DTIC FILE COPY

(2)

WORKING MATERIAL

FOR THE LECTURES OF

A. NERODE

SOME LECTURES ON MODAL LOGIC

SUPPLEMENT

AD-A213 957

INTERNATIONAL SUMMER SCHOOL

ON

LOGIC, ALGEBRA AND COMPUTATION

MARKTOBERDORF, GERMANY, JULY 25 - AUGUST 6, 1989

THIS SUMMER SCHOOL IS ORGANIZED UNDER THE AUSPICES OF THE TECHNISCHE UNIVERSITÄT MÜNCHEN AND IS SPONSORED BY THE NATO SCIENCE COMMITEE AS PART OF THE 1989 ADVANCED STUDY INSTITUTES PROGRAMME. PARTIAL SUPPORT FOR THE CONFERENCE WAS PROVIDED BY THE EUROPEAN RESEARCH OFFICE AND THE NATIONAL SCIENCE FOUNDATION AND BY VARIOUS INDUSTRIAL COMPANIES.

SOME LECTURES ON MODAL LOGIC
DRAFT

Anil Nerode* **
Mathematical Sciences Institute
Cornell University
Ithaca, New York 14853

§1. **Preface.** Modal logics arise throughout computer science. It is worthwhile to find the best means of exposition of theory and applications for mathematics and computer science students. The classical modal logic texts are neither oriented toward computer science nor mathematics. The computational content of proof procedures and how the notions apply in computer science and AI applications has to be brought out. Here we start an exposition without proofs of propositional modal logic using a tableaux method easy to remember for hand computation and suitable for automated reasoning. The exposition is analogous to the exposition in the author's lectures on intuitionistic logic (Nerode [1990]), also directed at computer science applications. Here one application exposited at length, also without proofs, is the autoepistemic logic of Moore. The outline of §9 was supplied by W. Marek. But any defects of exposition are solely due to present author. We outline classical constant domain modal predicate logic briefly. We conclude with dynamic logic. We give a brief introduction to a new intuitionistic dynamic logic due to D. Wijesekera, which is suitable for dealing with concurrency. (KR)

§2. **Propositional modal logic.** Propositional modal logic is based on connectives which construct new propositions from old. We treat propositional logic first. The modal propositional connectives are

"and"	$\wedge$
"or"	$\vee$
"implies"	$\rightarrow$
"not"	$\neg$
"box"	$\Box$
"diamond"	$\Diamond$

The primitive symbols will be:

An infinite list of propositional constants.
The list of logical connectives $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\Box$, $\Diamond$.
parentheses (,) and the comma.

The inductive definition of (modal) proposition is:

1) Propositional constants are propositions,
2) If α, β are propositions, then $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$, $(\neg \alpha)$ are propositions.
3) If α is a proposition, then $(\Box\alpha)$ is a proposition.
4) If α is a proposition, then $(\Diamond\alpha)$ is a proposition.

Sometimes we omit parentheses, but just as often we put extra ones in for legibility in complex expressions. We are indiscriminate in using both upper case Roman letters <u>and</u> lower case Greek letters for propositions.
Propositions constructed by rules 1), 2) alone are called classical propositions and constitute the language L. Propositions constructed using rules 1), 2), 3) are called modal propositions and constitute the language $L_{\Box}$. Propositions constructed using 1), 2), 3), 4) are also called modal propositions and constitute the language $L_{\Box,\Diamond}$.

The classical propositions of L are intended as truth functional modes of statement composition, that is the truth or falsity of a compound statement is determined by the truth or falsity of the parts. This is the import of the truth tables of propositions. Conversely, any truth table is the truth table of a proposition built from $\wedge$, $\vee$, $\neg$. Classical propositional logic was defined to deal with exactly all truth functional connectives.
As for the modal connectives,

"$\Box$P" is read "box P", or sometimes "necessarily P",
"$\Diamond$P" is read "diamond P", or sometimes "possibly P".

We prefer the readings "box" and "diamond", simply because the interpretations of the connectives $\Box$ and $\Diamond$ symbols in applications are often quite different from those associated with "necessary" and "possible". For example, "I know that", "I believe that", "John knows that", "John believes that" are often axiomatized using box with appropriate axioms. A further reason for neutral terminology is that the question as to what are the properties of necessity and possibility has been debated since the golden age of Greece.

Remark. Modal propositional connectives, unlike the classical connectives, never entered into the foundations of classical mathematics. These foundations rest only on the classical "truth functional" propositional connectives. The new connectives of modal logic are not intended to be truth–functional. "It is necessary that P " should not depend solely for its truth or falsity on the truth or falsity of P, otherwise it is merely P or $\neg$P.

3. Frames. C. I. Lewis [1918] introduced modal logic as a deductive subject and gave a notion of theorem based on axioms and rules of inference. Kanger [1957] and Kripke [1959, 1963] gave a semantics based on the notions of frame and model.

First let us review truth valuations of classical logic L. An L–assignment is a mapping A with domain the set of propositional constants to $\{T, F\}$. Let $\underline{A}$ be the set of all propositional constants mapped into T by A. Each assignment A has a unique extension to a classical L–valuation v of L, such that

0) $v(P) = A(P)$ for all propositional constants P.
1) $v(A \wedge B) = T$ iff $v(A) = T$ and $v(B) = T$.
2) $v(A \vee B) = T$ iff $v(A) = T$ or $v(B) = T$.
3) $v(A \rightarrow B)$ T iff $v(A) \neq T$ or $v(B) = T$.
4) $v(\neg A) = T$ iff $v(A) \neq T$.

Or equivalently,

0) For propositional constants P, P is true iff $P \in \underline{A}$.
1) $A \wedge B$ is true iff A is true and B is true.
2) $A \vee B$ is true iff A is true or B is true.
3) $A \rightarrow B$ is true iff A is not true or B is true.
4) $\neg A$ is true iff A is not true.

Frame semantics. For modal propositional logic Kripke introduced the notion of a frame $(\mathcal{F}, R)$ consisting of a non–empty set $\mathcal{F}$ of "possible worlds" and a binary relation $R \subseteq \mathcal{F} \times \mathcal{F}$. Then wRx is read "x is accessible from w". A model M is a triple $(\mathcal{F}, R, v)$, with $(\mathcal{F}, R)$ a frame and $v(w)$ a "valuation function" with domain the set of "possible worlds" $\mathcal{F}$ and range contained in the set of L–valuations, which assigns to each w in $\mathcal{F}$ an L–valuation $v(w)$. So the notation for the truth value of the valuation $v(w)$ assigned to world w at proposition A is $v(w)(A)$. Here is a definition of "φ is true at w in M". Reference to M is omitted when understood.

0) An atomic proposition A is true at w iff $v(w)(A) = T$.
1) $A \wedge B$ is true at w iff A is true at w and B is true at w.
2) $A \vee B$ is true at w iff A is true at w or B is true at w.
3) $A \rightarrow B$ is true at w iff A is not true at w or B is true at w.
4) $\neg A$ is true at w iff A is not true at w.
5) $\Box A$ is true at w iff for every x accessible from w, A is true at x.
6) $\Diamond A$ is true at w iff for some x accessible from w, A is true at x.

Fix M. We indiscriminately write "A is true at w" as "$w \vDash A$, or as "w forces φ". At times this avoids incorrect connotations of classical truth, and is a notation borrowed from set theory.

In the forcing notation, the inductive definition of $\vDash$ is: for all w in $\mathcal{F}$,

0) For an atomic proposition A, $w \vDash A$ iff $v(w)(A) = T$
1) $w \vDash A \wedge B$ iff $w \vDash A$ and $w \vDash B$.
2) $w \vDash A \vee B$ iff $w \vDash A$ or $w \vDash B$.
3) $w \vDash A \rightarrow B$ iff not($w \vDash A$) or $w \vDash B$.
4) $w \vDash \neg A$ iff not ($w \vDash A$).
5) $w \vDash \Box A$ iff for all x in $\mathcal{F}$ such that $w R x$, $x \vDash A$.
6) $w \vDash \Diamond A$ iff for some x in $\mathcal{F}$ such that $w R x$, $x \vDash A$.

Remark. Box "$\Box$" and diamond "$\Diamond$" are written as "propositional" connectives. But in model $(\mathcal{F}, R, v)$ the "$\Box$" in $w \vDash \Box A$ is a universal quantifier over possible worlds accessible from A (if any), while the "$\Diamond$" in $w \vDash \Diamond A$ is an existential quantifier over worlds accessible from w. So to construct models for modal propositional calculus, the appropriate method comes from classical predicate logic, not from classical propositional logic.

We assume as given an infinite sequence of constants to name worlds, "world constants". In the tableaux below these will be used ambiguously as names for worlds w and names for classical valuations w at worlds. That is, in the models associated with tableaux branches, $\mathcal{F}$ will be a set of world constants, and the accessibility relation R will be a relation between world constants. This means that for the models constructed by tableaux, the valuation map v will be such that $v(w) = w$. There we do not distinguish between names of worlds and names of valuations in the tableaux.

The tableaux rules are chosen to reflect exactly the definition of forcing in a model.

4. Propositional tableaux. A tableaux is a finite labelled tree, with apex at the top. Each node is labelled by a "signed formula" $Tv \Vdash A$ or $Fv \Vdash A$, with A a formula, v a world constant. These are read respectively "at world v, A is true", or "at world v, A is false". In addition, at the base of some branches is an "×", and these are called closed branches, the rest open. Tableaux are developed (extended to larger tableaux) by the rules below.

Here is the dynamic idea behind constructing a tableaux proof of A. To verify that A is valid in all models, we suppose not, and search for a counterexample by developing a tableaux with apex $Fw \Vdash A$, with w a world constant not occurring in A. If we develop such a tableaux according to the tableaux rules, all possible ways to falsify $w \Vdash A$ are taken into account. If an immediate contradiction occurs on every branch at some point of the tableaux construction, all ways of falsifying A have been exhausted. A is valid in all models. The resulting tableaux with contradictions on every branch is a proof of A. A closed branch is one with a contradiction "×" at the base. Open branches are those that are not closed. We develop the tableaux by <u>using</u> entries on <u>open</u> branches. An entry is <u>used</u> by placing an appropriate *atomic tableaux*, *omitting* its apex, at the base of some (or every) open branch through that entry. A branch is declared <u>closed</u> as soon as for some branch and some proposition B and some world constant w, that branch has entries of the form $Tw \Vdash B$, $Fw \Vdash B$. We place a cross "×" at the base of each branch so closed. A tableaux proof is a tableaux with all branches closed.

The tableaux proof system is based on the atomic tableaux for classical and modal connectives below, $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\Box$, $\Diamond$. It is the equivalent of the system K traditionally studied in modal logic. See Fitting [1983] for closely related systems of prefixed tableaux, from which these tableaux stem. We also will extend this system with additional tableaux development rules to deal with validity in special classes of frames.

