Knowledge Representation and Inference Two

Non-Classical Logics

or

Why Classical Propositions and Predicates are not Enough

1 Introduction

These notes are intended to provide sufficient detail for an introduction to the subject of non-classical logics via the description of a selected subset of the many possible different kinds. They assume a basic knowledge of propositional and first-order predicate logic. The idea is not that you should become intimately familiar with all the details presented here, but that you should obtain a clear understanding of the reasons for the different non-classical logics, and how the three kinds discussed below are constructed, and what their properties are.

2 Some Background

The use of logic in knowledge representation systems has received a certain amount of criticism from people working in AI. Various problems with using logic were raised, the most common being:

i) That logic is not expressive enough. In other words, that the knowledge that can be represented using it is too limited.

ii) That logic cannot deal with incomplete, uncertain, imprecise, vague, and/or inconsistent knowledge.

iii) That the algorithms for manipulating knowledge, which derive from logic, are inefficient.

Such criticisms are largely due to a common misconception that *logic* encompasses classical first order propositional and predicate logic only. This, however, is not the case. There are many other logics, most of which were specifically designed to overcome some of the deficiencies of classical logic listed above.

2.1 What is a Logic?

It is important to understand that any system for manipulating knowledge may be regarded as a logic if it contains:

19/1/88

Syntax operation (Turing machine).

a) A well-defined language for representing knowledge.

b) A well-defined model-theory (or semantics) which is concerned with the meaning of the statements expressed in the language. no ambiguity.

c) A proof theory which is concerned with the syntactic manipulation and derivation of statements from other statements.

In other words, a logic consists of a well-defined notation for the representation of knowledge, together with well-defined methods for interpreting and manipulating the knowledge which is represented. The important term here is *well-defined*. Therefore, it would appear that people who criticise logic are either unwittingly or deliberately condoning the use of ill-defined methods for knowledge representation and inference.

2.2 Some Non-Classical Logics (Non-standard).

The subject of non-classical logics is now a large and extensive one. These notes do not attempt to even survey the subject. The aim of these notes is to introduce three which are intended to illustrate the subject, and be more immediately accessible and useful than some of the others tend to be. A list of non-classical logics might include:

epistemic logic
fuzzy logic
higher-order logic
intensional logic
many-sorted logic
many-valued logic
modal logic
non-monotonic logic
situational logic
temporal logic

From this list many-sorted logic, situational logic, and modal logic will be considered in some detail. For further information on any of these you are referred to the bibliography at the end of these notes.

3 Some Distinctions

The terms *non-classical logics* or, as they are also sometimes called, *non-standard logics* are generic terms used to refer to logics other than those based upon classical propositional or predicate calculus. The term non-classical will be used here, rather than non-standard, but they are taken to be synonymous.

Non-classical logics can be divided into two kinds: those that attempt to replace classical logics, called rival logics, and those which extend classical-logics, called extended logics. The first kind includes multi-valued logics, fuzzy logic (if indeed this can be fairly described as a logic since it dispenses with a degree of formality normally associated with logics), and intuitionistic logic, for example. The second kind includes many-sorted logics, modal logics, and temporal logics, for example.

3.1 Rival Logics

Rival logics do not differ from classical logics in terms of the language employed. Rather, they differ in that certain theorems of classical logic are rendered false in the non-classical systems. Probably the most notorious example of this concerns the law of the excluded middle, A or not A. This is provable in classical propositional logic but not in either intuitionistic logic or in any of the standard three-valued logic systems. (See below for more on problems of classical logic.)

$(A \vee \neg A)$

3.2 Extended Logics

Extended non-classical logics sanction all the theorems of classical logic but, generally, supplement it in two ways. Firstly, the languages are an extension of those of classical logic, and secondly, the theorems of these non-classical logics supplement those of classical logic. Usually, such supplementation is provided by the enriched vocabulary. For example, modal logic (of which more latter) is enriched by the addition of two new operators $\Box$, for *it is necessary that*, and $\Diamond$, for *it is possible that*. Under this extension the sentence $\Box A \rightarrow A$ is taken as axiomatic. The addition of such axioms, and appropriate rules of inference involving these operators, facilitates the derivation of theorems which are not even expressible in first order predicate logic, for example.

This division of non-classical logics is not intended to be a definition, just a useful way of characterising two different kinds of non-classical logics.

4 Problems with Deduction in Classical Logics

In AI and Knowledge Representation in particular what *counts* as a deduction, or valid inference, or proof, in a representation language is of central importance. In classical logic deduction is based upon *material implication*, where $A \rightarrow B$ is taken to mean A *materially implies* B. This form of deduction can lead to some paradoxical theorems in classical logic. For example:

i) $p \rightarrow (q \rightarrow p)$

ii) $\neg p \rightarrow (p \rightarrow q)$

iii) $(p \rightarrow q) \vee (q \rightarrow p)$

PARADOXES OF MATERIAL IMPLICATION.

where (i) reads *that a true proposition is implied by anything*, (ii) reads *that a false proposition implies anything*, and (iii) reads *that given any two unrelated propositions, at least one will imply the other*. Non-classical logics have been developed to try to provide improved notions of implication which do not suffer from these problems. For example, the *strict* implications of the Lewis modal logics (of which more latter) were motivated by a desire to present a better account of implication, in that they avoided several of the paradoxical properties of the classical *material* implication since they were not theorems of the new modal logics. However, Lewis's attempts to deal with these problems were not entirely successful, and he seems to have given up on them. Work

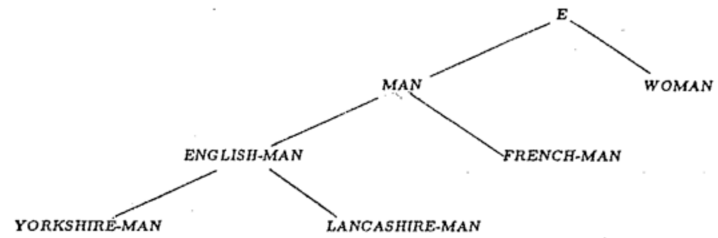
on modal logics subsequently went into decline until AI researchers became interested in improving upon classical logics.

We will start our brief look at the world of non-classical logic by introducing many-sorted logics.

5 Many-Sorted Logics (sorts)

In classical first order predicate logic a relational structure contains a single domain E of entities. Subsets of this domain are defined by use of *unary* (one-place) predicates. In many-sorted logics, the universe of discourse is regarded as comprising a relational structure in which the entities in the domain E are regarded as being of various *sorts*. The sorts are related to each other in various ways to form a *sort structure*. There are different kinds of sort structure:

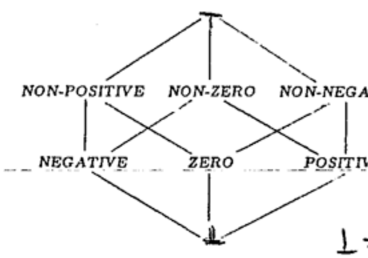
1. Structures in which the sorts are all disjoint. For example, E might contain enterties of the sorts: man, women, bicycle, car.
2. Structures in which the sorts are related in a *subset* tree structure. For example:



3. Structures in which the sorts are related in a lattice. This is the most general sort structure. The following is an example of a lattice type sort structure:

T => Top

Specialisation



Generalisation

⊥ => Bottom

The advantage of dividing the enterties in the domain of a relational structure into different sorts is that it can help to improve the efficiency of automated reasoning by reducing the *search space*.

For example, sorted logics prevent formulas from interacting as freely as they might in an unsorted logic. If the terms of two formulas have no sorts in common then they cannot interact directly. Also, meaningless assertions such as the *ford is married to the mercedes* can be easily detected. It should be noted, however, that many-sorted logics are no more expressive than unsorted logics.

Two kinds of many-sorted logic will now be described.

5.1 Many-Sorted Logics with Restricted Quantification

This kind of many-sorted logic is described in terms of a disjoint sort structure, since it is for this type of domain that it is most suitable.

