THIS PAGE INTENTIONALLY LEFT BLANK

Ordered Presentations of Theories

A Hierarchical Approach to Default Reasoning

Mark D. Ryan

Ph.D. thesis
1992

Author's address:
Department of Computing
Imperial College
London SW7 2BZ
England
E-mail: mdr@doc.ic.ac.uk

To my mother

Georgette Louise

1936–1991

Abstract

The thesis motivates and examines the properties of hierarchies of sentences in a logi where the hierarchy determines how any conflicts between the sentences should b resolved. In the thesis such hierarchies are called *ordered theory presentations* (OTPs). In an OTP, one sentence overrides another if it contradicts it and dominat it in the hierarchy. One of the principal contributions of the thesis is the ability allow such overriding to be partial. Thus, if a sentence in an OTP is dominated b another which contradicts it, those aspects of it which are contradicted are overridde but aspects of it which are not contradicted are preserved. Many properties of OTP are proved, of both a 'static' nature (relating to how conclusions can be drawn fro them) and a 'dynamic' nature (how they can be updated with new information).

OTPs have applications in Artificial Intelligence and Software Engineering. Th thesis concentrates mainly on the applications in AI, where OTPs provide a logi independent framework for representing and reasoning with default information, an for revising belief states with conflicting information. In SE, the topics of defaul and revision occur again in the context of specifications, and the ability to hand them mathematically is presented as an attempt to describe formally such concepts incremental specification and design by difference. These applications are described the thesis.

The machinery introduced for OTPs works for a wide class of logics given in tern of a language, a set of interpretations and a satisfaction relation. The class includ classical, intuitionistic and modal logics. The key definition gives, for each OTP, pre-order on interpretations which orders interpretations according to how well the satisfy the OTP. Models of the OTP are defined to be the interpretations which a maximal in the ordering. Consequences of the OTP are sentences which are true in a its models. Under the natural notion of adding new sentences to an OTP presentatio this consequence relation is *non-monotonic*, which means that the set of conclusion may shrink as the hierarchy of premises is extended. But if the underlying logic compact, the consequence relation retains the property of *weak monotonicity* prevale in the literature.

Contents

Acknowledgements

My supervisor Steve Vickers provided much support, both technical and moral, for which I am very grateful. In spite of a considerable workload of his own, he was always prepared to see me and give consideration to my problems. I am also indebted to others who took a supervisory role, especially Tom Maibaum and Martin Sadler. They have shown great interest in my work which at times I thought was unjustified but which was always very much appreciated. Innumerable friends and colleagues gave me support at various times, but I must single out (alphabetically): Abbas Edalat, José Fiadeiro, Lex Holt, Mark Dawson, and Murray Shanahan. I have also benefitted from discussions with many people at Imperial College and elsewhere, including André Fuhrmann, Anthony Finkelstein, Dov Gabbay, Ian Hodkinson, Johan van Benthem, Frank Veltman, Marcelino Pequeno, Paul Taylor, Pierre-Yves Schobbens, and Samson Abramsky.

The following people kindly read drafts of this thesis and provided valuable feedback: Anthony Finkelstein, Krysia Broda, Mark Dawson, Jonathan Moffett, Murray Shanahan, Paul Taylor, Steve Vickers, and Tom Maibaum.

My family also provided moral support. My father has great confidence in me which albeit sometimes unwarranted has the beneficial effect of spurring me on. My mother also had much faith in me, and her words on the subject of my Ph.D. are one of the reasons that it has come into being. Therefore, and for other reasons, I dedicate it to her. John Finnegan has guarded my sanity.

Chapter 1

Introduction

Logic has been used since antiquity to study correct human reasoning. But until t
last few decades, it had been successful only in representing very precise reasonin
such as that found in mathematics. Logic is appropriate for mathematical reasoni
because conclusions, when they follow from certain premises, do so inexorably. O
never has to deal with conflicting evidence in mathematics; it is always possible
resolve apparent conflicts by further investigation. Furthermore, once a conclusion h
been shown to follow from a certain set of premises, the addition of further premis
in the argument cannot eliminate it.

But recently, a variety of systems have been proposed for aspects of *practical* re
soning. The reasoning humans perform in everyday life does not have the precise ar
exact flavour of mathematical reasoning, but rather is often based on conflicting ev
dence, on assumptions which are known not always to be valid or on prejudices fro
past experience. Much of the motivation for such research has come from artifici
intelligence. Specifically, at least two phenomena have been studied:

Default reasoning. A default sentence is one which expresses a generality or prej
dice but which may be overridden by other, more certain information. Exampl
include *birds can fly* and *tigers have four legs*. Reasoning with defaults mea
being able to use such sentences to draw conclusions, taking account of wheth
they are overridden by other sentences or not. In a more general setting, the
could be a hierarchy of sentences to consider.

Belief revision. This is about incorporating new information about a situation whi
possibly conflicts with the older information an agent already has. The inform
tion an agent has is encoded in its 'belief state'. To incorporate new informatio
successfully, the agent must arrive at another belief state which supports the ne
information while keeping as much of the old as is consistent.

Default reasoning is concerned with a static aspect of reasoning, namely how best
use information to arrive at conclusions. Belief revision, on the other hand, is abo
the dynamics of new information arriving. Nevertheless, there are strong relationshi
between these subjects. These relationships have already been explored by exhibitin
equivalences between properties of default systems and properties of belief revisio
systems [48]. In this thesis we present a framework which treats default reasoni
and belief revision in a uniform way, thus providing a further way to see relationshi
between the systems.

There are other aspects of practical reasoning which can also be handled by the framework of this thesis. We will also discuss:

Prioritised evidence. Suppose there are a number of sources providing information about a particular topic. If the sources contradict each other, but we have an ordering as to their reliability, we may wish to use the ordering to resolve conflicts and get as near to a consensus as possible.

Verisimilitude, or closeness-to-the-truth. Given two descriptions of a situation, can we say that one of them is closer to the truth (or perhaps to a third description) than the other one is?

These topics have also been studied before, and we will compare our results with the existing work. The contribution of this thesis is a uniform framework for handling at least these four topics in practical reasoning, and perhaps others as well.

1.1 Ordered theory presentations

This thesis describes a new way of packaging sentences together to present a logical theory. It turns out that this provides a uniform way of dealing with the kinds of practical reasoning described above. As well as providing a framework for studying the relationships between topics, it also gives better results in the areas mentioned.

- In default reasoning, we get an improved way of expressing defaults and their interrelationships by using the proposed framework (chapter 5).
- In belief revision, we obtain a system which improves on existing systems by allowing *repeated* revision instead of just a single revision step. We depart from two widely accepted postulates of belief revision, however; but this departure is justified with examples (chapter 4).
- In prioritised evidence, we improve on the expressive power of the logic programming setting of D. Vermeir's work [40]. (As this topic is rather small compared with the others it is not dealt with beyond this chapter.)
- In verisimilitude, we get results which more closely match our intuitions (chapter 7).

We propose the concept of an **ordered theory presentation** for dealing with these phenomena. An ordered theory presentation, or OTP for short, is a multi-set of sentences equipped with a partial ordering. The partial ordering is read as 'dominates' or 'overrides'. The exact definition will be given and discussed in chapter 2. The reason for considering *multi*-sets instead of sets is that the same sentence may occur twice in different parts of the ordering. In some circumstances we will impose the restriction that the multi-set be finite.

We will use a graphical notation for representing OTPs. For example, the notation

represents the ordered theory presentation with the three sentences

$$p, \qquad \neg p \wedge q, \qquad \text{and} \qquad r$$

such that r dominates both p and $\neg p \wedge q$. Notice that *sentences lower in the orderi dominate those above*. Thus, the arrow is read as 'dominates'. Neither p nor $\neg p \wedge$ dominates the other. In the following OTP:

$$\left(p \wedge q \quad \neg p \vee r\right)$$

the sentences $p \wedge q$ and $\neg p \vee r$ are incomparable in the ordering. Therefore, they a written side-by-side with no arrows. The large brackets aren't really necessary, b they are useful in delimiting the OTP on the page in the absence of any arrows.

Examples motivating how to reason with OTPs will be given later (§1.3) and t definitions will be given in chapter 2. Examples to show how OTPs can be used f the topics mentioned above are also given later.

The concept of an ordered theory presentation may be seen as an extension of t concept of theory presentation, thus explaining the nomenclature. A theory is a s of sentences closed under logical consequence. A theory presentation is a finite way presenting a theory. Usually it is just a finite set of sentences. It presents the theo obtained by taking its closure under consequence.

As stated, an ordered theory presentation is a finite partially-ordered multi-set sentences. It can be viewed as a theory presentation equipped with a partial order such a way that the same sentence can, if necessary, be present in two different places the ordering. If all the sentences in an OTP are consistent then the theory it presen is just the closure under consequence of that set, analogously to the non-ordered cas But if the sentences are not mutually consistent then the ordering has to be taken in account to arrive at a consistent theory which the OTP presents. The way in which do this is defined in chapter 2 and motivated later in this chapter (§1.3).

From a logical point of view, there are issues for ordered theory presentations whi did not arise in the context of ordinary theory presentations.

- What are the natural ways of adding a sentence to an OTP? What are t properties of these ways?
- What are the natural ways of putting OTPs together to make bigger ones, a again, what are the properties?

These questions will be answered during the course of the thesis.

From a technical point of view, the main question addressed by this thesis is ho the conflicts between the sentences of an OTP are resolved in arriving at the presente theory. In order to answer this satisfactorily we introduce the idea of *degrees of sat faction* between interpretations and sentences. This works as follows. In ordinary logi given a language, an interpretation of the language and a sentence in the language, may say that the interpretation satisfies (or is a model of) the sentence or that it do not. Suppose we have two interpretations which fail to satisfy a sentence. In ord nary logic there is usually nothing more to be said. But using the idea of degrees satisfaction introduced in this thesis we can consider whether one interpretation *mo*

nearly satisfies the sentence than the other. This seems to represent a significant departure from classical logic. The machinery for doing this is introduced in chapter 2, and examples of interpretations satisfying sentences to varying degrees are given.

The remainder of this chapter is organised as follows. In the next section the four examples of practical reasoning described above are presented again, with more detail of how ordered theory presentations can be used in each case. In §1.3 some specific examples of OTPs are given, together the theories they present. These examples are used to build (or test) the reader's intuitions; the definitions are given in the next chapter. In §1.3.1, the criteria which a system for handling OTPs should satisfy are discussed.

1.2 Applications in 'practical' reasoning

1.2.1 Default Reasoning

Default reasoning is about using prejudices (or defaults) about the world to arrive at plausible conclusions in such a way that the conclusions can be withdrawn if evidence to the contrary emerges. To take the most hackneyed example, everyone would accept that *birds can fly*, which we write as

1. $\forall x.\ \big(b(x) \rightarrow f(x)\big)$.

It is undeniably true that *penguins are birds*:

2. $\forall x.\ \big(p(x) \rightarrow b(x)\big)$,

but it is also the case that *penguins cannot fly*, that is

3. $\forall x.\ \big(p(x) \rightarrow \neg f(x)\big)$.

Of course these statements conflict[1]. The world is not contradictory, however, and most people would agree that statement (1) should really say *birds (other than penguins, ostriches and some others) can fly*:

1′. $\forall x.\ \big(b(x) \wedge \neg(\cdots \vee \cdots) \rightarrow f(x)\big)$.

But it is not practical to spell out the exceptions, for almost every premise of interest in everyday reasoning is a generalisation for which it is infeasible to specify all the exceptions. What is needed is a way of seeing the initial set of statements (numbered 1, 2 and 3 above) as a way of presenting a consistent theory. There is an implied way of resolving the conflict, namely that statement (3) should override statement (1) whenever the two conflict. In this particular example, the overriding comes from the *specificity principle*:

> Statements about a specific class of things should override statements about a more general class.

[1] Strictly speaking, there are models in which nothing satisfies the predicate p.

In this case, the specific class is the class of penguins, and the general class is the cla of birds. This fact is represented by sentence 2, which expresses something about th *definition* of penguins. We will have more to say about this principle elsewhere in th thesis. For the present, it is possible to take a perhaps naïve view of this example, whic is that we will obtain intuitively correct results if we simply order the sentences in th presentation according to their relative 'priorities'. We can encode the informatio about this example with the following ordered theory presentation.

Sentence 2 is the strongest—it dominates or overrides 1 and 3, because it is true 'b definition'. Also, 3 overrides 1. The fundamental question of this thesis is: give such an OTP, how can a consistent theory be obtained, which includes as much of th sentences in the OTP as possible, taking account of their ordering? It should be see that this is not a trivial question. For example, one can*not* argue as follows: becau sentence 1 conflicts with 2 and 3 taken together, and because it is weaker than the are in the ordering, we can ignore it. And because 2 and 3 are consistent, the theo which the ordered presentation is intended to denote is given by their conjunction. Th reason that this argument is wrong is that one cannot prove from the resulting theo that birds which are not penguins can fly. In logical terms, we cannot ignore the who of sentence 1; we must retain any 'components' which are consistent with 2 and Exactly what is this notion of 'component' is one of the main questions addressed b this thesis.

There are, of course, hundreds of proposed ways of handling this example whic can be found in the literature [25, 44]. This one is particular because of the explic prioritisation of the sentences involved. The other approaches to default reasoning a classified in chapter 5, where OTPs are compared with other formalisms.

1.2.2 Belief revision

The basic question in belief revision is: how should new information be incorporate into a belief state to result in a belief state which contains the new information an as much of the original belief state as is consistent? The best-known work on th subject is called the AGM theory (after its originators, C. Alchourrón, P. Gärdenfo and D. Makinson), which models belief states as deductively-closed sets of sentence Here is an example from Gärdenfors' book [23, page 1] on the subject[2]:

> Oscar used to believe he had given his wife a gold ring at their wedding. He had bought it from a jeweller who claimed it was made of 24 carat gold, and had taken it to the jeweller next door who had testified to its gold content.

[2] This is not an exact quotation; I have simplified the story slightly.

> However, some time after the wedding Oscar noticed that the sulphuric acid his wife was using in her laboratory stained her ring. He remembered from school chemistry that the only acid that affected gold was aqua regia. So he had to revise his beliefs because they entailed a contradiction. He toyed with the idea that his wife had used aqua regia in the laboratory instead of sulphuric acid, but soon gave up that idea. Having greater confidence in his school chemistry than his own smartness, he concluded the ring was not gold after all. He became convinced that the jewellers had been lying, and guessed they were in collusion with each other.

There are several morals to this story, but we will restrict attention to those that have to do with belief revision. The essential points of the story seem to be:

- Revision (rather than *expansion*) is demanded in the face of inconsistency. (Expansion means just adding beliefs without removing others to keep consistency.)
- There are several ways of doing any particular revision (in the story, Oscar toyed with the alternatives), and the choice of which to do depends on how 'entrenched' other beliefs are. For example, school chemistry was more entrenched than Oscar's belief in his own smartness, in the sense that he is more prepared to give up the latter than the former in the face of inconsistency.
- The new beliefs combine with the remaining old ones to give rise to further beliefs (he concludes that the jewellers were lying), which may themselves carry less than total certainty (he suspects that they were colluding with each other).

All of these ideas will be discussed in chapter 4.

One notable point about Oscar's story is that he revises his beliefs but once. Indeed, the AGM theory of belief revision only handles this kind of one-off revision. Real agents (human or computer) revise their beliefs continually, and the theory we offer is able to model this easily. The reason why the AGM theory can only model single revisions is that the revision functions do not return a fully specified belief state of the kind they demand as an argument. This point will be amply expanded in the chapter, but the crucial problem is that belief states are represented by deductively-closed sets of sentences in the AGM theory.

It will come as no surprise that we advocate representing belief states by ordered theory presentations. To revise an OTP with a contradicting sentence (whether the sentence contradicts the theory presentation or not), simply add the new sentence at the bottom of the presentation. Thus, the presentation directly represents the relative degrees of certainty. New information is placed in the most certain position. Of course, this is not desirable for all kinds of revision, and in the chapter we will attempt to characterise the applications for which this notion of revision is suitable.

A fundamental notion in the topic of belief revision is that of *minimal change*. When revising a belief state, as much of the belief state should persist through the revision as possible. We will show that the AGM theory fails to capture this notion, but that the theory of belief revision based on OTPs scores highly on this point.

The following example of belief revision concerns the understanding of explanations. Explanations are often structured so that broad generalisations are stated first and then more specific information which may contradict the earlier generalisations is given.

Imagine an agent—a human, perhaps, or a robot—which acquires information abo the world in a sequential fashion. As stated, later information may contradict th which was learned earlier, and the agent wants to resolve these conflicts giving priori to the later information. If explaining the operation of a motor car, for example, might say:

1. When you turn the ignition key the starter motor turns the engine.
2. The engine then catches and turns by itself.

Of course this is not the full story, and if you want to know more I might say

3. If the battery is flat, the starter motor won't turn.
4. And if there's no petrol, the engine won't catch.

These latter statements partially override the earlier ones. Taken as a whole, the contradict each other. For example, 1 is supposed to be true in any situation, wheth the sun shines or not, whether it is a weekday or a weekend and whether the batte is flat or not, since none of these are mentioned as exceptions. Sentence 3 contradic this. Similarly, 2 and 4 contradict each other. The way to understand this explanatio and all such explanations for that matter, is as an example of belief revision:

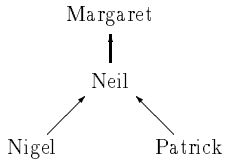
As more information arrives, the agent simply adds it to the end of its belief stat If later information happens to be consistent with earlier information, as sometim happens, then the ordering will be ignored and the conjunction of the sentences w be used.

Note that, as before, we cannot ignore sentences higher in the ordering because the contradict later ones. For example, 1 contradicts 3, but we still have to take accou of 1 when the contradiction does not arise.

1.2.3 Prioritised evidence

In the examples given so far, it has been necessary only to consider linearly ordere theory presentations. This is necessarily the case in belief revision examples, since t presentations are just revision histories. In examples of defaults, however, more co plex structures can be appropriate; this will be seen in chapters 5 and 6. For anoth example of general partial orderings between sentences, consider several advisors wi different degrees of credibility. We imagine a situation in which we are seeking t consensus of four politicians who have opposing points of view, but we have our ow opinion on the relative priorities we should give them and we want to use this to arri

at a conclusion. Our politicians are called Patrick, Neil, Nigel and Margaret. Suppose it is believed that Neil is considered more believable than Margaret, and although no priority is expressed between Nigel and Patrick, they both do better than Neil. These considerations lead to the following order; the arrows mean 'is more believable than'.



The issues of the day are the prospects for the ruling party at the next election, and what is likely to happen to interest rates and inflation. Let r mean that the ruling party is restored to power, i that interest rates increase and f that inflation goes up.

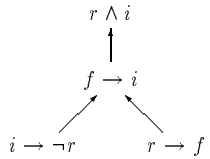
Margaret believes the ruling party will be re-elected, but also that interest rates will rise: $r \wedge i$.

Nigel believes that the party will lose unless interest rates come down: $i \rightarrow \neg r$.

Neil thinks that if inflation is high then interest rates will be high too: $f \rightarrow i$.

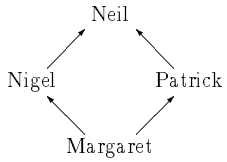
Patrick predicts that re-election of the ruling party will lead to inflation: $r \rightarrow f$.

To take account of our preference between advisors, we have to consider the following presentation:

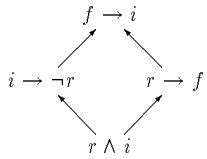


From this OTP we should expect to be able to deduce $\neg r \wedge i$, but nothing about f. To see that this is so, remember that we want to satisfy the constraints which occur lower in the ordering first, and then subject to satisfying those we want to satisfy the higher ones. In this case the three lowest sentences are consistent; their conjunction is $\neg r \wedge (\neg f \vee i)$. Now we want to satisfy the top sentence, or at any rate as much of it as we can. It says $r \wedge i$. We are already committed to $\neg r$, but we can accept the i and conclude $\neg r \wedge i$. In doing so we loose the ability to say anything about f.