<u>Atomic tableaux. The classical connectives.</u>

and

$$\begin{array}{c} Tw \Vdash \varphi \wedge \psi \\ | \\ Tw \Vdash \varphi \\ | \\ Tw \Vdash \psi \end{array} \qquad\qquad \begin{array}{ccc} & Fw \Vdash \varphi \wedge \psi & \\ \swarrow & & \searrow \\ Fw \Vdash \varphi & & Fw \Vdash \psi \end{array}$$

or

$$\begin{array}{c} Tw\vdash\varphi \vee \psi \\ \diagup \quad \diagdown \\ Tw\vdash\varphi \qquad Tw\vdash\psi \end{array} \qquad\qquad \begin{array}{c} Fw\vdash\varphi \vee \psi \\ | \\ Fw\vdash\varphi \\ | \\ Fw\vdash\psi \end{array}$$

implies

$$\begin{array}{c} Tw\vdash\varphi \rightarrow \psi \\ \diagup \quad \diagdown \\ Fw\vdash\varphi \qquad Tw\vdash\psi \end{array} \qquad\qquad \begin{array}{c} Fw\vdash\varphi \rightarrow \psi \\ | \\ Tw\vdash\varphi \\ | \\ Fw\vdash\psi \end{array}$$

not

$$\begin{array}{c} Tw\vdash\neg\varphi \\ | \\ Fw\vdash\varphi \end{array} \qquad\qquad \begin{array}{c} Fw\vdash\neg\varphi \\ | \\ Tw\vdash\varphi \end{array}$$

<u>Example</u>. Here is a tableaux proof of $\neg B \wedge (A \vee B) \rightarrow A$.

$$\begin{array}{lccl} 1 & & Fw\vdash\neg B \wedge (A \vee B) \rightarrow A & \\ & & | & \\ 2 & & Tw\vdash\neg B \wedge (A \vee B) & \text{by } 1 \\ & & | & \\ 3 & & Fw\vdash A & \text{by } 1 \\ & & | & \\ 4 & & Tw\vdash(\neg B) & \text{by } 2 \\ & & | & \\ 5 & & Tw\vdash(A \vee B) & \text{by } 2 \\ & & \diagup \quad \diagdown & \\ 6 & & Tw\vdash A \quad Tw\vdash B & \text{by } 5 \\ & & | \qquad\quad | & \\ 7 & & \times \quad Fw\vdash B & \text{by } 3, 6 \qquad \text{by } 4 \\ & & | & \\ 8 & & \times & \text{by } 6, 7 \end{array}$$

The number annotations on the left and the reason annotations on the right are not part of the formal tableaux proof, but are useful for reading a finished proof. Since "w⊢" plays no role in tableaux for propositions in the classical propositional calculus L, it can be omitted, getting the tableaux below.

$$\begin{array}{c} F\neg B \wedge (A \vee B) \rightarrow A \\ | \\ T\neg B \wedge (A \vee B) \end{array}$$

```
   FA
    |
   T¬B
    |
  T(A∨B)
   /    \
 TA      TB
  |       |
  ×      FB
          |
          ×
```

This is a classical tableaux in the sense of Smullyan [1968].

Example. (Peirce's Law) Here is a classical tableaux proof of another classical proposition.

```
1   F((A→B)→A)→A)
            |
2      FA              by 1
        |
3    T((A→B)→A)           by 1
       /        \
4  F(A→B)        TA       by 3
     |            |
5   FB            ×    by 4    by 2, 4
     |
6   TA                 by 4
7    ×               by 2, 6
```

Example. Here is another classical tableaux.

```
1        T((A∧(¬A))∨(B∨(C∧D))))
           /                \
2    T(A∧¬A)            T(B∨(C∧D))     by 1
        |                 /      \
3      TA              TB     T(C∧D)   by 2
        |                       /
4     T¬A                     TC      by 3
        |                      |
5      FA                     TD      by 4    by 3
```

6 ×

This is not a proof. Here we get a contradiction on the left branch. Each of the other branches exhibiting valuations making the topmost signed statement true. That is, any valuation making B true makes the topmost signed statement true, any valuation making C, D both true makes the topmost signed statement true. This exhibits the fact that counterexamples can be read off tableaux.

Atomic tableaux. The modal connectives.

Box.

$Tw\Vdash\Box\varphi$	$Fw\Vdash\Box\varphi$
$\mid$	$\mid$
$Tv\Vdash\varphi$	$Fv\Vdash\varphi$
	$\mid$
	$TwRv$
provided TwRv occurs on the branch already	for a new v not yet occurring on the branch

Diamond.

$Tw\Vdash\Diamond P$	$Fw\Vdash\Diamond P$
$\mid$	$\mid$
$Tv\Vdash P$	$Fv\Vdash P$
$\mid$	
$TwRv$	
for a new v not yet occurring on the branch	provided TwRv occurs on the branch already

Explanation. Recall the definition of forcing at a world in clauses 5), 6) above. For any tableaux entry $Tw\Vdash\Box\varphi$ on an open branch, if v is a valuation constant already occurring in a signed formula on that branch, we wish to be able to adjoin $Tv\Vdash\varphi$ to the end of that branch. For any tableaux entry $Fw\Vdash\Box\varphi$ on an open branch, and any valuation constant v not occurring on that branch, we wish to be able to adjoin to the end of that branch TwRv followed by $Fv\Vdash\varphi$. These are the last of the rules of proof for modal propositional logic $L_{\Box,\Diamond}$.

Remark. In these notes, diamond $\Diamond$ will not be mentioned again. We concentrate on $L_{\Box}$.

The definition of semantic validity must be expressed with care. A proposition P is valid in a model $(\mathcal{F}, R, v)$ if P is forced by every w in $\mathcal{F}$. A proposition P is valid in a frame $(\mathcal{F}, R)$ if for every possible valuation function v for that frame, P is valid in the model $(\mathcal{F}, R, v)$. A proposition is valid if valid in every frame. So P is valid if for every frame $\mathcal{F}$, every valuation v, every world w of $\mathcal{T}$, w forces P.

It is useful to have the notion of a deduction of proposition B from premises $A_1, ..., A_n$. The notion of deduction is supposed to be a syntactical equivalent of the semantical assertion that for all frames $(\mathcal{T}, R)$ for which $A_1, ..., A_n$ are valid in $(\mathcal{T}, R)$. B is also valid for $(\mathcal{T}, R)$. The hypothesis is that for all i, all w in $\mathcal{T}$, $w \Vdash A_i$. This is not reflected in the tableaux proof rules above. We need an additonal

<u>Atomic tableaux for deductions</u>.

For any premise A_i, any world constant v,

the tableaux below may be appended to any open branch

$$
\begin{array}{c}
| \\
Tv \Vdash A_i
\end{array}
$$

Then a deduction of conclusion B from premises $A_1, ..., A_n$ is a tableaux with all branches closed in which

1) the apex is $Fw \Vdash B$
2) The atomic tableaux for proofs are allowed.
3) Application of the tableaux deduction rule for premises $A_1, ..., A_n$ is allowed– i.e., the atomic tableaux for deduction indicated can be appended to the base of any open branch at will for any premise A_i and any valuation constant v.

<u>Theorem</u>. (Correctness). Every proposition with a tableaux proof (by rules 1–6) is valid. If proposition B has a tableaux deduction from $A_1, ..., A_n$, then B is valid in any frame in which $A_1, ..., A_n$ are valid.

<u>Theorem</u>. (Completeness) Every valid proposition has a tableaux proof. If B is valid in every frame in which $A_1, . ., A_n$ are valid, then there is a tableaux deduction of B from $A_1, ..., A_n$.

The completeness and correctness proofs mimic the classical case. They are straightforward by a "complete systematic tableaux procedure" like that of Smullyan for classical tableaux and of Fitting [1983] for prefixed tableaux. They will be supplied in a more complete version of these notes.

<u>Example</u>. All classical tautologies have tableaux proofs. For simply substitute "$Tw\vdash\varphi$" for "$T\varphi$", "$Fw\vdash\varphi$" for "$F\varphi$" throughout the classical tableaux proof of the tautology.

<u>Example</u>. Here is a tableaux proof of $\Box(A\to B)\to(\Box A\to\Box B)$. It is the axiom used to prove that [P : $\Box$P is a theorem] is closed under modus ponens in Hilbert–style systems of propositional modal logic based on axioms and modus ponens as the sole rule of inference.

1	$Fw\vdash\Box(A\to B)\to(\Box A\to\Box B)$		
2	$Tw\vdash\Box(A\to B)$		by 1
3	$Fw\vdash\Box A\to\Box B$		by 1
4	$Tw\vdash\Box A$		by 3
5	$Fw\vdash\Box B$		by 3
6	$TwRv$	new v	by 5
7	$Fv\vdash B$		by 5
8	$Tv\vdash A$		by 4, 6
9	$Tv\vdash A\to B$		by 2, 6
10	$Fv\vdash A$ $Tv\vdash B$		by 9
11	× ×		by 8, 7

<u>Example</u>. (Modus Ponens) From premises A, $A\to B$, deduce B.

1	$Fw\vdash B$	
2	$Tw\vdash A\to B$	Premise
3	$Tw\vdash A$	Premise
4	$Fw\vdash A$ $Tw\vdash B$	By 2
	× ×	By 1, by 3

The semantical equivalent is that if A and $A\to B$ are valid in a frame, then so is B.

Example. From premise A, deduce $\Box A$. This is called the rule of necessitation. (Line 4 of the deduction below uses the deduction atomic tableaux.)

1 $Fw \Vdash \Box A$

2 $TwRv$ by 1

3 $Fv \Vdash A$ by 1

4 $Tv \Vdash A$ premise

×

Example. In contrast, $A \to \Box A$ is not valid.

1 $Fw \Vdash A \to \Box A$

2 $Tw \Vdash A$ by 1

3 $Fw \Vdash \Box A$ by 1

4 $TwRv$ by 3

5 $Fv \Vdash A$ by 3

This produces a frame $\mathcal{F} = \{w, v\}$, $R = \{(w, v)\}$, and a valuation v in which A is true at w but not at v.

$$\overset{A}{w} \longrightarrow v$$

In this frame, w does not force $A \to \Box A$, so $A \to \Box A$ is not valid in all frames. But $A \to \Box A$ is valid in those models $(\mathcal{F}, R, v)$ such that for all $w, x \in \mathcal{F}$, if wRx and under the valuation, if A is true at w, then A is true at x. So one has to be very careful in formulating any sort of "deduction theorem" saying that under certain circumstances, if B can be deduced from A, then $A \to B$ is provable. One would have to decode the forcing meaning, whicn unwinds the modal operators as quantifiers, and look at their scopes.

Compare

– "From A as premise, deduce conclusion B." This says the following. Suppose that in a frame $(\mathcal{F}, R)$, all possible valuations v that can be assigned to worlds in $\mathcal{F}$ give models $(\mathcal{F}, R, v)$ such that for all w in F, w forces A. Then conclude that for all possible valuations v of that frame $(\mathcal{F}, R)$, in the model $(\mathcal{F}, R, v)$, every world w in $\mathcal{F}$ forces B.

– "$A \to B$ is a theorem." This says the following. In any model $(\mathcal{F}, R, v)$, for any w in $\mathcal{F}$,

if w forces A, then w forces B.
The quantifier structure of the two statements is quite different.

5. Some modal axioms.

Example. $\Box A \to A$ is not valid. It is traditionally called T. If $\Box$ is interpreted as "I know", then T says "knowledge is truth", so it is called the "knowledge axiom. If $\Box$ is interpreted as "I believe", then T says "What I believe is true". One <u>can</u> have false beliefs.

1	$Fw \Vdash \Box A \to A$	
2	$Tw \Vdash \Box A$	by 1
3	$Fw \Vdash A$	by 1

There is no contradiction. Reading off this tableaux the worlds and the forced atomic statements at those worlds, a one world frame $(\mathcal{W}, R)$. $\mathcal{W} = \{w\}$ with empty accessibility relation R and A false at w makes $\Box A \to A$ false. A reflexive frame is one in which wRw for every world w. Looking at the tableaux line 2, we would get $Tw \Vdash A$, contradicting line 3. So $\Box A \to A$ is valid in all reflexive frames. Conversely, any proposition valid in all reflexive frames can be deduced from $\Box A \to A$.

<u>Reflexive tableaux development rule</u>.

If w is any world occurring in an entry, at the base of any open branch through that entry we may append the tableaux

$$TwRw$$

A proposition is valid in all reflexive frames if and only if provable by the standard modal tableaux plus the reflexive tableaux development rule.

Example. $\Box A \to \Box\Box A$ is not valid. Traditionally, this proposition is called "4". In newer papers, this is called the "positive introspection axiom", "What I believe, I believe I believe".

1	$Fw \vdash \Box A \to \Box\Box A$	
2	$Tw \vdash \Box A$	by 1
3	$Fw \vdash \Box\Box A$	by 1
4	$TwRv$	new v by 3
5	$Fv \vdash \Box A$	by 3
6	$TvRu$	new u by 5
7	$Fu \vdash A$	by 5
8	$Tv \vdash A$	by 2, 4

There is no contradiction. But reading off the true atomic statements from the tableaux, we get a three world frame $\mathcal{F} = \{w, v, u\}$, with wRv, vRu, and in the model with A true at v, but A not true at w or u. This is a counterexample to the validity of $\Box A \to \Box\Box A$, which is not true at w.

$$\begin{array}{ccccc} & & A & & \\ w & \longrightarrow & v & \longrightarrow & u \end{array}$$

The labelled graph above has branches representing all accessibility relations and nodes representing all worlds. Labels of nodes are atomic propositions true at that world.