In classical first order predicate logic universal quantification concerns *every* entity in the relational structure in question. That is, every entity is taken into account when determining the truth value of a universally quantified formula. For example, consider the following formula:

$$\forall X\ [\mathrm{car}(X) \rightarrow \mathrm{numberOfWheels}(X,4) \vee \mathrm{numberOfWheels}(X,3)]$$

This formula, when written in clausal form is:

$$\{\neg \mathrm{car}(X),\ \mathrm{numberOfWheels}(X,4),\ \mathrm{numberOfWheels}(X,3)\}$$

which means that for all entities in the domain, either *e* is not a car, or *e* has four or three wheels.

Instead of using universal quantification, an alternative method of representing the knowledge given above is to use a formula containing a *restricted quantifier*:

FORMULA IN CLAUSAL FORM

SORT SYMBOL

RESTRICTION

SUBSET OF DOMAIN

$$\forall X/\mathrm{car}\ [\underline{\mathrm{numberOfWheels}(X,4) \vee \mathrm{numberOfWheels}(X,3)}]$$

SORT

The restricted quantifier, ∀X/car, ranges over a subset of the domain of the relational structure, ie over only those entities which are a sort of car. The symbol which expresses the restriction on the quantifier (in this example car), is called a *sort symbol*, and the subset of the domain which denotes is called a *sort*.

5.2 Using Restricted Quantification to Improve the Efficiency of Query Answering

Assume we have a *sorted* knowledge base containing the following:

Disjoint sorts: women = {mary, jane}
men = {peter, paul}

Proper axioms: {likes(mary, mary)
likes(mary, jane)
likes(peter, mary)
likes(jane, paul)}

Consider the following query:

$$Q1 = X \mid \forall Y/\mathrm{women}\ \mathrm{likes}(X,Y)$$

which is read as: *find all entries which like all women*. The answer, denoted A1, can be obtained from the completely open query $Q2 = \{X, Y \mid likes(X,Y)\}$, denoted by A2, as follows:

$$A1 = \delta Y/\mathrm{women}\ A2$$

where $\delta Y/women$ is a sorted relational algebraic division operator. We obtain all tuples $\langle X, Y\rangle$ from which likes(X,Y) is true, and then extract those values of X which are related to all values of Y taken from the sort *women*. That is:

A2

X	Y
mary	mary
mary	jane
peter	mary
jane	paul

and

A1 = δY/women

X
mary

giving *mary* as the answer.

Applications of sorted division operators as described above is, in general, more efficient than application of unsorted division operators. Also, certain, *negative* facts need not be stored (the assumption being that, if an entity is not represented as being of a particular sort, then it is assumed not to be of that sort - an extension of the closed world assumption used in first order predicate logic databases). These two factors mean that certain types of query evaluation, and consistency checking can be made more efficient using a sorted logic to represent the knowledge of a domain compared to using an unsorted logic.

5.3 More Expressive Many-Sorted Logics

A sorted logic in which the sortal behaviour of functions and predicates can be defined and in which restricted quantification is *not* used is described next. This logic is described in terms of a lattice sort structure, since this is the kind of sort structure for which it is most suitable.

5.3.1 Sort Lattices

Consider the sort lattice presented earlier. The symbol ⊤ at the top of the lattice is interpreted as the sort containing all entities in the domain of the relational structure. This sort is called *top*. The symbol ⊥ at the bottom of the lattice is interpreted as the empty sort. This sort is called *bottom*.

The sorts immediately above bottom, *NEGATIVE*, *ZERO*, and *POSITIVE*, from the previous example, are ***disjoint*** sorts. Bottom is the most ***specific*** sort. Moving up from bottom to top, the sorts become more general, top being the most general. A sort which is higher in the lattice than another sort, and which is connected to that sort by downward arcs, is called a ***supersort*** of the more specific sort. The more specific sort, in turn, is called a ***subsort*** of the more general sort. The subsort/supersort relationship is denoted by:

$$S1 \subseteq S2$$

indicating that S1 is a subsort or is equal to S2. For example, *POSITIVE* ⊆ *NON-ZERO* in the lattice example given above.

Sorts are related in other ways; for example, by dyadic operators ***lub***, ***glb***, and ***comp***:

a) ***lub*** – least upper bound: The sort S3 = S1 lub S2 is the most specific sort in the lattice which is a supersort of both S1 and S2. For example, in the lattice above, NON-POSITIVE = NEGATIVE lub ZERO. lub is related to the ***union*** operator in set theory.

b) ***glb*** – greatest lower bound: The sort S3 = S1 glb S2 is the most general sort in the lattice which is a subsort of both S1 and S2. For example, in the lattice above, NEGATIVE = NON-POSITIVE glb ZERO. glb is related to the ***intersection*** operator in set theory.

c) ***comp*** – complement: The sort S3 = S1 comp S2 is the sort containing all entities in S1 minus those in S2. For example, in the above lattice, NEGATIVE = NON-POSITIVE comp ZERO.

lub, glb, and comp can be used to define sorts without having to explicitly name them.

5.3.2 Sorting Functions *F*

Associated with every function symbol f in the language of the sorted logic is a ***sorting function*** f, whose purpose is to define the sort of f's output given the sorts of f's inputs. For example, suppose f is a multiply function which takes arguments such that:

Sorts of arguments	Sorts of result
ZERO, ZERO	ZERO
NON-POSITIVE, ZERO	ZERO
NEGATIVE, POSITIVE	NEGATIVE
.	.
.	.
.	.

⊤
SUPERSORT
↓
SUBSORT
⊥

MOST GENERAL
↑ GENERAL
↓ SPECIAL
MOST SPECIAL

The sorting function multiply is then defined by:

multiply(ZERO, ZERO) = ZERO
multiply(NON-POSITIVE, ZERO) = ZERO
multiply(NEGATIVE, POSITIVE) = NEGATIVE
. .
. .
. .

That is, sorting functions map the set of sorts S onto itself. Sorting functions are necessary to accommodate ***polymorphic*** functions; functions which take arguments of different sorts, or types. Sorting functions may also be used to define the sortal behaviour of predicate symbols, but, in this case the sorting functions map the set of sorts S onto the ***boolean*** set {EE, TT, FF, UU} where:

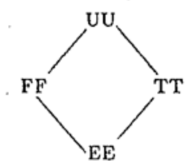
EE means that the atomic formula is ill-sorted; ie the arguments of the predicate are of the wrong sort.

TT means that the atomic formula is well-sorted and is true.

FF means that the atomic formula is well-sorted and is false.

UU means that the atomic formula is well-sorted but that its truth value is not defined.

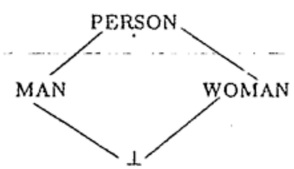
These ***boolean*** sorts also form a lattice:



5.3.3 Well-Sortedness of Functions, Expressions, Atomic Formulas, and Arbitrary Formulas

A function expression is well-sorted iff [1] the sorts of its terms match the sorts required by their respective arguments positions. In some of the many-sorted logics that have been defined, the sort S1 of a term only matches the sort S2 of an argument position if $S1 \subseteq S2$. However, a more expressive logic may be obtained if ***match*** is defined such that a match fails only if $S1 glb S2 = \perp$. For example, consider the function *husbandOf* and the following sort lattice:

SUBSORT



[1] iff denotes if and only if.

Suppose that the sorting function of *husbandOf* is defined such that:

$$\begin{aligned} \text{husbandOf(MAN)} &= \perp \\ \text{husbandOf(WOMAN)} &= \text{MAN} \end{aligned}$$

Suppose, further, that pat is of sort PERSON but it is not known if pat is a man or a woman. From the *more expressive* definition of match, it follows that husbandOf(pat) is well-sorted even though if pat is interpreted as a man then husbandOf(pat) fails to denote.