That conclusion was based on a particular ordering of our advisors. Now suppose we chose to re-order them according to a new opinion on their credibility. We may decide that we have more confidence in Margaret than we did before. Perhaps we think she's even more honest than Nigel and Patrick. We might re-order the views as follows:



Given this new ordering, the right presentation to use is:



In this case we deduce $r \wedge i \wedge f$, which is the conjunction of the views of Neil, Patrick an Margaret. We cannot accept Nigel's view because it is incompatible with Margaret' and she takes priority in the ordering. Although Neil has been assigned a low priorit we can accept his view because it does not conflict with any view given a higher priorit

The application of ordered theory presentations to prioritised evidence will not b discussed further in the thesis, but it is not as light-hearted or impractical as the read might think. There is an implementation of the definitions for ordered presentation for propositional logic (the code is given in appendix A). On an issue with a larg number of inter-dependent propositions and with a large number of different view I consider that this idea would be a practical aid to gaining a feel for the 'receive opinion'. Of course the conclusion one might reach is much dependent on the orderin of the views, and a well-designed reasoning tool would offer a graphical interface f changing them around.

1.2.4 'Closeness to the truth'

Another outcome of the techniques developed in this thesis is the ability to measure th 'distance' between competing theories to another theory, which may be thought of representing the true situation. This may be illustrated by means of an example to c with the economy. The state of the economy is often described by certain paramete which take numeric values, such as unemployment, inflation, the rate of interest, th gross domestic output, per-capita income, the value of the pound against other cu rencies and so on. Typically, it is desirable for some of these to be high (e.g., domest output, per-capita income) and for others to be low (e.g., interest rates, inflation) whi yet others are best kept within certain bounds (e.g., the value of the pound). One ma simplify the representation of the state of the economy (as politicians are wont to d by considering a family of atomic sentences expressing propositions about the valu of these parameters, like the following:

- u means unemployment is high;
- i, interest rates are high;
- c, per-capita income is high;
- p_1, p_2 and p_3 mean the pound is too low, within acceptable bounds, or too hig

and so on. There may be undisputed relationships between the propositions, such the fact that precisely one of p_1, p_2 and p_3 is true at a time.

Now imagine that we are performing a post-hoc comparison of several economis theories about what would be the case in the economy at the present time. We ha

to hand the truth of the matter, a theory T which says how things actually are. Most probably this will be a *logically complete* theory, that is, one which contains every sentence in the language, or its negation. It need not be complete, however, if not everything about the current state of the economy is known. The ideas being motivated work whether it is complete or not.

The economists' predictions are set out in theories $T_1, T_2, \ldots, T_n$. These will probably *not* be logically complete, and may have any of the usual boolean combinations of the atomic sentences, like conditionals (such as $i \rightarrow u$), disjunctions (e.g. $u \vee c$), negations and so on.

Even if a certain T_i is not the same as (or a superset) of T, it may be closer to it than some other theory T_j. In view of the expressive power of the T_is mentioned, this is not just a matter of comparing sets of atoms. We want T to induce a pre-order on all the theories over the language in question, so that $T_i \leqslant_T T_j$ means that T_j is as close to T as T_i is. We expect certain principles, such as:

$$\text{If } T \subseteq T' \text{ then } T' \text{ is } \leqslant_T\text{-maximal.}$$

For example, if one of our economists predicted as much or more as is known about the present economy, he or she must get full marks.

This application has not yet been fully developed, but the beginnings of it are described in chapter 7 and appendix B. The provisional definitions for theory comparison have also been implemented for propositional logic. Indeed the diagrams given in appendix B were computed by the program. Thus, the idea of using this as a practical means of ranking predictions against a known outcome is not unrealistic.

1.2.5 Software engineering

Finally, many of the ideas mentioned above can be applied to software engineering; a chapter of the thesis is devoted to exploring these issues, although this work is still at an early stage. In software engineering one is interested in specifications and how to construct them. In logical terms, specifications denote theory presentations, and we will advocate in chapter 6 that this be changed to *ordered theory presentations*. This will enable us to deal with

- Specifications involving default information. For example, certain components of the specification may have default characteristics that we wish to accept or override.
- The re-use of components which were specified for a similar (but not identical) purpose to the one at hand.
- Design by difference; that is, a system may be specified as being like another except in certain specifically mentioned respects.
- Fault tolerant systems. These systems have a *normative* behaviour which may be violated if the system goes into a faulty state. We want to specify what happens in these states.

There are relations between these ideas, which will be explored in chapter 6.

1.3 Examples

To recap: an ordered presentation of a theory is a *partially ordered multi-set of se*
tences. It is a 'multi-set' rather than a 'set' because the same sentence may occ
twice, in different places in the order (for example, two of our advisors might say th
same thing).

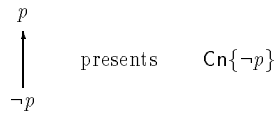
Ordered theory presentations are a simple approach to studying a variety of ph
nomena in practical reasoning. We will see in other chapters how they relate to oth
frameworks for practical reasoning. The good thing about OTPs are that they a
intuitively very simple; it is easy to see what a particular OTP should mean, as th
following examples show.

An informal syntax of graphs for OTPs was used in §1.1, and we will use th
here and indeed in the majority of the thesis. (In §2.2 we will introduce a more form
notation.) We will start with some linear examples from propositional logic and proce
to more general ones, and then consider examples from predicate logic. If Φ is a set
sentences, we write $\mathsf{Cn}(\Phi)$ for the set of consequences of Φ.

This section is intended to illustrate by example the intended behaviour of OTP
The reader can check the examples against his or her intuitions. All of them work o
successfully in the theory described in chapter 2. While reading these examples, it
important to keep the following points in mind:

1. In an OTP, sentences lower in the ordering take precedence over those above.
2. When a sentence lower in the ordering contradicts a sentence above it in th
ordering, the lower sentence overrides the higher one. But in general, this ove
riding is only partial. The lower sentence need not cancel the effect of the high
one completely.
3. The ordering of sentences is a partial ordering. We can have sentences in an OT
which are incomparable in the ordering.
4. In evaluating an OTP (that is, in working out the theory it presents), the idea
to use as much of the available information as possible but to avoid contradiction

Example 1.1



This OTP says: we want $\neg p$ (remember, the bottom sentences are the most important
and, subject to that, we want as much of p as possible. Since p is atomic, we can
extract anything of it which does not conflict with $\neg p$, so all we can deduce is $\neg$
(Later, it will be seen that this analysis is not valid if p is replaced by an arbitrary ϕ
Of course, the partial order is important here. If the two sentences were incomp
rable in the ordering, nothing interesting could be deduced. If the ordering was th
other way around, the ordered presentation would be equivalent to p:

$\left(p \quad \neg p\right)$ presents $\mathsf{Cn}\{\top\}$; $\begin{array}{c}\neg p\\ \uparrow\\ p\end{array}$ presents $\mathsf{Cn}\{p\}$

The notation on the left is the OTP with p and $\neg p$ incomparably ordered. In that case we must remain agnostic about p. On the right, we see that p dominating $\neg p$ is equivalent to p. The idea is to extract what we can from an ordered presentation without allowing contradictions. Notice that this means that an ordered presentation in which all the sentences are incomparable is not the same as the flat presentation formed from the same sentences; for the flat presentation $\{p, \neg p\}$ is equivalent to $\bot$, not $\top$.

Example 1.2

$\begin{array}{c}p \wedge q\\ \uparrow\\ \neg p\end{array}$ presents $\mathsf{Cn}\{\neg p, q\}$

We want $\neg p$, and subject to that, as much of $p \wedge q$ as possible. $p \wedge q$ does conflict with $\neg p$, so we can't have it all. But we can have the q component. Of course the ordering is significant:

$\begin{array}{c}\neg p\\ \uparrow\\ p \wedge q\end{array}$ presents $\mathsf{Cn}\{p, q\}$; $\left(\neg p \quad p \wedge q\right)$ presents $\mathsf{Cn}\{q\}$

Example 1.3

$\begin{array}{c}p \wedge q\\ \uparrow\\ \neg p \vee \neg q\end{array}$ presents $\mathsf{Cn}\{p \leftrightarrow \neg q\}$

This seems similar to example 1.1, since $\neg p \vee \neg q$ is identical to $\neg(p \wedge q)$ in the underlying logic (classical propositional logic in this case). But the analysis given there doesn't scale up to this case. Here, we want $\neg(p \wedge q)$, and subject to that we want as much of $p \wedge q$. What we can have is either p or q but not both.

Example 1.4

$\begin{array}{c}p\\ \uparrow\\ q\\ \uparrow\\ \neg p \vee \neg q\end{array}$ presents $\mathsf{Cn}\{\neg p, q\}$

This is like example 1.3, except now there is a priority expressed between p and q. Th priority is expressed by their location in the ordering. The bottom sentence (the mo important) says that we want one of p and q to fail; but subject to that we want This gives us $\neg p \wedge q$, since they are consistent. Then, subject to all *that*, we want But we've ruled that out by now, so we end up with $\neg p \wedge q$.

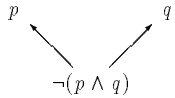
Example 1.5

$\begin{array}{c}p \vee q\\ \uparrow\\ \neg q\end{array}$ presents $\mathsf{Cn}\{p, \neg q\}$

Here, since $p \vee q$ and $\neg q$ are consistent with each other, we can simply have them bo and it doesn't matter how they are ordered:

$\begin{array}{c}\neg q\\ \uparrow\\ p \vee q\end{array}$ presents $\mathsf{Cn}\{p, \neg q\}$; $\left(\neg q \quad p \vee q\right)$ presents $\mathsf{Cn}\{p, \neg q\}$

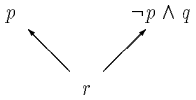
Example 1.6



presents $\mathsf{Cn}\{p \leftrightarrow \neg q\}$

There seems no reason to treat this differently from example 1.3. Therefore one mig ask whether it is in general possible to squash trees into linear orders in this way? Th following example answers this question negatively.

Example 1.7



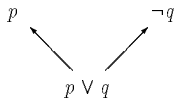
presents $\mathsf{Cn}\{q, r\}$

It is not possible to reduce non-linear partial orders to linear ones by zipping them u with $\wedge$s, since

$\begin{array}{c}\bot\\ \uparrow\\ r\end{array}$ presents $\mathsf{Cn}\{r\}$

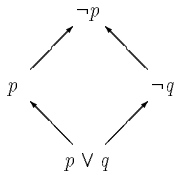
Indeed, the remark that $\left(p \quad \neg p\right)$ presents $\mathsf{Cn}\{\top\}$ while $p \wedge \neg p$ presents $\mathsf{Cn}\{\bot$ in example 1.1 was already an example of this. The intuitions for non-linear parti orders seem to depend on whether the branches share non-logical language or not. Th is important in specification theory applications (§1.2.5 and chapter 6).

Example 1.8



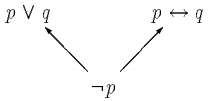
presents $\mathsf{Cn}\{p, \neg q\}$

Example 1.9 Adding $\neg p$ at a higher level cannot affect the outcome.



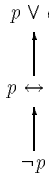
presents $\mathsf{Cn}\{p, \neg q\}$

Example 1.10



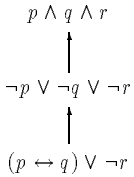
presents $\mathsf{Cn}\{\neg p\}$

Example 1.11 Of course if the defaults in the last example had an order, the situation would be different.



presents $\mathsf{Cn}\{\neg p, \neg q\}$

Example 1.12 This example will turn out to have crucial importance in chapter 4.

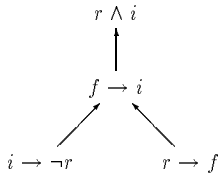


presents $\mathsf{Cn}\{p \leftrightarrow q, p \leftrightarrow \neg r\}$

To see this is correct, separate the cases of r and $\neg r$. If r, then we must have $p \leftrightarrow q$ in order to satisfy the most important sentence (the bottom one). To satisfy the next sentence, we must have $\neg p$ or $\neg q$. Since we already have $p \leftrightarrow q$, this means we have $\neg p \wedge \neg q$. Now we have determined the value of all three atoms, for we have $\neg p \wedge \neg q \wedge r$. On the other hand, if $\neg r$ then both the bottom sentence and the middle one are satisfied. We want as much of the top one as possible, which is $p \wedge q$. Therefore, we get $p \wedge q \wedge \neg r$. The presentation is thus equivalent to $(\neg p \wedge \neg q \wedge r) \vee (p \wedge q \wedge \neg r)$, which is elementarily equivalent to $(p \leftrightarrow q) \wedge (p \leftrightarrow \neg r)$.

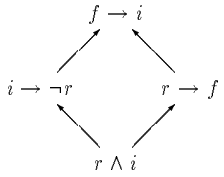
The next two examples were seen in §1.2.3

Example 1.13



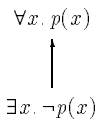
presents $\mathsf{Cn}\{\neg r, i\}$

Example 1.14



presents $\mathsf{Cn}\{r, i, f\}$

Example 1.15



presents $\mathsf{Cn}\{\exists x.\, (\neg p(x) \wedge \forall y.\, (x \neq y \rightarrow p(y)))\}$

The more important sentence (the bottom one) says that there is at least one individu which has not got the property p. But, subject to satisfying that, we want to satisfy much of the upper sentence as possible; it says that all individuals have the proper p. We conclude therefore, that precisely one individual fails p; all the others satis it. As one would expect, different orderings give different results. If the two sentenc $\forall x.\, p(x)$ and $\exists x.\, \neg p(x)$ are incomparable in the ordering (as shown below), then o can conclude that there is one element whose claim to the property p is disputed, b that all other elements have the property p.

$$\Big(\forall x.\, p(x) \quad \exists x.\, \neg p(x)\Big) \quad \text{presents} \quad \mathsf{Cn}\{\exists x.\, \forall y.\, (x \neq y \rightarrow p(y))\}$$

1.3.1 Criteria for the definitions for OTPs

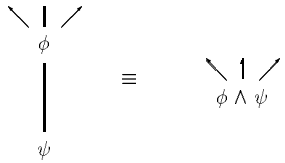
The examples above serve as a benchmark for the development of the system for dealin with ordered presentations given in this thesis. Some of the ideas mentioned there ar

1. Sentences lower in the ordering override those higher. But the overriding is on *partial* (examples 1.2, 1.15, and others).

2. We should be able to handle arbitrary partial orders.

3. Inclusion of the sentence lowest in the ordering (if there is one) is a minimal requirement on the theory being presented.

4. If two sentences are at the bottom an OTP and are consistent, it doesn't matter how they are ordered (example 1.5). Graphically: if $\phi \wedge \psi \neq \bot$ then



That is to say, these two graphs present the same theories; $\equiv$ is formally defined in §3.3. We do not expect this to extend to the case that ϕ and ψ are inconsistent (example 1.1) or are not at the bottom of the OTP (example 1.4).

Other requirements which we may add are:

5. There should be no 'hacks' to the connectives. The system we define for handling OTPs should not change the meanings of the connectives or interfere with the mechanism of the underlying logic.

6. The system should be defined as independently of the underlying logic as possible. For example, substitution of logical equivalents at any point of an ordered presentation should not change its meaning, as mentioned in the discussion of example 1.3. We would like to define the behaviour of ordered theory presentations over *any* logic meeting certain minimal requirements. These requirements on the underlying logic will be spelled out in chapter 2.

7. Infinite OTPs should be allowed, provided there are no infinite descending chains. Such an OTP would mean stronger and stronger sentences overriding earlier ones without any means of establishing what is ultimately wanted, which is clearly counterintuitive. On the other hand, weaker and weaker sentences do not appear to pose a problem.

1.4 Related work

There is no single chapter covering related work in this thesis. Discussion of related work is contained in chapters 4, 5, 6, and 7.

1.5 Outline of the rest of the thesis

The definitions and principal results concerning ordered presentations of theories are set out in chapters 2 and 3. Chapter 4 considers their application to the topic of belief revision, and comparisons are drawn with the standard work in that topic. Chapter 5 compares OTPs with other frameworks for reasoning with defaults. Chapter 6 represents work in progress to do with applying OTPs to software engineering. The idea

is well motivated, though some technical details remain to be resolved. Conclusion related work and future work are described in the final chapter.

Parts of this thesis have been published or will be published as follows. Abo half of the content of chapters 2 and 3 appeared as [61]. The content of chapter 4 contained in [59]. Some of chapter 6 appeared as [60].

Chapter 2

Ordered theory presentations

As seen in chapter 1, an ordered presentation of a theory is a bag (or multi-set) of sentences equipped with a partial order. We saw that if the sentences are mutually consistent, it is safe to ignore the partial order. The models of such an ordered presentation are just the models of the set of sentences. But if the sentences conflict, sentences lower in the ordering are to be treated as having greater weight or priority. This does not mean that a sentence high in the ordering can be ignored, even if it conflicts with sentences below it; some 'components' of it may still be needed in determining the models of the presentation. In §1.3, examples of ordered presentations were given to illustrate their intended behaviour, and criteria for judging a theory of ordered presentations were established.

In this chapter we formally define ordered theory presentations and establish a framework for reasoning from them. We prove many properties of the framework.

In §2.1, the logical setting and notation is established, and the class of logics is characterised for which the behaviour of OTPs will be specified. In §2.2 the models of an OTP are defined, through two kinds of ordering, $\sqsubseteq^{\Gamma}$ and $\sqsubseteq_{\phi}$ (§2.2.1 and §2.2.2). The second of these relies on a relation between sentences which we call 'natural consequence'. The sequence of definitions is motivated and elucidated in §2.2.1 to §2.2.4, and a summary is given in §2.2.5.

2.1 Logical setting

The definitions which will be given in §2.2 apply to any logic which is given in terms of language interpretations and a satisfaction relation, subject to being able to define the standard notion of positive and negative occurrences of non-logical symbols. Such logics include classical, intuitionistic and modal logics, in their propositional and predicate forms; Horn clause logic; equational logic, action logic and a host of others. We keep to this level of generality for most of the chapter as far as the definitions and results are concerned.

In this section, some we will recap on some standard definitions to establish notation. It will be useful to refer back to these later.

Definition 2.1 A *language* L is

1. a finite set of logical connectives;

2. a (possibly sorted) collection of non-logical symbols; and
3. a set of rules for forming L-sentences.

L considered as a set is the set of L-sentences.

Definition 2.2 A *interpretation system* $\langle \mathcal{M}, \Vdash \rangle$ for a language L is a set $\mathcal{M}$ of *interpretations* and a relation (called *satisfaction*) $\Vdash \subseteq \mathcal{M} \times L$.

Definition 2.3 A *logic* $\langle L, \mathcal{M}, \Vdash \rangle$ is a language L together with an interpretation system $\langle \mathcal{M}, \Vdash \rangle$ for L.

Of course this definition is not broad enough to capture every 'logic' encountered in the literature. For example, it excludes logics for default reasoning [47], linear logic [26], relevance logics [2], since any logic satisfying this definition is monotonic. As already mentioned, it *includes* propositional and predicate classical, intuitionistic and modal logics, Horn clause logic and others. For a variety of logics defined in this way including logics of partiality, see [70, 71]. It should also be noted that there are many other characterisations of logic (see eg. [30]). Definition 2.3 delineates the logics we consider in this thesis.