A transitive frame $(\mathcal{F}, R)$ is one such that for all w, v, u, if wRv and vRu, then wRu. Then from the tableaux we get TwRv, TvRu, so we get TwRu. Then we could apply line 2 and get $Tu \vdash A$, contradicting line 7. So $\Box A \to \Box\Box A$ is valid in transitive frames. Conversely, any proposition valid in all transitive frames is deducible from $\Box A \to \Box\Box A$. If we wish to deal only with transitive accessibility relations, we can add the following rule of tableaux development directly to those already given.

<u>Transitivity tableaux development rule</u>.

If TwRu and TuRv occur on a branch, we may append to all (some) open branches through that pair, the tableaux

TwRv

Then a proposition is valid in all transitive frames if and only if it has a tableaux proof using the standard modal tableaux rules plus the transitive tableaux development rule.

Example. $\neg\Box\varphi \to \Box\neg\Box\varphi$ is not valid. In older papers this is often abbreviated E for the Euclidean axiom, or 5. In newer papers, this is called the "negative introspection axiom". "What I don't believe, I believe I don't believe"

1	$Fw \Vdash \neg\Box\varphi \to \Box\neg\Box\varphi$		
2	$Tw \Vdash \neg\Box\varphi$		by 1
3	$Fw \Vdash \Box\neg\Box\varphi$		by 1
4	$Fw \Vdash \Box\varphi$		by 2
5	$TwRv$		by 4
6	$Fv \Vdash \varphi$	new v	by 4
7	$TwRu$	new u	by 3
8	$Fu \Vdash \neg\Box\varphi$		by 3
9	$Tu \Vdash \Box\varphi$		by 8

If we read off the true atomic sentences, they are wRv, wRu. With φ declared false in all three valuations, we get a model

$$\begin{tikzcd} & v \\ w \arrow[ur] \arrow[r] & u \end{tikzcd}$$

in which w does not force $\neg\Box\varphi \to \Box\neg\Box\varphi$. An Euclidean frame is one such that for all w, v, u in $\mathcal{F}$, wRu and wRv imply uRv. Looking at the tableaux, we had TwRu, TwRv. With the Euclidean property, we get also TuRv, by line 9 this gives $Tv \Vdash \varphi$, contradicting line 6. So $\neg\Box\varphi \to \Box\neg\Box\varphi$ is valid in all Euclidean frames. Conversely, any proposition true in all Euclidean frames is deducible from $\neg\Box\varphi \to \Box\neg\Box\varphi$. For a Euclidean R , for any world w in $\mathcal{F}$ the restriction of R to $[v \in \mathcal{F}: w \mathrel{R} v]$ is an equivalence relation, but this set does not necessarily contain w itself.

Euclidean tableaux development rule.

If a branch contains entries TwRu and TwRv, then we may append to every open branch through these two entries the tableaux

$$\begin{array}{c} | \\ TuRv \end{array}$$

Then a proposition is true in all Euclidean frames if and only if it has a tableaux proof using the standard modal tableaux plus the Euclidean Tableaux development rule.

Example. $\Box P \to \neg\Box\neg P$ is not valid. In the older literature, this is axiom D. In newer papers, this is called the serial axiom. "What I believe, I don't believe the negation of"

$$\begin{array}{lll} & Fw\Vdash \Box P \to \neg\Box\neg P & \\ & | & \\ 2 & Tw\Vdash \Box P & \text{by } 1 \\ & | & \\ 3 & Fw\Vdash \neg\Box\neg P & \text{by } 1 \\ & | & \\ 4 & Tw\Vdash \Box\neg P & \text{by } 3 \end{array}$$

There is no contradiction. The model with a single world w and empty accessibility R and P false at w will do to falsify $\Box P \to \neg\Box\neg P$. A serial frame is one such that for every world w, there is a world v such that wRv. In this case from $Tw \Vdash \Box P$, $Tw \Vdash \Box\neg P$, we get $Tv \Vdash P$, $Tv \Vdash \neg P$, a contradiction, so $\Box P \to \neg\Box\neg P$ is valid in serial frames. In fact, any proposition valid in all serial frames is deducible from $\Box P \to \neg\Box\neg P$.

Serial tableaux deduction rule.

For any world constant v occurring in an entry on an open branch, and any valuation constant u not on that branch, we may append to that open branch the tableaux below.

$$\begin{array}{c} | \\ TvRu \end{array}$$

Thus a proposition is valid in every serial frame if and only if it has a tabeaux proof using the standard modal tableaux plus the serial tableaux deduction rule.

Hilbert systems for modal logic. A standard set of axioms and rules of inference for a Hilbert style modal logic called K defines the theorems (of K) as the smallest set of propositions such the following hold.

1. All classical tautologies with modal propositions substituted for variable are theorems. (These tautologies are the "axioms".)

2. All propositions $\Box(A \to B) \to (\Box A \to \Box B)$ are theorems.
3. If A, $A \to B$ are theorems, then B is a theorem.
4. If A is a theorem, then $\Box A$ is a theorem.

The notion of deductive closure of a set of premises S would replace "are theorems" by "are consequences of S" in 1, 2, 3, 4, and add
5) Premises in S are consequences of S.

We have already shown that each axiom and rule of inference holds for tableaux provability using the standard modal tableaux. Propositions proven by the tableaux method are valid in all frames. Any proof of completeness for the system based on 1)–4) shows that tableaux provability coincides with provability in this system.
Here is a list of commonly occurring systems.

– K is the proof system using the classical and modal atomic tableaux.
This tends to be a substructure of modal systems used for computer science.
– T is K plus the schema $\Box A \to A$ as premises for deductions. T tends to be regarded as the logic of knowledge (true beliefs).

• A proposition is provable in T iff valid in all reflexive frames iff provable by the tableaux of K plus the reflexive tableaux development rule.

– S4 is T plus the additional schema $\Box A \to A$ and $\Box A \to \Box\Box A$ added as premises for deductions.

• A proposition is provable in S4 iff valid in all reflexive, transitive frames iff provable by the tableaux of K plus the reflexive and transitive tableaux development rules.

– S5 is S4 plus the additional schema $\neg\Box A \to \Box\neg\Box A$ added as premises for deductions. A relation R on $\mathcal{F}$ is transitive, Euclidean and reflexive iff R is an equivalence relation.

• A proposition is provable in S5 iff valid in all frames with an equivalence relation on $\mathcal{F}$ as accesibility iff provable by the tableaux rules of K plus the reflexive, transitive, and Euclidean tableaux development rules.

There is more to say for S5.

Lemma. Suppose $(\mathcal{F}, R, v)$ is a model and $w \in \mathcal{F}$. Define a model $(\mathcal{G}, R', v')$ by setting $\mathcal{G} = [w' \in \mathcal{F}: wRw']$, $R' = R \cap \mathcal{G} \times \mathcal{G}$, $v' = v$ restricted to $\mathcal{G}$. Then w forces φ in $(\mathcal{F}, R, v)$ iff w forces φ in $(\mathcal{G}, R', v')$.
The proof of the lemma is by induction on the definition of forcing.

According to this lemma, φ is forced by all v in all models $\mathcal{F}$ with R an equivalence relation on $\mathcal{F}$ iff forced by all v in all models with R an equivalence relation on $\mathcal{F}$ which has a single equivalence class, that is $R = \mathcal{F} \times \mathcal{F}$. A complete frame is one where the accessibility on $\mathcal{F}$ is $R = \mathcal{F} \times \mathcal{F}$.

The system S5 was used by Moore [1985] for autoepistemic logic (see below). The system S5 is suitable for reasoning about knowledge in distributed systems, provided that there are many S5 modal connectives $\Box_A$, one for each agent or machine A. This takes one beyond complete frames, the lemma no longer works for multiple agents, one is stuck with many equivalence relations, one for each agent, see Halpern Moses [1984, 198?], and also Lehmann [1984].

Complete tableaux development rule.

If world constants u, v occur in entries, then we may append to the base of any open branch through those entries the tableaux below.

$$\begin{array}{c} | \\ TuRv \end{array}$$

• A proposition is provable in K5 iff valid in all complete frames iff provable by the tableaux of K plus the complete tableaux development rule.

– K45 is K plus the additional schema $\neg\Box A \rightarrow \Box\neg\Box A$, $\Box A \rightarrow \Box\Box A$.

• A proposition is provable in K45 iff true in all transitive Euclidean frames iff provable by the tableaux of K plus the transitive and Euclidean development rules.

K45 is a candidate (Halpern and Moses [1986], Moore [1988]) for a logic of belief for a "logically omniscient completely introspective rational agent", see below.

6. Non–monotonic reasoning. An important computer science class of modal logics arise in artificial intelligence in the area called "non–monotonic reasoning". In monotonic reasoning, a consequence drawn by a deduction from a set of axioms is also drawn by the same deduction from any larger set of axioms. That is, the consequence and the deduction are never withdrawn later however the set of axioms is enlarged. Monotonic reasoning is the only reasoning in classical mathematics and in constructive mathematics as well. The axioms upon which mathematics is based have been extended from those for Euclidean geometry in Euclid's time (300 B. C.) to those for calculus in the time of Newton and Leibnitz (1680's) to those for analysis in the time of Weierstrass (1850's), to those for set theory in the time of Cantor (1880's). Gaps in proofs may have to be filled, but complete proofs are never withdrawn. This is the monotone nature of mathematics, in which mathematicians never disagree as to what is a proof and never reject the proofs of their predecessors, but build on their results instead. This characteristic may, in fact, be unique to mathematics if one looks at the history of all other disciplines, scientific or scholarly.

Think of each logic as having propositions. These propositions are certain strings from a fixed alphabet. The logic also has rules of inference. What is a monotone rule of inference? By instantiating the rules of inference, each monotonic rule of inference can be cast in the form

"From $\alpha_1, ..., \alpha_n$, infer γ",

where $\alpha_1, ..., \alpha_n$ are propositions (premises), γ is a proposition (conclusion).

The rules of inference with no premises we think of as the "logical axioms". In a logic with monotone rules of inference, if A is a set of propositions, then a set D of propositions is called a <u>deductively closed theory containing A</u> if

"For each rule of inference, if $\alpha_1, ..., \alpha_n$ are in D, then γ is in D."

In a logic with monotone rules of inference, for every set A of propositions there is a smallest deductively closed theory containing A. This property is lost in the non–monotonic logics below.

For non–monotonic logics we allow a more general form of rule of inference. We label premises purely formally as "positive" or "negative". Each non–monotonic rule of inference can be cast in the form

"If $\alpha_1, ..., \alpha_n$ are positive premises and $\beta_1, ..., \beta_k$ are *negative premises, infer* γ",
where $\alpha_1, ..., \alpha_n, \beta_1, ..., \beta_k, \gamma$ are propositions.

So each monotonic rule can be recast as a non–monotonic rule by labelling its premises positive and having an empty set of negative premises.

For a system based on non–monotonic rules of inference, a set D of propositions containing set A *is called a* *deductively closed set containing A* if for all rules of inference, if the positive premises $\alpha_1, ..., \alpha_n$ are in D and negative premises $\beta_1, ..., \beta_k$ are not in D, then γ is in D.

In non–monotonic reasoning a consequence drawn by a deduction from a set of axioms may not be a consequence of a larger set of axioms, due to radically different deductively closed sets containing the changed axioms in the non–monotonic case.

7. **Informal Belief**. Commonsense reasoning is often non–monotonic. I, a rational introspective agent, have beliefs in my current complete set of beliefs B based on incomplete information. Later I have to change to another belief set B' in which we may no longer have some of the previous beliefs in B. We assume my belief set is closed under classical logical consequence, that is, I believe the logical consequences of what I believe (principle of logical omniscience). We assume that my belief set contains all of my beliefs. We suppose that the facts about the external world (objective facts) and rules I know for sure (our knowledge base) are in all my belief sets.

Example. In my current belief set B might be the propositions

R: "If x is a bird, and I do not believe that x cannot fly, then x can fly"
F: "Tweety is a bird."

Suppose

"Tweety cannot fly."

is not derivable from my belief set B. My beliefs are assumed closed under classical deduction so we conclude that

"I do not believe that Tweety cannot fly."

is in B. So applying rule R of B, we deduce that "Tweety can fly" and thus also we deduce that

"I believe that Tweety can fly."