An atomic formula is well-sorted iff the sorts of its terms match the sorts required by their respective argument positions. For example, consider the predicate *marriedTo*:

$$\begin{aligned} \text{marriedTo(MAN, WOMAN)} &= \text{UU} \\ \text{marriedTo(WOMAN, MAN)} &= \text{UU} \\ \text{marriedTo(MAN, MAN)} &= \text{EE} \\ \text{marriedTo(WOMAN, WOMAN)} &= \text{EE} \end{aligned}$$

If the more expressive definition of match is used then it follows that:

$$\text{marriedTo(PERSON, PERSON)} = \text{UU}$$

Note, however, that when this more expressive definition of *match* is used, the most specific sort of an entity must be used when determining the well-sortedness of the expressions in which it occurs.

An arbitrary formula is well-sorted if there exists an assignment of the sorts to its terms such that:

a) All sub-expressions are well-sorted.

b) The assignment is compatible with the predefined sorts of the constant symbols.

For example, suppose that john and peter are of sort MAN, that mary is of sort WOMAN, and that the marriedTo predicate has the sortall behaviour as defined above. Then, the following formulas are well-sorted:

F1: married(john, mary) $\wedge \neg$married(peter, mary)

F2: $\forall X \forall Y$ [married(X, Y) $\rightarrow$ married(Y, X)]

Note that in this approach restricted quantification is not used. This would reduce the value of allowing polymorphic functions since an instance of a variable would then have a unique sort associated with it. Instead of using restricted quantification, the sorts of variables are determined by the sorts of the argument positions in which they occur. When variables occur as arguments of a polymorphic function or predicate symbol, they may range over several sorts and the sort of the entire formula may then vary as a function of the sorts of such variables. For example, an instantiation of the second formula, F2, above is well sorted if the sort of X is MAN or PERSON and the sort of Y is WOMAN or PERSON, or vice versa. In all other cases the formula is ill-sorted.

atomic formula

(A, B, C) calling # (h, i, j) will set the

Function type sorts as follow h:A; i:B; j:C etc

5.3.4 Using a More Expressive Many-Sorted Logic for Integrity Checking

Suppose that we are constructing a knowledge base using a language of a many-sorted logic as the representation formalism. We could proceed as follows:

a) We begin by defining the sort lattice. This involves naming the sorts and indicating the relationships between them.

b) We then define the sorts of the entities which are to be represented. Errors can be detected if an entity is specified as being of two disjoint sorts.

c) We then define the sorting functions for the required functions and predicates. Errors can be detected if an inconsistent definition is given. For example, the following definition is incompatible with the lattice defined above;

$$\begin{aligned} \text{related(MAN, WOMAN)} &= \text{UU} \\ \text{related(WOMAN, MAN)} &= \text{UU} \\ \text{related(MAN, MAN)} &= \text{UU} \\ \text{related(WOMAN, WOMAN)} &= \text{UU} \\ \text{related(PERSON, PERSON)} &= \text{EE} \end{aligned}$$

d) We then input assertions into the knowledge base. Errors can be detected if formulas are ill-sorted. For example, suppose that the *marriedTo* predicate has the sortal behaviour defined above and that *hasBrother* and *hasBrotherInLaw* are defined as follows:

$$\begin{aligned} \text{hasBrother(MAN, WOMAN)} &= \text{EE ... etc.} \\ \text{hasBrotherInLaw(MAN, WOMAN)} &= \text{EE ... etc.} \end{aligned}$$

Then the following formula is ill-sorted:

$$\forall X \forall Y \forall Z\ [\text{marriedTo}(X, Y) \wedge \text{hasBrother}(Z, Y)] \rightarrow \text{hasBrotherInLaw}(Z, X)$$

5.3.5 Using a More Expressive Many-Sorted Logic to Improve the Efficiency of Automated Reasoning

The efficiency of an automated reasoning system can be improved if it is designed so that it does not attempt to perform inferences with ill-sorted formulas. For example, consider the formula F2 above. If this is converted to clausal form, we get:

$$\{\neg\text{married}(X, Y), \text{married}(Y, X)\}$$

Supposed that pat and jan are both of sort MAN. If the deductive system is designed as suggested it would never generate the following instantiation:

$$\{\neg\text{married(pat, jan)}, \text{married(jan, pat)}\}$$

5.4 Some Concluding Remarks

Although it is well established that many-sorted logics can be used to improve the efficiency of automated reasoning, there is some controversy as to whether this is the best way of achieving such improvements.

In the many-sorted logic approach, the knowledge represented by unary predicates in an unsorted logic is regarded as *meta-knowledge* and held in a sort structure. An alternative approach would be to treat unary predicates, and formulas defining relationships between unary predicates, in a different way to other predicates in implementation rather than in principle. An advantage of the latter approach is that it does not result in the rather messy interface which exists, in many-sorted logics, between the sort structures and formulas. For example, consider the following formula in an unsorted logic:

$$\begin{aligned}\forall X\ [\text{bloodTemp}(X, \text{warm}) \ &\wedge\ \text{numberOfLegs}(X, 4)\\ &\wedge\ \text{skinCovering}(X, \text{fur})\\ &\wedge\ \text{eats}(X, \text{eucalyptusLeaves})\\ &\wedge\ \text{australian}(X)]\\ \rightarrow \text{koalabear}(X)\end{aligned}$$

In defence of the use of many-sorted logic for knowledge representation it does provide a formal way of expressing sort, or type, knowledge about the entities being represented. This is often knowledge that is known, or should be known, at representation, or assertion, time, but which it is often difficult to express in representation languages. A related issue arises in the subject of computer programming languages where some people prefer untyped languages, such as Prolog, while others prefer fully typed languages, such as Standard ML.

SITUATIONS & STATES OF BELIEF.

HAYES FRAME PROBLEM

6 Situational Logic

CHANGE IN UNIVERSE OF DISCOURSE

Classical logics and the many-sorted logics described above are primarily concerned with *static* relational structures. However, for many applications there is a need to be able to represent and reason about a change in universe of discourse. *Situational logic* was developed by McCarthy and Hayes for this type of application.

In situational logic all predicates are given an extra argument which denotes the *situation* in which the formula is true. For example, consider the following formula:

$$\text{on}(b1, b2, s1)$$

This formula states that b1 is on b2 in situation s1. Suppose that b1 and b2 are blocks (inevitably — this is AI after all). In a subsequent situation block b2 might have been moved elsewhere, resulting in the following formula:

$$\neg\text{on}(b1, b2, s2)$$

The transformation of s1 to s2 is assumed to have been caused by an *event*: the event of b2 being moved from being on b1 to somewhere else.

Situations and events are related by a relation R, where R(e, s) denotes the situation which is obtained when event e occurs in situation s. For example, consider the following assertion concerning the movement of blocks:

$$\forall X\ [\text{on}(b1, b2, s) \wedge \neg\text{on}(X, b3, s) \rightarrow \text{on}(b2, b3, R(\text{move}(b2, b3), s))]$$

This is read as, that if b2 is on b1 and no block is on b3, then the new situation denoted by R(move(b2, b3), s), which results from moving a block from b1's tower to b3's tower, will have b1 on b3.

The assertion above adequately describes the relative positions of b1 and b3 in the new situation s' = R(move(b2, b3), s). However, we can infer nothing about the relative positions of all other blocks in s'. A solution to this problem would be to state that a block stays where it is unless it is moved. In general, we would then need to make assertions of the form:

FRAME PROBLEM.

$$\varphi\,[[s]] \wedge \alpha(e) \rightarrow \varphi 2[[R(e, s)]]$$

where φ [[s]] denotes a set of formulas, every situation in which is an occurrence of s. $\alpha(e)$ is a set of formulas which are affected by the event e. Hayes calls such assertions *frame axioms*, or *frame laws*, and refers to the problem of determining adequate collections of such axioms as *the frame problem*. This problem will be referred to again in the lecture on planning.

Other problems arise as a consequence of a changing universe of discourse. Beliefs must change to accommodate a changing world. Consider the following example first presented by Hayes (but de-anthropomorphisised by me).