Example 2.4 *Classical propositional logic.* An appropriate language L has

1. the connectives $\{\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \bot, \top\}$;
2. a set atoms(L) of propositional atoms; and
3. the following rules for sentence formation:
   - $\top$ and $\bot$ are sentences;
   - if $p \in$ atoms(L) then p is a sentence; and
   - if ϕ and ψ are sentences then $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$ and $\phi \leftrightarrow \psi$ are sentences.

Brackets are used to disambiguate expressions involving nested connectives; but we also adopt the convention that $\neg$ binds more closely than $\wedge$ and $\vee$, which are in turn more binding than $\rightarrow$ and $\leftrightarrow$.

$\mathcal{M}$ consists of assignments of truth values to propositional atoms; if $M \in \mathcal{M}$ then $M : \text{atoms}(L) \rightarrow \{\mathbf{t}, \mathbf{f}\}$. The satisfaction relation is defined in the following (standard) way:

$$
\begin{array}{rll}
M \Vdash \top & & \\
M \nVdash \bot & & \\
M \Vdash p & \text{if} & M(p) = \mathbf{t} \text{ and } p \in \text{atoms}(L) \\
M \Vdash \neg\phi & \text{if} & M \nVdash \phi \\
M \Vdash \phi \wedge \psi & \text{if} & M \Vdash \phi \text{ and } M \Vdash \psi \\
M \Vdash \phi \vee \psi & \text{if} & M \Vdash \phi \text{ or } M \Vdash \psi \\
M \Vdash \phi \rightarrow \psi & \text{if} & M \Vdash \phi \text{ implies } M \Vdash \psi \\
M \Vdash \phi \leftrightarrow \psi & \text{if} & (M \Vdash \phi \text{ iff } M \Vdash \psi)
\end{array}
$$

Example 2.5 *Classical predicate logic, with equality.* L has

1. each of the connectives of example 2.4 plus $\{\forall, \exists\}$;
2. a set of predicate symbols, each with an arity $n \geqslant 0$, a set of function symbols, also each with an arity $n \geqslant 0$, and a set of variables; and
3. the following rules for term formation, formula formation, and sentence formation:
   - if x is a variable, f a function symbol with arity n and $t_1, \ldots, t_n$ are terms then x and $f(t_1, \ldots, t_n)$ are terms.
   - if $t_1, t_2, \ldots, t_n$ are terms, p a predicate symbol with arity n, and ϕ and ψ are formulas and x is a variable then $p(t_1, \ldots, t_n)$, $t_1 = t_2$, $\top$, $\bot$, $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, $\phi \leftrightarrow \psi$, $\exists x.\, \phi$ and $\forall x.\, \phi$ are formulas.
   - if ϕ is a formula with no free variables then ϕ is a sentence.

   The definition of free variables is the standard one. See, for example, [31, definition 3.8].

Each $M \in \mathcal{M}$ has

- a domain of individuals D_M;
- for each predicate symbol p with arity n, a subset $M[\![p]\!]$ of D_M^n (D_M^n is $\underbrace{D_M \times \ldots \times D_M}_{n \text{ times}}$);
- for each function symbol f with arity n a function $M[\![f]\!]$ from D_M^n to D_M; and
- for each variable x an element $M[\![x]\!]$ of D_M.

$M[\![\cdot]\!]$ is extended to terms by

$$M[\![f(t_1, \ldots, t_n)]\!] = M[\![f]\!](M[\![t_1]\!], \ldots, M[\![t_n]\!])$$

for each function symbol f with arity n.

For each variable x of L, an equivalence relation $\sim_x \subseteq \mathcal{M} \times \mathcal{M}$ is defined as follows: $M \sim_x N$ if $D_M = D_N$ and for each predicate symbol p and function symbol f, $M[\![p]\!] = N[\![p]\!]$ and $M[\![f]\!] = N[\![f]\!]$ and for each variable y with the possible exception of x, $M[\![y]\!] = N[\![y]\!]$. That is to say, M and N are alike in every way except possibly in how they assign the variable x.

The satisfaction relation is defined as follows: if χ is of the form $\top$, $\bot$, $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, or $\phi \leftrightarrow \psi$, then $M \Vdash \chi$ according to example 2.4. Otherwise,

$$\begin{aligned} M \Vdash p(t_1, \ldots, t_n) &\quad \text{if} \quad \langle M[\![t_1]\!], \ldots, M[\![t_n]\!] \rangle \in M[\![p]\!] \\ M \Vdash t_1 = t_2 &\quad \text{if} \quad M[\![t1]\!] = M[\![t_2]\!] \\ M \Vdash \forall x.\, \phi &\quad \text{if} \quad N \Vdash \phi \text{ for each } N \text{ s.t. } M \sim_x N \\ M \Vdash \exists x.\, \phi &\quad \text{if} \quad N \Vdash \phi \text{ for some } N \text{ s.t. } M \sim_x N \end{aligned}$$

We now return to standard definitions and a result:

Definition 2.6 A (flat) theory presentation over a language L, or an L-theory pr sentation, is a finite set of L-sentences.

Definition 2.7 Let Φ be a theory presentation. Then $M \Vdash \Phi$ if $M \Vdash \phi$ for ea $\phi \in \Phi$.

Definition 2.8 ϕ is a *consequence* of Φ, or Φ *entails* ϕ, written $\Phi \models \phi$, if for ea $M \in \mathcal{M}$, $M \Vdash \Phi$ implies $M \Vdash \phi$.

An expression like $\Phi \models \phi$ is called a *sequent*. Simple though these definitions ar there are some well known consequences.

Proposition 2.9 Let L be a language and $\models$ the consequence relation defined fro an interpretation system $\langle \mathcal{M}, \Vdash \rangle$. The following properties hold of $\models$:

1. Inclusion: $\Phi, \phi \models \phi$
2. Monotonicity: $$\frac{\Phi \models \psi}{\Phi, \phi \models \psi}$$
3. Cut: $$\frac{\Phi, \phi \models \psi \quad \Psi \models \phi}{\Phi, \Psi \models \psi}$$

As usual, Φ, Ψ and Φ, ϕ are abbreviations for $\Phi \cup \Psi$ and $\Phi \cup \{\phi\}$ respectively. Th horizontal rule means: if the top sequent holds then so does the bottom one.

The last standard definition to consider is that of positive and negative occurrenc of non-logical symbols in formulas. The exact definition depends on the connectives ar their interpretations. We will give examples for propositional and predicate classic logic.

Example 2.10 Let L, $\mathcal{M}$ and $\Vdash$ be classical propositional logic (example 2.4) wi $p \in \text{atoms}(L)$.

- p occurs positively in p.
- If p occurs positively (negatively) in ϕ then it occurs negatively (positively) $\neg\phi$.
- If p occurs positively (negatively) in ϕ or in ψ then it occurs positively (negativel in $\phi \wedge \psi$ and $\phi \vee \psi$.
- If p occurs negatively (positively) in ϕ or positively (negatively) in ψ then occurs positively (negatively) in $\phi \rightarrow \psi$.
- If p occurs at all in ϕ or ψ then it occurs both positively and negatively in $\phi \leftrightarrow$
- p does not occur in either $\top$ or $\bot$.

Note, therefore, that p can occur positively, or negatively, or positively and negatively, or p need not occur at all. In

$$\Big(p \to (q \leftrightarrow q \wedge r)\Big) \wedge (q \to \neg p)$$

p occurs negatively (twice), q occurs positively (twice) and negatively (three times) and r occurs positively and negatively (once). s does not occur.

Example 2.11 In the case of predicate logic, if p is a predicate symbol and $t_1, \ldots, t_n$ are terms then p occurs positively in $p(t_1, \ldots, t_n)$. Each of the clauses for the propositional connectives above applies. Moreover, if p occurs positively (negatively) in ϕ then it occurs positively (negatively) in $\forall x.\, \phi$ and $\exists x.\, \phi$. In the sentence

$$\forall x.\, \exists y.\, \Big(x \neq y \wedge \Big(p(x) \to q(x, y) \vee p(y)\Big)\Big)$$

p occurs positively and negatively, q positively and r not at all. We need not talk of the occurrence of $=$ as it is built in to the language.

Thus, the class of logics for which OTPs are defined in this chapter is quite wide. (For other examples of such logics, see [63].) An interesting question is whether this can be broadened still further. For example, a natural but abstract class of logics are the so-called *institutions* [27] used in specification theory. Whether OTPs can be defined over arbitrary institutions is a matter of ongoing research.

2.2 Ordered theory presentations

The purpose of this section is to define satisfaction for ordered presentations of theories, so that consequence for such presentations can be defined by definition 2.8. As before we assume we are working with a fixed language L and interpretation system $\langle \mathcal{M}, \Vdash \rangle$.

We have seen that an ordered theory presentation is a collection of sentences equipped with a partial order. But to cover the case that the same sentence occurs several times in different places in the presentation, it is necessary to posit a 'carrier set' on which the order is defined and whose points are labelled by sentences.

Definition 2.12 An ordered theory presentation ? over a language L is a tuple $\langle X, \leqslant, F \rangle$ where

1. X is a set (the carrier set).
2. $\leqslant$ is a well-founded partial order on X (that is, there are no infinite descending chains $x_1 > x_2 > x_3 > \ldots$).
3. F is a function mapping X to L-sentences.

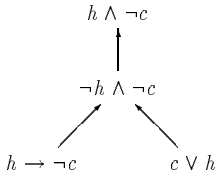
A *finite* ordered theory presentation is one whose carrier set is finite. Some of the results given in this and the next chapter work only for finite OTPs (this will be explicitly stated each time).

As usual, $x < y$ means $x \leqslant y$ and $y \not\leqslant x$, and $x \geqslant y$ and $x > y$ mean $y \leqslant$ and $y < x$ respectively. The letters Φ and Ψ were used for 'flat' theory presentatio (definition 2.6); we shall use ? and Δ for ordered presentations.

The intuitive meaning of the ordering is: if $x < y$ then the sentence $F(x)$ has great priority (or more influence) than $F(y)$. This information is used when $F(x)$ and $F($ conflict.

We have already seen many ordered presentations in chapter 1 using the inform notation of graphs; definition 2.12 is the formal definition.

Example 2.13 The ordered presentation



is formally written as follows:

1. $X = \{1, 2, 3, 4\}$.
2. $\leqslant = \{(1,1), (1,3), (1,4), (2,2), (2,3), (2,4), (3,3), (3,4), (4,4)\}$
3. $F(1) = h \to \neg c$; $F(2) = c \vee h$; $F(3) = \neg h \wedge \neg c$; $F(4) = h \wedge \neg c$.

The requirement that X have no infinite descending chains means that there is n infinite sequence of ever more important sentences in the presentation, which obvious would not make sense. There is no need to exclude infinite sequences of ever *le* important sentences, however; an example of a situation in which this would be usef will be seen in chapter 5.

A consequence of the requirement on X is that it is always possible to find minim elements of any subset of X. Indeed, it will be useful to prove the slightly strong result:

Lemma 2.14 Let $\langle X, \leqslant, F \rangle$ be an ordered theory presentation, and let $X' \subseteq X$ ar $x \in X'$. Then there is a $y \in X'$ such that y is minimal in X' and $y \leqslant x$.

Proof If x is minimal in X' then set $y = x$. Otherwise, pick $x_1 \in X'$ such tha $x_1 < x$. If x_1 is minimal in X' then set $y = x_1$; otherwise, pick $x_2 \in X'$ such tha $x_2 < x_1$. Proceed in this way until a minimal element is found. If none is found, v have constructed an infinite descending chain $x > x_1 > x_2 > \ldots$, a contradiction.

We want to define the models of an ordered theory presentation, that is, to exter the satisfaction relation to ordered presentations analogously to its extension to fl presentations in definition 2.7. Let $? = \langle X, \leqslant, F \rangle$ be an ordered theory presentatio over $\langle L, \mathcal{M}, \Vdash \rangle$. If all the sentences of ? are mutually consistent, then the models ? are just the models of that set of sentences. The interesting case is when sentenc in ? are inconsistent with each other and we have to use the ordering to resolve th

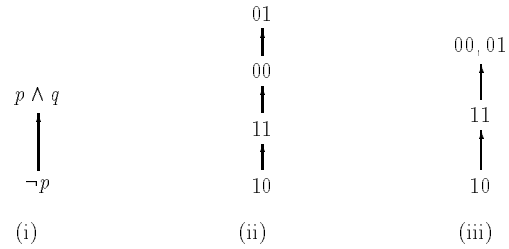


Figure 2.1: an ordered theory presentation and candidate interpretation orderings

conflict. In this case we cannot hope to satisfy all the sentences but models of ? should satisfy as many of them as possible, taking account of their ordering.

The technique to be adopted is to order interpretations of L according to ?, so that those higher up the ordering are better at satisfying ?. This ordering is written $\sqsubseteq^{\Gamma}$. $M \sqsubseteq^{\Gamma} N$ means N is at least as good as M at satisfying ?. Models of ? are then taken to be the interpretations which are maximal according to $\sqsubseteq^{\Gamma}$.

The remainder of §2.2 is structured as follows. In §2.2.1 we consider a proposal for the definition of $\sqsubseteq^{\Gamma}$ and find it to be wanting. The correct definition relies on what we call 'satisfaction orderings', which are motivated in §2.2.2. They rely on a restriction of ordinary consequence which is defined in §2.2.3. With this to hand, satisfaction orderings are defined in §2.2.4. Finally, in view of this plethora of definitions and considerations, the situation is summarised in §2.2.5.

2.2.1 The ordering $\sqsubseteq^{\Gamma}$

The question addressed in this section is how $\sqsubseteq^{\Gamma}$ is defined. If ? were not itself ordered, this task would be easier. For example, one might say $M \sqsubseteq^{\Gamma} N$ if N satisfies all the sentences of ? that M does. But ? is ordered, and our definition must take account of that. Consider again the interpretations M and N. If $M \sqsubseteq^{\Gamma} N$, but there is a sentence ϕ in ? such that M satisfies ϕ and N does not, then there must be a more important sentence ψ which is satisfied by N but not by M. Thus we might be tempted to define $\sqsubseteq^{\Gamma}$ as follows:

Proposal 2.15 $M \sqsubseteq^{\Gamma} N$ if $\forall x \in X.\, M \Vdash F(x)$ and $N \nVdash F(x)$ implies $\exists y \leqslant x.\, M \nVdash F(y)$ and $N \Vdash F(y)$.

To see that this is wrong, consider the ordered presentation given in example 1.2. A model of this theory is an interpretation which satisfies $\neg p$ and as much of $p \wedge q$ as it can. Let $\langle \mathcal{M}, \Vdash \rangle$ be the usual interpretation system for this logic (see example 2.4). An interpretation M of $\mathcal{M}$ is specified by whether it satisfies the atoms p and q. Let us write 10 for the interpretation which satisfies p but not q; 11, 01 and 00 are defined analogously.

Intuitively we expect the interpretation 01 to be the only model of ?. To see this, notice that it must be either 00 or 01 since $\neg p$ is the most important sentence of ?. Of these two 01 is better at satisfying ? overall because, while neither of them satisfy

$p \wedge q$, it at least satisfies half of $p \wedge q$. Further reasoning along these lines results the conclusion that figure 2.1(ii) is the correct interpretation ordering for the theory question. There, the arrows mean $\sqsubseteq^{\Gamma}$.

But since neither of the interpretations 01 and 00 fully satisfy $p \wedge q$, and proposal 2.1 just looks at what sentences are satisfied by the various interpretations, the propos cannot distinguish between 01 and 00. In fact, according to the proposal $\sqsubseteq^{\Gamma}$ is t order given in figure 2.1(iii). 01 and 00 are both maximal in this ordering, so bo would be models of ? according to the proposal.

The problem is that we were not able to take account of the fact that, while neith 01 nor 00 satisfy $p \wedge q$, 01 is actually better at it than 00; at least it satisfies q, which a consequence of $p \wedge q$. This thought leads us to the idea that, given a sentence and a interpretation, there is more we can say than whether the interpretation satisfies t sentence or not. We can compare two interpretations as to the degree to which the satisfy the sentence.

This intuition, about degrees of satisfaction, is formalised in the following wa We suppose the existence of an ordering $\sqsubseteq_{\phi}$ on interpretations (for each sentence ϕ and use that to define $\sqsubseteq^{\Gamma}$. $M \sqsubseteq_{\phi} N$ means that N is as good as M at satisfyi ϕ. The example discussed above shows that we should be interested in ordering t interpretations which fail to satisfy ϕ according to how nearly they do; for example, is better than 00 at satisfying $p \wedge q$ (therefore, $00 \sqsubset_{p \wedge q} 01$). $\sqsubseteq_{\phi}$ is called a 'satisfactio ordering', and we suppose it satisfies the following assumption.

Assumption 2.16 Let $\langle L, \mathcal{M}, \Vdash \rangle$ be a logic, and for each $\phi \in L$ let $\sqsubseteq_{\phi}$ be a satisfaction ordering. Then

1. $\sqsubseteq_{\phi}$ is a pre-order (i.e. reflexive and transitive);
2. M is $\sqsubseteq_{\phi}$-maximum iff $M \Vdash \phi$.

Recall that a point M in an order $\langle \mathcal{M}, \sqsubseteq \rangle$ is maximum if for each $N \in \mathcal{M}$, $N \sqsubseteq M$.

We will define suitable orderings which meet this assumption in §2.2.4. A cons quence of the assumption is

Lemma 2.17 If $M \nVdash \phi$ and $N \Vdash \phi$ then $M \sqsubset_{\phi} N$.

Proof We show (i) $M \sqsubseteq_{\phi} N$ and (ii) $N \not\sqsubseteq_{\phi} M$. (i) $M \sqsubseteq_{\phi} N$ since N is $\sqsubseteq_{\phi}$-maximu by the assumption. (ii) $N \not\sqsubseteq_{\phi} M$, for $N \Vdash \phi$ and $M \nVdash \phi$.

We have used some standard notation in this lemma. It is as well to fix the derived orderings once and for all.

Notation 2.18

1. $M \sqsubset_{\phi} N$ if $M \sqsubseteq_{\phi} N$ and $N \not\sqsubseteq_{\phi} M$.
2. $M \equiv_{\phi} N$ if $M \sqsubseteq_{\phi} N$ and $N \sqsubseteq_{\phi} M$.
3. $\sqsubseteq_x$ will abbreviate $\sqsubseteq_{F(x)}$ when in the context of a particular OTP; similarly f $\equiv_x$ and $\sqsubset_x$.

4. $M \sqsubset^\Gamma N$ if $M \sqsubseteq^\Gamma N$ and $N \not\sqsubseteq^\Gamma M$; also, $M \equiv^\Gamma N$ means $M \sqsubseteq^\Gamma N$ and $N \sqsubseteq^\Gamma M$.

5. $M \sqsupseteq^\Gamma N$ means $N \sqsubseteq^\Gamma M$; and similarly for $M \sqsupset^\Gamma N$, $M \sqsupseteq_\phi N$ and $M \sqsupset_\phi N$.

Given the satisfaction orderings of assumption 2.16, we can define the interpretation ordering induced by ?. The definition captures the flavour of proposal 2.15, which is that if a sentence in ? makes the 'wrong' choice of two interpretations then there is a sentence with greater priority which makes the 'right' choice. But now, the choice that the sentence ϕ makes is determined by $\sqsubseteq_\phi$.

Let $? = \langle X, \leqslant, F\rangle$ be an OTP over $\langle L, \mathcal{M}, \Vdash\rangle$.

Definition 2.19 $M \sqsubseteq^\Gamma N$ if for each $x \in X$, $M \not\sqsubseteq_x N$ implies there exists $y \leqslant x$ such that $M \sqsubset_y N$.

One can read this as saying: N is as good as M overall $[M \sqsubseteq^\Gamma N]$ if whenever it appears not to be so at a point x $[M \not\sqsubseteq_x N]$ then there is a more important point y $[y \leqslant x]$ where N is doing better than M $[M \sqsubset_y N]$.

