meets the first question. It can be checked that incorporating time does not change the analysis, but it is a detail to be noted.)

Formally, we enrich the language in the following way:

$LKT_n ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid \mathbf{K}_i\ \alpha, i \in \{1, \ldots, n\} \mid \bigcirc\ \alpha \mid \alpha\ \mathbf{U}\ \beta$.

The semantics is given for the modalities as follows:

- $M, (r,m) \models \mathbf{K}_i\alpha$ iff $\forall (r',m') \sim_i (r,m), M, (r',m') \models \alpha$.
- $M, (r,m) \models \bigcirc\alpha$ iff $M, (r, m+1) \models \alpha$.
- $M, (r,m) \models \alpha\mathbf{U}\beta$ iff $\exists k \geq m$ such that $M, (r,k) \models \beta$ and for all $l : m \leq l < k$, $M, (r,l) \models \alpha$.

This is standard linear time temporal logic added on to LK_n. Note that the temporal operators refer to future instants within the same run, whereas the knowledge operator refers to points across runs. We could also refer to instants only within the same run to get a more restricted knowledge modality: this issue is discussed in [R96a], and the resulting modality is similar to a *present tense* modality studied in [R96b].

The satisfiability problem for linear time temporal logic is PSPACE-complete [SC85]. The following theorem shows that adding knowledge operators does not add more complexity. On the other hand, as we saw before, for $LK_n + C_G$, satisfiability is EXPTIME-complete, and in such a situation, adding temporal operators cannot make things worse.

Theorem 7.2 *The satisfiability problem is PSPACE-complete for LKT_n and EXPTIME-complete for $LKT_N + C_G$.*

In terms of axiom systems, we only need to put together axioms and rules of any complete system for linar time alongwith the system for LK_n. In this sense, when considered over the class of all systems, there is little interaction between the knowledge and temporal modalities.

On the other hand, when we look at systems with *prefect recall*, the situation becomes dramatically worse. These are systems where an agent's action depends not only on its local state but also on the sequence of states that it has gone through till then. Such memory-based actions are common both in distributed protocols and trading agents in markets, the most common applications for these logics. Formally, we say that an agent i has perfect recall if whenever we have $(r, m) \sim_i (r', m')$ then the local state sequence for i seen in r upto m is the same as the sequence in r' upto m'. In such systems, the following implication is valid: $\mathbf{K}_i \bigcirc \alpha \supset \bigcirc \mathbf{K}_i \alpha$ Halpern and Vardi [HV89] show the following:

Theorem 7.3 *Over systems where agents have perfect recall, the satisfiability problem for LKT_n is nonelementary in complexity, and the problem becomes highly undecidable (Π^1_1-complete) for the logic $LKT_n + C_G$.*

7.2 Knowledge and action

In some sense, both the theorems above are unsatisfactory ! One of them says that there is little interaction between knowledge and time, when we *know* this is not the case in distributed systems, and the other says that perfect recall is computationally horrible, which is hardly a surprise. To take up the first matter, a little bit of reflection shows that the modelling suggested above indeed leaves out the effect of knowledge on action, as well as the effect of actions in acquiring knowledge, and then it is hardly surprising that the modalities do not interact. Hence, it seems reasonable to explicitly study how knowledge changes due to actions without assumptions of perfect recall, and see how the modalities interact.

For this, we first consider a simple **dynamic logic of knowledge** before looking at temporal operators, as dynamic logic is a natural vehicle for studying how actions change truth values. $DK ::= p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid \langle a \rangle \alpha \mid \diamond\ \alpha \mid \mathbf{K}_i \alpha$.

What are the frames for the logic ? Dynamic logic is usually studied over state transition systems, except that the states we want to study here are *states of knowledge* of agents. What difference does that make ? It means that distinct states must be distinguishable by some agent in the system. Considering states as information about knowledge of agents has further implications: when an agent performs a local action, this should not affect the state of knowledge of any other agent who is totally uninvolved in that action. Moreover, the enabling and effect of an action should be completely determined by the knowledge of the acting agent. These are assumptions often implicitly made in the context of distributed protocols and games in economics. (Incidentally, the frames defined below are basically the same as *asynchronous automata* studied in concurrency theory [Zie87], and we have here a knowledge-theoretic account of these automata.)