A robot concludes from a theory, which includes its beliefs as assertions, that it can drive to the airport. However, when it attempts to do so it finds that it has a flat tyre. A human would simply add a new assertion, *a tyre is flat*, to her knowledge base and conclude that she cannot now drive to the airport until it is fixed. Adding the new belief renders an earlier conclusion false even though it was a valid conclusion from the earlier set of beliefs. If the robot is using classical logic, then the only way in which it can make such an amendment to its beliefs is if its earlier conclusion were *if none of the tyres are flat, then it is possible to drive to the airport*. However, there are many potential mishaps which might prevent the robot from driving to the airport and it would be unreasonable to qualify the conclusion with all such possibilities.

QUALIFICATI PROBLEM.

Hayes calls this the *qualification problem* and states that belief logics cannot be expected to obey the monotonicity property of classical logics. In order to overcome the qualification problem associated with monotonic logics, Hayes introduced a new unary connective called *proved* which means *can be proved from the current set of beliefs*. Using this connective, it is possible to write assertions such as:

$$\neg\text{proved flat}(\text{tyres}, s) \rightarrow \text{at}(\text{robot}, \text{airport}, R(\text{drive}(\text{airport}, s))$$

which reads, *if it cannot be proved that any tyres are flat in situation s, then the robot can drive to the airport giving situation R(drive(airport), s), and in this new situation the robot is at the airport.*

Proved could be defined as follows, where α and φ stand for arbitrary sets of formulas:

$$\begin{array}{lcl} \text{R1} & : & \alpha \vdash \text{proved } \alpha \\ \text{R2} & : & \varphi \vdash \alpha \text{ if } \varphi \nvdash \alpha \end{array}$$

Unfortunately, R1 and R2 are inconsistent. Suppose that B $\nvdash$ A but that A is consistent with B. By R2 we can conclude $\neg provedA$ from B. However, if we now add A to B (which we can do without obtaining an immediate inconsistency) then, by R1 we can conclude proved A. A solution to this problem is to tag proved with a belief state marker in a similar way to the way in which predicates are tagged with external situation markers. R1 and R2 then become:

$$\begin{array}{lcll} \text{R1'} & : & \alpha \vdash \text{proved(s) } \alpha\text{(s)} & \\ \text{R2'} & : & \varphi \vdash \neg\text{proved(s)}\alpha\text{(s)}, & \text{where } \varphi \nvdash \alpha\text{(s)}, \\ & & & \text{and every member of } \varphi \text{ has index s} \end{array}$$

Therefore, assertions of the form proved α now have an extra index which identifies the *state of belief* at the time the inference was made.

Use of this extended logic requires that:

a) Whenever R2' is applied, φ contains all assertions with index s.

b) Whenever an assertion is added, every belief index s is replaced by a new one s' except those on *proved* assertions.

These ideas are what led to the development of non-monotonic logics which are the subject of next week's lecture.

7 Modal Logics

STATE OF AFFAIRS

MUST BE
MAY BE

Classical logics and the non-classical logics described so far are called *truth-functional* logics. When we determine consistency or prove theorems in theories of such logics, we consider interpretations each of which assigns a value of true or false to the atomic formulas of the theories concerned. For example, consider the following, where N is some number:

A stands for *N is divisible by eight*
B stands for *N is divisible by four*
C stands for *N is divisible by two*

It is intuitively obvious that A $\rightarrow$ C is a logical consequence of A $\rightarrow$ B and C $\rightarrow$ C, irrespective of what the Number N actually is.

However, suppose that:

A stands for *Reagan was born in France*
B stands for *Reagan speaks French*
C stands for *Reagan speaks French in the White House*

then, in this case, it is *not* intuitively obvious that A $\rightarrow$ C is a logical consequence of A $\rightarrow$ B and B $\rightarrow$ C. In other words, given the two sentences S1 and S2, defined below, it is not reasonable to infer the sentence S3:

S1: *Reagan was born in France implies that Reagan speaks French*

S2: *Reagan speaks French implies that Reagan speaks French in the White House*

S3: *Reagan was born in France implies that Reagan speaks French in the White House*

STATES OF AFFAIRS

This inference is intuitively wrong because the sentences S1 and S2 relate to different states of affairs or *possible worlds*. The first sentence, S1, has to do with a state of affairs in which, other things being as close as possible to the actual state of affairs, Reagan was born in France rather than the USA. The second sentence, S2, has to do with states of affairs in which, other things being as close as possible to the actual state of affairs, Reagan was a French speaking man living in the White House. But the *other things* aren't the same in the two cases, and as a result, the state of affairs that the S1 has to do with do not overlap with those of the sentence S2: if Reagan had been born in France he wouldn't have been President of the USA and thus presumably would not have been living in the White House.

Classical logics and the non-classical logics considered so far cannot accommodate the distinction between states of affairs, or possible worlds, such as those which occur in the example above; neither can they accommodate states of affairs which exist in people's beliefs, moral codes, etc. In order to deal with such things logicians have developed logics called *modal logics*. These can be thought of as logics of *necessity* and *possibility*, or logics of *must be* and *may be*.

7.1 What is a Modal Logic?

Modal logics are distinguished by the use of *modal operators*. A formal feature of modal operators is that they form statements whose truth values are not a function of the truth values of the statement(s) being operated on. For example, consider the following statements:

a) John has appendicitis.

b) It is the case that John has appendicitis.

c) It is possible that John has appendicitis. MODAL

Statements (a) and (b) are not modal. Statement (b) is true iff (a) is true. Statement (c) is modal. It is true if (a) is true but may be interpreted as true or false if (a) is false.

Early work on modal logic was primarily concerned with statements containing the operators *it is necessary that* and *it is possible that* and their negations. Later, logicians considered statements containing modal operators such as:

it will always be the case
it is obligatory that
it is permissible that
it is known that
it is believed that

and so on. Modal logic, then, is concerned with states of affairs or possible worlds in addition to the one that exists.

7.2 Monadic and Dyadic Modal Operators

Monadic modal operators range over single statements. All of the examples above are monadic. Dyadic modal operators form new statements from pairs of statements. Various attempts have been made to formalise a dyadic modal *if … then* operator. Two operators in particular have been defined: *strict implication* and *entailment*. In classical logic the material implication formula P → Q is equivalent by definition to the negation of the conjunction P ∧¬Q. In other words, P materially implies Q iff it is not the case that P is true and Q is false. The strict implication formula:

$P \to Q \models\!\dashv P \wedge \neg Q$

P strictly implies Q

$P \Rightarrow Q \vdash\!\dashv \neg\Box P \wedge \neg Q.$

is, however, equivalent to the *impossibility* of the conjunction P ∧¬Q.

The following sections briefly described a number of types of modality.

7.2.1 Alethic Modality

Alethic modality is concerned with necessity and possibility. The name comes from the Greek word for truth. In the same way that our intuitions demand certain properties of the logical connectives of truth-functional logics, they also demand certain properties of modal operators. For example, adequate systems of alethic modality would be expected to have the following theorems:

AS1	:	if necessary P then possible P
AS2	:	if necessary P then P
AS3	:	if P then possible P
AS4	:	if not possible P then not necessary P
AS5	:	if not P then not necessary P
AS6	:	if not possible P then not P
AS7	:	possible not P iff not necessary P
AS8	:	necessary not P iff not possible P
AS9	:	either possible P or possible not P
AS10	:	not both necessary P and necessary not P

There are various types of alethic modality depending upon the interpretation of *necessary* and *possible*. For example, consider the following statements:

NECESSITY

a) It is necessary that it will snow tomorrow or it will not snow tomorrow. LOCAL

b) It is necessary that a bachelor be male. DEFINITIONAL

c) It is necessary that an action have an equal and opposite reaction. PHYSICAL

The first example is one of logical necessity, the second of *definitional* necessity, and the third of physical necessity.

Modality	operations	
Alethic	necessity	possibility
Temporal	always	sometimes
Deontic	obligatory	permissible
Epistemic	known	believed.