Informally, the definition says: if things appear to go wrong at a particular x, then they go well at some y in a more important position than x. The condition that there be no descending chains in OTPs guarantees that the process of finding 'more important ys' terminates. To be precise:

Lemma 2.20 $M \sqsubseteq^\Gamma N$ iff $\forall x \in X.(M \not\sqsubseteq_x N$ implies $\exists y \leqslant x. M \sqsubset_y N$ and $\forall z < y. M \equiv_z N)$.

Proof (If) Immediate. (Only if) Suppose $M \sqsubseteq^\Gamma N$ and $M \not\sqsubseteq_x N$ for some x. Let $X' = \{y \in X \mid M \sqsubset_y N$ and $y \leqslant x\}$. $X' \neq \emptyset$ since $M \sqsubseteq^\Gamma N$. Let y be a minimal point in X' (this is possible by lemma 2.14). Then $M \sqsubset_y N$, and if $z < y$ then $z \notin X'$, so $M \not\sqsubset_z N$. Either $M \not\sqsubseteq_z N$ or $M \equiv_z N$. If $M \not\sqsubseteq_z N$ then $\exists z' \leqslant z. z' \in X'$, a contradiction since then $z' < y$. Therefore, $M \equiv_z N$. ♢

Definition 2.19 is only one out of four possible ways of capturing proposal 2.15. We might just as easily have said:

- $M \not\sqsubseteq_x N$ implies $\exists y \leqslant x. N \not\sqsubseteq_y M$, or
- $N \sqsubset_x M$ implies $\exists y \leqslant x. M \sqsubset_y N$, or
- $N \sqsubset_x M$ implies $\exists y \leqslant x. N \not\sqsubseteq_y M$.

Indeed, replacing $y \leqslant x$ with $y < x$ gives us another four plausible definitions. Some of these eight are equivalent. Without going into details, it turns out that only the one chosen for definition 2.19 has good formal properties. In particular, it is the only one with the following property, which I consider clear-cut grounds for choosing it.

Proposition 2.21 $\sqsubseteq^\Gamma$ is a pre-order.

Proof Reflexivity is obvious. For transitivity, suppose $L \sqsubseteq^\Gamma M \sqsubseteq^\Gamma N$, and l
$L \not\sqsubseteq_x N$. We shall show $L \sqsubset_y N$ for some $y \leqslant x$.

Suppose $L \sqsubseteq_x M$. Either $M \sqsubseteq_x N$ or $M \not\sqsubseteq_x N$. If $M \sqsubseteq_x N$ then $L \sqsubseteq_x N$,
contradiction. If $M \not\sqsubseteq_x N$, let $y_2 \leqslant x$ be such that $M \sqsubset_{y_2} N$ and $M \sqsubseteq_z N$ for $z \leqslant$
(lemma 2.20). If $L \not\sqsubseteq_{y_2} M$, then let $y \leqslant y_2$ be such that $L \sqsubset_y M$. Then $y \leqslant x$ ar
$L \sqsubset_y N$ follows from $L \sqsubset_y M$ and $M \sqsubseteq_y N$. If $L \sqsubseteq_{y_2} M$, set $y = y_2$. Then $y \leqslant x$, ar
$L \sqsubset_y N$ follows from $L \sqsubseteq_y M$ and $M \sqsubset_y N$ and assumption 2.16.

On the other hand, suppose $L \not\sqsubseteq_x M$ and let $y_1 \leqslant x$ be such that $L \sqsubset_{y_1} M$ ar
$L \sqsubseteq_z M$ for all $z \leqslant y_1$ (lemma 2.20). Again, consider separately the two cases $M \sqsubseteq_{y_1}$
and $M \not\sqsubseteq_{y_1} N$. If $M \sqsubseteq_{y_1} N$, set $y = y_1$. Then $y \leqslant x$, and $L \sqsubset_y N$ follows from $L \sqsubset_y$
and $M \sqsubseteq_y N$. If $M \not\sqsubseteq_{y_1} N$ then let $y \leqslant y_1$ be such that $M \sqsubset_y N$. Then $y \leqslant x$, ar
$L \sqsubset_y N$ follows from $L \sqsubseteq_y M$ and $M \sqsubset_y N$ and assumption 2.16.

Proposition 2.22 Then $M \sqsubset^\Gamma N$ implies $\exists z \in X. M \sqsubset_z N$.

Proof Suppose $M \sqsubset^\Gamma N$. Then $N \not\sqsubseteq^\Gamma M$, so by definition 2.19 $\exists x. N \not\sqsubseteq_x M$.
$M \not\sqsubseteq_x N$ then by the definition $\exists y. M \sqsubset_y N$, so set $z = y$. Otherwise, $M \sqsubseteq_x N$,
$M \sqsubset_x N$, so set $z = x$.

Proposition 2.23 Then $M \equiv^\Gamma N$ iff $M \equiv_x N$ for all $x \in X$.

Proof (If) immediate. (Only if) Suppose $M \not\equiv_x N$. Then $M \not\sqsubseteq_x N$ or $N \not\sqsubseteq_x M$
Without loss of generality, assume $M \not\sqsubseteq_x N$. Since $M \sqsubseteq^\Gamma N$, by lemma 2.20 pick
such that $M \sqsubset_y N$ and $\forall z < y. M \equiv_z N)$. Since $N \sqsubseteq^\Gamma N$, by definition 2.19 pick $z \leqslant$
such that $N \sqsubset_z M$. Clearly, $z \neq y$; therefore, $z < y$ so $M \equiv_z N$, a contradiction.

The definition of $\Vdash$ on flat presentations (definition 2.7) can now be extended
ordered presentations in the way already described.

Definition 2.24 $M \Vdash ?$ if M is $\sqsubseteq^\Gamma$-maximal.

Definition 2.24 further overloads $\Vdash$. (To determine whether $M \Vdash A$, we have
check whether A is a sentence, a flat theory presentation or an ordered theory prese
tation and use definitions 2.2, 2.7 or 2.24 accordingly.) This overloading is justified
that for the most part the different senses of $\Vdash$ correspond well. To be precise, we ha
that $M \Vdash \phi$ iff $M \Vdash \{\phi\}$, where ϕ is a sentence. Also, $M \Vdash \Phi$ implies $M \Vdash$ "Φ" whe
Φ is a set of sentences and "Φ" is the OTP with the same sentences and the discre
ordering. If Φ is consistent then we have the converse, that $M \Vdash$ "Φ" implies $M \Vdash$
The one case of disagreement, then, is when Φ is inconsistent, in which case we ha
$M \nVdash \Phi$ and $M \Vdash$ "Φ" for all M. An example of this was given (case 3 of example 1.1

Finally, consequence is defined in the standard way:

Definition 2.25 $? \models \phi$ if for each $M \in \mathcal{M}$, $M \Vdash ?$ implies $M \Vdash \phi$.

Now we give some results to continue to get the feel for the behaviour of OTP
Naturally we expect that the minimum sentence (if there is one) is satisfied by mode
of the theory:

Definition 2.26 ϕ is *minimum* in $? = \langle X, \leqslant, F\rangle$ if $\langle X, \leqslant\rangle$ has a minimum point 0 and $F(0) = \phi$.

Proposition 2.27 Let $? = \langle X, \leqslant, F\rangle$ be an ordered presentation and $M \in \mathcal{M}$ such that $M \Vdash ?$. If ϕ is minimum in $?$ and $\phi \neq \bot$ then $M \Vdash \phi$.

Proof Let 0 be the minimum point in X. $F(0) = \phi$. Suppose for a contradiction that $M \nVdash \phi$. Since $\phi \neq \bot$, let $N \Vdash \phi$. By lemma 2.17, $M \sqsubset_0 N$. We show $M \nVdash ?$ by showing $M \sqsubset^{\Gamma} N$. To show $M \sqsubseteq^{\Gamma} N$, suppose x is such that $M \not\sqsubseteq_x N$. Let $y = 0$. Then $y \leqslant x$ and $M \sqsubset_y N$. To show $N \not\sqsubseteq^{\Gamma} M$, let $x = 0$. $N \not\sqsubseteq_x M$. If $y \leqslant x$, then $y = 0$ since 0 is minimum; but $N \not\sqsubset_y M$. ♢

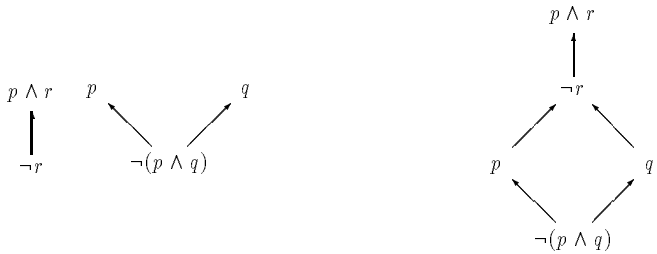
Already we have enough to look at some effects of putting ordered theory presentations together. Let $?\Delta$ be $?$ and Δ 'side by side', and let $\overset{\Gamma}{\Delta}$ be $?$ on top of Δ. Formally:

Definition 2.28 Let $? = \langle X, \leqslant_X, F_X\rangle$ and $\Delta = \langle Y, \leqslant_Y, F_Y\rangle$, with X and Y disjoint.

1. $?\Delta = \langle Z, \leqslant_Z, F_Z\rangle$, with $Z = X \cup Y$, $F_Z(x) = F_X(x)$ if $x \in X$, otherwise $F_Z(x) = F_Y(x)$, and $x \leqslant_Z y$ if $x \leqslant_X y$ or $x \leqslant_Y y$.
2. $\overset{\Gamma}{\Delta} = \langle Z, \leqslant_Z, F_Z\rangle$, with Z and F_Z as above and $x \leqslant_Z y$ if $x \leqslant_X y$ or $x \leqslant_Y y$ or $(x \in Y$ and $y \in X)$.

Example 2.29 If $?$ and Δ are respectively

then $?\Delta$ and $\overset{\Gamma}{\Delta}$ are respectively



Proposition 2.30

1. $M \sqsubseteq^{\Gamma\Delta} N$ iff $M \sqsubseteq^{\Gamma} N$ and $M \sqsubseteq^{\Delta} N$.
2. $M \sqsubset^{\Gamma\Delta} N$ iff $(M \sqsubset^{\Gamma} N$ and $M \sqsubseteq^{\Delta} N)$ or $(M \sqsubseteq^{\Gamma} N$ and $M \sqsubset^{\Delta} N)$.
3. $M \sqsubseteq^{\Gamma}_{\Delta} N$ iff $M \sqsubset^{\Delta} N$ or $(M \equiv^{\Delta} N$ and $M \sqsubseteq^{\Gamma} N)$.

4. $M \sqsubset^{\Gamma}_{\Delta} N$ iff $M \sqsubset^{\Delta} N$ or $(M \equiv^{\Delta} N$ and $M \sqsubset^{\Gamma} N)$.

Proof 1. and 2. follow easily from the definitions, and 4. follows easily from 3.

For 3., suppose $M \sqsubseteq^{\Gamma}_{\Delta} N$ and $M \not\sqsubseteq^{\Gamma} N$. We show $M \sqsubset^{\Delta} N$. (a) $M \sqsubseteq^{\Delta} N$. Pic $x \in \Delta$[1]. Since $M \sqsubseteq^{\Gamma}_{\Delta} N$, we can find $y \in \Delta$ satisfying the conditions of definitio 2.19. (b) $N \not\sqsubseteq^{\Delta} M$. Suppose $N \sqsubseteq^{\Delta} M$; we derive a contradiction. Using the fact tha $M \not\sqsubseteq^{\Gamma} N$, pick $x \in ?$ such that $M \not\sqsubseteq_x N$ and $\forall y \in ?$ with $y \leqslant x$, $M \not\sqsubset_y N$. But sin $M \sqsubseteq^{\Gamma}_{\Delta} N$, $\exists y \in \Delta . M \sqsubset_y N$ and $\forall z < y . M \equiv_z N$. But $N \sqsubseteq^{\Delta} M$ and $N \not\sqsubseteq_y M$, $\exists z < y . N \sqsubset_z M$, a contradiction.

Conversely, we show: (i) $M \sqsubset^{\Delta} N$ implies $M \sqsubseteq^{\Gamma}_{\Delta} N$. Let $x \in \overset{\Gamma}{\Delta}$ be such tha $M \not\sqsubseteq_x N$. By proposition 2.22, pick $z \in \Delta$ such that $M \sqsubset_z N$. Since $z \leqslant x$, we ha $M \sqsubseteq^{\Gamma}_{\Delta} N$. (ii) $M \equiv^{\Delta} N$ and $M \sqsubseteq^{\Gamma} N$ imply $M \sqsubseteq^{\Gamma}_{\Delta} N$. Let $x \in \overset{\Gamma}{\Delta}$ be such tha $M \not\sqsubseteq_x N$. By proposition 2.23, $x \in ?$. Since $M \sqsubseteq^{\Gamma} N$, pick $y \in ?$ such that $M \sqsubset_y$

Propositions 2.27 and 2.30 are meant to convince the reader that the definitio of $\sqsubseteq^{\Gamma}$ is the right one. The next chapter contains further evidence, but we end th section with a final remark in this direction. As before, let us write "ϕ" for the OT with the single sentence ϕ. Then we have, as a consequence of definition 2.19:

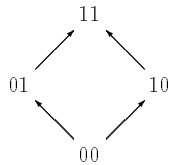
Remark 2.31 $M \sqsubseteq^{\text{“}\phi\text{”}} N$ iff $M \sqsubseteq_{\phi} N$.

2.2.2 The ordering $\sqsubseteq_{\phi}$ (motivation)

In §2.2.1 we assumed so-called *satisfaction orderings* $\sqsubseteq_{\phi}$ satisfying the conditions assumption 2.16. In this section we show how such an ordering may be defined, ar give examples.

Given a sentence ϕ and an interpretation M, we are interested in how well M satisfies ϕ. If $M \Vdash \phi$, then this is the best one could hope for; M satisfies ϕ to th fullest possible extent. But if $M \nVdash \phi$, all is not lost; for it may more nearly satis ϕ than some other interpretation N which also fails to satisfy ϕ. In that case w write $N \sqsubset_{\phi} M$. The aim of $\sqsubseteq_{\phi}$ is to order the interpretations which do not satisfy according to how *nearly* they do.

The definition of $\sqsubseteq_{\phi}$ is motivated by the example given at the beginning of §2.2 (see figure 2.1). We concluded there that we wanted to have $00 \sqsubset_{p\wedge q} 01$, and we ca extend the argument for the following diagram for $\sqsubseteq_{p\wedge q}$:



In other words, we wish that interpretations which satisfy p or q are better at satisfyin $p \wedge q$ than that which satisfies neither p nor q.

As p and q are consequences of $p \wedge q$, one might consider the following basis for definition of $\sqsubseteq_{\phi}$:

[1] We should really say: let $\Delta = \langle X, \leqslant_X, F\rangle$ and pick $x \in X$. But $x \in \Delta$ is a convenient shorthan

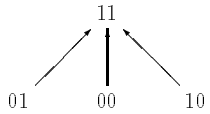
The more consequences of ϕ that M satisfies, the higher it should be in $\sqsubseteq_\phi$.

Thus one might consider the following definition for $\sqsubseteq_\phi$:

Proposal 2.32 $M \sqsubseteq_\phi N$, if for each ψ,

$$\phi \models \psi \Rightarrow (M \Vdash \psi \Rightarrow N \Vdash \psi)$$

However, one can immediately see that not all the consequences of ϕ are appropriate to take into account in the definition of $\sqsubseteq_\phi$. Consider again example 2.47. p, $p \leftrightarrow q$ and q are all consequences of $p \wedge q$, but none of each other. Therefore proposal 2.32 gives the following for $\sqsubseteq_{p \wedge q}$:



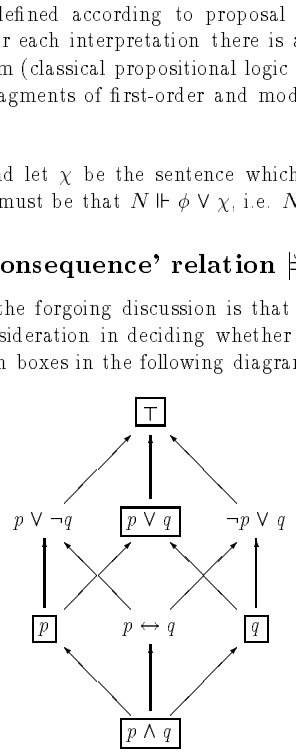
This is wrong according to the intuition mentioned. Indeed, it turns out that under this definition $\sqsubseteq_\phi$ always has a height of just 2. To be precise:

Proposition 2.33 If $\sqsubseteq_\phi$ is defined according to proposal 2.32 and the underlying logic has the property that for each interpretation there is a sentence which picks it out uniquely up to isomorphism (classical propositional logic over a finite language has this property, as do certain fragments of first-order and modal logics), then $M \sqsubseteq_\phi N$ implies $N \Vdash \phi$ or $M = N$.

Proof Suppose $M \sqsubseteq_\phi N$ and let χ be the sentence which characterises M. Since $\phi \models \phi \vee \chi$ and $M \Vdash \phi \vee \chi$, it must be that $N \Vdash \phi \vee \chi$, i.e. $N \Vdash \phi$ or $N = M$. $\diamondsuit$

2.2.3 The 'natural consequence' relation $\models^\smile$

The problem encountered in the forgoing discussion is that not all the consequences of ϕ should be taken into consideration in deciding whether $M \sqsubseteq_\phi N$. In the case of $p \wedge q$, only the consequences in boxes in the following diagram are appropriate.



What distinguishes these consequences of $p \wedge q$ is that they are *monotonic* in p and That is to say, if a model M satisfies such a consequence ψ, then so does the model obtained from M by increasing the 'extension' of p or of q. To define this we need define positive and negative occurrences. As stated previously, we assume that the are given by the underlying logic (examples 2.10 and 2.11).

Definition 2.34 If ϕ is an L-sentence other than $\bot$ and p a non-logical symbol in

1. ϕ is *monotonic in* p if it is equivalent to a sentence in which all occurrences of (if any) are positive.
2. ϕ is *anti-monotonic in* p if it is equivalent to a sentence in which all occurrenc of p are negative.
3. ϕ^+ and ϕ^- are the sets of symbols in which ϕ is monotonic and anti-monoton respectively.

The case that $\phi = \bot$ is handled separately, for reasons which will be explained late we define $\bot^+ = \bot^- = \emptyset$.

Notice that although the definition uses the *syntactic* notion of positive and negati occurrences, it is *semantic* in the sense that it is not sensitive to the way ϕ is writte Let us write $\phi \models\!\models \psi$ if $\phi \models \psi$ and $\psi \models \phi$.

Proposition 2.35 If $\phi \models\!\models \psi$ then $\phi^\pm = \psi^\pm$.

Proof If $p \in \phi^+$ then there is a sentence χ such that $\phi \models\!\models \chi$ and p occurs on positively in χ. But then, $\psi \models\!\models \chi$, so $p \in \psi^+$. The converse, and the case for ϕ^-, a proved similarly.

The justification for the terminology of 'monotonic' and 'anti-monotonic' is as fo lows. One may define the *extension* of a non-logical symbol p in a model to be t set of tuples of which p is true in the model. (In the propositional case, if p is tr in a model then its extension is defined to be the singleton $\{*\}$; if p is false, it is $\emptyset$ Extensions are naturally ordered by inclusion. Let us write $M \leqslant^p N$ if M and N a exactly alike except that N has possibly a greater p-extension than M. It follows th ϕ is monotonic in p iff $(M \leqslant^p N \Rightarrow (M \Vdash \phi \Rightarrow N \Vdash \phi))$, i.e. increasing p-extension a model preserves ϕ-satisfaction. Similarly, ϕ is anti-monotonic in p iff $(N \leqslant^p M$ $(M \Vdash \phi \Rightarrow N \Vdash \phi))$.