A *distributed action alphabet* is a pair (Σ, θ), where Σ is a finite nonempty set of actions, and $\theta : \Sigma \to (2^{\{1,\ldots,n\}} - \emptyset)$. For $a \in \Sigma$, we talk of $\theta(a)$ as the *locations* of a. We can think of agents in $\theta(a)$ jointly performing the action a.

Definition 7.4 *A* **synchronizing knowledge transition system (SKTS)** *over the distributed alphabet (Σ, θ) is a tuple $\mathcal{K} = (S, \to, \sim_1, ..., \sim_n)$, where*

1. *S is a set of* states*,*
2. *$\to \subseteq S \times \Sigma \times S$ is the* transition relation*,*
3. *$\sim_i \subseteq S \times S, i \in \{1, \ldots, n\}$ are equivalence relations satisfying for all s, s' in S:*
   (a) *if $(\forall i, s \sim_i s')$ then $s = s'$.*

(b) *if* $s \xrightarrow{a} s'$ *and* $i \notin \theta(a)$ *then* $s \sim_i s'$.

(c) *if* $s_1 \xrightarrow{a} s_2$, *and* $\exists s'_1 \in S$ *such that for every* $i \in \theta(a), s_1 \sim_i s'_1$, *then there exists* s'_2 *such that* $s'_1 \xrightarrow{a} s'_2$, *and* $\forall i \in \theta(a), s_2 \sim_i s'_2$.

S can be thought of as containing the global states of the system, and $\sim_i$ gives the indistinguishability relation for agent i. Since $[s]_i$, the equivalence class of s under $\sim_i$, specifies all states in which agent i has the same view as in s, we can think of $[s]_i$ as a *local* state of agent i. Note that every transition carries the labels of those agents in the system which participate in that transition: therefore, no external changes are described. In this sense, S represents the states of knowledge of agents rather than the "states of the world".

A word about the communication mechanism here. SKTSs can display a good deal of asynchrony : while agents 1 and 2 are participating in action a, say, agents 3 and 4 could be performing b. Subgroups of agents may be proceeding at different speeds. The reference to synchrony here is only to emphasize that when agents do synchronize, it is by way of handshaking rather than message passing (deals made on telephone rather than via letters). Condition (a) above is reasonable only for such systems. In a message based system, a global state needs to have more information than merely given by a tuple of local states ; for instance, the status of buffers containing undelivered messages needs to be recorded. This can be bypassed by modelling buffers as additional agents, but condition (c) breaks down more crucially. When $s \xrightarrow{a} s'$ and a constitutes the sending of a message by agent 1 to 2, $s \sim_2 s'$ but receipt of the message by 2 is enabled at s' but not in s. Indeed this is why knowledge transition systems are defined in [KR 94] using condition (b) above and the (weaker) first condition below.

Proposition 7.5 *In every SKTS* $\mathcal{K} = (S, \rightarrow, \sim_1, ..., \sim_n)$ *, the following properties hold:*

- *if* $s \xrightarrow{a} s_1, s \xrightarrow{b} s_2$ *and* $\theta(a) \cap \theta(b) = \emptyset$, *then there exists* s_3 *such that* $s_1 \xrightarrow{b} s_3$ *and* $s_2 \xrightarrow{a} s_3$.
- *if* $s \xrightarrow{a} s_1 \xrightarrow{b} s_2$ *and* $\theta(a) \cap \theta(b) = \emptyset$, *then there exists* s_3 *such that* $s \xrightarrow{b} s_3 \xrightarrow{a} s_2$.

Unfortunately the logic DK is too expressive : formulas can force models to contain "grids", tilings of the plane. This leads to undecidability. The following theorem [R94] is proved by recursive reductions of instances of colouring problems (variants of tiling problems) to (finite) satisfiability of $\mathcal{L}$ formulas.

