CHAPTER 1: INTRODUCTION

1.1: What is Nonmonotonic Reasoning?

The goal of Artificial Intelligence (AI) is to improve our understanding of intelligent behavior through the use of computational models. One of the few things researchers in this young science commonly agree upon is the importance of knowledge for intelligence. Thus the study of techniques for representing knowledge in computers has become one of the central issues in AI.

Of course, it would be convenient if we could tell our computers what we want them to know in natural language. But so far this is just a dream. We thus need artificial, formal languages for representing knowledge which can be handled more easily by computers. Formal languages have the advantage of allowing for a much higher degree of precision and clarity than any natural language - admittedly at the cost of flexibility and adaptability.

If our formal knowledge representation languages are to be more than collections of meaningless strings, if they are to represent anything at all, we have to show how expressions and symbols are related to the (part of the) world we want to represent, that is we have to define a semantics for these languages. At this point formal logic plays an important role.

Logic is, first of all, the study of inference. But the perspective of the logician is normative, not empirical or descriptive. The separation between knowledge given in an explicit, declarative form and knowledge which is implicit, that is can be inferred from the given premises, makes it necessary to come up with a criterion for the validity of inferences. This criterion itself must be based in some way on the meaning of the formulae used for expressing the knowledge.

The discussion about the role of logic in AI is as old as AI itself. The relation between human reasoning and the theoretically sound reasoning formalized in logic has always been a matter of debate, particularly in the light of Gödel's famous impossibility theorems. We share the views expressed by Pat Hayes (Hayes 77), Robert Moore (Moore 82) and Wolfgang Bibel (Bibel 84). Logic - and logic here does not necessarily mean classical first order logic - is of fundamental importance to AI since,

besides providing a proof theory, it gives us a clear and precise way of assigning meaning to symbols and of judging the validity of inferences based on this meaning.

We do not claim that logic is the only possible way of doing this, nor do we claim that logic by itself solves the problems of AI. Logic separates valid from invalid conclusions, but it says nothing about what beliefs to adopt in specific situations where limited resources may be available and the cost of computation has to be taken into account. Moreover, logic is based on ontological assumptions (the existence of a domain of identifiable nonchanging objects, the existence of nonchanging relations between these objects) which may not be adequate for some purposes. But it should be clear that whoever proposes a 'nonlogical' representation formalism has to find other, hopefully equally clear and intuitive, ways of assigning meaning to his language and hence providing a criterion for distinguishing between semantically justified and unjustified inferences.

The motivation behind the development of classical logic at the end of the last century was to put mathematical reasoning on a precise formal foundation. Of course, the reasoning of a mathematician trying to establish a mathematical result differs from everyday reasoning. The knowledge we base our decisions on in real life is never as precise and complete as in the ideal setting of this analytical science. We should, therefore, not be astonished that classical logic does not model all forms of everyday reasoning adequately. The main topic of this book is to show how some of the less than ideal forms of human reasoning can be formalized. What we are looking for is a precise mathematical theory of commonsense reasoning.

Classical logic has the following property: if a formula p is derivable from a set of premises Q then p is also derivable from each superset of Q. The reason should be clear: every proof of p from Q is, by the definition of proof in classical logic, also a proof of p from each superset of Q. This property is called the *monotonicity* of classical logic.

To formalize human commonsense reasoning something different is needed. Commonsense reasoning is frequently not monotonic. In many situations we draw conclusions which are given up in the light of further information. The 'canonical' example is the flying ability of birds. If we know that Tweety (one of the most famous animals in AI circles) is a bird, we tend to draw the conclusion that it flies since birds typically fly. Given the information that it is a penguin we certainly withdraw our former conclusion but - and this is important - without withdrawing any of our former premises. We still believe Tweety is a bird and still believe that birds, typically, fly. Such forms of reasoning which allow additional information to invalidate old conclusions are called *nonmonotonic*.

If the notion of nonmonotonic reasoning is understood in a broad sense, then probabilistic reasoning can also be subsumed: additional evidence, obviously, can decrease the conditional probability of a statement in a probabilistic setting. However, probabilistic reasoning - as possibilistic or fuzzy reasoning - has usually been treated

numerically. Numbers are used to represent the degree of plausibility, certainty, confirmation or whatever. The standard use of the term nonmonotonic reasoning is more restricted, being confined to nonnumerical, logic based approaches.