Thus, the monotonicities of ϕ is a pair $\langle \phi^+, \phi^- \rangle$ of sets of non-logical symbols su that, if in any model of ϕ the extension of any symbol of the first set is increased, the extension of any in the second set is decreased, the resulting interpretation is st a model of ϕ.

Example 2.36 Let $(L, \mathcal{M})$ be classical propositional logic over $\{p, q\}$. For several examples of ϕ, ϕ^+ and ϕ^- are shown in the following table.

ϕ	ϕ^+	ϕ^-
$\top$	$\{p, q\}$	$\{p, q\}$
p	$\{p, q\}$	$\{q\}$
q	$\{p, q\}$	$\{p\}$
$p \wedge q,\ p \vee q$	$\{p, q\}$	$\emptyset$
$p \rightarrow q$	$\{q\}$	$\{p\}$
$p \leftrightarrow q$	$\emptyset$	$\emptyset$
$\bot$	$\emptyset$	$\emptyset$

Example 2.37 Let $(L, \mathcal{M})$ be classical predicate logic over p (unary) and q (binary).

ϕ	ϕ^+	ϕ^-
$\forall x.\, p(x)$	$\{p, q\}$	$\{q\}$
$\exists x.\, p(x)$	$\{p, q\}$	$\{q\}$
$\forall x.\, \exists y.\, q(x, y)$	$\{p, q\}$	$\{p\}$
$\forall x.\, (p(x) \rightarrow \exists y.\, q(x, y))$	$\{q\}$	$\{p\}$
$\forall x.\, \forall y.\, (q(x, y) \rightarrow q(y, z))$	$\{p\}$	$\{p\}$

We are interested in the consequences of ϕ which preserve these monotonicities.

Definition 2.38 A consequence ψ of ϕ is a *natural consequence* (written $\phi \mathrel{|\!\!=} \psi$) if it preserves the monotonicities of ϕ:

$$\phi \mathrel{|\!\!=} \psi \text{ if } \phi \models \psi,\ \phi^+ \subseteq \psi^+ \text{ and } \phi^- \subseteq \psi^-$$

Natural consequence is a sub-relation of ordinary consequence; in addition to ordinary entailment we require that the monotonicities of the premise be preserved by the conclusion.

Proposition 2.39 $\mathrel{|\!\!=}$ is reflexive and transitive. ♢

Example 2.40 The relations $\models$ and $\mathrel{|\!\!=}$ on the set of sentences formed from the language containing the propositions $\{p, q\}$ are shown in figure 2.2 for comparison. These figures are the Lindenbaum algebras of $\models$ and $\mathrel{|\!\!=}$. The nodes are the $\models$ (resp. $\mathrel{|\!\!=}$) equivalence classes, and the 'arrows'[2] are the relation $\models$ (resp. $\mathrel{|\!\!=}$). (We will prove in proposition 2.41 that the equivalence classes are the same for $\mathrel{|\!\!=}$ as for $\models$ — this justifies the second diagram.)

Thus: $p \wedge q \mathrel{|\!\!=} p$ and $p \wedge q \mathrel{|\!\!=} p \vee q$, but $p \wedge q \not\models p \leftrightarrow q$ and $p \not\models p \vee q$. Moreover, $\bot \mathrel{|\!\!=} \phi$ for all ϕ.

The definition of natural consequence is perhaps not very satisfying, because (one might ask), what is so special about preserving monotonicities? One way to answer this is purely pragmatic: as we will see, it is essential for the next definition, which does

[2]For TEXnical reasons the arrowheads are not shown in the diagram.

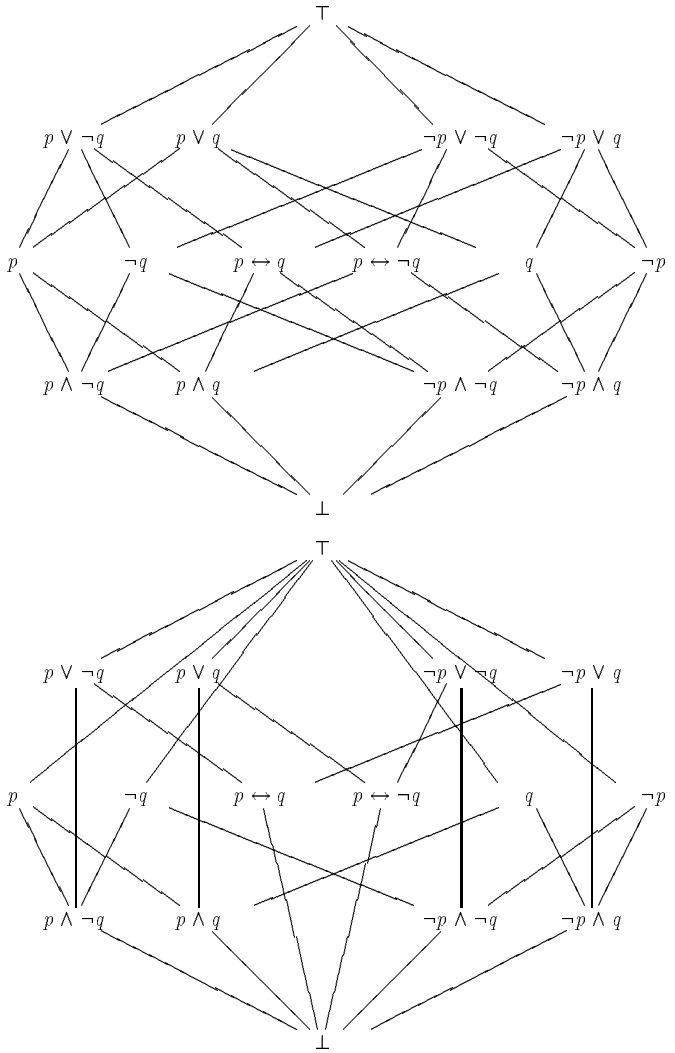


Figure 2.2: The *ordinary* and *natural* consequence relations over $\{p, q\}$

have a satisfying feel. But first, we justify the term *natural consequence* by showing examples of how much more natural this consequence really is.

Natural consequence is something like relevant consequence; it stops us adding irrelevant disjuncts in our conclusions. (This is not the same notion of relevance as Anderson/Belnap [2], for there one is interested in stopping irrelevant conjuncts in the premises.) The following sequents, which are ordinarily valid, are not naturally valid:

$$\begin{array}{lll} p \models p \vee q & p \models q \rightarrow p & p \wedge q \models p \leftrightarrow q \\ p \models p \vee \neg q & \neg p \models p \rightarrow q & \end{array}$$

There are well-known objections to the classical validity of these entailments, so it is rather pleasing that they are not naturally valid. Regarding the first pair, the premise p tells us nothing about q, and therefore it is suspect to introduce q or $\neg q$ as a disjunct. The second pair are the standard inelegancies of material implication, and are rejected by 'resource' logics like linear logic and relevance logics. Finally, we dislike $p \wedge q \models p \leftrightarrow q$ because the right-hand side suggests that p and q are in some way bound together, whereas the left-hand side only says that they are both true.

On the other hand, the simplicity of the definition and the fact that it is based on satisfaction by models ensures that there is nothing untoward going on. In particular, if ϕ and ψ are classically equivalent then they are naturally equivalent; indeed:

Proposition 2.41 $\phi \dashv\models \psi$ iff $\phi \Bumpeq \psi$.

($\phi \Bumpeq \psi$ means $\phi \models \psi$ and $\psi \models \phi$.)

Proof Suppose $\phi \dashv\models \psi$. Then, by proposition 2.35, $\phi^{\pm} = \psi^{\pm}$. Therefore, $\phi \Bumpeq \psi$. The converse is immediate from definition 2.38. ♢

Proposition 2.42 If $\phi \dashv\models \psi$ then $\phi \models \chi$ iff $\psi \models \chi$.

Proof Suppose $\phi \models \chi$. If $\phi \dashv\models \psi$ then $\psi \models \phi$ by proposition 2.41, and by proposition 2.39, $\psi \models \chi$. The converse is proved similarly. ♢

We can also examine the structural properties of $\models$. Clearly it is *substructural*, that is, it fails the usual properties of inclusion, monotonicity and cut:

Example 2.43 (See proposition 2.9 for the statement of the rules.)

1. Inclusion fails: $p \wedge (\neg p \vee q) \not\models \neg p \vee q$, since the left hand is equivalent to $p \wedge q$ and is monotonic in p, while the right hand is not.
2. Monotonicity fails: $\neg p \vee q \models \neg p \vee q$, but, as above, $p \wedge (\neg p \vee q) \not\models \neg p \vee q$.
3. Cut fails: Monotonicity is also built in to Cut, so the same example goes through. We have $p \wedge q \models \top$ and $\top \wedge (\neg p \vee q) \models \neg p \vee q$, but (cutting $\top$) we also have $(p \wedge q) \wedge (\neg p \vee q) \not\models \neg p \vee q$ (the left-hand side is equivalent to $p \wedge q$ and is monotonic in p, which the right-hand side is not).

We do, however, have their weak varieties:

Proposition 2.44 1. Reflexivity: $\phi \models \phi$.

2. Weak monotonicity [10]: $\dfrac{\phi \models \psi_1 \quad \phi \models \psi_2}{\phi \wedge \psi_1 \models \psi_2}$

3. Weak cut: $\dfrac{\phi \models \psi_1 \quad \phi \wedge \psi_1 \models \psi_2}{\phi \models \psi_2}$

Proof 1. (proposition 2.39.)

2. Suppose $\phi \models \psi_1$ and $\phi \models \psi_2$. By definition of $\models$, $\phi \models \psi_1$, so by classic properties $\phi \models \phi \wedge \psi_1$. But also, $\phi \wedge \psi_1 \models \phi$, and since $\phi \models \psi_2$, we ha $\phi \wedge \psi_1 \models \psi_2$ by proposition 2.42.

3. Suppose $\phi \models \psi_1$ and $\phi \wedge \psi_1 \models \psi_2$. By classical properties, $\phi \models \psi_2$. Now suppo $p \in \phi^{\pm}$. Then $p \in \psi_1^{\pm}$, since $\phi \models \psi_1$. Therefore, $p \in (\phi \wedge \psi_1)^{\pm}$, and sin $\phi \wedge \psi_1 \models \psi_2$, $p \in \psi_2^{\pm}$, thus proving $\phi \models \psi_2$.

2.2.4 The ordering $\sqsubseteq_\phi$ (definition)

Finally we can define $\sqsubseteq_\phi$. As expected, the definition is just like proposal 2.32, b with $\models$ instead of $\models$.

Definition 2.45 $M \sqsubseteq_\phi N$, if for each ψ,

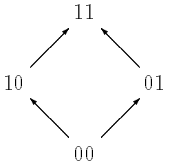
$$\phi \models \psi \Rightarrow (M \Vdash \psi \Rightarrow N \Vdash \psi)$$

Proposition 2.46 For each L-sentence ϕ, $\sqsubseteq_\phi$ is a pre-order.

Proof Reflexivity is obvious. For transitivity, suppose $L \sqsubseteq_\phi M \sqsubseteq_\phi N$, and let ψ such that $\phi \models \psi$ and $L \Vdash \psi$. Then, since $L \sqsubseteq_\phi M$, $M \Vdash \psi$. And since $M \sqsubseteq_\phi$ $N \Vdash \psi$.

Some examples of this ordering now follow. We omit the details except in the fir case; but the propositional examples have been checked by the Miranda program give in appendix A.

Example 2.47 Consider again the propositional language over $\{p, q\}$ and the inte pretations $\{00, 01, 10, 11\}$ as before. The ordering $\sqsubseteq_{p \wedge q}$ is as follows:



Thus, interpretations which satisfy p or q are better than that which satisfies neither nor q. To see that this is so, first consider the natural consequences of $p \wedge q$: they a $\{p \wedge q, p, q, \bot\}$. Since 00 satisfies none of these, it is $\sqsubseteq_{p \wedge q}$ everything else; 01, on t other hand, satisfies q so it is $\sqsubseteq_{p \wedge q}$ only others which satisfy q, namely itself and 1 An analogous argument holds for 10; and since 11 satisfies $p \wedge q$ it is $\sqsubseteq_{p \wedge q}$ only itsel

Example 2.48 If ϕ is just p, then $\sqsubseteq_\phi$ is as follows:

This is because the only natural consequences of p are $\top$ and p. Intuitively, either an interpretation satisfies p or it doesn't; there is no question of partial satisfaction. Notice that $\sqsubseteq_\phi$ is not necessarily antisymmetric. For here, 10 and 11 are equivalent as far as satisfying p is concerned, but they are not equal.

Example 2.49 $\sqsubseteq_{\neg p}$ is simply $\sqsubseteq_p$ turned upside down:

(The natural consequences of $\neg p$ are $\neg p$ and $\top$.) But $\sqsubseteq_{\neg(p \wedge q)}$ (or, equivalently, $\sqsubseteq_{\neg p \vee \neg q}$) bears little resemblance to $\sqsubseteq_{p \wedge q}$ (which was given in example 2.47):

(The natural consequences of $\neg(p \vee q)$ are itself and $\top$.)

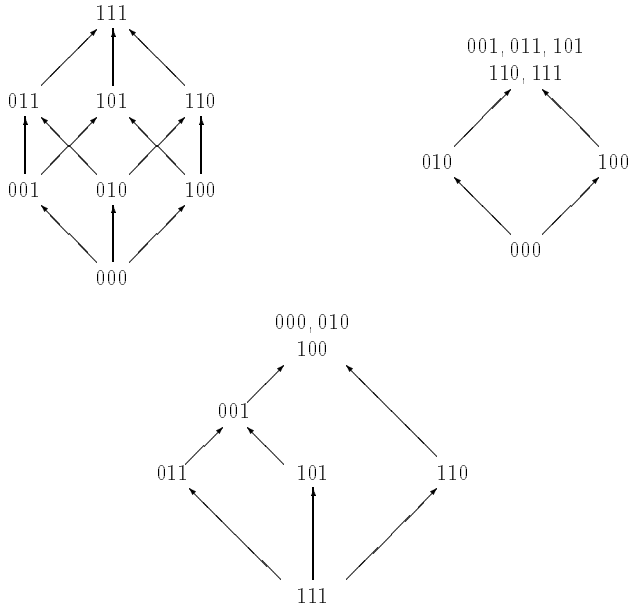
It should be clear that the ordering is only concerned with the interpretations which fail to satisfy the sentence in question.

Example 2.50 If ϕ is $\top$, then the ordering is the indiscrete one in which everything is equivalent, for no model is any better at satisfying $\top$ than any other. That is because they all satisfy it.

Example 2.51 If ϕ is $\bot$, the ordering is the discrete one in which nothing is related; we have $M \sqsubseteq_\_ N$ iff $M = N$. For suppose $M \neq N$; pick any ϕ such that $M \Vdash \phi$ and $N \nVdash \phi$. We have $\bot \models \phi$. Therefore, $M \not\sqsubseteq_\_ N$.

As far as the theory of OTPs is concerned, the difference between $\sqsubseteq_\top$ and $\sqsubseteq_\_$ is of no importance. The fact that their strict versions, $\sqsubset_\top$ and $\sqsubset_\_$, are both the empty relation is significant, and is what one would expect. The reader may be concerned about the fact that we stipulated that $\bot^+ = \bot^- = \emptyset$ in definition 2.34. The reason for this is simply that we thereby obtain $\bot \models \phi$ for all ϕ, and therefore $\bot$s position in the second diagram of figure 2.2. It is true that if we had not treated $\bot$ in any special way in definition 2.34 we would have obtained that $\bot \models \phi$ implies $\phi = \bot$ or $\phi = \top$; we then would have obtained that $\sqsubseteq_\_$ is the indiscrete ordering ($M \sqsubseteq_\_ N$ for all M, N) rather than the discrete one; but the rest of the theory of OTPs would remain the same. It turns out that $\bot$ has a rather unusual rôle in OTPs; we will return later to this topic (proposition 3.18). The relevant point here is that the question of how $\bot$ should be treated at this level has no significant impact.

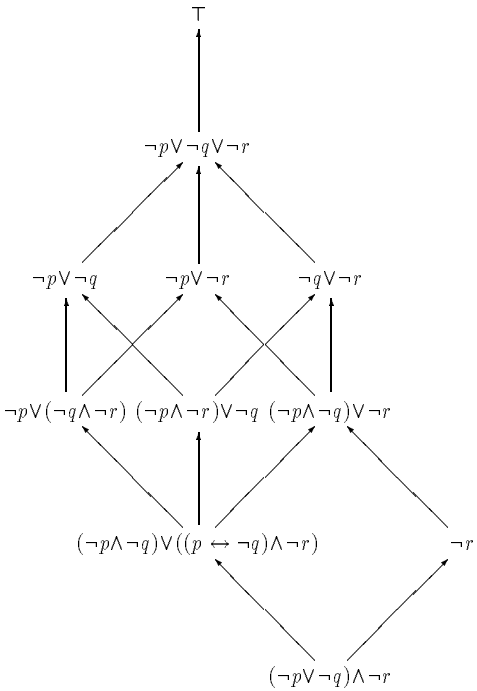
Example 2.52 The orderings $\sqsubseteq_{p \wedge q \wedge r}$, $\sqsubseteq_{(p \wedge q) \vee r}$ and $\sqsubseteq_{(\neg p \vee \neg q) \wedge \neg r}$ are



We will show the working for just the last of these three diagrams.

The positive and negative monotonicities of $(\neg p \vee \neg q) \wedge \neg r$ are respectively and $\{p, q, r\}$. Its natural consequences are therefore the sentences in the followir diagram; the diagram orders them by logical strength (that is, in this diagram tl

arrow means $\models$).



To derive the model ordering from this diagram, the definitions say in effect to consider each interpretation as the upwards-closed set of sentences it satisfies in this diagram. The model ordering is then given by the inclusion ordering on these sets. For example, to check that 101 should appear lower than 011 in the diagram (as it does), we must check that the natural consequences which 101 satisfies form a subset of those satisfied by 011. This is indeed so, since 101 satisfies $\{\top, \neg p \vee \neg q \vee \neg r, \neg p \vee \neg q, \neg q \vee \neg r, (\neg p \wedge \neg r) \vee \neg q\}$; and 001 additionally satisfies $\{\neg p \vee \neg r, \neg p \vee (\neg q \wedge \neg r), (\neg p \wedge \neg q) \vee \neg r, (\neg p \wedge \neg q) \vee ((p \leftrightarrow \neg q) \wedge \neg r)\}$.

Further examples, including ones in predicate logic, are given in §3.1.

We finish this subsection with a few definitions and results to reassure us that everything is according to plan:

Proposition 2.53 If $\phi \rightleftharpoons \psi$ then $\sqsubseteq_\phi = \sqsubseteq_\psi$.

Proof Suppose $M \sqsubseteq_\phi N$, and $\psi \models \chi$ and $M \Vdash \chi$. By proposition 2.42, $\phi \models \chi$, so $N \Vdash \chi$. Therefore, $M \sqsubseteq_\psi N$. The converse is proved similarly. $\diamondsuit$

Proposition 2.54 M is $\sqsubseteq_\phi$-maximum iff $M \Vdash \phi$.

Proof (If) If $M \Vdash \phi$ then $M \Vdash \psi$ whenever $\phi \models \psi$. Therefore, $N \sqsubseteq_\phi M$ for any N

(Only if) If $\phi = \bot$ then M is not maximum by the argument given in example 2.5 Suppose $\phi \neq \bot$ and $M \nVdash \phi$. We show that M is not $\sqsubseteq_\phi$-maximum. Let $N \Vdash$ We show that $M \sqsubset_\phi N$. (i) $M \sqsubseteq_\phi N$, since by the (If)-part N is $\sqsubseteq_\phi$-maximum. (i $N \not\sqsubseteq_\phi M$, since $\phi \models \phi$, $N \Vdash \phi$ and $M \nVdash \phi$.