7.2.2 Temporal Modality

The simplest temporal logics interpret the operators *necessary* and *possible* as *always* and *sometimes*. Formulas which are not in the scope of such operators are assumed to be represented in the present state of affairs. The axioms in such logics are the same as in alethic modality, as given above. More complex temporal logics include:

a) Logics with tense operators such as *it has been*, *it will be*, *it has always been*, and *it will always be*. These logics have appropriate sets of logical axioms defined for them.

b) Logics which include time variables as well as variables for entities.

7.2.3 Deontic Modality

Deontic logics contain the modal operators *it is obligatory* and *it is permissible*. They differ from alethic logics in that the ten logical axioms, AS1 to AS10, given above are not all appropriate. Whether or not something happens to be true has no bearing upon whether it is obligatory or permissible from a moral or legal stand point. The axioms AS2, AS3, AS5, and AS6 have no counterparts in deontic logic. However, the following logical axioms should be theorems of any deontic logic:

DS1	:	if obligatory P then permissible P
DS2	:	if not permissible P then not obligatory P
DS3	:	permissible not P iff not obligatory P
DS4	:	obligatory not P iff not permissible P
DS5	:	either permissible P or permissible not P
DS6	:	not both obligatory P and obligatory not P

7.2.4 Epistemic Modality

Epistemic logics are concerned with knowledge and belief. Simple epistemic logics involve modal operators *it is known that* and *it is believed that*. These two operators are not inter-definable as are the operators *necessary* and *possible*. Also, our intuitions vary as to what we mean by *know* and *believe*. However, most simple epistemic logics contain the following logical axioms:

ES1	:	if known P then believed P
ES2	:	if not believed P then not known P
ES3	:	not both known P and not know P

More complex epistemic logics include notions of agents or indexed modal operators which allow them to be used to represent statements like *John knows P*.

7.3 Possible Worlds, Accessibility Relations, and the Notion of Necessity

Rather than discuss each of the above modal logics individually which have a good deal of overlap this section will be based upon the modal operators *necessity* and *possible*. The possible worlds interpretation and the relationship between possible worlds, and why some logical axioms are more appropriate for some types of modal logic are presented. Following that some modal operators are defined more formally and a definition of a modal propositional logic given. A description of the various categories of modal logics which are related to different properties of the *accessibility* relation between possible worlds, is also presented.

The world in which we live is the *actual*, or *real* world. However, it is not the only world we are interested in (indeed some would no doubt say that AI researchers are hardly interested in the real world at all). Often we use the notion of non-real *possible* worlds in our thinking and discussions. For example, we make statements like *if X were the case then it would follow that Y would be true, or possible*, and we discuss such possible world in much the same way as we discuss the real world. Indeed we often make very little attempt to distinguish between the real world and the possible worlds we discuss.

The conjecture of possible worlds is useful in a number of circumstances. For example, if politicians are considering a change in the voting system, then it is useful for them to consider the implications of such a change, ie the possible worlds which could arise if the voting system were really changed.

7.3.1 Types of Possible Worlds

The term *possible world* can be used in different ways to distinguish *possible* worlds from *impossible* worlds. For example:

a) A *logically possible* world might be defined as a world which conforms to the rules of logic. A world in which Edinburgh is or is not the capital of Scotland is logically possible whereas a world in which Edinburgh is and is not the capital of Scotland is a logically impossible world. LOGIC

SYMMETRIC

b) A *physically possible* world might be defined as a world which has the same physical properties as the real world. An example of a physically impossible world is one in which gravitational mass is distinct from inertial mass. Another example of a physically impossible world is one in which particles (other than tachyons that is) can travel faster than the velocity of light. PHYSICS

c) A *morally possible* world might be defined as one in which all laws of some particular moral code are obeyed. MORALS

d) A *conceivably possible* world might be one defined as a world which can be conceived. We might conceive of a world in which there is no concept of colour, for example. CONCEPTUAL

NOT SYMMETRIC

e) A *temporally possible* world might be defined as a world which is at the same time or is in the future of the world under consideration. The definition captures, to some extent, the idea that we cannot go back in time. TIME !?!

The last two examples indicate the need for worlds to be defined as being possible with respect to other worlds rather than absolutely defined. For example:

EXAMPLES

a) Given the above definition of *temporally possible worlds*, a world of 1989 is temporally possible with respect to a world of 1988, but not with respect to a world of 1998.

b) A world W in which there is no notion of colour is conceivably possible with respect to the real world (which has colour for most people). However, the real world would not be conceivably possible with respect to W since the inhabitants of W would have no concept of colour.

7.3.2 Accessibility Relations

A concise way of describing relative possibilities between the worlds in some set of worlds W is to define a binary relation R, called an *accessibility relation*, over W, such that for any wi, wj ∈ W: DEF$^{N.}$

$$\langle wi, wj \rangle \in R \text{ iff } wj \text{ is possible with respect to } wi$$

In other words, a pair of worlds ⟨wi, wj⟩ is a member of the relation R iff wj is possible with respect to wi.

The terminology of relational theory can now be used to describe R. For example:

A→A

a) A relation R is *reflexive* in the set W iff for all wi ∈ W, ⟨wi, wi⟩ ∈ R. In other words, the accessibility relation is reflexive in a set of worlds W iff all worlds in W are possible with respect to themselves. In most of the examples above the accessibility relation is reflexive. However, consider a set of worlds W, each member of which has a moral code. We might define R such that ⟨wi, wj⟩ ∈ R iff wj obeys the moral code of wi. It is most likely that ⟨wi, wj⟩ ∉ R for some wi ∈ W (else why would it be necessary to create moral codes? - if you happen to believe in the necessity of such codes that is). In this case R is not reflexive. EXCEPTION

b) A relation R is *transitive* in the set W iff it satisfies the following condition for all wi, wj, wk ∈ W: A→B B→C

$$\text{if } \langle wi, wj \rangle \in R \text{ and } \langle wj, wk \rangle \in R \text{ then } \langle wi, wk \rangle \in R$$

A→C

The accessibilty relation for temporal possibility defined above is transitive.

c) A relation R is *symmetric* in the set W iff it satisfies the following condition for all wi, wj ∈ W: A→B

$$\text{if } \langle wi, wj \rangle \in R \text{ then } \langle wj, wi \rangle \in R$$

B→A

The accessibility relation for physical possibility as defined above is symmetric, whereas the accessibility relation for temporal possibility is not symmetric.

d) A relation R is *connected* in the set W iff it satisfies the following condition for all wi, wj ∈ W: CONTEXT

$$\text{if } wi \neq wj \text{ then } \langle wi, wj \rangle \in R \text{ or } \langle wj, wi \rangle \in R$$

$A \to A \vdash A \to A$
$A \to B \vdash B \to A$
$A \to B \; B \to C \vdash A \to C$
$\overline{\qquad}$
$A = B$

e) A relation R is an *equivalence* relation in the set W iff it is reflexive, symmetric, and transitive in W. For example, the accessibility relation for *logical* possibility is an equivalence relation.

7.3.3 The Notions of Necessary and Possible Truth

The terms *necessary* and *possible* truth may be informally defined as:

a) A proposition P is necessarily true in a world w iff P is true in *all* worlds which are accessible from w. Necessary truth is denoted by $\Box$.

b) A proposition P is possibly true in a world w iff P is true in at least one world which is accessible from w. Possible truth is denoted by $\Diamond$.

Possible truth can be defined in terms of necessary truth:

$$\text{DF1} : \Diamond P \text{ for } \neg\Box\neg P$$

In other words, a proposition is possibly true iff it is not necessary that it is not true.

As examples of the use of necessary and possible truth, consider the following propositions which we shall assume to be true in the current world. Possible worlds in this example are taken to be temporally possible worlds. In this case, the accessibility relation R is reflexive and transitive, but not symmetric:

EXAMPLES

PA1 : johnSmithIsAlive
read as *John Smith is alive in the current world*

PA2 : $\Diamond$johnSmithIsDead
read as *in the current world or in some future world of the current world John Smith is dead*

PA3 : $\Box$[johnSmithIsDead $\to$ $\Box$johnSmithIsDead]
read as *in the current world and in all worlds in the future of the current world, if John Smith is dead in that world, then he will be dead in all worlds in the future of that world.* ie once John Smith is dead, he will remain dead.