I shall not discuss any of the numerical approaches in this book. This does not mean, however, that I think they are unimportant. Numbers certainly have their advantages. But one of the common views underlying the research in this area is that we should try to find out how far we can get without them. Some interesting recent results, for example in (Geffner, Pearl 88), indicate a close relation between nonmonotonicity and infinitesimal probability. The theory of infinitesimal probabilities might turn out to be one way of providing nonmonotonic formalisms with a semantics. I shall not, however, investigate further this possibility in this book. For a survey of results about the relationship between nonmonotonicity and probability see (Pearl 89) which also contains references to further relevant literature.

Some readers may wonder whether it really is impossible to handle the Tweety example in classical logic. How about the following representation?

(1) $\forall x.\ \text{BIRD}(x) \wedge \neg\text{EXCEPTIONAL-BIRD}(x) \supset \text{FLIES}(x)$

(2) $\forall x.\ \text{EXCEPTIONAL-BIRD}(x) \equiv \text{PENGUIN}(x) \vee \text{DEAD}(x) \vee \ldots.$

(3) $\text{BIRD}(\text{TWEETY})$

However, this makes it necessary to list all possible exceptions explicitly. There are so many unforeseeable circumstances in which something potentially can go wrong with a bird's flying ability that this in itself is an impossible task. And even if we were able to come up with a complete list of exceptional birds the above representation still would be unsatisfactory: we have to show that TWEETY is not a penguin, not dead, etc. in order to derive from these formulae that TWEETY flies.

We would like to be able to derive that TWEETY flies when there is no information that TWEETY is exceptional without having to prove that TWEETY is not exceptional. This is beyond the power of classical logic.

It is interesting to compare this first order representation with a corresponding representation in the programming language Prolog. One feature that distinguishes Prolog from first order logic is the treatment of negation. NOT P is true in Prolog whenever P cannot be derived (negation as failure). This makes the following representation of the Tweety example possible:

```
FLIES(_x) :- BIRD(_x), not EXCEPTIONAL-BIRD(_x).

EXCEPTIONAL-BIRD(_x) :- PENGUIN(_x).

BIRD(TWEETY).
```

Since EXCEPTIONAL-BIRD(TWEETY) cannot be proven Prolog derives NOT EXCEPTIONAL-BIRD(TWEETY). Note that this derivation is not possible in first order

logic from the corresponding set of implications. This together with BIRD(TWEETY) allows us to derive FLIES(TWEETY) via the first rule. If we add

PENGUIN(TWEETY).

then FLIES(TWEETY) can no longer be derived since now EXCEPTIONAL-BIRD(TWEETY) is provable. Prolog, hence, is nonmonotonic.

There have always been AI systems which, like Prolog, were able to draw some sort of nonmonotonic conclusions. An early example was the planning system PLANNER (Hewitt 72) with its nonmonotonic THNOT-operator. This operator, applied to a proposition, failed when the proposition could be proven and succeeded otherwise.

Another type of nonmonotonic system in common use for many years, particular in expert system tools, is the frame system. Frames are representations of object classes consisting of a collection of slot-value pairs. These pairs describe typical values of certain attributes (slots) of members of the particular class. Moreover, the frames form a sub/superclass hierarchy. In case of a conflict the most specific information wins. For example, using the frame language of the expert system tool BABYLON (di Primio, Brewka 85) one might define two frames as follows:

(defframe CAR

(slots (WHEELS 4) (SEATS 5) (CYLINDERS 4)))

(defframe SPORTSCAR

(supers CAR)

(slots (CYLINDERS 6)))

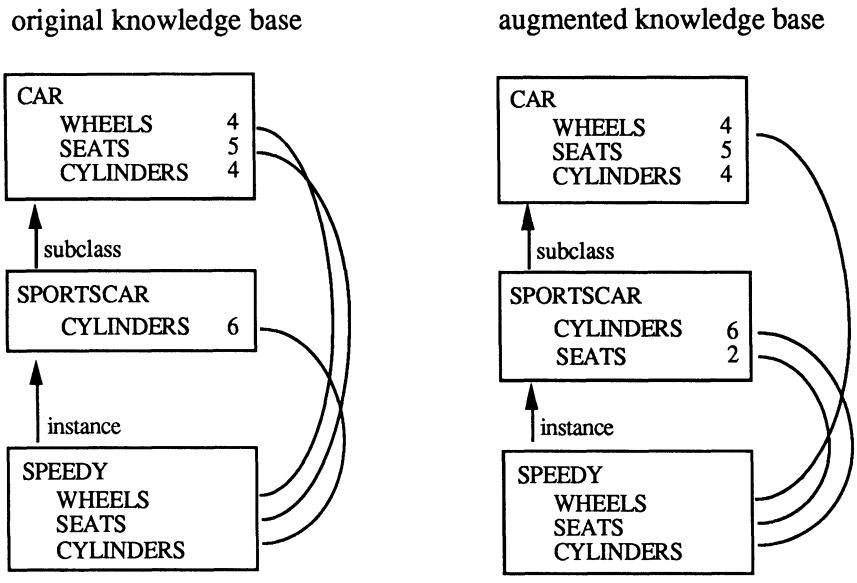
The supers-specification in the second definition states that SPORTSCAR is a subclass of CAR. Given a particular instance of SPORTSCAR, say SPEEDY, we derive that it has 6 cylinders, 5 seats and 4 wheels. If we add information that sportscars typically have 2 seats, i.e. if we change the second frame definition to