Theorem 7.6 *The satisfiability problem for* DK *is highly undecidable (*Σ_1^1*-complete).*

This means that the set of valid formulas is not recursively enumerable, and hence we cannot hope for a finite axiomatization. Thus we have no option but to change the logic.

An analysis of the proof of undecidability shows that the problem is mainly due to the use of *global* propositions and modalities. For a simplified explanation, consider a system with four states s_1, s_2, s_3, s_4 and transitions $s_1 \xrightarrow{a} s_2 \xrightarrow{b} s_4, s_1 \xrightarrow{b} s_3 \xrightarrow{a} s_4$, $\Sigma = \{a, b\}$, $\theta(a) = \{1\}, \theta(b) = \{2\}$. The partitions for 1 and 2 are respectively $(\{s_1, s_3\}, \{s_2, s_4\})$, and $(\{s_1, s_2\}, \{s_3, s_4\})$. Consider the valuation V which makes p true at s_2 and s_4 and q true at s_3 and s_4. At s_1, the formula $(\neg p \wedge \neg q) \wedge \langle a \rangle (p \wedge \neg q) \wedge \langle b \rangle (\neg p \wedge q)$ holds, and this is a definite assertion which cannot be made by any agent in the system. Agent 1 can see the change from $\neg p$ to p but would be uncertain about the status of q, and similarly for agent 2. They could pool in their information together, but that would require communication between them which is not possible in this example system. Thus we have a God-like ability to make global assertions, which are not obtained by composing individual knowledge assertions. It is this ability that creates trouble.

In [R96a], we therefore propose a logic where the modalities are *local*, and we compose such assertions globally mainly using boolean connectives. The logical language defined below called

$KPDL$ has two levels of syntax, consisting of *local formulas* and *global formulas*. Fix P_i the set of local atomic propositions for each agent i. We use α, β, γ etc. (with or without subscripts) to denote local formulas. The syntax of **i-local formulas** is given by:

$$\Phi_i \ ::= \ p \in P_i \mid \neg\, \alpha \mid \alpha \ \vee \ \beta \mid \langle a \rangle \ \alpha \mid \Diamond \alpha \mid \mathbf{K}\phi$$

where, ϕ is a global formula, defined below.

$$KPDL ::= \alpha @ i, \ \alpha \in \Phi_i, \ i \in \{1, \ldots, n\} \mid \neg\, \phi \mid \phi_1 \ \vee \ \phi_2 \mid \langle a \rangle \phi, \ Voc(\phi) \subseteq \theta(a)$$

$Voc(\phi)$, the *agent vocabulary* of ϕ is inductively defined in the obvious manner :

$$Voc(\alpha @ i) \stackrel{\text{def}}{=} \{i\}; \ Voc(\neg \phi) \stackrel{\text{def}}{=} Voc(\phi);$$

$$Voc(\phi_1 \vee \phi_2) \stackrel{\text{def}}{=} Voc(\phi_1) \cup Voc(\phi_2);$$

$$Voc(\langle a \rangle \phi) \stackrel{\text{def}}{=} \theta(a).$$

At the global level, we have but a restricted modality. This is because, a is a synchronization between agents who participate in a and hence they can exchange their views of the system state. The vocabulary restriction ensures that this is all they get to see. Additional modalities at the global level lead to excessive power, so we stop at this.

A *model* is a tuple $M = (\mathcal{K}, V_1, \ldots, V_n)$, where $\mathcal{K} = (S, \rightarrow, \sim_1, ..., \sim_n)$, and $V_i : S \rightarrow \wp(AP)$ is the i^{th} valuation function such that $s \sim_i s'$ implies $V_i(s) = V_i(s')$. Thus, atomic propositions are evaluated at local states.

The formula ϕ being satisfied in a model M at a state is defined below. We first define the notion for i-local formulas. Let $M_i \stackrel{\text{def}}{=} (K_i, V_i)$, where $K_i \stackrel{\text{def}}{=} \{[s]_i \mid s \in S\}$.