7.4 Special Inference Rules and Logical Axioms for Particular Modal Logics

Alethic modal propositional logic includes (a) all of the machinery of classical propositional logic extended to include the symbols $\Box$ and $\Diamond$, (b) the definition DF1 above, and (c) some additional rules of inference and logical axioms. In this section, we describe one inference rule and one logical axiom which are found in many basic modal logics.

The rule described is called the Gödel rule, or the rule of *necessitation*. This rule is appropriate for most modal logics since it captures the notion that if something is logically true then it is necessarily true. Since most modal systems deal only with possible worlds which are also logically possible, Gödel's rule is found (or can be proved) in most modal logics. The rule is defined as follows:

$$\text{from} \vdash P \text{ infer } \Box P$$

This rule means that if P is a theorem of the modal logic being used, ie a logical axiom, then $\Box$P is also a theorem. Note that this does not mean that if P is a proper axiom of some modal theory T, then $\Box$P is also a theorem of T.

~~In addition to the above rule of inference, many modal logics include the following~~ logical axiom and an extension to classical propositional logic:

LOGICAL AXIOM MODAL

$$\text{LAM} : \Box[P \to Q] \to [\Box P \to \Box Q]$$

That is, if it is necessary that whenever P is true Q is also true, it follows that if P is necessary then Q is necessary. The axiom LAM follows from our intuitive understanding of the notion of necessity.

7.4.1 Additional Logical Axioms for Particular Modal Systems

Basic modal systems can be extended by the addition of logical axioms. Adding these axioms increases the number of valid formulas in such systems and therefore can increase the number of deductions which can be made in their theories. the axioms described below are related to the various properties of the accessibility relation. They can therefore be added, as required, to a basic modal system to construct a logic which is more appropriate for a given domain, or application. For example, if the application has a reflexive and transitive, but not symmetric, accessibility relation, then an appropriate modal system can be constructed from a basic modal system plus the axioms LA1 and LA2 defined below:

a) If R is reflexive then the following axiom is appropriate: LOGICAL AXIOM.

$$\text{LA1} : \Box P \to P$$

That is, if $\Box$P is true in some world w, then P is true in w since w is accessible from itself.

b) If R is transitive then the following axiom is appropriate:

$$\text{LA2} : \Box P \to \Box\Box P$$

That is, if P is true in all worlds wj which are accessible from some world wi, then P is true in all worlds wk, which are accessible from wj.

c) If R is symmetric then the following axiom is appropriate:

$$\text{LA3} : P \to \Box\Diamond P$$

This means that if P is true in some world w, then $\Diamond$P is true in all worlds that are accessible from w.

d) If R is an equivalence relation then the following axioms are appropriate:

$$\text{LA1}: \Box P \rightarrow P$$
$$\text{LA4}: \Diamond P \rightarrow \Box\Diamond P$$

The following example concerns temporally accessible worlds. According to our earlier definition of temporal possibility, the accessibility relation R is both reflexive and transitive. (This need not, however, necessarily be the case for temporal logics).

Since R is reflexive and transitive it is appropriate to use axioms LA1 and LA2 from above. For example, consider a modal theory T which contains the following proper axioms:

$$\begin{array}{lll} \text{TA1} & : & \text{johnSmithIsAlive} \\ \text{TA2} & : & \Diamond\text{johnSmithIsDead} \\ \text{TA3} & : & \Box[\text{johnSmithIsDead} \rightarrow \blacksquare\text{johnSmithIsDead}] \\ \text{TA4} & : & \Box[\text{johnSmithIsAlive} \rightarrow \neg\text{johnSmithIsDead}] \end{array}$$

where TA4 can be read as *in all worlds accessible from the current world, if John Smith is alive then John Smith is not dead.*

Since the accessibility relation being used is reflexive and transitive, the axioms LA1 and LA2, defined above, can be used, together with classical propositional logic, when proving theorems in T. For example, to prove ¬(johnSmithIsDead), we proceed as follows:

$$\text{TA4}: \Box[\text{johnSmithIsAlive} \rightarrow \neg\text{johnSmithIsDead}]$$
$$\Big\downarrow \; \textit{using LA1}$$
$$\text{johnSmithIsAlive} \rightarrow \neg\text{johnSmithIsDead}$$
$$\Big\downarrow \; \textit{using TA1 and modus ponens}$$
$$\neg\text{johnSmithIsDead}$$

in this example we have shown how different modal logics can be constructed by the inclusion or exclusion of appropriate logical axioms. It has also been shown that these axioms relate to the properties of the accessibility relation between the worlds in some set of possible worlds. Thus, appropriate modal logics can be constructed for different applications in which different meanings are ascribed to the term *possible world.* This has resulted in the construction of various types of modal logic such as temporal logic and epistemic logics, for example.

7.5 Modal Properties of Propositions and Formulas

A more formal description of the properties of the modal logics described above will now be presented.

In each possible world, each atomic proposition has one or other of the values *true* or *false*. Consequent upon this other *modal* properties of propositions, which determine the way in which the truth-values of propositions are distributed across the set of all possible worlds can be defined. For example:

a) A proposition is ***possibly true*** iff it is true in at least one possible world. If this possible world is the real world, then the proposition is ***really true***. For example, *John Smith lived for 108 years* is possibly true, and *Reagan is the President of the USA* is really true. (At the time of writing these notes!) POSSIBLY TRUE REALLY TRUE

b) A proposition is ***possibly false*** iff it is false in at least one possible world. POSSIBLY FALSE

c) A proposition is ***contingent*** iff it is true in at least one possible world and false in at least one possible world. For example, the proposition about Reagan in (a) is contingent and happens to be true in the real world. (Again at the time of writing these notes!) CONTINGEN

d) A proposition is ***necessarily true*** if it is true in all possible worlds. The following is an example of a necessarily true proposition: NECESSARI TRUE

John Smith lived 108 years or John Smith did not live 108 years

e) A proposition is ***necessarily false*** if it is false in all possible worlds. For example, John Smith lived 108 years and John Smith did not live 108 years, is a necessarily false proposition. NECESSARILY FALSE

f) A proposition is ***non-contingent*** if it is necessarily true or necessarily false. NON CONTINGENT

Note that the examples presented above are restricted to notions of ***logical possibility*** and ***logical necessity***. As an example of another type of necessity, consider the proposition:

If a good man knows that his neighbour is in difficulty, then he should help his neighbour.

This proposition would be morally necessarily true in a moral modal logic based upon the ten commandments, for example.

7.5.1 Some Symbolisation

In these notes the modal properties of propositions are symbolised as follows:

$\blacklozenge P$	means P is possibly true
$\blacklozenge\neg P$	means P is possible false
$\clubsuit P$	means P is contingent
$\blacksquare P$	means P is necessarily true
$\blacksquare\neg P$	means P is necessarily false
$\blacktriangle P$	means P is non-contingent

7.5.2 Modal Relationships Between Pairs of Propositions

Propositions can be related to each other in various ways. Some of these ways are described below:

a) A proposition P1 is a *contradictory* of a proposition P2 if P1 is false in all possible worlds in which P2 is true and P1 is true in all possible worlds in which P2 is false. For example, the proposition, John Smith lived 108 years, is a contradictory of the proposition, John Smith did not live 108 years.

b) A proposition P1 is a *contrary* of a proposition P2 if although both may be false in some world, both may not be true in any possible world. For example, the proposition, it is Wednesday, and, it is Friday, are contrary propositions. Note that a necessarily false proposition is a contrary of any and every other proposition including itself.

c) Two propositions are *inconsistent* iff they are contradictory or contrary. A pair of propositions which includes a necessarily false proposition is always inconsistent.

d) Two propositions are *consistent* iff they are not inconsistent.

e) A proposition P strictly *implies* a proposition Q iff Q is true in all those possible worlds, if any, in which P is true. For example, the proposition, John is married to Sue, strictly implies the proposition, Sue is married to John. Note, that false propositions may have implications according to this definition. The difference between implication of a false proposition and the implication of a true proposition is that a false proposition has implications some of which may be false, whereas a true proposition has implications all of which are true. Note also, that a necessarily false proposition implies any and every proposition and a necessarily true proposition is implied by any and every proposition. These consequences of the definition of implication given above may appear somewhat counterintuitive, however, arguments can be made for their acceptance. You should try to form some.

f) Two propositions are *equivalent* iff they imply one another, ie P is equivalent to Q iff P is true in all possible worlds in which Q is true and Q is true in all possible worlds in which P is true.