(defframe SPORTSCAR

(supers CAR)

(slots (SEATS 2) (CYLINDERS 6)))

then it is concluded that SPEEDY has 2 seats and not 5, an example of the nonmonotonic behaviour of frame systems based on the principle that more specific information is to be preferred. The following diagram shows from where SPEEDY inherits information in each case:



But if such systems and programming tools have been in use already, why was it necessary to think about such complicated things as nonmonotonic logics at all? The answer is that the existing tools have either been too restrictive and could not be generalized without a formal theory, or, when they were more general, they produced results which were not understood well enough. This was, for instance, the case with PLANNER where the user was responsible for avoiding circular dependencies. Such dependencies could lead to groundless belief or non-terminating programs (McDermott, Doyle 80).

We do not want to depend only on the system developer's intuitions. They may be commonly agreed in simple, restricted examples, but in more complicated cases intuition is no longer a sufficient guide. Moreover, the logics will enable us to prove theorems about the behaviour of various systems and tell us how far we can go with extending them. There seems to be no alternative: we have to give our intuitions a formal grounding. We need precise mathematical definitions of nonmonotonic inference which can be given a semantic justification.

1.2: Types of Nonmonotonicity

Nonmonotonic reasoning is not a single phenomenon. Various different types can be distinguished. Let us first have a somewhat closer look at four of them:

1. Default Reasoning

The need for nonmonotonic reasoning arises whenever our knowledge is incomplete and does not allow for the sound derivation of the conclusions necessary to base our decisions, plans and actions on. Of course, in less ideal settings than mathematics this

is the rule, not an exception. Very often we are forced to act in spite of such gaps in our knowledge, i.e. we have to fill those gaps, to 'jump' to conclusions which do not follow logically from what we know ('logically' here is to be taken in the sense of classical logic, of course). These conclusions, then, are less than certain. They may be called beliefs or assumptions. In the light of additional information it may turn out that we have 'jumped' to a wrong conclusion. The conclusion then has to be retracted.

We do not choose blindly among possible extensions of our knowledge. We do not simply flip a coin. There are many cases where our choice can be rationally guided. Much of our experience of the world is available in the form of general rules which are not universally true; they may have exceptions but they express what is true under normal conditions. 'Birds (typically) fly' is one example.

Such rules are very convenient, easy to learn and remember, and they can guide our choices of how to fill gaps in our knowledge when necessary. In the absence of conflicting information, the rule about the flying ability of birds justifies preferring the belief 'Tweety flies' to the belief 'Tweety doesn't fly'. Rules with exceptions are also called defaults and reasoning based on them default reasoning. The conclusions obtained from defaults are less than certain, i.e. default reasoning is a form of plausible reasoning.

Some researchers additionally distinguish between

(1) rules of the form 'An A is typically B' or 'a normal A is B' and

(2) rules expressing statistical facts like 'most A are B'.

They claim that prototypical reasoning based on the first type of rules has nothing at all to do with statistical reasoning (Reiter, Criscuolo 82), (Nutter 87).

This claim seems somewhat overstated. In psychology the term 'prototype' is used to denote an instance, possibly imaginary, of a class of objects with the characteristic properties of the class members. Prototypes can be used for two purposes: we match objects against them in order to decide class membership, and if we know that an object is a member of a certain class we tend to ascribe to it the properties of the prototype.

But what makes properties characteristic? How do we create our prototypes? We still do not understand these phenomena very well. However, the role of prototypes is to enable reasonable guesses to be made. They would certainly not be very useful if they did not lead to good decisions in most cases. This indicates that there must be at least some intricate, possibly very indirect, connection between the notions 'typically' and 'most'. Admittedly, 'most' here has to be understood as relative to a certain context, as 'most amongst those objects we will possibly encounter in every day life'. This excludes all the dead birds that ever lived on earth from consideration: most of these certainly do not fly. We can possibly see a prototype as a compilation of (nonnumerical) probabilistic knowledge into a form which allows for efficient use.