- $M_i, [s]_i \models_i p$ iff $p \in V_i(s)$.
- $M_i, [s]_i \models_i \neg \alpha$ iff $M_i, [s]_i \not\models_i \alpha$.
- $M_i, [s]_i \models_i \alpha \vee \beta$ iff $M_i, [s]_i \models_i \alpha$ or $M_i, [s]_i \models_i \beta$.
- $M_i, [s]_i \models_i \langle a \rangle \alpha$ iff $\exists s_1 \in [s]_i, s_2 \in S$ such that $s_1 \xrightarrow{a} s_2$ and $M_i, [s_2]_i \models_i \alpha$.
- $M_i, [s]_i \models_i \Diamond \alpha$ iff $\exists s_1 \in [s]_i, s_2 \in S$ such that $s_2 \in R(s_1)$ and $M_i, [s_2]_i \models_i \alpha$.
- $M_i, [s]_i \models_i \mathbf{K}\phi$ iff for every $s' \in S$, if $s \sim_i s'$, then $M, s' \models \phi$.

In the last clause above, $\models$ refers to the global satisfaction relation defined below.

- $M, s \models \alpha @ i$ iff $M_i, [s]_i \models_i \alpha$.
- $M, s \models \neg \phi$ iff $M, s \not\models \phi$.
- $M, s \models \phi_1 \vee \phi_2$ iff $M, s \models \phi_1$ or $M, s \models \phi_2$.
- $M, s \models \langle a \rangle \phi$ iff there exists $s' \in S$ such that $s \xrightarrow{a} s'$ and $M, s' \models \phi$.

[R96a] then proves the following theorem and also provides a complete axiomatization of the valid formulas.

Theorem 7.7 *Satisfiability of a formula ϕ in $KPDL$ can be decided in (nondeterministic) time $2^{O(m)}$, where m is the length of ϕ.*

7.3 Relating knowledge and time

We can unfold the runs of knowledge transition systems and consider temporal modalities on the runs. That is one way of studying interaction between knowledge and time, and the programme is partially carried out in [R96a] (partial, as we only have decidability results there, axiom systems being yet elusive). Another way is to investigate the relationship between passage of time and changes in knowledge of agents, in a categorical sense. [KR94] presents such a study, and we briefly describe the approach here.

Lamport [Lam78] discussed the ordering of event occurrences in such systems and argued that each agent 'locally' sees a linear order of event occurrences whereas 'globally', only a partial order consistent with the local linear ones is available. In this discussion, the ordering refers to causal dependency between event occurrences and thus incomparability under the ordering denotes causal independence, and therefore (in a sense) concurrency.

If we see concurrency as causal independence, one way of phrasing the assertion "event occurrences e_1 and e_2 can be concurrent" is: "no agent in the system *knows* that e_1 must precede e_2 or that e_2 must precede e_1". In a sense, this identifies the states of the system with the states of knowledge of agents in the system.

We can ask: to what extent these notions of agents' knowledge (specified as equivalence relations on states, one relation for each agent) and partial orders on event occurrences are *dual*. Can we go back and forth between these structures without losing information about agents' behavior? This question is meaningful, because, given a temporal structure describing how events occur, we can compute how the knowledge of agents changes. Therefore, given a transition system which describes knowledge changes, can we recover the temporal picture ?

[KR94] answers this question positively, and in the process, we also understand more precisely assertions like the following ones: "an agent cannot lose knowledge by receiving a message", "an agent cannot gain knowledge by sending a message" and so on. Such statements are commonly used in the analysis of distributed protocols, and do make intuitive sense.