7.5.3 Symbolisation of Relationships

The relationships defined above are symbolised as follows:

$P\ o\ Q$	means P is consistent with Q
$P \varphi Q$	means P is inconsistent with Q
$P \Rightarrow Q$	means P strictly implies Q
$P \Leftrightarrow Q$	means P is equivalent to Q

Note that modal implication denoted by $\Rightarrow$ and modal equivalence denoted by $\Leftrightarrow$ are distinct from *material implication* or *conditionality*, denoted by $\rightarrow$, and *material biconditionality*, denoted by $\leftrightarrow$, in truth-functional logics.

The modal connective $\Rightarrow$, called ***strict implication***, corresponds to ordinary language words such as *if* and *implies* better than the material implication $\rightarrow$ of truth-functional logic. $A \Rightarrow B$ is equivalent to $\Box[A \rightarrow B]$ and therefore avoids some of the more bizarre theorems involving $\rightarrow$.

7.6 The Syntax of a Modal Propositional Logic

The following context-free grammar defines the syntax of a language of modal logic called L3:

terminals	=	$\{p, q, r, s, \ldots, \wedge, \neg, [,], \Box\}$
wff	::=	$\neg$wff | wff $\wedge$ wff | $\Box$ wff | atomic formula
atomic formula	::=	p | q | r | s, ...

The atomic formulas of L3 are denoted by the use of letters such as p, q, r, or by character strings. If P and Q are wffs, then the following abbreviations are also defined for L3:

$P \vee Q$	for	$\neg[\neg P \wedge \neg Q]$
$P \rightarrow Q$	for	$\neg P \vee Q$
$P \leftrightarrow Q$	for	$[P \rightarrow Q] \wedge [Q \rightarrow P]$
$\Diamond P$	for	$\neg\Box\neg P$
$P \Rightarrow Q$	for	$\Box[P \rightarrow Q]$
$P \Leftrightarrow Q$	for	$\Box[P \leftrightarrow Q]$

The following are some examples of wffs of L3 in which strings are used to denote atomic formula:

a) $\Box$[the moon is made of green cheese $\vee$ the moon is not made of green cheese] — read as: it is necessarily true that the moon is or is not made of green cheese.

b) $\Box$all triangles have three sides — read as: it is necessarily true that all triangles have three sides.

c) $\Box$[[John has a child $\wedge$ John is male] $\rightarrow$ John is a father] — read as: it is necessarily true that if John has a child and John is male, then John is a father. Note that this formula may be rewritten as:

[John has a child $\wedge$ John is male] $\Rightarrow$ John is a father

d) $\Diamond$[Reagan was born in France $\rightarrow$ Reagan speaks French] — read as: it is possible that if Reagan were born in France then Reagan would speak French.

e) $\Diamond$[Reagan speaks French $\rightarrow$ Reagan speaks French in the White House] — read as: it is possible that if Reagan speaks French then Reagan speaks French in the White House.

f) $\Box$[N is divisible by 8 $\rightarrow$ N is divisible by 4] — read as: it is necessarily true that if N is divisible by 8 then N is divisible by 4.

g) $\Box$[N is divisible by 4 $\rightarrow$ N is divisible by 2] — read as: it is necessarily true that if N is divisible by 4 then N is divisible by 2.

The modal operators $\Box, \Diamond, \nabla$, and Δ, and the modal connectives $o, \varphi, \Rightarrow$, and $\Leftrightarrow$ are not truth-functional as are the operators $\neg, \vee, \wedge, \rightarrow$, and $\leftrightarrow$. For example, given the truth value of P it is not possible in general to determine the truth value of $\Box$P.

7.6.1 Equivalence Rules

For the purposes of *regularising* modal formulas for subsequent processing, the following equivalence rules can be used in addition to the abbreviation definitions defined above:

$$
\begin{aligned}
P \circ Q &\leftrightarrow \Diamond[P \wedge Q] \\
P \varphi Q &\leftrightarrow \neg\Diamond[P \wedge Q] \\
\nabla P &\leftrightarrow \Diamond P \wedge \Diamond\neg P \\
\triangle P &\leftrightarrow \Box P \vee \Box\neg P
\end{aligned}
$$

For example:

$P \Rightarrow P \varphi Q$ may be rewritten as $\Box[P \rightarrow \neg[P \wedge Q]]$

7.7 The Modal Logics of Lewis

The five modal logics of Lewis, called S1 to S5 are now defined. The logic S5 dates back to Leibnitz but was named S5 by Lewis to indicate its place in the Lewis hierarchy. The modal logics S1 to S5 all contain the same language and rules of inference but differ in their sets of logical axioms. The systems were defined independently of the notion of accessibility relations.

7.7.1 The Languages S1 to S5

All five languages use the language L3 defined above.

Rules of Inference

All five languages use the following rules of inference:

a) Modus ponens for strict implication: given P and $P \Rightarrow Q$, infer Q.

b) Uniform substitution: given P, infer Q where Q is the result of substituting some wff for a propositional variable uniformally throughout P.

c) Conjunction: given P and Q, infer $P \wedge Q$.

d) Replacement of equivalents: given $P \Leftrightarrow Q$ and some propositional context ...P... involving P, infer ...Q..., where Q has replaced P in one or more of its occurrences in the initial context.

Logical Axiom Schemas for S1

$$
\begin{aligned}
&\text{AS1} : P \wedge Q \Rightarrow Q \wedge P \\
&\text{AS2} : P \wedge Q \Rightarrow P \\
&\text{AS3} : P \Rightarrow P \wedge P \\
&\text{AS4} : P \wedge [Q \wedge R] \Rightarrow Q \wedge [P \wedge R] \\
&\text{AS5} : [P \rightarrow Q] \wedge [Q \Rightarrow R] \Rightarrow [P \Rightarrow R] \\
&\text{AS6} : P \Rightarrow \Diamond P
\end{aligned}
$$

Logical Axiom Schemas for S2

Those for S1 plus AS7 : $\Diamond[P \wedge Q] \Rightarrow \Diamond P$.

Logical Axiom Schemas for S3

Those for S1 plus AS8 : $[P \Rightarrow Q] \Rightarrow [\Diamond P \Rightarrow \Diamond Q]$.

Logical Axiom Schemas for S4

Those of S1 plus AS9 : $\Diamond\Diamond P \Rightarrow \Diamond P$.

Logical Axiom Schemas for S5

Those of S1 plus AS10 : $\Diamond P \Rightarrow \Box\Diamond P$.

The modal logics S1 to S5 were defined and categorised before their relationship to accessibility relations and possible worlds was fully appreciated. To some extent they were simply regarded as modal logics in which different theorems could be proven. S5 is said to be *stronger* than S4 and S4 *stronger* than S3, etc. This is because all universally valid formulas of S4 can be derived in S5, and all universally valid formulas of S3 can be derived in S4, and so on.

In 1963 Kripke provided a new understanding of modal logics which related them to properties of accessibility relations discussed earlier. He did this by defining various modal logics as extensions to a basic logic which is variously called M, T, S2', or Feys/Van Wright system. This logic is called M here and corresponds to a reflexive accessibility relationship.