Propositions 2.46 and 2.54 show that $\sqsubseteq_\phi$ satisfies assumption 2.16.

2.2.5 Summary of definitions for OTPs

To recap, we started with a logic given in terms of a language and a set of interpretatio in the standard way. Ordered presentations of theories in this logic consist of a poset points, each one labelled by a sentence in the language (definition 2.12). To define t models of ordered presentations, we first define, for each sentence ϕ in the languag an ordering on the interpretations written $\sqsubseteq_\phi$ (definition 2.45). $M \sqsubseteq_\phi N$ intuitive means that N satisfies ϕ at least as well as M. To define $\sqsubseteq_\phi$, we need the notion natural consequence (definition 2.38). Then we define the ordering $\sqsubseteq^\Gamma$ (definition 2.19 $M \sqsubseteq^\Gamma N$ intuitively means that N is as good as M at satisfying ?, taking account ?'s own ordering. Finally, models of ? are the $\sqsubseteq^\Gamma$-maximal elements, and consequen is defined in the standard way (definition 2.25).

Here is a summary, for reference:

1. We assume the underlying logic defines the notions of *satisfaction* (written $\Vdash$ and *positive and negative occurrence*.
2. $\phi \models\!\!\models \psi$ if $\phi \models \psi$ and $\phi^\pm \subseteq \psi^\pm$.
3. $M \sqsubseteq_\phi N$ if $\phi \models\!\!\models \psi$ implies ($M \Vdash \psi$ implies $N \Vdash \psi$).
4. $M \sqsubseteq^\Gamma N$ if $\forall x \in X. \exists y \in X. (M \not\sqsubseteq_x N$ implies $y \leqslant x$ and $M \sqsubset_y N)$.
5. $M \Vdash ?$ if M is $\sqsubseteq^\Gamma$-maximal.
6. $? \models \phi$ if, for all M, $M \Vdash ?$ implies $M \Vdash \phi$.

This chapter has, I hope, motivated and explained the definitions for ordered theo presentations. The next chapter considers some of their properties.

Chapter 3

Examples and Properties of OTPs

In the last chapter the definitions of ordered theory presentations and their semantics were given, in terms of an arbitrary logic defined in terms of interpretations and a satisfaction relation. In this chapter, some of the properties of these definitions are considered. An important result shows that there always *are* models of an OTP if the underlying logic is compact. This is done in §3.2 by showing that, for each ? , there are maximal interpretations in the $\sqsubseteq^{\Gamma}$ ordering. The question of how to add sentences to OTPs is examined in §3.3. We show that the operation extending an OTP by adding new sentences to its bottom has natural properties.

We begin with a section giving details of worked examples for propositional and predicate calculus.

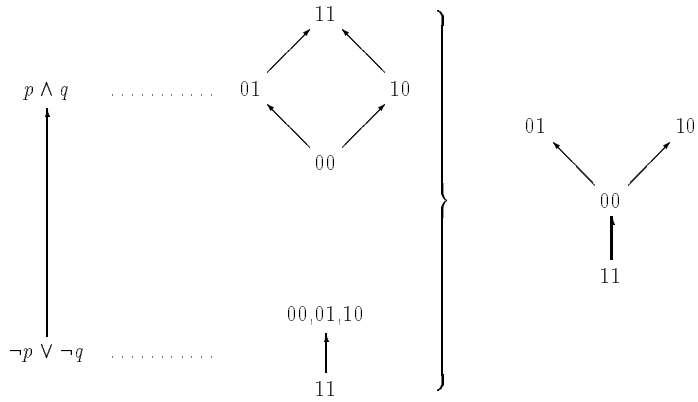
3.1 Worked examples

In §1.3 a number of examples were given which we expect our theory to satisfy, and indeed it does. In this section we recall some of the examples.

3.1.1 Examples in propositional logic

For each sentence we illustrate $\sqsubseteq_{\phi}$. Then we show $\sqsubseteq^{\Gamma}$, where ? is the whole presentation. The reader can check that the $\sqsubseteq^{\Gamma}$-maximal elements are precisely the models of the sentence claimed to be equivalent to the ordered presentation in §1.3. The notation of 0s and 1s was introduced in §2.2.1.

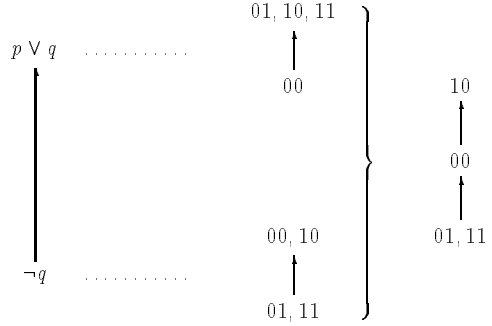
Example 3.1 (Example 1.3)



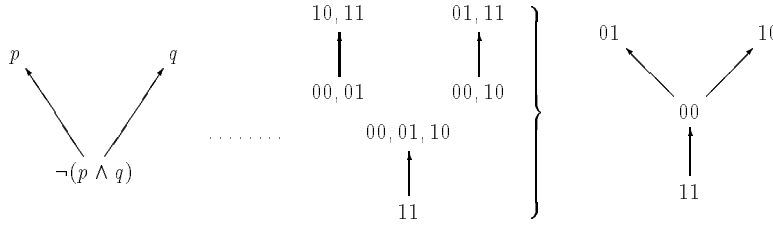
Cf. examples 2.47 and 2.49 (second diagram) for how the left-hand orderings are com
puted. They are put together to obtain the right-hand ordering by definition 2.19.

Again, it is worth emphasising that these diagrams can be computed by the co
given in appendix A. Having already given worked examples for $\sqsubseteq_{\phi}$ in the last chapte
we omit the working from the next three examples.

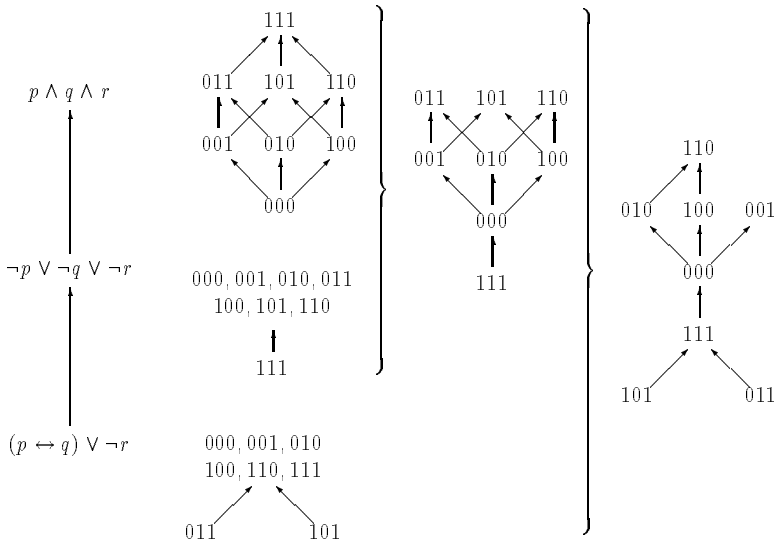
Example 3.2 (Example 1.5)



Example 3.3 (Example 1.6)



Example 3.4 (Example 1.12)



3.1.2 Examples in predicate logic

Let ? be the presentation

$$\begin{array}{c} \forall x.\, p(x) \\ \uparrow \\ \exists x.\, \neg p(x) \end{array}$$

We will show that the models of ? are the interpretations with precisely one element which is *not* in the extension of p.

The way to do this is to work out the orders $\sqsubseteq_{\forall x.\, p(x)}$ and $\sqsubseteq_{\neg\exists x.\, p(x)}$ (which we abbreviate to $\sqsubseteq_\forall$ and $\sqsubseteq_\exists$ respectively). This is done using definition 2.45. Then $\sqsubseteq^\Gamma$ is obtained from this by definition 2.19. We will restrict our attention to countable models.

Notation 3.5 For all a, b in $\{1, 2, 3, \ldots, \omega\}$, let the expression (a, b) denote the class of interpretations with $a + b$ elements, of which a satisfy p and b do not.

We start with the order $\sqsubseteq_\forall$. To compute this we are interested in the natural consequences of $\forall x.\, p(x)$.

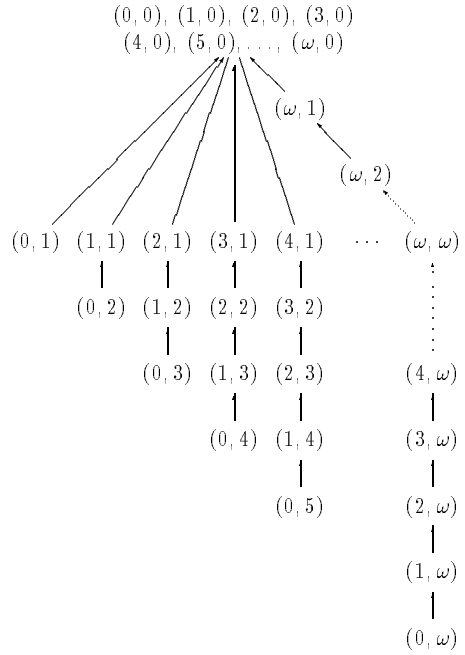
Lemma 3.6 If ϕ is a non-tautologous natural consequence of $\forall x.\, p(x)$ then it can be expressed in the form

$$\forall \underline{x}_1 \exists \underline{x}_2 \forall \underline{x}_3 \exists \underline{x}_4 \ \cdots .\, \psi$$

where each $\underline{x}_i$ is a tuple of variables and ψ is quantifier-free and contains only positive occurrences of p.

Proof Consider ϕ in *prenex form*, that is, with all quantifiers at the beginning. Eve first-order sentence can be written in this form [31, proposition 4.28]. Since ϕ is consequence of $\forall x.\, p(x)$ it must begin with a $\forall$. Since $p \in (\forall x.\, p(x))^+$, we have $p \in \psi$ If $p \in \psi^-$ then ψ can be written with no occurrences of p and if $p \notin \psi^-$ then ψ can b written with only positive occurrences of p (by definition 2.34).

Proposition 3.7 The order $\sqsubseteq_\forall$ is the following.



Proof It is sufficient to show:

1. If $n \neq \omega$ or $m \neq \omega$ then $(n, m) \sqsubset_\forall (n + 1, m \perp 1)$.

   Suppose $M \in (n, m)$ and $N \in (n + 1, m \perp 1)$. Let $f : M \to N$ be a bijecti function which maps all elements satisfying p in M to elements satisfying p M. Then there is precisely one element, say $a \in M$, not satisfying p in M b such that $f(a)$ satisfies p in N.

   Suppose ϕ is a natural consequence of $\forall x.\, p(x)$ and $M \Vdash \phi$. We must show th $N \Vdash \phi$. Consider ϕ in the form specified in lemma 3.6. We must show th for any tuple $\underline{b}_1$ of elements of N there is a tuple $\underline{b}_2$ of elements, such that f any tuple $\underline{b}_3$ there is ... such that $\psi[\underline{b}_i/\underline{x}_i]$. Consider such a tuple $\underline{b}_1$, and l $\underline{a}_1 = f^{-1}(\underline{b}_1)$. Since $M \Vdash \phi$, we can find a tuple $\underline{a}_2$ such that

   $$M \Vdash \forall \underline{x}_3 \exists \underline{x}_4 \ \cdots .\, \psi[\underline{a}_1/\underline{x}_1, \underline{a}_2/\underline{x}_2]$$

Let $\underline{b}_2 = f(\underline{a}_2)$. Similarly, given $\underline{b}_3$ we can find $\underline{b}_4$ by mapping into M and back. Proceed in this way until all the quantifiers have been dealt with.

Since $\psi[\underline{a}_i/\underline{x}_i]$ is true in M and p only occurs positively in ψ, $\psi[\underline{a}_i/\underline{x}_i]$ cannot assert $\neg p(a)$. Therefore, $\psi[\underline{b}_i/\underline{x}_i]$ is true in N. But the tuples $\underline{b}_1$, $\underline{b}_3, \ldots$ were arbitrary; therefore, $N \Vdash \phi$.

This shows that $M \sqsubseteq_\forall N$. To show $M \sqsubset_\forall N$ we have additionally to exhibit a natural consequence satisfied by N but not by M. If $m \neq \omega$, such a one is

$$\forall x_1, x_2, \ldots, x_m. \left(\bigwedge_{i \neq j} x_i \neq x_j \rightarrow \bigvee_k p(x_k) \right)$$

which says that in any selection of m distinct elements, one must satisfy p. If $n \neq \omega$, we can take

$$\forall x_1, \ldots x_{n+1}. \exists y_1, \ldots y_{n+1}. \left(\bigwedge_{i \neq j} x_i \neq x_j \rightarrow \bigwedge_{h \neq k} y_h \neq y_k \wedge \bigwedge_l p(y_l) \right)$$

which says that if there are $n+1$ distinct elements then there are $n+1$ distinct ones satisfying p. It is not hard to verify that these two sentences are indeed natural consequences of $\forall x.\, p(x)$.

2. If $n + m \neq n' + m'$ and $n, n' > 0$ then (n, m) and (n', m') are incomparable in the ordering.

   Suppose $M \in (n, m)$ and $N \in (n', m')$. We can exhibit a natural consequence of $\forall x\, p(x)$ satisfied by M and not by N, and another satisfied by N but not M. First, let us adopt the notation that

   $$\text{size} \geqslant n \qquad \text{abbreviates} \qquad \exists x_1, \ldots x_n. \left(\bigwedge_{i \neq j} x_i \neq x_j \right).$$

   This formula expresses the fact that there are at least n elements. Additionally, let

   $$\text{size} = n \qquad \text{abbreviate} \qquad \text{size} \geqslant n \wedge \neg(\text{size} \geqslant n+1)$$

   Then we have, for any n,

   $$\forall x.\, p(x) \models\!\!\!\sim \forall x.\, p(x) \vee \text{size} = n$$

   We also have

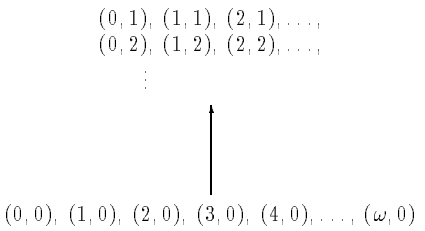
   $$M \Vdash \forall x.\, p(x) \vee \text{size} = n + m, \qquad \text{but} \qquad N \not\Vdash \forall x.\, p(x) \vee \text{size} = n + m$$

   In the former case M satisfies the second disjunct. In the latter, N fails both the first disjunct (since $n' > 0$) and the second (since $n' + m' \neq n + m$). On the other hand, we also have for similar reasons:

   $$M \not\Vdash \forall x.\, p(x) \vee \text{size} = n' + m', \qquad \text{but} \qquad N \Vdash \forall x.\, p(x) \vee \text{size} = n' + m'$$

   ♢

Proposition 3.8 The order $\sqsubseteq_\exists$ is the following.



Proof Easy.

Now note that if

$$? = \begin{array}{c} \phi \\ \uparrow \\ \psi \end{array}$$

then it follows from proposition 2.30(4) and remark 2.31 that

$$M \sqsubset^\Gamma N \text{ iff } M \sqsubset_\psi N \text{ or } (M \sqsubset_\phi N \text{ and } M \sqsubseteq_\psi N)$$

Proposition 3.9 Let $?$ be

$$\begin{array}{c} \forall x.\, p(x) \\ \uparrow \\ \exists x. \neg p(x) \end{array}$$

as above. Then for each n, the interpretation $(n, 1)$ is maximal.

Proof Suppose $(n, 1) \sqsubset^\Gamma (a, b)$. Then **either** $(n, 1) \sqsubset_\exists (a, b)$, which is impossib by inspection of the diagram; **or** $(n, 1) \sqsubseteq_\exists (a, b)$ and $(n, 1) \sqsubset_\forall (a, b)$. The second these conditions implies $b = 0$, which contradicts the first. Therefore $(n, 1) \sqsubset^\Gamma (a, b)$ contradictory, therefore $(n, 1)$ is maximal.

3.2 Existence of models for OTPs

As stated, models of an ordered presentation $?$ are $\sqsubseteq^\Gamma$-maximal interpretations of th language of $?$. When is it possible to find such maximal interpretations? In th section we show that, if the underlying logic is compact, *every* ordered presentatio has a model.

First, it is worth noting that there are simple cases of ordered presentations wi no models, when compactness fails.

Example 3.10 Let $?$ be the OTP

$$\begin{array}{c} \forall x.\, p(x) \\ \uparrow \\ \text{domain is infinite} \wedge \\ [\![p]\!] \text{ is finite} \end{array}$$

The bottom sentence says that the domain of individuals is infinite, but that only finitely many of its elements satisfy the predicate p. But the top sentence says that *all* the individuals must satisfy p. These are sentences in second order predicate logic; it is not possible to express finiteness of the interpretation of a predicate or infiniteness of the domain in first order logic. (For details of how precisely to state these constraints in second order logic, see [73].)

There are no models of this theory, because every candidate model M can be improved to obtain an interpretation which is closer to being a model, *ad infinitum*. That is to say, for all $M \in \mathcal{M}$ there is an $N \in \mathcal{M}$ such that $M \sqsubset^\Gamma N$. To see this, suppose M pretends to be a model of $?$.

- If the domain of individuals of M is finite, then construct N by adding infinitely many new individuals which do not satisfy p.
- If $M[\![p]\!]$ is infinite, then construct N from M by using the same domain but removing all but finitely many elements from $[\![p]\!]$.
- If $M[\![p]\!]$ is finite but the domain is infinite, then N is obtained by adding one more element to $[\![p]\!]$.

In each of these cases, $M \sqsubset^\Gamma N$.

Now we turn to the proof that if the underlying logic is compact (which second-order logic is not), then every ordered presentation has a model. The proof strategy is to use Zorn's lemma to find $\sqsubseteq^\Gamma$-maximal interpretations.

Let L be a language and $\langle \mathcal{M}, \Vdash \rangle$ its interpretation system, and let $? = \langle X, \leqslant, F \rangle$ be an ordered presentation over L.

Definition 3.11 The logic $\langle L, \mathcal{M}, \Vdash \rangle$ is *compact* if for all sets of sentences $\Phi \subseteq L$, Φ has a model if each of its finite subsets has a model.

Definition 3.12 For each M, N in $\mathcal{M}$, the (M, N)-frontier, written $\mathrm{fr}(M, N)$, is the set of minimal elements of the set $\{x \in X \mid M \not\equiv_x N\}$.

Lemma 3.13 For all $M, N \in \mathcal{M}$ and $x \in X$, either $M \equiv_x N$ or $\exists y \leqslant x.\ y \in \mathrm{fr}(M, N)$.

Proof By lemma 2.14, $\{x \in X \mid M \not\equiv_x N\}$ has minimal elements. ♢

Lemma 3.14 $M \sqsubset^\Gamma N$ iff $\mathrm{fr}(M, N) \neq \emptyset$ and $\forall x \in \mathrm{fr}(M, N).\ M \sqsubset_x N$.

Proof (If) First we show $M \sqsubseteq^\Gamma N$. Suppose $x \in X$ with $M \not\sqsubseteq_x N$. By lemma 3.13, $\exists y \in \mathrm{fr}(M, N)$ with $y \leqslant x$. By hypothesis, $M \sqsubset_y N$. Next, we show $N \not\sqsubseteq^\Gamma M$. Let $x \in \mathrm{fr}(M, N)$. Then $N \not\sqsubseteq_x M$, but for each $y < x$, $M \equiv_y N$.