But B is closed under classical deduction, so this proposition is in B. I now visit New Zealand and see a Kiwi, and realize that Tweety is a Kiwi, and conclude

N: "Tweety cannot fly"

In my new belief set B' I retain rule R and fact F and put new fact N into B'.

Since

"Tweety can not fly"

is in B', and is therfore a belief,

"I believe that Tweety can not fly"

is in B', and the hypothesis of rule R is not satisfied for B', and we cannot conclude, using rule R, that "Tweety can fly" is in B'. We have withdrawn a conclusion of B. This is the non-monotonicity of the reasoning. If indeed

"Tweety can fly"

is <u>not</u> derivable from B', since B' consists of all beliefs, we can conclude that

"I don't believe that Tweety can fly"

is in B' as well.

8. Autoepistemic logic. The complete set B of all beliefs of an agent is the subject of Moore's autoepistemic logic [1984]. His is an account of how an agent reasons about the agent's <u>own</u> beliefs. This is the origin of the use of "autoepistemic", the notion of self knowledge. Let $L_\Box$ be the set of all modal propositions based on classical connectives and $\Box$. Let L be the subset of classical propositions. Moore reads $\Box P$ as "P is in the agent's complete current belief set B". In his exposition he begins with $L_\Box$ regarded as a classical propositional logic with every proposition of the form $\Box\varphi$ as an additional propositional letter along with the usual ones. Thus a classical deductive closed set of this classical $L_\Box$ is merely one closed under classical tableaux deductions, or one closed under tautologies and modus ponens. A classical $L_\Box$–assignment maps all propositional letters and all propositions $\Box\varphi$ into $\{T, F\}$, and each of these is extendible to a classical $L_\Box$–valuation with domian $L_\Box$ and values in $\{T, F\}$.

<u>Definition</u>. An autoepistemic theory is a set B of $L_\Box$–propositions for which there is a classical $L_\Box$–valuation v such that B consists of all P in $L_\Box$ such that $v(\Box P) = T$. Also v *is said to* be an autoepistemic interpretation of B.

Since v can be $L_\Box$–valued arbitrarily on any atomic proposition P and any modal $\Box Q$, there is no necessary connection between the truth values of these propositions.

<u>Example</u>. For propositional letters A, B, we can define an $L_\Box$–valuation with $\Box A$ true, $\Box B$ true, $\Box(A \wedge B)$ false. So A, B are in the corresponding autoepistemic theory, but $A \wedge B$ is not. This is simply an instance of the fact that we can $L_\Box$–value propositions of the form $\Box P$ arbitrarily and independently. So Moore allows in his definition of an autoepistemic theory B that an agent may be incapable of any reasoning from beliefs to beliefs. This makes it possible in this framework to study adding in reasoning abilities of limited strength by suitable axioms restricting the allowed $L_\Box$–valuations. So the notion of autoepistemic theory allows the study of agents with varied reasoning abilities by introducing additional modal axioms reflecting these abilities.

<u>Definition</u>. A model of autoepistemic theory B is an autoepistemic interpretation of B such *that all propositions in* B *are true.*

<u>Definition</u>. An autoepistemic theory B is <u>semantically complete</u> if B contains every

proposition true in all autoepistemic models of B.

Theorem (Moore [1985]). B is semantically complete iff
1) B is closed under classical $L_{\Box}$–consequence.
2) If $P \in B$, then $\Box P \in B$.
3) If $\neg(P \in B)$, then $\Box P \notin B$.

These three properties were the definition of a stable set B of modal propositions given by Stalnaker [1980, 1989].

Example. We informally used the stability of B and B' in the Tweety example.
– We applied 3) to verify that rule R could be applied to yield that "Tweety can fly" is in B, with P the proposition "Tweety cannot fly".

– We applied 2) to verify that "I believe that Tweety cannot fly" is in B'. with P the proposition "Tweety cannot fly".

Is stability a reasonable condition for the complete belief set B of a rational agent?

Requirement 1) is that the agent should be "logically omniscient", that is, any classical logical consequence of the agent's belief set B should also be in B. This is a simplyfying idealization, since to recognize that a given proposition is a classical logical consequence of known axioms for a given B is at least an NP–complete problem (Halpern and Moses [1985]). Verifying conditions 2) and 3) for specific propositions both involve this NP–hard problem.

Reformulating, the condition that B is closed under $L_{\Box}$–consequence means exactly that a tautology with beliefs substituted for variables is a belief, and that beliefs be closed under modus ponens. For a modal point of view this commits us exactly to the closure conditions on B imposed by deductive closure in the Hilbert style version of system K described in section. Equivalently, this commits us exactly to system K, of modal atomic tableaux for $\Box$, together with the deduction rule for tableaux.

Definition. An autoepistemic theory B is sound with respect to a set of premises A iff every autoepistemic interpretation of B in which all the propositions of A are true is an autoepistemic model of B.

Definition. An autoepistemic theory B is grounded in a set of premises A iff B is contained in

$$Cn[A \cup \{\Box p : p \in B\} \cup \{\neg\Box P : \neg(p \in B)\}],$$

where Cn is the classical $L_\Box$–consequence relation.

Theorem (Moore [1985]). An autoepistemic theory B is grounded in A iff sound with respect to A.

Theorem (Moore [1985]). If A is a set of premises, then an autoepistemic theory T extending A is sound and semantically complete with respect to A iff

$$T=Cn[A \cup \{\Box p : p \in T\} \cup \{\neg\Box P : p \notin T\}]$$

Definition. An autoepistemic theory B is a stable expansion of a set of premises A if B contains A and is grounded in A.

Moore identifies the possible complete sets of beliefs that a rational agent might hold after accepting A as the stable expansions of A. The problem in dealing with stable expansions is that there can be none, one, two, or many, and the are not so easy to identify.

Example. $\{\neg\Box P \rightarrow Q, \neg\Box Q \rightarrow P\}$ has at least two stable expansions, one containing P but not Q, one containing Q but not P.

Example. $\{\neg\Box P \rightarrow P\}$ has no stable expansions. Let B be a purported stable expansion. If P is in B , then B is not grounded and therefore not a stable expansion. Any stable If P is not in B, then $\neg\Box P$ is in B (B is stable), so P would be in B (B is grounded), a contradiction.

9. Autoepistemic logic 2. I am indebted to W. Marek for the outline of this section. We will repeat from scratch some of the same ground as in the previous section, but from a different viewpoint. This viewpoint starts out with a "rational agent", it does not lend itself as immediately to partially rational agents with limited reasoning powers as did the Moore exposition of the previous section. We begin with "list semantics" for $\Box$. We work again in

$L_{\Box}$. We let L be the corresponding classical language without $\Box$. This exposition emphasizes the role of classical L–valuations v. These are valuations of the classical propositions only, obtained from assignments to the classical propositional letters (not the $\Box P$ propositions). If S is a set of modal propositions (called the "list"), a "list" consequence relation "$\Vdash_{v,S}$" is defined from S.

1. For propositional constants P, $\Vdash_{v,S} P$ iff $v(P) = T$.
2. $\Vdash_{v,S} \neg\varphi$ iff not $\Vdash_{v,S} \varphi$.
3) $\Vdash_{v,S} (\varphi \vee \psi)$ iff $\Vdash_{v,S} \varphi$ or $\Vdash_{v,S} \psi$.
4) $\Vdash_{v,S} \Box\varphi$ iff $\varphi \in S$.

Remark. We can interpret $\Box\varphi$ as "the agent believes φ", we can interpret S as the list of the agent's beliefs, we can interpret 4) as expressing that if the agent is asked if the agent believes φ, the agent consults the "list", and answers yes in case φ is on the list.

Definition (S–entailment). Let I be a set of modal propositions. Then

$I \Vdash_S \varphi$ iff for all valuations v, $\Vdash_{v,S} I$ implies $\Vdash_{v,S} \varphi$.

Definition. An expansion of I is a collection S of modal propositions such that the fixed point condition $S = \{\varphi : I \Vdash_S \varphi\}$.

Let I be given, suppose that S is being guessed by the agent. What does it mean for the guess to be correct?

1) Whatever is S–entailed should be in S (an adaquacy requirement).
2) Whatever is in S should be S–entailed (a completeness requirement).

Theorem (Moore). The following are equivalent.
a) S is an expansion of I.
b) $S = Cn(I \cup \{\Box\varphi : \varphi \in S\} \cup \{\neg\Box\varphi : \varphi \notin S\})$.
(Here Cn is classical consequence).

A set S of modal propositions is called <u>stable</u> (Stalnaker, 1980) if

1) closed under classical deduction,
2) $\varphi \in S$ implies $\Box\varphi \in S$,
3) $\varphi \notin S$ implies $\neg\Box\varphi \in S$.

Condition 3) makes the reasoning non–monotonic. Stable theories are supposed to represent the set of all beliefs of a *completely rational* introspective agent.

The objective part of a set of modal propositions is its subset of classical propositions without $\Box$.

<u>Theorem</u>. (Moore [1984]).
(i) If S is an expansion of I, then S is stable.
(ii) If S is stable, then S is an expansion (and in fact the unique expansion) of its objective part.

<u>Theorem</u> (Marek [1986], Konolige). Every collection of L–propositions closed under classical L–consequence is the objective part of a stable $L_\Box$–theory.

So stable $L_\Box$–theories are in a 1–1 correspondence with classical objective L–theories.

We now discuss how to generate expansions.
Let $L_{\Box,n}$ be the propositions of $L_\Box$ with $\Box$'s nested to at most depth n.

<u>Operation E.</u> Given a set A of L–propositions,
let E(0, A) be the set of classical L–consequences of A,
let E(n+1, A) be the set of classical $L_\Box$–consequences in $L_{\Box,n+1}$ of

$$E(n, T) \cup \{\Box\varphi : \varphi \in E(n, T)\} \cup \{\{\neg\Box\varphi : \varphi \in L_{\Box,n} - E(n, T)\}.$$

Let E(T) be the union of all E(n, T).

<u>Theorem</u> (Marek [1986]). If A is a set of propositions in L, then E({A}) is the unique expansion of A.

Thus to find the expansion one has to find the objective part.

Example. Let I consist of $\neg\Box P \rightarrow P$ alone. This I has no expansion. The only candidates are: 1) $E(\emptyset)$ and 2) $E(\{P\})$.

Re 1: P is not in $E(\emptyset)$, so $\neg\Box P \in E(\emptyset)$, and if it is an expansion, then $P \in E(\emptyset)$ by modus ponens.
Re 2: One can check that P is not an $L_\Box$ classical consequence of
$I \cup \{\Box\varphi : \varphi \in E(P)\} \cup \{\Box\varphi : \text{not } (\varphi \in E(P)\}$.

Example. Let I consist of $\neg\Box P \rightarrow Q$ and $\neg\Box Q \rightarrow P$. This I has two expansions, E(P) and $E(\{Q\})$. (There are two more candidates $E(\emptyset)$ and $E(\{P, Q\})$, but they are discarded by the same reasoning as above.) Why is $E(\{P\})$ an expansion? Since Q is not in $E(\{P\})$, we get that $\neg\Box Q$ is in $\{\neg\Box\varphi : \varphi \notin E(\{P\})$. From this

$$E(\{P\}) = Cn[I \cup \{\Box\varphi : \varphi \in E(\{P\})\} \cup \{\neg\Box\varphi : \varphi \notin E(\{P\})\}]$$

can be proved. The non–trivial inclusion is from left to right, proved by induction using $E(n, \{P\})$.

Example. Let I consist of P and $\Box P \rightarrow Q$. This has the unique expansion $E(P \wedge Q)$.

Theorem (Moore [1984]). If S_1 and S_2 are two different stable theories, one cannot be contained in the other.

Thus stable theories act a little like classical complete theories.
Clearly if $Cn(I) = Cn(J)$, then I, J have exactly the same expansions.

Autoepistemic normal form. An autoepistemic (ae) clause is a modal statement of the form $A \rightarrow \sigma$, where

σ is in L, and
A is of the form $\Box\varphi_1 \wedge \ldots \wedge \Box\varphi_r \wedge \ldots \wedge \neg\Box\psi_1 \wedge \ldots \wedge \neg\Box\psi_s$,

where $\varphi_1,\ldots\varphi_r$, $\psi_1,\ldots\psi_s$ are in $L_\Box$.