[KR94] studies a subclass of KTSs; as opposed to the synchronizing KTSs defined above, there we consider the asynchronous subclass, where agents may communicate only by messages. These systems are shown to correspond to a natural partial order model of event occurrences in distributed systems (the model of Asynchronously Communicating Sequential Agents, abbreviated ACSAs, introduced by [LRT91]). This correspondence is precise in the following sense: we associate a KTS with an ACSA and conversely an ACSA with a KTS in such a way that ACSA $\rightarrow$ KTS $\rightarrow$ ACSA is an isomorphism, and KTS $\rightarrow$ ACSA $\rightarrow$ KTS is a simulation. This can be set up in a categorical framework, showing that there is a co-reflection between the category of ACSAs and that of KTSs, or rather, we can embed the category of ACSAs in the category of KTSs. Thus, we have a duality of sorts between knowledge and temporal structures.

References

[AB75] Anderson, A. and Belnap, M.D., *Entailment: the logic of relevance and necessity*, Princeton Univ. Press, 1975.

[Ben84] Benthem, J.F.A.K. van, "Correspondence theory", in [GG84], Vol 2, 167-247.

[BS84] Bull, R.A., and Segerburg, K., "Basic modal logic", in [GG84], Vol 2, 1-88.

[CM86] Chandy, K.M., and Misra, J., "How processes learn", *Distributed Computing*, vol 1(1), 40-52.

[Che80] Chellas, B.F., *Modal logic*, Cambridge Univ Press, Cambridge, 1980.

[DM90] Dwork, C., and Moses, Y., "Knowledge and common knowledge in a byzantine environment: crash failures", *Information and Computation*, vol 88(2), 156-186.

[FHMV95] Fagin, R., Halpern, J., Moses, Y. and Vardi, M., *Reasoning about knowledge*, M.I.T. Press, 1995.

[FL79] Fischer, M.J., and Ladner, R.E., "Propositional dynamic logic of regular programs", *Journal of Computer and System Sciences*, vol 18(2), 194-211.

[Fi83] Fitting, M., *Proof methods for modal and intuitionistic logics*, D.Reidel, Dordrecht, 1983.

[Fi93] Fitting, M., "Basic modal logic", in Gabbay, D.M., Hogger, C. and Robinson, J.A., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford Science Pub, Oxford, 1993, 368-448.

[GG84] Gabbay, D. and Guenthner, F., (Eds), *Handbook of Philosophical Logic*, Reidel, Dordrecht, 1984.

[Gol92] Goldblatt, R., *Logics of time and computation*, CSLI Lecture Notes No. 7, Stanford University, 1992.

[HF89] Halpern, J., and Fagin, R., "Modelling knowledge and action in distributed systems", *Distributed Computing*, vol 3, #4, 1989, 159-177.

[HM84] Halpern, J., and Moses, Y., "Knowledge and Common Knowledge in a Distributed Environment", *PODC 84*, journal version in *JACM*, vol. 37, pp. 549-578.

[HMV94] Halpern, J., Moses, Y., and Vardi, M., "Algorithmic knowledge", *TARK V*, Theoretical Aspects of Reasoning about Knowledge, 1994, 255-266.

[HV89] Halpern, J. Y., and Vardi, M., "The complexity of reasoning about knowledge and time, I. Lower Bounds", *JCSS*, **38** (1989) pp. 195-237.

[HV91] Halpern, J., and Vardi, M., "Model checking vs theorem proving: a manifesto", in *Artificial Intelligence and Mathematical Theory of Computation*, (ed) V. Lifschitz, Acad Press, 1991, 151-176.

[Har84] Harel, D., "Dynamic logic", in [GG84], Vol 2, 497-604.

[Hin62] Hintikka, J., *Knowledge and belief*, Cornell Univ Press, Ithaca, NJ, 1962.

[HU79] Hopcroft, J.E. and Ullman, J.D., *Introduction to automata theory, languages and computation*, Addison-Wesley, New York, 1979.

[HC68] Hughes, G.E., and Cresswell, M.J., *An introduction to modal logic*, methuen, London, 1968.

[HC84] Hughes, G.E., and Cresswell, M.J., *A companion to modal logic*, methuen, London, 1984.