7.7.2 An Axiomatisation of M

M includes any complete axiomatisation for classical propositional logic extended to include $\Box$, plus:

a) The definitions:

$$
\begin{aligned}
&\text{DF1} : \Diamond P \text{ for } \neg\Box\neg P \\
&\text{DF2} : P \Rightarrow Q \text{ for } \neg\Diamond[P \wedge \neg Q] \\
&\text{DF3} : P \Leftrightarrow Q \text{ for } [P \Rightarrow Q] \wedge [Q \Rightarrow P]
\end{aligned}
$$

strict implication

b) The logical rules:

$$
\begin{aligned}
&\text{AS1} : \Box P \rightarrow P \text{ (the reflexiveness axiom)} \\
&\text{AS2} : \Box[P \rightarrow Q] \rightarrow [\Box P \rightarrow \Box Q]
\end{aligned}
$$

c) The inference rules:

R1 : modus ponens for $\rightarrow$
R2 : uniform substitution
R3 : Gödel's rule: from $\vdash P$ infer $\vdash \Box P$

The addition of extra logical axioms to M results in various other modal logics. In particular, Kripke showed that S4 and S5 can be built from M as follows:

a) S4 can be built from M by adding the axiom:

$$\Box P \rightarrow \Box\Box P \text{ (the transitivity axiom)}$$

b) S5 can be built from M by adding the axiom:

$$\Diamond p \rightarrow \Box\Diamond P$$

It should be noted that M, S4, and S5 are not the only modal logics that can be so constructed. Other modal logics can be built which correspond to other types of accessibility relation.

8 Theorem-Proving in Modal Logics

An extension to the tableaux method can be used to prove theorems in modal logics, but this is considered to be beyond the scope of this course. For a presentation of this method see Frost, page 349. The subject of theorem proving for Non-classical logics in general is a major research field within AI and Computing Science. See Thistelwaite et al for an uptodate statement on work in this area.

9 Some Comments on the Logical Modelling of Belief

As a final section to these notes on non-classical logics some work on the modelling of *belief* will be outlined by way of indicating one particular research area in which modal logics are being further developed by application.

The work of Konolige (see bibliography for reference) is concerned with trying to define a *deduction* model belief. A model is an abstract characterisation of the actual object under consideration. It is an abstraction because, for the sake of simplicity, it normally does not have all the properties of the object it is modelling. Models are, therefore, useful for reasoning about a concept, especially if they retain the most important or relevant properties of the concept, while discarding confusing details.

Konolige says that for planning and problem-solving agents there are two important properties of belief which therefore need to be represented by a modal of belief. These are that:

- agents can draw conclusions from an initial set of beliefs, but that
- they do not necessarily derive all logically possible conclusions.

The first of these properties reflects the need for agents to represent facts about their world, and to make inferences from these facts and to draw conclusions from them. However, agents, particularly artificial ones, are also computational devices, and, as

such, they have limitations — constraints upon the time and space available to perform inferences. Thus arises the second property of the belief modal: certain inferences may be logically possiblle, but an agent may not make them.

The best formal models of belief can capture the first property. These models represent the beliefs of an agent as a set of possible worlds using an appropriate modal logic. The possible worlds model is successful in addressing a number of representational issues concerning knowledge and belief, and also as an elegant and concise axiomatization in terms of modal logic. However, a problem with the possible worlds model is that it is inconsistent with the second property defined above. The notion that agents are ideal reasoners (or as Hintikka puts it, are *logically omniscient* is inherent in the analysis of belief in terms of a set of possible worlds, because all logical consequences of an agent's beliefs are also true at each compatible world. Thus, while the possible worlds models are good at predicting what consequences an agent could *possibly* derive from its beliefs, they are not capable of predicting what an agent *actually* believes, given that the agent may have resource or other limitations restricting or preventing the derivation of all the consequences of its beliefs.

The Konolige deduction model of belief was developed in an effort to define accurate models of the beliefs of AI robot planning systems. For these systems, reasoning about the world is an inferential process — that is, they perform syntactic manipulations of the internal language of representation to derive new facts from an original set of beliefs. From a logical point of view such planning systems are often incomplete in their reasoning in exactly the way claimed above in property two; there are simple deductions that are never performed, even with adequate space and time esources. The reasons for this vary, but an important one is that a complete set of deduction rules for an internal language with the expressive power of first order logic is not decidable, so there is no computational procedure that is guaranteed to answer the question of whether a sentence is a consequence of a set of beliefs in a finite amount of time. So system-builders design and build deduction systems that are incomplete, but which are computationally efficient for a particular domain. Prolog is an example of this approach, and some would say a successful one!

Because Konologie's system represents beliefs in a computational, symbol-processing based paradigm, the model is compatible with current philosophical theories of human cognitive states. It should not be supposed, though, that the deduction model gives a completely accurate account of human belief. Our current understanding of human cognitive processes is not even remotely capable of providing formal theories that describe the intricasies of human behaviour. The deduction model is explicitly *not* an attempt to provide such theories, and makes no pretense of being able to model behaviour of this sort. However, the deduction belief model can capture what Konolige claims are the two most important properties of human belief and commonsense reasoning: the fact that people can draw conclusions from their beliefs, and that they do not necessarily derive all the logically possible ones.

The importance of having a *formal* model of belief or other components of an intelligent system is often underestimated. Formal models have the advantage of concreteness: it is possible to prove rigorously what properties the model has and what predictions it makes. By starting with a formal model, we can outline clearly how the model fits or does not fit its intended domain. Since all attempts at representing the world involve abstraction of greater or lesser degree, having the abstraction *on the table* and open to mathematical scrutiny seems to be the only way we can understand, in a precis way, the nature of the abstraction.

For a formalism to be useful in building a working AI system, it must be *heuristically adequate*, that is, there must be some way of efficiently computing useful results from the formalism. As experience with AI systems has shown, a heuristically adequate system can usually be built only when the characteristics of a particular application domain are taken into account.

10 Bibliography

1. Frost, R.A., *Introduction to Knowledge Base Systems*, Collins, 1986.
2. Hughes, G.E. and Cresswell, M.J., *An Introduction to Modal Logic*, Methuen, 1968.
3. Konolige, K., *A Deduction Model of Belief*, Research Notes in Artificial Intelligence, Pitman, 1986.
4. Thistelwaite, P.B., McRobbie, M.A., and Meyer, R.K., *Automated Theorem-Proving in Non-Classical Logics*, Research Notes in Theoretical Computer Science, Pitman, 1988.
5. Turner, R., *Logics for Artificial Intelligence*, Ellis Horwood, 1984.

11 Required Reading:

From Brachman and Levesque, *Readings in Knowledge Representation*:

1. Chapter 2: McCarthy, *Epistemological Problems of Artificial Intelligence*, page 23.
2. Chapter 14: Hayes, *The Logic of Frames*, page 287.
3. Chapter 18: Moore, *The Role of Logic in Knowledge Representation and Commonsense Reasoning*, page 335.

12 Class Work Exercise

- Starting with the example lattice type sort structure given in section 5, page 5, constructed an appropriate lattice sort structure which contains all the relationship between the object that can be built from the following kinds of numbers: Zero, Positive, Negative, Integer, Real, and Complex. For this sort structure define appropriate sorting functions for multiply, divide, addition, and subtraction, functions. Show, by example, that the sorting functions defined are well defined over the range of sorts included in the sort structure.

Briefly explain how you might implement your sort structure and sorting functions in a polymorphic arithmetic package in either Prolog or Lisp.

- Using the definitions, logical axioms, and inference rules defined for the modal logic M, see section 7.7.2, page 27, together with any of those you may require from classical propositional logic, prove that the following are theorems of M:

  1. $\Box P \Leftrightarrow \Box[\neg P \rightarrow P]$
  2. $\Diamond[P \vee Q] \Leftrightarrow [\Diamond P \vee \Diamond Q]$
  3. $[P \wedge \neg P] \Rightarrow Q$

Tim Smithers
January 1988