We can take the theories we are concerned with to be generated by ae clauses. Here, imitating logic programming, we call "σ" the <u>head</u> and "A" the <u>body</u> of the ae clause.

<u>Theorem</u> (Marek and Truszczynski [1988, 1989]).

1) For every $I \subsetneq L_\Box$ there exists an $I' \subsetneq L_\Box$ with the same expansions such that the ae–clauses $A \to \sigma$ of I' all have A of $\Box$–nesting depth 1.

2) Such a I' can be computed in polynomial time.

This tells us that the problem of Byzantine generals does not exist in autoepistemic logic.

<u>Normal form for expansions.</u>

<u>Theorem</u> (Marek and Truszczynski [1988, 1989]).

Let $I = \{O_i = A_i \to \sigma_i\colon\ 1 < i < k\}$. Then

1) Every expansion of I is of form

$$E(\{O_i : i \in J\}) \text{ for suitably chosen } J \subset \{1,\ldots,k\}$$

2) A theory $S \subsetneq L$ has the property that $E(S)$ is an expansion of I if and only if there exists a representation of S in the form $S = Cn(\{O_i\colon i \in J\})$ such that

(i) $I \subsetneq E(S)$

(ii) For all $i \in J$, we have that $A_i \in E(S)$.

The problem is that a theory S may have numerous representations as $Cn(\{O_i\colon i \in J\})$ for various J. It is enough that <u>one</u> of these representations has the property 2(ii)..

<u>Example</u>. $I = \{\neg\Box P \to (P \wedge Q),\ \neg\Box R \to P,\ \neg\Box R \to Q\}$. Then $E(\{P \wedge Q\})$ is an expansion of I for the following reason. The second and third clause give "epistemic support". If we select our representation of $E(\{P \wedge Q\})$ $(=E(\{P, Q\}))$ from first clause, then we would not have the necessary epistemic support. Fortunately only one representation is required so it is an expansion.

<u>Theorem</u> (Marek [1986], Moore [1988]). There is an algorithm which, given $T \subsetneq L$ and $\varphi \in L_\Box$, tests whether or not $\varphi \in E(T)$.

This algorithm cannot be polynomial time, but it is polynomial time in the characteristic function of Cn(T). The last two theorems imply that we can effectively compute all expansions in the propositional logic case.

Definition. φ is ae–consequence of I iff φ belongs to all expansions of I.

The previous two theorems imply that this notion is decidable.

Example. $I = \{\neg\Box P \rightarrow Q, \neg\Box Q \rightarrow P\}$. Proposition $P \vee Q$ is an ae consequence of I. Proposition $\Box P \vee \Box Q$ is ae–consequence of I. Formula $(\Box P \wedge \neg\Box Q) \vee (\neg\Box Q \wedge \Box P)$ is ae–consequence of I. Neither $\Box P$ nor $\Box Q$ are ae–consequence of I.

Example. (a) $I = \{\neg\Box P \rightarrow P\}$. I is consistent but it is ae–inconsistent (since there is no expression, intersection of expansions is $L_{\Box}$).

(b) $I = \{\neg\Box P \rightarrow P, \Box P \rightarrow P\}$ has a unique expansion E(P), thus it is ae–consistent. Its subtheory $\{\neg\Box P \rightarrow P\}$ has no extensions. Thus there are *ae–consistent theories with* inconsistent subtheories. The subtheory $\{\Box P \rightarrow P\}$ has two expansions: E(TAUT), E(P). Thus we have a situation in which the smaller theory has a smaller set of consequences (previously had bigger...). The fact that I can have many or no expansions is disturbing. Are there conditions that imply uniqueness of expansions?

Gelfand stratification. A G–clause is a propositon of the form

$$(P_1 \wedge ... \wedge P_K \wedge \Box Q_1 ... \wedge \Box Q_r \wedge \neg\Box S_1 \wedge ... \wedge \neg\Box S_m) \rightarrow (T_1 \vee ... \vee T_u),$$

where all P_j's, Q_i's, S_k's, and T_t's are atoms. A theory I consisting of G–clauses is G–stratified if there exists a representation.
as a disjoint union $I = I_0 \cup ... \cup I_n$ such that

a) I_0 consists of the classical propositions in I.

b) Whenever clause

$$P_1 \wedge ... \wedge P_k \wedge \Box Q_1 \wedge ... \wedge \Box Q_1 \wedge \neg\Box S_1 \wedge ... \wedge \neg\Box S_m \rightarrow T_1 \vee ... \vee T_u$$

belongs to I_j, then

(i) $Q_1, ..., Q_1, S_1, ..., S_n$ do not appear on the right hand side of implication in any I_m, $m \geq j$ (that is, they are "defined" in $I_0 ... I_{j-1}$).

(ii) $P_1 \ldots P_k$ do not appear on the right hand side of implications in any I_m, $m > j$.

Theorem (Gelfond [1987]). If a theory I consisting of G–clauses is G–stratified, then it possesses a unique expansion.

Stratification. There is another notion of stratification. Theory I consisting of ae–clauses is stratified if there is a representation

$$I = I_0 \cup \ldots \cup I_k$$

such that for all $A \to \sigma \in I_j$,

(i) If an atom appears in σ then it does not appear in any formula in any I_k, $k < j$.

(ii) If an atom appears in A then it does not appear in the "head" of any formula in I_k, $k > j$

Theorem (Marek and Truszczynski [1988]). If I is stratified and $I = I_0 \cup \ldots \cup I_n$, then

(a) I has at most one expansion

(b) If $S = E(T)$ is an expansion of I and T is closed under Cn, and if T_i is an intersection of T with the language whose atoms are those appearing in I_i, then

(i) $E(T_i)$ is an expansion of I_i

(ii) S is an expansion of $T_i \cup I_{i+1} \cup \ldots \cup I_n$.

This theorem tells us how to compute expansions recursively: Compute an expansion of I_0. S_0. Then compute an expansion of $S_0 \cup I_1$, say S_1. Then compute an expansion of of $S_1 \cup I_2$, say S_2. At each step we are guaranteed at most one expansion. If we do not get one at any stage, there is no expansion for I.

Fixed Points. Let $\mathcal{S}$ be a modal logic such as K, S_4, S_5, etc. S is called an $\mathcal{S}$-fixed point over I iff $S = Cn_{\mathcal{S}}(I \cup \{\neg\Box\varphi : \varphi \notin S\})$. This definition is due to McDermott.

Theorem (Svarts [1989]). Expansions of I are precisely the K45 fixed points over I.

Example. $I = \{\Box P \to Q, \Box Q \to P\}$ has two expansions, $E_1 = E(\phi)$ and $E_2 = E(P, Q)$. But P is in the second expansion because $\Box Q$ is there, that is, because Q is there, that is because $\Box P$ is there, that is because p is there. Hence the evidence for p being in E_2 is that "p is there", and there is definitely a circularity.

Let us eliminate this circularity. Define an operator A as follows. For $S \subseteq L_\Box$ put $A(S) = Cn(S \cup \{\Box\varphi : \varphi \in S\})$. and define

$$A_0(S) = S$$
$$A_{n+1}(S) = A(A_n(S))$$
$$A_\infty(S) = \cup_1 A_n(S)$$

Call T an iterative expansion over I if $T = A_\infty(I \cup \{\neg\Box\varphi : \varphi \notin T\})$

Theorem (Marek and Truszczynski [1988]). If T is iterative expansion over I then T is an expansion of I.

Iterative expansions are fixed points with respect to the simplest modal logic in which there is classical tautologies, modus ponens, and necessitation, but no specific modal axiom such as K, T, 4 or 5.

Connection with Logic Programming. Given a logic program P, let $\Pi = \text{ground}(P)$ be the set of all ground instances of P. Then Π consists of expressions of the form

$$C: P \leftarrow Q_1,\ldots,Q_r,\neg S_1,\ldots,\neg S_1$$

To such clause assign its Gelfond translation

$$G(C) = Q_1 \wedge\ldots\wedge Q_r \wedge \neg\Box S_1 \wedge \ldots \wedge \neg\Box S_t \to P$$
$$G(\Pi) = \{G(C) : C \in \Pi\}$$

If P is stratified in the sense of Apt–Blair–Walker then $G(\Pi)$ is G–stratified.
Theorem.

(a) (Gelfond [1987]). Let P be stratified. Let M_p be its "perfect" model in sense of Apt–Blair–Walker. Then $E(Cn(M_p))$ is the only expansion of $G(\Pi)$.

(b) (Marek and Truszczynski [1988]) $E(Cn(M_p))$ is an iterative expansion of $G(\Pi)$.

10. Autoepistemic logic and Euclidean transitive frames. The semantics of §8, §9 using $L_\square$ and L valuations respectively is the classical logic way of doing things. It is natural that there is an equivalent in frame semantics.

Theorem (Moore [1984]). T is a stable autoepistemic theory if and only if T is the set of all valid modal propositions of a complete frame.

This was also proven by Halpern and Moses and Levesque.

Since the complete graphs (in which directed branches extend from every node to every node) are determined up to isomorphism by the cardinality of the nodes alone, one can restrict the complete frames for this theorem to those of the form $K = (\mathcal{S}, R)$, where the set of worlds $\mathcal{S}$ is a set of classical valuations (of the propositional letters) and R is $\mathcal{S} \times \mathcal{S}$. Introduce for each classical valuation V a copy (V, 0), to be used as a new world distinguished from world V if the latter is present in $\mathcal{S}$. Call it the distinguished V. Each pair consisting of K and a distinguished V gives rise to an ordinary Euclidean frame $K_V = (\mathcal{S}', R')$, where $\mathcal{S}' = \mathcal{S} \cup \{(V, 0)\}$ and $R' = R \cup (\{(V,0)\} \times \mathcal{S})$. That, is, every W in K is accessible from (V,0) (including V if the latter is in $\mathcal{S}$), but (V, 0) is not accessible from any world in K. There is a natural extension of K_V to a model, where each world V in K is assigned valuation V, and (V, 0) is assigned valuation V.

Now let B be an autoepistemic theory. Such a model K_V with the assignment above, arising from a complete model, is called a

– "possible worlds" interpretation of B iff B consists of all the propositions valid in this model.

– "possible worlds" model of B iff every proposition of B is true in K_V.

<u>Theorem</u> (Moore [1988]). The "possible worlds" models K_V of B are precisely those "possible worlds" interpretations in which V is a member of $\mathcal{F}$

This affords a back and forth translation of autoepistemic interpretations and autoepistemic models of stable theories to "possible worlds" interpretations and "possible worlds" models as defined above. Moore [1984, 1988] uses these semantic "possible world" characterizations to investigate stable expansions, and decision methods for semantic entailment. His method amounts to the use of the "list semantics" of the last section. There is a nice tableaux–based approach which can be redone in the style of the present lectures due to Niemelä [1986]. We omit these applications, which will be in an expanded version of these lectures, for lack of space.

§11. Modal predicate logic with constant domains. We introduce very briefly a modal predicate logic within classical logic intended to describe a single "constant" domain, with different true atomic statements at different worlds. This logic can be extended to have several modalities $\Box_i$, we do not do this here. Dynamic logic uses this model, there the constant domain is the set of all states of a machine, the $\Box_i$ are induced by programs or commands. This formulation covers theories of beliefs or knowledge for several agents at once about a fixed domain of individuals as well, one $\Box_i$ for each agent. The earliest example of such a theory is Hintikka [1962], see also Konolige [1986] or Halpern and Moses [1985] for further references. Here is the list of primitive symbols.

Predicate letters of degree n,
An infinite list of variables
an infinite list of (individual) constants
Logical connectives $\wedge$, $\vee$, $\rightarrow$, $\neg$, , $\exists$, $\forall$
parentheses (,) and a comma.

The inductive definition of formula and free occcurence of variables is:

1) If R is a predicate letter of degree n and $\alpha_1, \ldots \alpha_n$ are variables or constants, then $R(\alpha_1, \ldots, \alpha_n)$ is a formula. (These are called the atomic formulas.) In atomic formulas all occurrences of all variables are free.

2) If α, β are formulas, then $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$, $(\neg\alpha)$ are formulas. Occurrences of variables in these formulas are free or bound as they were in α, β.