I shall not pursue these issues further here. I use the term 'default' in the sense of a rule with possible exceptions, be the reasons for adopting the rule of a statistical, prototypical, methodological ('working hypotheses'), or decision-theoretic nature (taking the costs of possible errors into account as in the case of the presumption of innocence in law).

As we shall see, the standard formalizations of default reasoning do not syntactically distinguish between safe, irrefutable knowledge and plausible or tentative knowledge. The only way to distinguish between the certain and the defeasible parts of a knowledge base is to inspect the proofs. Consequently, these logics do not model any loss of plausibility when, for instance, long chains of defaults are needed to derive a conclusion. The logics are based on the view that jumping to a conclusion means assuming that the conclusion is true and has the same impact on further derivations as any other premise. In this sense the handling of defaults differs from any probabilistic treatment - unless infinitesimal probabilities are used (Pearl 89).

2. Autoepistemic Reasoning

Assume one of your colleagues asks you whether John McCarthy is going to give a talk at your department next week. You probably say no (unless your department is at Stanford). But nobody told you that there will be no such talk. How did you know the answer? You probably have reasoned along the following lines: if there were a talk there would have been an announcement, or I would have received an electronic mail, or one of my colleagues would have told me about it, or I would have heard about it from somewhere else. Anyway, if there were such a talk I would know about it. But I do not. Hence, there (unfortunately) is no talk given by John McCarthy next week in our department.

Moore (Moore 85) has called this form of reasoning *autoepistemic*, since it involves reasoning about one's own knowledge. Autoepistemic reasoning follows the pattern

I. *If statement x were true I would know it.*

II. *I don't know whether x is true.*

III. *Therefore x is not true.*

It is important that the second antecedent in this pattern is not a premise but can be derived from the knowledge at hand. Otherwise we would not have nonmonotonicity.

Here is the standard example. Let us assume the knowledge base of an agent contains

(1) *If someone is my brother I know it.*

(2) *John is my brother.*

and there is no information in the knowledge base which allows to conclude that Peter is my brother. Then we conclude

Peter is not my brother.

since

I don't know that Peter is my brother.

can, in an intuitive sense to be made precise later, be derived from the premises. However, adding the information

(3) *Peter is my brother.*

to the premises makes, obviously, the previous conclusion impossible.

Autoepistemic reasoning is a form of sound reasoning: after the addition of (3) we know that (1) must have been wrong when we used it to derive *Peter is not my brother*. The wrong conclusion was possible only because the premise was wrong.

What, then, has this to do with nonmonotonic reasoning? Look at (1) again. We are in a position to say that this proposition was wrong with respect to the former knowledge base. But this does not mean that we have to throw (1) away for that reason. It is quite possible (and reasonable to assume) that (1) is absolutely right now, in the new state of knowledge. The meaning of the proposition has simply changed. It refers to our knowledge, and if the knowledge changes, its meaning changes correspondingly.

The nonmonotonicity of autoepistemic reasoning is thus a consequence of the fact that the meaning of statements about one's knowledge is context-sensitive, or - in other words - these statements are indexical. For a more detailed analysis of this type of nonmonotonic reasoning we refer to (Moore 85).

Since plausible default reasoning and autoepistemic reasoning are so very different one would not necessarily expect that a common formalization is possible. Konolige, however, - as we shall see in Section 3.3 - came up with a quite surprising result: default logic, one of the most important formalizations of default reasoning, and the logic developed by Moore for autoepistemic reasoning are equivalent, in a sense to be made precise later. This seems to suggest that different types of reasoning do not necessarily imply different formalizations.

3. Representation Conventions

Assume you want to know whether there is a train from Bonn to Munich at 10.00 a.m. At the station you find a timetable. Let us assume that the timetable mentions no train to Munich leaving at 10.00 a.m. You will conclude that there is no train to Munich at that time. This conclusion is probably not based on a default like 'There is typically no train at 10.00 to Munich'.

You know that railroad officials follow a certain implicit convention: the convention that information about train connections missing from the timetables is simply false, i.e. if there is no train connection mentioned then there is none.[1]

Such conventions are economical and convenient because they make the exchange of information very efficient. The conventions are usually left implicit: there is no extra

[1] Of course, combinations of conventions and defaults are possible and frequent. Our focus here, however, is the *distinction* between different types of commonsense reasoning.

note on the timetable that connections missing from the timetable don't exist. It is assumed that everybody has learned how to use such timetables in the right way.

If, on the other hand, the convention were made explicit, then sound logical reasoning would lead to our conclusion: if there were a 10.00 train to Munich then our 'all trains are there' convention would have been violated, i.e. the premise that connections missing from the timetable do not exist would be false.