(Only if) If $\mathrm{fr}(M, N) = \emptyset$ then $M \equiv^\Gamma N$, a contradiction. Let $x \in \mathrm{fr}(M, N)$. Either $M \not\sqsubseteq_x N$ or $N \not\sqsubseteq_x M$. In the former case, $\exists y \leqslant x$ with $M \sqsubset_y N$; since $x \in \mathrm{fr}(M, N)$, y must equal x. In the latter case, $N \not\sqsubseteq_x M$ and if $M \sqsubseteq_x N$ then $M \sqsubset_x N$. Therefore, in both cases $M \sqsubset_x N$ as required. ♢

Lemma 3.15 Let $?$ be a finite OTP and $\mathcal{N}$ be a non-empty chain in $\mathcal{M}$ with n
maximal element (i.e. for every $M, N \in \mathcal{N}$, if $M \neq N$ then $M \sqsubset^\Gamma N$ or $N \sqsubset^\Gamma M$; ar
for each $M \in \mathcal{N}$ there is an $N \in \mathcal{N}$ such that $M \sqsubset^\Gamma N$). There is a non-empty s
$Y \subseteq X$ and a non-empty chain $\mathcal{L} \subseteq \mathcal{N}$ such that

1. For each $a \in Y$ and $M, N \in \mathcal{L}$, if $M \sqsubset^\Gamma N$ then $M \sqsubseteq_a N$; and
2. For each $a \in Y$ and $M \in \mathcal{L}$ there exists $P \in \mathcal{L}$ such that $M \sqsubset^\Gamma P$ and $M \sqsubset_a P$

Proof Let $X' = \{x \in X \mid \forall M \in \mathcal{N}\ \exists M_1, M_2 \in \mathcal{N}\ (M \sqsubset^\Gamma M_1 \sqsubset^\Gamma M_2 \text{ and } x \in \mathrm{fr}(M_1, M_2))\}$.

If $X = X'$, let $\mathcal{L} = \mathcal{N}$. Otherwise, for each $x \in X \perp X'$ let M_x be such that, for a
$M_1, M_2 \in \mathcal{N}$, if $M_x \sqsubset^\Gamma M_1 \sqsubset^\Gamma M_2$ then $x \notin \mathrm{fr}(M_1, M_2)$. That such an M_x can be foun
follows immediately from the definition of X'. Let $M_X = \max(\{M_x \mid x \in X \perp X'\}$
we can take this maximum because X is finite; and let $\mathcal{L} = \{M \in \mathcal{N} \mid M_X \sqsubseteq^\Gamma M$
$\mathcal{L} \neq \emptyset$ since $M_X \in \mathcal{L}$.

Thus, whether $X = X'$ or not, we have that $\mathcal{L} \neq \emptyset$. Also, $\mathcal{L}$ is upwards close
(i.e. for all $M, N \in \mathcal{N}$, $M \in \mathcal{L}$ and $M \sqsubseteq^\Gamma N$ imply $N \in \mathcal{L}$). Let $M_1, M_2 \in \mathcal{L}$ wit
$M_1 \neq M_2$. Then either $M_1 \sqsubset^\Gamma M_2$ or $M_2 \sqsubset^\Gamma M_1$. In either case, $\mathrm{fr}(M_1, M_2) \neq \emptyset$. Bu
$\mathrm{fr}(M_1, M_2) \subseteq X'$, so $X' \neq \emptyset$. Let Y be the minimal points of X'.

1. Suppose $a \in Y$, $M, N \in \mathcal{L}$, and $M \sqsubset^\Gamma N$. If $a \in \mathrm{fr}(M, N)$ then $M \sqsubset_a N$.
   $a \notin \mathrm{fr}(M, N)$ and $M \not\sqsubseteq_a N$ then $\exists y \in \mathrm{fr}(M, N).\ y \leqslant a$ by lemma 3.13, so $a \notin Y$
   a contradiction.
2. Suppose $a \in Y$ and $M \in \mathcal{L}$. Since $a \in X'$, $\exists M_1, M_2.\ M \sqsubset^\Gamma M_1 \sqsubset^\Gamma M_2$ and a
   $\mathrm{fr}(M_1, M_2)$. Since $M \sqsubseteq^\Gamma M_1 \sqsubseteq^\Gamma M_2$, $M \sqsubseteq_a M_1 \sqsubseteq_a M_2$; and since $a \in \mathrm{fr}(M_1, M_2$
   we have $M \sqsubseteq_a M_1 \sqsubset_a M_2$. Let $P = M_2$.

Lemma 3.16 If $\langle L, \mathcal{M}, \Vdash \rangle$ is compact and $?$ is finite then for each $M \in \mathcal{M}$, the
exists $N \in \mathcal{M}$ such that $M \sqsubseteq^\Gamma N$ and N is $\sqsubseteq^\Gamma$-maximal.

Proof Let $M \in \mathcal{M}$. We show that $\{N \mid M \sqsubseteq^\Gamma N\}$ has maximal elements. Let
be a non-empty chain in that set. By **Zorn's lemma** it suffices to show that eve
such chain has an upper bound. If $\mathcal{N}$ has a maximal element, that element is also a
upper bound. Suppose, then, that $\mathcal{N}$ does not have a maximal element. Let Y and
be as given by lemma 3.15. Let $Z = Y \cup \{x \in X \mid \forall y \in Y.\ y \not\leqslant x\}$. We now show tha
for each $x \in Z$ and $M, N \in \mathcal{L}$, $M \sqsubseteq^\Gamma N$ implies $M \sqsubseteq_x N$. If $x \in Y$, this follows fro
lemma 3.15 part 1. If $x \in Z \perp Y$, then $\forall y \in Y.\ y \not\leqslant x$ by definition of Z. By lemm
3.13, $\exists y' \leqslant x.\ y' \in \mathrm{fr}(M, N) \subseteq X'$, so $\exists y \in Y.\ y \leqslant y'$, a contradiction.

For each $M \in \mathcal{L}$ let M^* be $\{\psi \mid M \Vdash \psi \text{ and } \exists x \in Z.\ F(x) \not\models \psi\}$. M^* has a mode
since it has M as a model. Also, $M \sqsubset^\Gamma N$ implies $M^* \subseteq N^*$. For suppose $\psi \in M$
Then $M \Vdash \psi$, and there is an $x \in Z$ s.t. $F(x) \not\models \psi$. Since $M \sqsubseteq_x N$, we have $N \Vdash$
Therefore, $\psi \in N^*$.

Let $\Phi = \bigcup_{M \in \mathcal{L}} M^*$. Φ has a model, since every M^* and therefore every finite subs
of Φ has a model, and the underlying logic is compact. Let $K \Vdash \Phi$. It remains
show that $\forall M \in \mathcal{L}.\ M \sqsubseteq^\Gamma K$, i.e. that K is an upper bound. Since $\mathcal{L}$ is a non-emp
upwards-closed subchain of $\mathcal{N}$, it is sufficient to consider the case $M \in \mathcal{L}$. Let $M \in$

The fact that $M^* \subseteq \Phi$ implies that for each $x \in Z$, $M \sqsubseteq_x K$. Suppose $M \not\sqsubseteq_x K$. Then $x \notin Z$. We require that $M \sqsubset_y K$ for some $y \leqslant x$. Since $x \notin Z$, $\exists y \in Y.\, y \leqslant x$. We now show that $M \sqsubset_y K$ for every $y \in Y$, completing the proof. By lemma 3.15, pick P such that $M \sqsubset^\Gamma P$ and $M \sqsubset_y P$. It suffices to show that $P \sqsubseteq_y K$. Suppose $F(y) \models \psi$ and $P \Vdash \psi$. Then $\psi \in P^*$, so $\psi \in \Phi$, so $K \Vdash \psi$. ♢

As an immediate corollary, we get:

Proposition 3.17 Every finite ordered presentation ? over a compact logic has a model.

Proof By lemma 3.16, $\sqsubseteq^\Gamma$ has maximal elements. ♢

A consequence of this result is that contradictions can never be derived from an ordered presentation, not even one with the contradictory sentence in it! Indeed, *nothing* can be derived from the theory with one sentence which is $\bot$. That is because every interpretation is a model of that theory. This may come as a surprise, but really it is quite rational.

Proposition 3.18 If $? \models \phi$ then $\phi \neq \bot$.

Proof Let $M \Vdash ?$. Since $M \Vdash \phi$, $\phi \neq \bot$. ♢

Our policy about $\bot$ in the definitions in this thesis has been: "let $\bot$ do what it wants". That is to say, we have tried to avoid giving $\bot$ any special treatment in the definitions; a consequence of this is that it has perhaps surprising properties in the theorems. If it had turned out that $\bot$ had positively unpleasant properties one might be inclined to return to the definitions and try to change them to avoid those properties. As it is, $\bot$ has turned out completely benign: we cannot derive anything from it, and it makes no difference to an OTP no matter where it is placed within it.

Proposition 3.19 If "$\bot$" $\models \phi$ then $\phi = \top$.

Proof Every M is $\sqsubseteq_-$-maximal (example 2.51), so is $\sqsubseteq^{\text{"}\bot\text{"}}$-maximal (remark 2.31). ♢

3.3 Adding information to OTPs

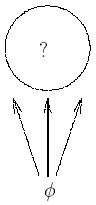
A natural way to add information to an ordered presentation is to add it at the bottom. This is not the only way, but it is obviously one with many interesting properties. Other ways of adding information will be considered in chapter 6.

Definition 3.20 Let $? = \langle X, \leqslant, F\rangle$ be an ordered theory presentation, let $0 \notin X$ and let ϕ be a sentence. The ordered presentation $? * \phi$ is $\langle X', \leqslant', F'\rangle$ where

1. $X' = X \cup \{0\}$,
2. $\leqslant' = \leqslant \cup \{(0, x) \mid x \in X'\}$, and

3. $F'(x) = \begin{cases} \phi & \text{if } x = 0 \\ F(x) & \text{otherwise} \end{cases}$

This situation is graphically illustrated as follows:



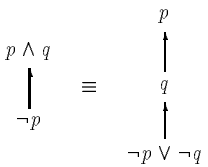
Definition 3.21 Let ? and Δ be OTPs.

1. ? and Δ are *statically equivalent*, written $? \equiv \Delta$, if they have the same extensio

$$? \equiv \Delta \text{ if for all } M, (M \Vdash ? \text{ iff } N \Vdash \Delta).$$

2. ? and Δ are *dynamically equivalent*, if, for all ϕ, $? * \phi \equiv \Delta * \phi$.

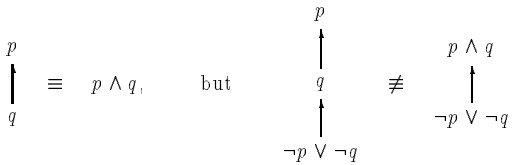
Example 3.22



Compare examples 1.2 and 1.4.

Dynamic equivalence implies static equivalence, but the converse is not so as t
following example shows.

Example 3.23



Proposition 3.24 $M \sqsubseteq^{\Gamma * \phi} N$ iff $M \sqsubset_\phi N$ or $(M \sqsubseteq_\phi N$ and $M \sqsubseteq^\Gamma N)$.

Proof Proposition 2.30 and remark 2.31.

Corollary 3.25 $M \sqsubset^{\Gamma * \phi} N$ iff $M \sqsubset_\phi N$ or $(M \sqsubseteq_\phi N$ and $M \sqsubset^\Gamma N)$.

If $? \models \phi$ we would not expect that revising $?$ by ϕ should change the set of models:

Proposition 3.26 If $\langle L, \mathcal{M}, \Vdash\rangle$ is compact and $?$ is finite then $? \models \phi$ implies $? \equiv ? * \phi$.

Proof Suppose $M \Vdash ?$ and $M \not\Vdash ? * \phi$. Since $M \Vdash ?$ and $? \models \phi$, $M \Vdash \phi$ (definition 2.25). Since $M \not\Vdash ? * \phi$, there is an N such that $M \sqsubset^{\Gamma * \phi} N$. By proposition 3.24,

- either $M \sqsubset_\phi N$, a contradiction since $M \Vdash \phi$ (proposition 2.54);
- or $M \sqsubseteq_\phi N$ and $M \sqsubset^\Gamma N$, contradicting $M \Vdash ?$.

Conversely, suppose $M \Vdash ? * \phi$ and $M \not\Vdash ?$. By lemma 3.16, take N such that $M \sqsubset^\Gamma N$ and N is $\sqsubseteq^\Gamma$-maximal, i.e. $N \Vdash ?$. By $N \Vdash ?$ and definition 2.25, $N \Vdash \phi$. By proposition 2.54, $M \sqsubseteq_\phi N$. Therefore, by proposition 3.24, $M \sqsubset^{\Gamma * \phi} N$, contradicting $M \Vdash ? * \phi$. ♢

Let $[\![\phi]\!] = \{M \mid M \Vdash \phi\}$.

Proposition 3.27 If $\langle L, \mathcal{M}, \Vdash\rangle$ is compact and $?$ is finite and $\phi \neq \bot$ then $M \Vdash ? * \phi$ iff M is $\sqsubseteq^\Gamma$-maximal in $[\![\phi]\!]$.

Proof (If.) We show that $M \not\Vdash ? * \phi$ implies M is not $\sqsubseteq^\Gamma$-maximal in $[\![\phi]\!]$. Suppose $M \not\Vdash ? * \phi$. Then $M \sqsubset^{\Gamma * \phi} N$ for some N. By lemma 3.16 we can take such an N such that $N \Vdash ? * \phi$, and by proposition 2.27, $N \Vdash \phi$, i.e. $N \in [\![\phi]\!]$. By corollary 3.25, either $M \sqsubset_\phi N$, in which case $M \notin [\![\phi]\!]$, or $M \sqsubset^\Gamma N$, in which case M is not $\sqsubseteq^\Gamma$-maximal in $[\![\phi]\!]$.

(Only if.) Suppose M is not $\sqsubseteq^\Gamma$-maximal in $[\![\phi]\!]$. If $M \notin [\![\phi]\!]$, pick any $N \in [\![\phi]\!]$. If $M \in [\![\phi]\!]$ and is not $\sqsubseteq^\Gamma$-maximal, pick $N \in [\![\phi]\!]$ with $M \sqsubset^\Gamma N$. In either case, we have $M \sqsubset^{\Gamma * \phi} N$ (by corollary 3.25), and so $M \not\Vdash ? * \phi$. ♢

We also obtain what we might loosely describe as weak analogues of proposition 2.9:

Proposition 3.28

1. Weak inclusion: if $\phi \neq \bot$ then $? * \phi \models \phi$

If $\langle L, \mathcal{M}, \Vdash\rangle$ is compact and $?$ is finite then

2. Weak monotonicity: $\dfrac{? \models \phi \quad ? \models \psi}{? * \phi \models \psi}$
3. Weak cut: $\dfrac{? * \phi \models \psi \quad ? \models \phi}{? \models \psi}$

Proof

1. Follows from proposition 2.27.
2. and 3. Follow from proposition 3.26.

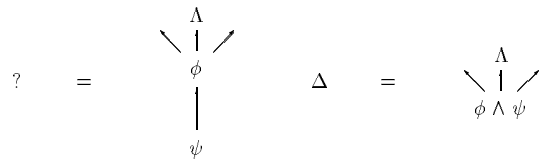
♢

These principles are accepted as being requirements which a default system should have (see for example [10, 46]).

Proposition 3.29 Suppose the underlying logic is compact. Let $? = \langle X, \leqslant, F_X\rangle$ l
a finite OTP and ϕ and ψ be mutually consistent sentences such that $\{1, 2\} \subseteq X$ ar
$\leqslant|_{\{1,2\}} = \{(1,1), (1,2), (2,2)\}$ and $F(1) = \psi$ and $F(2) = \phi$; and 1 is minimum in
and 2 is minimum in $X \perp \{1\}$.

Let $\Delta = \langle Y, \leqslant|_Y, F_Y\rangle$ be such that $Y = X \perp \{1\}$, and $F_Y(2) = \phi \wedge \psi$ and $F_Y(x)$
$F_X(x)$ if $x \neq 2$. Then $? \equiv \Delta$.

Graphically, this seemingly complicated state of affairs is simply illustrated:



Compare requirement 4 in §1.3.1.

Proof Let $Z = X \perp \{1, 2\}$ and $\Lambda = \langle Z, \leqslant|_Z, F|_Z\rangle$. ($\Lambda$ is shown in the diagram.) W
have (by corollary 3.25):

(A) $M \sqsubseteq^\Gamma N$ iff $M \sqsubset_\psi N$ or ($M \sqsubseteq_\psi N$ and ($M \sqsubset_\phi N$ or ($M \sqsubseteq_\phi N$ and $M \sqsubseteq^\Lambda N$))

(B) $M \sqsubseteq^\Delta N$ iff $M \sqsubset_{\phi\wedge\psi} N$ or ($M \sqsubseteq_{\phi\wedge\psi} N$ and $M \sqsubseteq^\Lambda N$)

We will use the following **intermediate result**: if $M \Vdash \phi \wedge \psi$ then the following a
equivalent:

1. $M \sqsubseteq^\Gamma N$;
2. $M \sqsubseteq^\Delta N$;
3. $M \sqsubseteq^\Lambda N$ and $N \Vdash \phi \wedge \psi$.

Proof: First we note that by hypothesis $M \Vdash \phi$ and $M \Vdash \psi$. By proposition 2.
we have $M \not\sqsubset_\phi N$, $M \not\sqsubset_\psi N$, and $M \not\sqsubset_{\phi\wedge\psi} N$. ($1 \Rightarrow 2$) By (A), we have (in view
the forgoing) that $M \sqsubseteq_\phi N$, $M \sqsubseteq_\psi N$ and $M \sqsubseteq^\Lambda N$. Therefore $N \Vdash \phi$ and $N \Vdash$
Therefore by (B) we have $M \sqsubseteq^\Delta N$. ($2 \Rightarrow 3$) By (B) we have $M \sqsubseteq_{\phi\wedge\psi} N$ and $M \sqsubseteq^\Lambda$
the former assures $N \Vdash \phi \wedge \psi$. ($3 \Rightarrow 1$) We have $M \sqsubseteq_\phi N$ and $M \sqsubseteq_\psi N$ since M ar
N both satisfy $\phi \wedge \psi$, so by (A), $M \sqsubseteq^\Gamma N$.

Now suppose $M \Vdash ?$. We will show $M \Vdash \Delta$. Suppose $M \sqsubseteq^\Delta N$; we will sho
$N \sqsubseteq^\Delta M$. By proposition 2.27, $M \Vdash \psi$. Also, $M \Vdash \phi$. For suppose not; then pick a
$P \Vdash \phi \wedge \psi$ (since they are consistent). Then $M \sqsubseteq_\psi P$ and $M \sqsubset_\phi P$, and so $M \sqsubset^\Gamma$
by (A), a contradiction. By the intermediate result, $M \sqsubseteq^\Gamma N$ and $N \Vdash \phi \wedge \psi$. B
$M \Vdash ?$, so $N \sqsubseteq^\Gamma M$; so again by the intermediate result, $N \sqsubseteq^\Delta M$.

Conversely, suppose $M \Vdash \Delta$; we will show $M \Vdash ?$. Again, if we suppose $M \sqsubseteq^\Gamma N$
is sufficient to show $N \sqsubseteq^\Gamma M$. By proposition 2.27, $M \Vdash \phi \wedge \psi$; so by the intermedia
result, $M \sqsubseteq^\Delta N$ and $N \Vdash \phi\wedge\psi$. But $M \Vdash \Delta$, so $N \sqsubseteq^\Delta M$; so again by the intermedia
result, $N \sqsubseteq^\Gamma M$.