3) If α is a formula, x is a variable, then $((\exists x)\alpha)$, $((\forall x)\alpha)$ are formulas. Occurrences of variables other than x are free or bound in these formulas as they were in α. variable x is bound in all its occurences in these formulas.

3) If α is a formula, then $(\Box\alpha)$ and $(\Diamond\alpha)$ are formulas. Variables are free or bound in these statements as they are in α.

A <u>statement</u> is a formula in which all occurrences of all variables are bound.
This determines a language $L_{\Box,\Diamond}$. It has a purely classical sublanguage L obtained by omitting all reference to clause 3).
We need the notion of substitution. If we write a formula φ as $\varphi(x)$ for x a variable, and c is a constant, then $\varphi(c)$ is the result of substituting c for all free occurrences of x throughout φ.

For the sake of defining the usual notion of "relational system" in a form exactly appropriate for tableaux, assume that L <u>has no constants itself</u>. Let C be a set of individual constants and extend L to a language L(C) by adding in C to L. An <u>assignment</u> A for L(C) is a map of the atomic statements of L(C) to $\{T, F\}$. Each assignment A is extended uniquely to a <u>valuation</u> V mapping the statements of L(C) to $\{T, F\}$, by the inductive definition below.

0) $V(P) = A(P)$ for all atomic statements P.
1) $V(A \wedge B) = T$ iff $V(A) = T$ and $V(B)$ T.
2) $V(A \vee B) = T$ iff $V(A) = T$ or $V(B) = T$.
3) $V(A \rightarrow B)$ T iff $V(A) \neq T)$ or $V(B) = T$.
4) $V(\neg A) = T$ iff $V(A) \neq T$.
5) $V((\exists x)\varphi(x)) = T$ iff for some constant c of C, $V(\varphi(c)) = T$.
6) $V((\forall x)\varphi(x)) = T$ iff for all constants c of C, $V(\varphi(c)) = T$.

In the notation common in predicate logic, an assigment defines a relational system for L with <u>domain</u> C. This relational system has each relation symbol R of degree n of $L(C)_{\Box,\Diamond}$ denote

$$\{(c, ..., c_n) \in \times^n C : V(R(c_1, ..., c_n)) = T\}.$$

The definition of <u>model</u> $(\mathcal{F}, R)$ for modal predicate logic with constant domains goes as follows. Frames are the same as in propositional calculus, a pair $(\mathcal{F}, R)$ consisting of a non-empty set $\mathcal{F}$ of "possible worlds" and an "accessibility relation" $R \subseteq \mathcal{F} \times \mathcal{F}$. A <u>model</u> is given by a set C of individual constants (the "constant domain") and a map v (the valuation function) assigning to each $w \in \mathcal{F}$ a valuation $v(w)$ of $L(C)$. The definition of "$w \Vdash \varphi$" for statements φ of $L(C)_{\Box,\Diamond}$ is as follows.

0) $w \Vdash P$ for atomic statements P iff $v(w)(P) = T$
1) $w \Vdash A \wedge B$ iff $w \Vdash A$ and $w \Vdash B$.
2) $w \Vdash A \vee B$ iff $w \Vdash A$ or $w \Vdash B$.
3) $w \Vdash A \rightarrow B$ iff $w \Vdash A$ implies $w \Vdash B$
4) $w \Vdash \neg A$ iff not $w \Vdash A$
5) $w \Vdash \Box A$ iff for all w' in $\mathcal{F}$ such that $w R w'$, $w' \Vdash A$.
6) $w \Vdash \Diamond A$ iff for some w' in $\mathcal{F}$ such that $w R w'$, $w, w \Vdash A$.
7) $w \Vdash ((\exists x)\varphi(x))$ iff $w \Vdash \varphi(c)$ for some c in C.
8) $w \Vdash ((\forall x)\varphi(x))$ iff $w \Vdash \varphi(c)$ for all c in C.

The reason these are called "constant domain" models is that the domain C of the relational system assigned to each world is precisely the same. In constant domain models we do not have to worry about any change in denotation of a constant from world to world. The constants are the same in every world and can be thought of as having the same denotation. and even may be thought of as denoting themselves. The big difference between worlds is that the atomic statements $R(c_1, ..., c_n)$ forced in one world may not be forced in another world.

The definition of "valid in a frame" and "valid in a model" and "valid" are as for propositional logic. Using the tableaux before, correctness and completeness are routine.

<u>Constant domain tableaux</u>. We need a countable list of world constants just as in modal propositional logic.. We also need a countable list of new individual constants, to be used in the tableaux to name elements of an intended constant domain. These individual constants are used, as in tableaux for classical predicate logic (Smullyan [1968]) as witnesses for existential quantifiers. Here is the motivation, similar to that for classical logic, but for

frames. Each branch b of a tableaux is viewed an attempt to build a model in which each forcing statement on the branch holds as stated. So the frame would consist of the set $\mathcal{F}$ of world constants w mentioned on b; the constant domain C would be the set of all constants occurring on b; the model based on this frame has the valuation at world w with atomic statement $R(c_1, ..., c_n)$ true iff $Tw \Vdash R(c_1, ..., c_n)$ occurs on b. When a branch b is contradictory in such a tableaux development, this attempt to build a model has failed. When all such attempts to build a model have failed on all branches, we have a tableaux proof.

We add the usual atomic tableaux rules for predicate logic quantifiers (Smullyan [1968]) to those of modal propositional logic. The rules for $T(\forall x)\varphi(x)$, $F(\exists x)\varphi(x)$ are set up to handle the constant domain situation only, since we are allowed to instantiate using <u>any</u> constant already on the branch. We have assumed that $L_{\Box,\Diamond}$ has no constants itself.

<u>Quantifier atomic tableaux for constant domains.</u>

Universal

$Tw \Vdash (\forall x)\varphi(x)$	$Fw \Vdash (\forall x)\varphi(x)$
$\mid$	$\mid$
$Tw \Vdash \varphi(c)$	$Fw \Vdash \varphi(c)$
For any c	For a new c not occurring on any entry above on the branch

Existential

$Tw \Vdash (\exists x)\varphi(x)$	$Fw \Vdash (\exists x)\varphi(x)$
$\mid$	$\mid$
$Tw \Vdash \varphi(c)$	$Fw \Vdash \varphi(c)$
For a new c not occurring on any entry above on the branch	For any c

<u>Example</u>.

1	$F w \Vdash (\forall x)\Box A(x) \rightarrow \Box(\forall x)A(x)$	
2	$T w \Vdash (\forall x)\Box A(x)$	by 1
3	$F w \Vdash \Box(\forall x)A(x)$	by 1
4	$T wRv$	by 3
5	$F v \Vdash (\forall x)A(x)$	by 3
6	$F v \Vdash A(c)$ new c	by 5
7	$T w \Vdash \Box A(c)$	by 2
8	$T v \Vdash A(c)$	by 7
	$\times$	

So $(\forall x)\Box A(x) \rightarrow \Box(\forall x)A(x)$ is provable.

<u>Example.</u> $(\forall x)\neg\Box\varphi \rightarrow \neg\Box(\exists x)\varphi$.

1	$F w \Vdash (\forall x)\neg\Box\varphi \rightarrow \neg\Box(\exists x)\varphi$	
2	$T w \Vdash (\forall x)\neg\Box\varphi$	by 1
3	$F w \Vdash \neg\Box(\exists x)\varphi$	by 1
4	$T w \Vdash \Box(\exists x)\varphi$	by 3
5	$T w \Vdash \neg\Box\varphi(c)$	by 2
6	$F w \Vdash \Box\varphi(c)$	by 5
7	$T wRv$	by 6
8	$F v \Vdash \varphi(c)$	by 6
9	$T v \Vdash (\exists x)\varphi$	by 4
10	$T v \Vdash \varphi(d)$ new d	by 9

This is not a proof. With domain $C = \{c, d\}$, and two worlds w, v, with v accessible from w and no atomic proposition holding in w and $\varphi(d)$ holding in v, we get a counterexample.

$$w \longrightarrow \overset{\varphi(d)}{v}$$

Just as in classical predicate tableaux, the constructions are as helpful for finding counterexamples as for finding proofs. We remark that the same tableaux method applies in case more general situations than constant domains are allowed. But the semantics intended for individual constants then has to be very precisely specified before it becomes obvious what the appropriate tableaux rules for quantifiers are.

§12. **Classical Dynamic Logic.** Hoare [1969] designed a logic for expressing program specifications and for proving "partial correctness" of programs. A basic construct of his logic was A{P}B, meaning that if A holds before the execution of program P, then B holds afterwards (Gries [1981]). Pratt [1976] was motivated by this to develop a modal logic of programs in which each command c in a computer language implemented on the machine is associated with two distinct modal connectives $\Box_c$ and $\Diamond_c$. See Pratt [1976, 1980], Harel [1984], and Kozen and Parikh [1982]. Dynamic logic will be well covered in a forthcoming article by Kozen and Tiuryn [1989] in the new *Handbook of Theoretical Computer Science*, to appear. We take the material in the next two paragraphs from that paper of Kozen and Tiuryn , to which the reader is referred.

Here is a brief explanation. A simple model of sequential computing is that the current state of a sequential machine is determined by an assignment of values in storage locations to variables. Call such an assignment a <u>store</u>. Let $\mathcal{S}$ be the set of all possible stores. Let c be a single command in the language. Corresponding to c introduce a relation $R_c \subseteq \mathcal{S} \times \mathcal{S}$ by the definition that wR_cw' iff

when the store is w and command c is executed,
at the end of execution, the store is w'.

(Of course, c could be a program taking many machine cycles to execute.) Let $\mathcal{S}$ be the set of all stores, let $\mathcal{C}$ be the set of all commands c of the computer language. Define a "multiple modal" propositional logic frame $(\mathcal{S}, \{R_c\}_{c \in \mathcal{C}})$ with an accessibility relation $R_c \subseteq \mathcal{S} \times \mathcal{S}$ for each command c in $\mathcal{C}$. Introduce a modal logic having a modal connective $\Box_c$ and $\Diamond_c$ for each $c \in \mathcal{C}$.

Propositional dynamic logic has propositions of the form $\Box_c\varphi$, $\Diamond_c\varphi$ for program or command c.

$-\Box_c\varphi$ is interpreted as meaning that if <u>any</u> execution of c terminates in a state s, then φ holds at s.

$-\Diamond_c\varphi$ is interpreted as meaning that there us an execution terminating in a state s with φ holding at s.

The connectives mentioned are not the only ones used in propositional dynamic logic. There are additional operations for constructing new commands or programs from old, stemming from the theory of regular events.

<u>Classical propositional dynamic logic.</u>

Syntax

Atomic program letters – (lower case Greek)

Propositional letters – (Upper case Roman)

1) Atomic program letters are programs.

2) If α, β, are programs then so are $\alpha;\beta$, α^*, $\alpha\cup\beta$, $\varphi?$, where φ is a proposition.

3) If α, β are programs and A, B are propositions, then

$A \wedge B$, $A \vee B$, $\neg A$, $\Box_\alpha A$, $\Diamond_\alpha A$ are propositions.

Semantics

A modal frame (Kripke model) consists of a set S of "states" or "possible worlds, together with a set of accessibility relations $\{R_\alpha\}$, one for each atomic program α.

Extend R to all programs by

$R_{\alpha;\beta} = \{(s, t) : (\exists u)((s, u) \in R_\alpha \wedge (u, t) \in R_\beta\}$,

$R_{\alpha\cup\beta} = R_\alpha \cup R_\beta$,

$R_{\alpha^*} = \cup_{n\in\omega} R_\alpha{}^n$.

$R_{\varphi?} = \{(u, u) : u \text{ satisfies } \varphi\}$.

Satisfaction is defined as for ordinary modal logic, except that different accessibilities are used for different programs.

Axiomatization

1) Axioms for propositional logic
2) $\diamond_{\alpha}\varphi \wedge \Box_{\alpha}\pi \to \diamond_{\alpha}(\varphi \wedge \pi)$,
3) $\diamond_{\alpha}(\varphi \vee \pi) \to \diamond_{\alpha}\varphi \vee \diamond_{\alpha}\pi$.
4) $\diamond_{\alpha\cup\beta}\varphi \leftrightarrow \diamond_{\alpha}\varphi \vee \diamond_{\beta}\varphi$,
5) $\diamond_{\alpha;\beta}\varphi \leftrightarrow \diamond_{\alpha}(\diamond_{\beta}\varphi)$,
6) $\diamond_{\varphi?}\pi \leftrightarrow \varphi \wedge \pi$,
7) $(\varphi \vee \diamond_{\alpha}(\diamond_{\alpha^*}\varphi)) \to \diamond_{\alpha^*}\varphi$,
8) $\diamond_{\alpha^*}\varphi \to (\varphi \vee \diamond_{\alpha^*}(\neg\varphi \wedge \diamond_{\alpha}\varphi))$.