Assume that, later, an additional connection between Bonn and Munich at 10.00 is established. The timetable has to be augmented accordingly. This makes our earlier conclusion underivable, but the convention is still the same: this timetable is complete. As in the case of autoepistemic reasoning nonmonotonicity is an effect of the indexical meaning of our convention: 'this timetable' refers to another timetable after the addition of an entry.

Such communication and information storing conventions are very common in the theory of databases and have also been studied formally there. In many cases the *closed-world assumption* (CWA) captures the effects of these conventions:

Definition 1.1: *Let T be a set of formulae. We say that p is derivable from T under the closed-world assumption iff*

$$T \cup ASS(T) \mid\!\sim p$$

where $ASS(T) := \{\neg q \mid q \text{ is atomic and not } T \mid\!\sim q\}$.

The time table from our train example, for instance, can be represented as a set of atomic formulae of the form CONNECTION(x,y,t) stating that there is a train connection from x to y starting in x at time t. Assume CONNECTION(BONN, MUNICH, 10.00) is not contained in this set. Then this formula is underivable from T, the description of the time table. Hence ¬CONNECTION(BONN, MUNICH, 10.00) is in ASS(T) which implies that this formula is derivable from T under the CWA.

Unfortunately, the CWA can lead to inconsistency. If, for instance, T={a v b}, then ¬a as well as ¬b are in ASS(T) which - together with a v b - is inconsistent. This shows that the CWA is not general enough to capture all the interesting cases of nonmonotonic reasoning. See (Genesereth, Nilsson 87) for an overview of various subcases where there is no danger of becoming inconsistent.

Conventions and autoepistemic reasoning certainly are closely related. Knowledge about our own knowledge is often based on knowledge about communication conventions. We know that we would know about a talk of John McCarthy since we know how people communicate. However, the fact that we would know about such a talk itself is not a convention (it may be a consequence of conventions) as it is no convention that we know all our brothers. And there are clearly cases of autoepistemic reasoning having nothing at all to do with conventions (for instance if you believe that your car did not explode since otherwise you would have heard it). It therefore seems justified to treat the two as different types of nonmonotonic reasoning.

4. Reasoning in the Presence of Inconsistent Information

Nonmonotonic reasoning, interestingly, not only arises when the knowledge is incomplete, but also when the knowledge is too complete, i.e. inconsistent. Assume you come home in the evening and find the following short note on the table: *Hi, I went to the cinema with John, the children are visiting Grandma, Peter will visit us on Monday 17th. See you later.* You immediately realize that Monday is not the 17th, i.e. the information at hand is inconsistent. But does that mean that you do not know where the children are?

You probably isolate the inconsistent subpart of the information and remain agnostic with respect to Peter's visit. But you do not throw away the rest. You believe that the children are visiting Grandma and that your husband is in the cinema.

However, if you suddenly remember that Grandma is on holiday in Paris, then this additional information will certainly cause you to withdraw the belief about where the children are. The additional information made other parts of your knowledge inconsistent and conclusions based on these parts are withdrawn.

The example shows that every agent who is able to draw reasonable conclusions based on possibly inconsistent information must reason nonmonotonically.

Another form of reasoning in which some parts of the knowledge have to be disregarded in certain cases is counterfactual reasoning (Lewis 73; Ginsberg 86). A counterfactual is a statement of the form 'if p then q' (denoted $p > q$) where p is known or expected to be false. Typical examples are 'If the electricity hadn't failed, dinner would have been ready on time' or 'If you had thought a little bit harder you wouldn't have made this mistake' or 'If you were not writing this book you could go to the beach with us'. If we were to interpret conditionals as material implications then they would be always true, because their preconditions are false.

We distinguish, however, between true and false counterfactuals. Roughly, the truth of a counterfactual $p > q$ can be determined as follows: add p to your world description. This renders the description inconsistent. Try to find consistent world descriptions which are as similar as possible to this inconsistent description and which contain p. If q holds in all of them, then $p > q$ is true, else it is false.

It often happens that a true counterfactual becomes invalid when additional information is obtained. 'If the electricity hadn't failed, dinner would have been ready on time', for instance, becomes false when we get the additional information that Peter forgot to go shopping.

I shall not give a precise formal account of these intuitive ideas here, especially of the notion of similarity. The reader is referred to (Ginsberg 87) for a discussion of various formalizations and an investigation of computational aspects of counterfactual reasoning.

In this section we have tried to isolate different types of reasoning. In real life they very often appear in combination. For instance, it may be the case that we believe that