[Ko83] Kozen, D., "Results on the propositional μ-calculus", *Theoretical Computer Science*, vol 27(1), 333-354.

[KNP90] Krasucki, P., Ndjatou, G. and Parikh, R., "Probabilistic knowledge and probabilistic common knowledge", in *ISMIS 90*, International Symp. on Methodology for Intelligent Systems, 1990, 1-8.

[KP81] Kozen, D., and Parikh, R., "An elementary proof of the completeness of PDL", *Theoretical Computer Science*, vol 14(1), 113-118.

[KR94] Krasucki, P., and Ramanujam, R., "Knowledge and the ordering of events in distributed systems", *TARK V*, Theoretical Aspects of Reasoning about Knowledge, 1994, 267-283.

[Kr59] Kripke, S., "A completeness theorem in modal logic", *Journal of Symbolic Logic*, vol 24, 1-14.

[Lad77] Ladner, R.E., "The computational complexity of provability in systems of modal propositional logic", *SIAM Journal on Computing*, vol 6(3), 467-480.

[Lam78] Lamport, L., "Time, clocks and the ordering of events in a distributed system", *Communications of the ACM*, vol 21(7), 558-565.

[MP92] Manna, Z., and Pnueli, A., *The temporal logic of reactive and concurrent systems*, Springer-Verlag, berlin, 1992.

[LRT91] Lodaya, K., and Ramanujam, R., and Thiagarajan, "Temporal Logics for Communicating Sequential Agents: I", Intl Jnl on Found. of Comp Sci, vol 3, #2, 1992, 117-159.

[Pa87a] Parikh, R., "Knowledge and the problem of logical omniscience", *ISMIS 87*, International Symp. on Methodology for Intelligent Systems, 1987, 432-439.

[Pa87b] Parikh, R., "Some recent applications of knowledge" in *FST and TCS 7*, Foundations of Software Technology and Theoretical Computer Science, 1987, LNCS #287, 528-539.

[Pa91] Parikh, R., "Monotonic and nonmonotonic logics of knowledge", *Fundamenta Informaticae*, vol 15(4), 255-274.

[Pa92] Parikh, R., "Finite and infinite dialogues", in Y.N.Moschovakis, (ed.), *Logic from computer science*, MSRI Pub. No.21, 481-497, Springer-Verlag, Berlin.

[Pa94] Parikh, R., "Logical omniscience" in *Logic and Computational Complexity*, LNCS 960, 22-29.

[PK92] Parikh, R. and Krasucki, P., "Levels of knowledge in distributed computing", *Sadhana* vol 17(1), 167-191.

[PR85] Parikh, R. and Ramanujam, R., "Distributed processing and the logic of knowledge", Springer *Lecture Notes in Computer Science*, vol 193, 256-266.

[Pra79] Pratt, V., "Models of program logics", Proc. 20th IEEE FOCS, 115-122, 1979.

[R94] Ramanujam, R., " Knowledge and the next state modality", *Indian National Seminar in TCS*, 1994, 62-80.

[R96a] Ramanujam, R., "Local knowledge assertions in a changing world", *Proc TARK VI*, Theoretical Aspects of Rationality and Knowledge, 1996, 1-17.

[R96b] Ramanujam, R., "Locally linear time temporal logic", *Proc LICS*, 1996, 118-127.

[R96c] Ramanujam, R., "View-based explicit knowledge", forthcoming *Rohit Parikh Fetschrift* (presented at FST&TCS 1996).

[Ras74] Rasiowa, H., *An algebraic approach to non-classical logics*, North-Holland, Amsterdam, 1974.

[Seg68] Segerberg, K., *Results in nonclassical logic*, Berlingska Boktryckeriet, Lund, Sweden, 1968.

[SC85] Sistla, A.P., and Clarke, E.M., "The complexity of propositional linear temporal logics", *Journal of the ACM*, vol 32(3), 733-749.

[Zie87] Zielonka, W., "Notes on finite asynchronous automata", RAIRO-Inf. Theor. et Appli., vol 21, 1987, 99-135.