Rules of Inference.
Modus ponens.
From A, A $\to$ B, infer B.

Modal generalisation.
From A, infer $\Box_{\alpha}A$ for all programs α.

Classical predicate dynamic logic. Here is a little about classical predicate dynamic logic. In classical first order logic, the truth or falsity of a formula in a relational system is determined as soon as values in the domain of the relational system are assigned to all free variables. In programming environments the values assigned to programming variables vary from stage to stage during the execution of a program. We need a predicate language which can handle changing assignments of values to programming variables for a sufficiently wide class of programs. Within classical logic, the propositional dynamic logic of Pratt, Harel, and Kozen was generalized by them as follows to a predicate modal logic. Let the set S of states be the set of all assignments F which map the set $\{x_i\}_{i\in\omega}$ of program variables into the domain of a relational system M. Each such $\bar{F}$ assigns values to terms of the language. A program can be viewed as inducing a transformation on states. Given an initial state, the program will go through a series of intermediate states, perhaps eventually halting in a final (output) state. In dynamic logic a program is a well-formed expression built inductively from primitive programs using a small set of program constructors which are usually taken to be ; (sequential composition) , * (iteration) and $\cup$ (non deterministic choice) . Dynamic logic interprets these programs semantically as input/output relations on a suitably chosen set of states which makes it a good formalism to describe those properties that manifests in the input/output relations of a program thereby making Dynamic logic undesirable to formalise properties of programs that are not supposed to halt. One is given an input-output specification, a formal relation between the input and output states that the program is supposed to maintain. The input/output relation of a program carries all the information necessary to determine whether the program is correct relative to such a specification. In dynamic logic, programs are first-class objects on a par with formulas, complete with a collection of operators for forming compound programs inductively from a basis of primitive programs. In the case of first order dynamic logic, the atomic programs are taken to be assignment statements

$$x_i <- t,$$

where x_i is a variable and t is a term. The states are taken as set S of total assignments of values in the relational system to the program variables. $R_{x_i <- t}$ denotes

$$\{(F, G) : F, G \in S \wedge G = F(F(t)/x_i)\}$$

The rest is taken from propositional dynamic logic.

§13. **Intuitionistic dynamic predicate logic** (Wijesekera). Note that in classical dynamic logic, propositional or predicate, the "states" are assumed as completely known in order to carry out these valuations. In most actual situations, we have only partial knowledge of the complete state of the machine, say the readings from a few pertinent registers and stacks. What kind of logic can make effective use of "partial knowledge" of states? Wijesekera proposes an intuitionistic system of dynamic logic, and the use of Kripke models, based on partial knowledge of assignments. See Nerode [1990] for explanations as to why Kripke models of intuitionistic reasoning reflect increasing partial knowledge of states. We also wish to be as constructive as possible for another reason. We believe that much more constructive systems have to be developed with term extraction for many of these logics to make them tools for automated reasoning. A beginning has been made by Duminda Wijesekera [1989] in modal intuitionistic logic with two different kinds of accessibilities, one the modal accesssibilities for the problem at hand, one for intuitionistic increase of knowledge. These logics have correctness and completeness theorems. They have been applied to model concurrency by using a constructivised version of Peleg's model of concurrency Peleg [1987]). They have also been applied to give a good intuitionistic dynamic logic with decent term extraction properties. Such features are characteristic of intuitionistic natural deduction systems and not characteristic of their classical counterparts, modal or otherwise. This may prove to be important for implementation as tools in systems such as Constable's NuPRL.

Let K be a Kripke frame for first order intuitionistic logic. Let S be the set of partial maps of assignments into worlds.

Definition. $(F, G) \in R_{x <-t}$ iff
$F, G \in S$ are mapped into the same world in the Kripke model, and
F, G are defined at x,
and $G = F(F(t) / x)$.

Definition. $F \leq G$ iff
1) the world that F is mapped into is below the world that G is mapped into in the intuitionistic partial order, and
2) If $F(x)$ is defined, then so is $G(x)$, and they take the same value.

The following conditions are consequences of the definitions above.

1) If $F \leq G$ and $(F, F') \in R_\alpha$, then there is a G' satisfying $F' \leq G'$, and $(G, G') \in R_\alpha$.
2) If $(F, F') \in R_\alpha$ and $F' \leq G'$, then there is a G such that $F \leq G$ and $(G, G') \in R_\alpha$.

The meaning of the ordinary logical connectives is the same as in intuitionistic logic.

Definition. We say that $w \Vdash \diamond_\alpha \varphi$ if
there is a w' such that
$(w, w') \in R_\alpha$ and $w' \Vdash \varphi$.

Definition. We say that $w \Vdash \Box_\alpha \varphi$ if
whenever $w \leq w'$ and $(w, w'') \in R_\alpha$,
we can conclude that $w'' \Vdash \varphi$.

The notion of satisfaction used here is the one usually called local satisfaction in a frame.

That is, $\Gamma \vdash \varphi$ semantically means that for all w in the frame, $w \Vdash \Gamma$ implies that $w \Vdash \varphi$.

We can prove correctness and completeness relative to this semantics for the following axiomatic systems.

(1) Axioms of Heyting predicate logic.
(2) Scott's axioms of the logic of existence, equality and strictness axioms.
(3) The propositional dynamic logic axioms.

We change the notation for modalities corresponding to programs to the standard notation of dynamic logic for the operators. It is otherwise hard to read the axioms.

Write $<\alpha>$ for $\diamond_\alpha$ and write $[\alpha]$ for $\Box_\alpha$.

$<\alpha;\beta>\varphi \longleftrightarrow <\alpha><\beta>\varphi,$
$<\alpha \cup \beta>\varphi \longleftrightarrow <\alpha>\varphi \vee <\beta>\varphi,$
$<\alpha>(\varphi \vee \theta) \longleftrightarrow <\alpha>\varphi \vee <\alpha>\theta,$
$<\varphi?>\theta \longleftrightarrow \varphi \wedge \theta,$
$<\alpha^*>\varphi \longleftrightarrow \varphi \vee <\alpha><\alpha^*>\varphi,$
$[\alpha : \beta]\varphi \longleftrightarrow [\alpha][\beta]\varphi,$
$[\varphi?]\theta \longleftrightarrow (\varphi \rightarrow \theta),$
$[\alpha^*]\varphi \longleftrightarrow \varphi \wedge [\alpha][\alpha^*]\varphi,$
$\neg<\alpha>\varphi \longleftrightarrow [\alpha](\neg\varphi),$
$(<\alpha>T \rightarrow [\alpha]\varphi) \rightarrow [\alpha]\varphi,$
$(<\alpha>T \rightarrow [\alpha]\varphi) \rightarrow [\alpha]\varphi,$
$<\alpha>\bot \rightarrow \bot,$
$(<\alpha>\varphi \rightarrow [\alpha]\theta_1 \vee [\alpha]\theta_2) \rightarrow [\alpha](\varphi \rightarrow \theta_1 \vee \theta_2).$

Here is what is needed for the deduction theorem.

$[\alpha](\varphi \rightarrow \theta) \rightarrow ([\alpha]\varphi \rightarrow [\alpha]\theta),$
$[\alpha](\varphi \rightarrow \theta) \wedge <\alpha>\varphi \rightarrow <\alpha>\theta,$
$[\alpha]\varphi \wedge <\alpha>(\varphi \rightarrow \theta) \rightarrow <\alpha>\varphi,$
$<x_i \leftarrow a>\varphi(x_i) \longleftrightarrow \exists t_1 t_2[t_1 = x_i \wedge t_2 = a \wedge \varphi(t_2)],$
$[x_i \leftarrow a]\varphi(x_i) \longleftrightarrow \forall t_1 t_2[t_1 = x_i \wedge t_2 = a \rightarrow \varphi(t_2)].$

Rules

Rules of quantification of E^{+} logic.

(This is Scott's logic of partial existence (Troelstra and van Dalen [1988])

Modus Ponens

Substitution

$$\frac{\varphi(x) \quad Et}{\varphi(t/x)}$$

Modal Rules

$$\frac{\Gamma, A \vdash B}{[\alpha]\Gamma, \quad \langle\alpha\rangle A \vdash \langle\alpha\rangle B}$$

$$\frac{\Gamma \vdash A}{[\alpha]\Gamma \vdash [\alpha]A}$$

Propositional dynamic logic rules.

$$\frac{\Gamma \vdash \varphi \rightarrow [\beta][\alpha^i]\theta \quad \text{for all } i}{\Gamma \vdash \varphi \rightarrow [\beta][\alpha^*]\theta}$$

$$\frac{\Gamma \vdash \langle\beta\rangle\langle\alpha^i\rangle\varphi \rightarrow \theta \quad \text{for all } i}{\Gamma \vdash \langle\beta\rangle \langle\alpha^*\rangle \varphi \rightarrow \theta}$$

Intuitionistic concurrent dynamic Logic (Wijesekera). This is a refinement of Peleg's model of concurrency. Here we add an extra program construct $\cap$. This $\alpha \cap \beta$ is supposed to mirror the fact that α and β are executed simultaneously, starting from a common state w. So each program α now denotes an $R_\alpha \subseteq S \times P(S)$, where P(S) is the power set of the set of states.

Definition. We say that $w \vDash \langle\alpha\rangle\varphi$ iff $\exists T \subseteq S$ such that

$(w, T) \in R_{\alpha}$, and

$w' \vDash \varphi$ for each $w' \in T$.

Definition. We say that $w \vDash [\alpha]\varphi$ if for all $w' \geq w$ and all $T' \subseteq S$ such that $(w', T') \in R_{\alpha}$ and all $w'' \in T'$, we have $w'' \vDash \varphi$.

We have to redefine composition and * all over again (see Peleg [1987]).

Definition. $R_{\alpha;\beta}$ is the set of all pairs (w, T) such that

$\exists T' \subseteq S$ with $(w, T') \in R_{\alpha}$, and

for all u in T', there is a T_u such that

$$(u, T_u) \in R_{\beta} \text{ and } T = \bigcup_{u \in T'} T_u.$$

Definition. $R_{\alpha}{}^{*}$ is $\cup_{n \in \omega} R_{\alpha}{}^{n}$.

There is a axiomatization for the concurrent case similar to those we have supplied for the sequential dynamic logic.

§14. Closing note. Other logics have been designed for non terminating and perpetual processes such as operating systems, and for concurrent programs. In temporal logic, the program is fixed and considered part of the structure over which the logic is interpreted. Such a logic is sometimes called an endogenous logic The current location in the program during execution is stored in a special variable for that purpose, called the program counter, and is part of the state along with the values of the program variables. Instead of program operators, there are temporal operators that describe how the program variables, including the program counter, change with time. Temporal logic lacks the ability of dynamic logic to combine programs,and deal with several programs in the same model , but because it deals with execution sequences, temporal logic (and another subject, process logic), can deal with correctness of perpetual programs and programs that sometimes halt, such as operating systems and communication networks. Pnuili [1977] suggested that temporal logics could be used to reason about concurrent programs, when the issue of termination ought to be suppressed from the discussion. Temporal logic began as a formal axiomatic subject (tense logic) in Prior [1955]. Temporal logic also has a natural "possible worlds" Kripke model theory. Syntax and semantics of various temporal logics from a philosophical point of view and without computer science may be found in the excellent texts of Rescher and Urquhart [1971], McArthur [1976], van Benthem [1983]. The 1960's introduced the problem of program specification (what a program is supposed to do), program development (find a program which is supposed to satisfy the specification), and program verification (verify that the program satisfies its specification). Floyd [1967] developed the "inductive assertion method" for verifying that a flowchart program (built up from conditional branching, join of control, and assignment) for computing such a function satisfies "partial correctness" (if the program terminates on an input, the resulting output satisfies the specification). Hoare [1969] turned this into a calculus, much investigated since, based on the construct $\{P\}S\{Q\}$ representing "if the assertion P is true when the program S is initiated, then assertion Q is true if and when the program S terminates." Burstall [1974] developed a method for showing "total correctness" (partial correctness plus the program always terminates). He follows the execution of the program using symbolic (variable) data, using mathematical induction to prove general assertions about what happens at loops. Burstall himself makes the point that modalities are involved. In Burstall's proofs of total correctness, assertions to be proved have the form

"($\exists$time t)(at time t, program line l is executed and $P(t)$)".

In Floyd's proofs of partial correctness, assertions to be proved have the form

"($\forall$ times t)(at time t, program line l is executed implies $P(t)$)".

Pnueli [1977] systematized the modal logic suggested by Burstall as a classical logic augmented by $\Box$, $\Diamond$ corresponding to Burstall's suggestion.

$\Box P(t)$ is read "always" means "now and in the future",

$\Diamond P(t)$ is read "eventually" and means "now or sometime in the future".

He assumed that time is the non–negative integers with the usual order and introduced a third operator $\circ$.

$\circ P(t)$ is read "next P" and means "$P(t+1)$".

These modal logics clarified program correctness proofs, and are equally suitable for concurrent or perpetual programs such as operating systems. An important topic is fairness. This takes many forms. A weak one is that a continuously active process will eventually be scheduled. A stronger requirement is that a process active infinitely often will be scheduled. Another is that a process which is active at least once will be scheduled. All these can be formulated in the Pnueli calculus mentioned, and treated as program specifications. But stronger notions, such as that of two processes, the one that is active first will be scheduled sooner, exceed the capacity of this calculus. Gabbay et al. introduced a binary connective U, vUw, read " u until w", such that

(uUw)(t) is true if v(w) is true at all times w
until a future time s when w(s) is true.

Computer science applications of temporal logic are a thriving specialty in their own right. There are many contributions to the specification and verification of sequential and concurrent systems.

Multiple believers. Now think of "agent 1 believes", "agent 2 believes", etc. We may want all these operators present at once in the same logic. After all, what one agent believes or knows does not necesssarily coincide with what another believes or knows, or with the common beliefs or knowledge of several agents. One objective of such studies is to analyze, model, and machine simulate rational behavior based on knowledge and belief. These applications generally start by putting down reasonable axioms for belief or knowledge, and continue by trying to develop methods of determining whether a given proposition is believed or known on the basis of other propositions. The agents themselves may be machines, and we may be trying to reconcile their databases (beliefs, knowledge).
Hintikka [1962, 1971] gave a Kripke model of beliefs of multiple agents. In its simplest form, there is one set of possible worlds, but a different accessibility relation for each agent, and an agent believes P if P is true at the worlds accessible to the agent. See Halpern and Moses [1985] for a survey of logics of knowledge and belief. Also see Konolige [1986, 1988].

* Supported by NSF grant MCS-83-01850 and ARO contract DAAG29-85-C-0018

** thanks to Prof. Wiktor Marek for §9 and to MSI fellow Duminda Wijesekera for §13, and to Prof. Andre Deutz for his help in the preparation of this paper.

BIBLIOGRAPHY

M. Abadi and Zohar Manna, "Modal theorem proving", LCNS 230, 172–188.

H. Barringer [1985], *A Survey of Verification Techniques for Parallel Programs*, lCNS 191, Springer–Verlag.

R. Brachman, H. Levesque, R. Reiter, eds. [1989], First Conference on Principles of Knowledge Representation and Reasoning, Toronto, Canada, May 15–18,1989.

R. Burstall [1974], "Program proving as hand simulation with a little induction", Information Processing 74, 308–312.

B. F. Chellas [1980], *Modal Logic: An Introduction*, Cambridge University Press.

E. Clark, E. Emerson, A. Sistla, [1986], "Automatic verification of finite state concurrent systems using temporal logic specifications",, ACM Trans. on Programming Languages and Systems, 8, 244–263.

D. McDermott and J. Doyle [1980], "Non–Monotonic Logic I", Artificial Intelligence 13, 41–72.

D. McDermott [1982], "Non–monotonic logic II: Nonmonotonic modal theories", J. ACM 29, 33–57.",

D. Etherington [1988], *Reasoning with Incomplete Information*, Pitman, London, 1988.

R. Fagin and J. Y. Halpern [1988], "Belief, awareness, and limited reasoning", Artificial Intelligence 34, 39–76.

L. Fariñas del Cerro [1985], "Resolution modal logics, logique et analysee" 110/111.

M. Fitting [1983], *Proof Methods for Modal and Intuitionistic Logics*, D. Reidel, Dordrecht, Holland.

R. Floyd [1967], "Assigning meaning to programs", Proc. Symp. in Applied Math., AMS 19, 19–32.

D. Gabbay, A. Pneuli, S. Shelah and J. Stavi [1980], "On the temporal analysis of fairness", Proc. 7th ACM Symp. on Prin. of Programming Languages, 163–173.

A. Galton, ed. [1987], *Temporal Logics and their Applications*, Academic Press, N.Y.

C. Geissler and K. Konolige, "A resolution method for quantified modal logics of knowledge and Belief", in J. Y. Halpern [1986]. 309–324.

M. Gelfond, H. Przymusinska, and T. Przymusinska [1986],"The stable model semantics for logic programming", in R. Kowalski and K. Bowen, eds., Proc. of the 5th Logic Programming Symposium, 1070–1080, Assoc. for Logic Programming, MIT Press, Cambridge, Mass.

M. Gelfond [1987], "On stratified autoepistemic theories", Proc. AAAI–87, Amer. Assoc. for Artificial Intelligence, Morgan–Kaufmann, Los Altos, CA, 1987.

M. Gelfond [1989], "Autoepistemic logic and the formalization of commonsense reasoning", LNCS 346, 176–186.

M. Ginsberg [1987], *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, Los Altos, Calif.

J. Y. Halpern and Y. O. Mose, [1984], "Knowledge and common knowledge in a distributed environment", 3rd ACM Conference on the Principles of Distibuted Computing, 50–61. (Revised as IBM RJ 4421, 1984).

J. Y. Halpern and Y. O. Moses [1985], "A guide to modal logics of knowledge and belief", in Proc. IJCAI, Los Angeles, 50–61.

J. Y. Halpern (Ed.) [1986] "Theoretical aspects of reasoning about knowledge", Proc. 1986 Conf.

J. Hintikka [1962], *Knowledge and Belief*, Cornell University Press, Ithaca, N.Y.

S. Hanks and D. McDermott [1986], "Nonmonotonic logic and temporal projection", Artificial Intelligence 33, 379–412.

D. Harel, [1979], "Dynamic Logic", in Gabbay and Guenthner, eds., *Handbook of Philosophical Logic* [1983/5].

J. Hintikka [1962], *Knowledge and Belief*, Cornell University Press, Ithaca, N.Y.

C. Hoare [1969], "An axiomatic basis for computer programming", Comm. ACM 12, 576–583.

G. E. Hughes and M. J. Cresswell [1968], *Introduction to Modal Logic*, Methuen, London.

S. Kanger [1957], "A note on quantification and modalities", Theoria 23, 133–4.

K. Konolige [1986], *A Deductive Model of Belief*, Morgan Kaufmann, Inc., Los Altos, California.

K. Konolige [1988], "On the relation between default logic and autoepistemic logic", Artificial Intelligence 35, 343–382.

D. Kozen and R. Parikh [1982], "An elementary proof of the completeness of PDL", Theor. Comp. Sci. 14 (1981), 113–118.

D. Kozen and J. Tiuryn [1989], "Logic of programs", Cornell University Department of Computer Science Report no. 89–962.

S. A. Kripke [1959], "A completeness theorem in modal logic", J. S. L. 24, 1–15.

S. A. Kripke [1963], "Semantical considerations on modal logic", Acta Phil. Fennica 16, 83–94.

R. E. Ladner [1977], "The computational complexity of provability in systems of modal propositional calculus", SIAM J. of Computing 6:3 467–480.

A. Loperic, "On the method of valuations in modal logic", Math. Logic Proc. 1st Brazilian Conf., N. Da Costa and R. Chuaqui, ed., Letter Notes in Pure and Applied Math 39, Marcel Dekker, 141–157.

Z. Manna and A. Pnueli [1981], "Verification of concurrent programs: the temporal framework", in R. S. Boyer and J. S. Moore, eds., *The Correctness Prooblem in Computer Science*, Academic Press, London, 215–273.

W. Marek [1986], "Stable theories in autoepistemic logic", to appear in Fundamenta Informaticae.

W. Marek and V. Subrahmanian [1989], "The relationship between logic program semantics and non–monotonic reasoning", Proc. 6th ICLP.

W. Marek and M. Truszczynski [1988], "Autoepistemic logic", Technical report 115–88, Computer Science Department, University of Kentucky.

W. Marek and M. Truszczynski [1989], "Stable semantics for logic programs and default theories", Technical Report 127–88, Computer Science Department, University of Kentucky.

W. Marek and M. Truszczynski [1989], "Relating autoepistemic and default logics", in Brachmann et al. [1989].

R. C. Moore [1985], "Semantical considerations on nonmonotonic logic", Artificial Intelligence 25 (1).

R. C. Moore [1988], "Autoepistemic logic", in Smets et al.

P. H. Morris [1989], "Autoepistemic stable closure and contradiction resolution", LNCS 346, 60–73.

I. Niemalä [1988], "Decision method for autoepistemic logic", Proc. 9th Inter. Conf. on Automated Deduction, Argonne, Ill.

I. Niemalä [1988], "Autoepistemic predicate logic", Research report A–6, Digital Systems Laboratory, Helsinki University of Technology, Espoo, Finland.

D. Peleg [1987], "Concurrent dynamic logic", JACM 34:2, 450–479.

A. Pnueli [1977], "The temporal logic of programs", Proc. 18th IEEE Symp. on Found. of Comp. Science, 46–67.

V. Pratt [1976], "Semantical considerations on Floyd–Hoare logic", 17th Annual IEEE Symp. on Found. Comp. Sci., New York, 109–121.

Pratt [1980], "Applications of modal logic to programming", Studia Logica 39, 257–274.

Prior [1967], *Past, Present, and Future*, Oxford: Clarendon Press.

M. Reinfrank, J. de Kleer, M. L. Ginsberg and E. Sandewall (Eds.) [1989] *Non–Monotonic Reasoning* Berlin: Springer–Verlag

R. Reiter [1980], "A logic for default reasoning", Artificial Intelligence 13 (1–2).

R. Reiter [1987], "Nonmonotonic reasoning", Ann. Rev. Comp. Sci. 2, 147–186.

Y. Shoham [1988], *Reasoning about Change*, MIT Press, Cambridge, MA

G. F. Shvarts [1989], "Fixed–points of non–monotonic modal theories", mss.

P. Smets, E. H. Mamdani, D. Dubois, H. Prade [1989], *Non–Standard Logics for Automated Reasoning*, Academic Press, New York.

R. Smullyan [1968], *First Order Logic*, Springer Verlag, New York.

R. Stalnaker, [1980], "A note on non–monotonic modal logic", Department of Philosophy, Cornell University, Ithaca, N.Y. (unpublished).

A. Troelstra and D. van Dalen [1988], *Constructivism in Mathematics*, vol. 1, 2, North

Holland, Amsterdam.

R. Turner [1984], *Logics for Artificial Intelligence*, Chichester:Ellis–Horwood.

J. van Benthem [1983], *The Logic of Time*, Reidel, Dordrecht.

J. van Benthem [1984], "Correspondence theory," in D. Gabbay and F. Guenther, eds., *Handbook of Philosophical Logic*, vol. II, Reidel, Dordrecht.

J. van Benthem [1985], *Modal Logic and Classical Logic*, Bibliopolis, Napoli.

J. van Benthem [1988], *A Manual of Intensional Logic*, 2nd ed., CSLI, Stanford.

M. Vardi [1985], "On epistemic logic and logical omniscience", in J. Y. Halpern, 293–305.

M. Vardi [1989], "On the complexity of epistemic reasoning", LICS 1989, IEEE, 243–246.

M. Xiwen and G. Weide [1983], "A modal logic of knowledge", 8th IJCAI, 398–401.