Chapter 4

Belief revision

4.1 Introduction

The central question in belief revision is the following: given a *belief state* and a *sentence*, how should one obtain a new belief state in which the sentence is *true*, but which preserves as much of the old belief state as possible? In other words, one wants a function

$$* : \text{belief states} \times \text{sentences} \longrightarrow \text{belief states}$$

such that

1. ϕ is true in $? * \phi$; that is, the revision has been *effective*; and
2. given this constraint, $? * \phi$ contains 'as much' of $?$ as is consistent; that is, old beliefs persist through revisions if they can.

The case of interest, of course, is that in which $\neg\phi$ is true in $?$, so that the revision is more than just *refinement*, or the addition of compatible information. We also hope that

3. $? * \phi$ does not contain any extraneous information which was present in neither $?$ nor ϕ.

In the above requirements, some things are easy to formulate and some are not. We assume that any satisfactory representation of belief states comes with a function $|\cdot|$ which takes a belief state and returns the set of sentences true in it. $|?|$ is called the *extension* of $?$. But formalising the 'as much' requirement and the requirement of no extraneous information (numbers 2 and 3) is not so easy, and is the subject of this chapter.

Belief revision has obvious applications in artificial intelligence (eg. robotics), computer science (eg. deductive databases—see eg. [14]), the philosophy of science, social theory and so on. It also has applications beyond the idea of revising 'beliefs'. For example, in specification theory and in AI, there is the well-known frame problem to do with the semantics of actions. Given (the representation of) a state of a system and the post-condition of an action performed in the state, what is the state which results from performing the action? The same requirements on the revision function

apply here too: the post-condition should be true in the resulting state, which (give this constraint) should preserve as much of the original state as possible.

We begin the next section by describing the standard theory of belief revisio known as the AGM theory. The AGM theory suffers from several disadvantages. O is that it represents belief states as *infinite* objects, namely deductively closed sets sentences. Another is that existing belief revision models make *too strong assumptio* about what information is available to guide revisions. A consequence of this is th repeated revisions are impossible.

Not surprisingly, we will advocate *ordered theory presentations* to represent beli states. The revision function will usually be the one which simply puts the revisi sentence in the most prominent position of the belief state (we say 'usually', becau there will be a special case to consider). We will be interested only in *linear* OTP that is, those in which the partial order is in fact total. In view of this, we can use simpler notation for them; a linear OTP is simply a list of sentences.

The result is a finite representation of belief states. The revision operator is show to satisfy some, but not all, of the AGM postulates. The counterexamples for the AG postulates which fail are motivated. The important point is that no information oth than that encoded in the OTP is needed to effect the revision; this makes repeate revision easy.

The remainder of the chapter is organised as follows. We look at the AGM theo in the next section, and find it to be wanting. Criteria for belief revision are set up §4.3, and also the axioms are rewritten in a more general form which allows compariso with systems of belief revision which do not model belief states as theories. Then linea OTPs are proposed as representations of belief states in §4.4; they are shown to satis the criteria. But, as shown in §4.5, they do not satisfy two of the AGM axioms. Th fact is discussed in §4.5.1. We end with some examples (§4.6).

The content of this chapter has been published as [59].

4.2 The AGM theory

The standard theory of belief revision—known as the AGM theory after its a thors, C. Alchourrón, P. Gärdenfors and D. Makinson [23]—models belief states *deductively-closed sets of sentences*. More recent developments of the AGM theory a described in S. O. Hansson's thesis [33]. It describes a small set of postulates whi any belief revision operator should satisfy (see below). If K is a belief state and a formula, then $K * \phi$ is a belief state, the result of revising K with ϕ. As alread noted, the case of interest is when $\neg\phi \in K$, that is, when the revising sentence conflic with the current belief state. We suppose we are utilising classical logic with the usu connectives, and the usual entailment relation $\models$.

Notation 4.1

1. Let Φ be a set of sentences. $\mathsf{Cn}(\Phi) = \{\phi \mid \Phi \models \phi\}$.
2. L is the set of all sentences in the language.
3. $K + \phi = \mathsf{Cn}(K \cup \{\phi\})$.

The AGM postulates are the following:

K1 $K * \phi$ is a deductively-closed theory;

K2 $\phi \in K * \phi$;

K3 $K * \phi \subseteq K + \phi$;

K4 If $\neg\phi \notin K$ then $K + \phi \subseteq K * \phi$;

K5 $K * \phi = L$ implies $\phi = \bot$;

K6 If $\models \phi \leftrightarrow \psi$ then $K * \phi = K * \psi$;

K7 $K * (\phi \wedge \psi) \subseteq K * \phi + \psi$;

K8 If $\neg\psi \notin K * \phi$ then $K * \phi + \psi \subseteq K * (\phi \wedge \psi)$.

K1 says that $K * \phi$ should be a belief state. K2 says that the revision should be successful, i.e. the resulting theory should at least contain ϕ. The third axiom says that $K * \phi$ should have no more than what we would get by just adding ϕ set-theoretically and closing under entailment. Of course, if ϕ is inconsistent with K then adding it in that way would yield the whole of L (the theory with every sentence in it). K4 asserts that if ϕ *is* consistent with K then we get precisely the result of adding it set-theoretically. We should point out that this is one of the (two) axioms with which we take issue in §4.5. K5 says that the revision yields the contradictory theory L only if ϕ is inconsistent. This is not just that ϕ is inconsistent *with* K, but is inconsistent on its own. The converse, that revising with an inconsistent sentence yields the inconsistent theory, is guaranteed by K2. K6 says simply that revising with logical equivalents yields the same theory.

K7 and K8 are more complicated, approximating what happens with repeated revisions. They are analogues of K3 and K4.

Note that K7 and K8 do not contain expressions like $K * \phi * \psi$, and therefore do not constrain repeated revision in any explicit way. The only constraints on repeated revision are those inherited from the more general case of revision which K1–8 describe.

I believe the AGM axioms to be neither *sound* not *complete* with respect to intuitively rational belief revision. Of course such a statement is necessarily imprecise, because 'intuitively rational' belief revision is not amenable to mathematical description. My argument to show lack of soundness is to give 'counterexamples' to K4 and K8 later in the chapter. (Again the scare quotes show that these are not counterexamples to any fully spelled-out conjecture.) My argument against completeness is the following proposition, which shows that K1–8 admit revision functions which have no element of the 'persistence' requirement (number 2 above).

Proposition 4.2 The revision function

$$K * \phi = \begin{cases} K + \phi & \text{if } \neg\phi \notin K \\ \mathsf{Cn}\{\phi\} & \text{otherwise} \end{cases}$$

satisfies axioms K1–8.

Proof K1–4 and K6 are immediate.

K5 Suppose $K * \phi = L$. By K3, $K + \phi = L$, so $\neg\phi \in K$. Therefore, $K * \phi = \mathsf{Cn}\{\phi$ so $\phi = \bot$.

K7 Suppose $\neg(\phi \wedge \psi) \in K$. Then $K * (\phi \wedge \psi) = \mathsf{Cn}(\phi \wedge \psi)$. If $\neg\phi \in K$ the $(K * \phi) + \psi = \mathsf{Cn}(\phi \wedge \psi)$; otherwise it is $(K + \phi) + \psi$, which contains $\mathsf{Cn}(\phi \wedge \psi$

Otherwise $\neg(\phi \wedge \psi) \notin K$, and so $\neg\phi \notin K$. $K * (\phi \wedge \psi) = K + (\phi \wedge \psi)$ $(K + \phi) + \psi = (K * \phi) + \psi$.

K8 Suppose $\neg\psi \notin K * \phi$. Suppose also that $\neg\phi \notin K$. Then $K * \phi = K + \phi$. Therefor $K * \phi + \psi = K + \phi + \psi$. Also, since $\neg\psi \notin K * \phi$ and $K * \phi = K + \phi$, we have tha $\neg\psi \notin K + \phi$ and therefore $\neg(\phi \wedge \psi) \notin K$. Therefore, $K * (\phi \wedge \psi) = K + (\phi \wedge \psi)$ $K + \phi + \psi$, as required.

Now suppose $\neg\phi \in K$. Then $\neg(\phi \wedge \psi) \in K$, so $K * \phi + \psi = K + \phi + \psi$ $K + (\phi \wedge \psi) = K * (\phi \wedge \psi)$.

Of course there are more interesting functions satisfying the axioms. The followi two are the most important in the AGM literature: *partial meet* revision; and revisio by *epistemic entrenchment*.

4.2.1 Selection functions

Suppose K is a belief state and ϕ is a sentence other than $\bot$. Let

$$K|_\phi = \text{ the } \subseteq\text{-maximal elements of } \{K' \subseteq K \mid \neg\phi \notin K\},$$

that is, the set of maximal subsets of K which are consistent with ϕ. $K|_\phi$ may b pronounced 'K without ϕ'. The operation of partial meet revision assumes a selectio function S_K which selects *some* of these subsets. Then revision is defined by

$$K * \phi = \begin{cases} \mathsf{Cn}\Big(\bigcap S_K(K|_\phi, \phi) \cup \phi\Big) & \text{if } \phi \neq \bot; \\ L & \text{otherwise.} \end{cases}$$

That is to say, if $\phi \neq \bot$ it is the intersection of those ϕ-consistent maximal subse chosen by S_K with ϕ added set-theoretically. If $\phi = \bot$ it is simply L, the inconsiste theory (the set of all sentences).

It should be clear that this is unsatisfactory, since the whole problem of how to ma a revision has just been packaged up in the existence of a selection function, and ha not been solved at all. Obviously, the selection function must depend on K. Therefor we need not bother with the information K alone provides us, since everything we ne might just as well be given by this magical S! The drawback of coding everything S is that repeated revisions are then impossible.

There is a limiting case of partial meet revision, in which $S_K(K|_\phi, \phi)$ is always singleton. This case is known as maxichoice contraction. There is another limiti case in which $S_K(K|_\phi, \phi) = \bigcap K|_\phi$, the intersection of all the candidate theories, whi is known as full meet revision. The first of these is unsatisfactory for the same reaso as the general case, namely that the selection function remains to be defined. (It ha other, worse, problems too, detailed in Gärdenfors' book.) The second limiting ca

does not have this problem, and is worth spelling out in full, since it fully specifies how to carry out a revision without the need for extra information. According to it,

$$K * \phi = \begin{cases} \mathsf{Cn}\Big(\bigcap K|_\phi \cup \phi\Big) & \text{if } \phi \neq \bot; \\ L & \text{otherwise.} \end{cases}$$

It is straightforward to check that this definition satisfies the postulates K1-K8. But there are problems. Consider, for example, how to revise $\mathsf{Cn}(\{p, q\})$ with $\neg p \vee \neg q$. Intuitively, there are at least three plausible answers: $\mathsf{Cn}(\{p\})$, $\mathsf{Cn}(\{q\})$ and $\mathsf{Cn}(\{p \leftrightarrow \neg q\})$. Full meet contraction gives us the last of these, because no information is available to chose whether to give up p or to give up q. But, in practice there may be criteria for choosing to give up one rather than the other. This is what leads to consideration of selection functions, since they could encode the extra information required. But then, as already remarked, repeated revision is impossible. The moral *we* draw from this situation is different. It is that *deductively closed theories are inadequate as representations of belief states*. We return to this point later, after considering the other main way of providing the information necessary to guide revisions, namely epistemic entrenchment orderings.

4.2.2 Epistemic entrenchment

Revision by epistemic entrenchment is effected as follows. First we require an epistemic entrenchment ordering on the current belief state. This is a linear pre-order on the sentences in the state, which represents the degree to which they are believed. Those less entrenched according the ordering are dispensed with more readily in the case of a revision which conflicts with the current state. An epistemic entrenchment ordering for a belief state K must satisfy the following axioms:

- **E1** If $\phi \leqslant_K \psi$ and $\psi \leqslant_K \chi$ then $\phi \leqslant_K \chi$ (*transitivity*);
- **E2** If $\phi \models \psi$ then $\phi \leqslant_K \psi$ (*dominance*);
- **E3** Either $\phi \leqslant_K \phi \wedge \psi$ or $\psi \leqslant_K \phi \wedge \psi$ (*conjunctiveness*);
- **E4** If K is consistent then $\phi \leqslant_K \psi$ for all ψ iff $\phi \notin K$ (*minimality*);
- **E5** If $\phi \leqslant_K \psi$ for all ϕ, then $\models \psi$ (*maximality*).

As in the case of the K postulates, these axioms are intended to encode rationality constraints on what an epistemic entrenchment ordering might be. For example, E2 says that it is always better to give up logically weaker sentences during the course of a revision; therefore, these should be less entrenched. E3 says that giving up a conjunction is at least as hard as giving up either of the conjuncts. Taken together, axioms E1–E3 imply that $\leqslant_K$ is a linear order, that is, either $\phi \leqslant_K \psi$ or $\psi \leqslant_K \phi$ (or both). E4 says that a sentence is minimally entrenched in K iff it is not in K. E5 says that just the tautologies are maximally entrenched.

Given a belief state K, an epistemic entrenchment ordering $\leqslant_K$ on K, and a sentence ϕ, the revision of K by ϕ is given by

$$K * \phi = \begin{cases} \mathsf{Cn}\Big(\{\psi \in K \mid \neg\phi <_K \neg\phi \vee \psi\} \cup \{\phi\}\Big) & \text{if } \phi \neq \bot; \\ L & \text{otherwise.} \end{cases}$$

($<$ is the usual strict counterpart of $\leqslant$, defined by: $\phi < \psi$ if $\phi \leqslant \psi$ and $\psi \not\leqslant \phi$.)

Full motivation for the K and EE axioms, as well as for the definition of $*$ in tern of $\leqslant_K$, can be found in Gärdenfors' book [23].

We now summarise the main weaknesses we have described of the AGM theor Belief states are represented as deductively closed theories. This means that they a (in general) impossible to write down fully, or to store on a computer. Moreover, noted, they are incapable of representing the necessary information required to cho between alternative revisions. Therefore, extra information in the form of a selectio function or an EE ordering is required. This information is not deemed part of t belief state, and is lost during the revision, making further revision impossible. It worth pointing out that this means that the real *intention* of axiom K1 is not satisfie by these revision functions. Its intention is that after a revision we should end up wi an object of the same type as the one with which we started. Obviously, both parti meet revision and revision by epistemic entrenchment fail this requirement. In tho cases we start off with a pair, respectively of type $\langle K, S_K\rangle$ and $\langle K, \leqslant_K\rangle$, and end u with something of type K.

There are some proposals for modifying the AGM theory to solve some of the problems. For example, some work has been done on theory base revision to addre the problem of the infinite nature of deductively closed sets of sentences. In th work, belief states are represented as finite sets of sentences (theory *bases* or theo *presentations*) [20, 34, 51]. But each of these authors assume the existence of somethi like a selection function or an EE ordering, so are subject to objections on tho grounds. There are proposals of non-deterministic revision [45], which alleviate t need for a selection function, but they rely on infinite belief state representations.

There are proposals to allow repeated revision using EE orderings, either by keepi a single EE ordering for all belief states or assuming the existence of a function whic for every belief state, gives an EE ordering [57, 65]. But as neither the single orderi nor this function is itself revised in the course of belief revisions, it is easy to fi examples which are in contradiction with intuitions about iterated belief change [33

Another modification of the AGM theory which allows EE orderings to be revise is given by H. Rott [58]. He defines revision of EE orderings as follows.

$$\psi \leqslant_{K*\phi} \chi \text{ if } \phi \to \psi \leqslant_K \phi \to \chi.$$

However, as he points out, this fails to capture much of the intuition of repeated revisio because any further revision of $K * \phi$ always includes ϕ.

4.3 Criteria for belief revision

In this section we enumerate what we claim are the criteria by which to judge a theo of belief revision.

1. Finite representation of belief states.
2. Persistence.
3. Iteration: what you put in is what you get out.

4. The "intentions" behind the K axioms of AGM.

The criterion of finite representation means that all belief states can be explicitly written down or represented on a computer. The advantages of this should be easy to see; one in particular is that one can give examples of belief revision in action! (See section 4.6.)

Persistence means that as much of the former belief state should survive a revision as possible. We rule out revisions like the one of proposition 4.2.

The iteration criterion says that you should get out of a revision an object of the same type as you put in. As mentioned, this is violated by AGM, since you put in either an EE ordering, or a theory coupled with a selection function; but, all you get out is a theory. We call this *iteration* since, if it obtains, it guarantees that revisions may be repeated. Its absence is a serious problem in AGM.

The last criterion, concerning the K axioms of AGM, is deliberately expressed in a vague way. Obviously, if belief states are not represented as deductively closed sets of sentences then it is impossible to test them literally. Also, as we have noted, they do not specify what should happen under repeated revision, in terms of expressions of the form $K * \phi * \psi$. This is presumably because the AGM models do not support repeated revision. Moreover, for reasons which we will discuss in section 4.5, we dispute two of the AGM axioms. In view of these reasons, we can only say that something like the intention of the AGM axioms is desirable.

The AGM axioms K1–8 rely on a particular representation of belief states (namely, deductively closed sets of sentences). Therefore, direct comparison with theories of belief revision which use other representations of belief states is impossible. To overcome this we can we rewrite the axioms in a more general way, which assumes only the following:

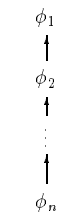
1. A set of belief states, together with a subset of 'contradictory' belief states.
2. A function $*$ (revision) which takes a belief state and a sentence to a belief state;
3. A function $|\cdot|$ (extension) which takes a belief state and returns the set of sentences true in it.

Here are the axioms rewritten in this way. We will write $\mathcal{K}$ for a typical 'abstract' belief state.

$\mathcal{K}1$ $\mathcal{K} * \phi$ is a belief state;
$\mathcal{K}2$ $\phi \in |\mathcal{K} * \phi|$;
$\mathcal{K}3$ $|\mathcal{K} * \phi| \subseteq |\mathcal{K}| + \phi$;
$\mathcal{K}4$ If $\neg\phi \notin |\mathcal{K}|$ then $|\mathcal{K}| + \phi \subseteq |\mathcal{K} * \phi|$;
$\mathcal{K}5$ $\mathcal{K} * \phi$ is contradictory implies $\phi = \bot$;
$\mathcal{K}6$ If $\models \phi \leftrightarrow \psi$ then $|\mathcal{K} * \phi| = |\mathcal{K} * \psi|$;
$\mathcal{K}7$ $|\mathcal{K} * (\phi \wedge \psi)| \subseteq |\mathcal{K} * \phi| + \psi$;
$\mathcal{K}8$ If $\neg\psi \notin |\mathcal{K} * \phi|$ then $|\mathcal{K} * \phi| + \psi \subseteq |\mathcal{K} * (\phi \wedge \psi)|$.

4.4 Linear ordered theory presentations

We present a system for belief revision which satisfies each of the criteria describe above. Belief states are represented by linear *ordered theory presentations*. A linea OTP is a finite list of formulas; $? = [\phi_1, \phi_2, \ldots, \phi_n]$. Elsewhere in the thesis we wri it as



Here, n is said to be the *length* of $?$. The *extension* of $?$ is the deductively-close theory which $?$ presents; that is, it is the set of sentences entailed by $?$, after takin account of the various conflicts in $?$. This was defined in chapter 2 (definition 2.25 To be precise, we set:

$$|?| = \{\phi \mid \ ? \models \phi\}$$

There is an easy intuition for linear ordered theory presentations. The OTP $?$ $[\phi_1, \phi_2, \ldots, \phi_n]$ presents the theory which first of all has ϕ_n, and then has as much ϕ_{n-1} as possible while retaining consistency, and then ... up to ϕ_1. Put another wa we start with ϕ_1. Then we 'force in' ϕ_2, overriding as necessary. Then ... and so o until ϕ_n.

The following are examples of belief states.

1. $[p \wedge q]$ 2. $[p, q]$ 3. $[p \wedge q, \neg p]$

Their lengths are 1, 2 and 2 respectively. States 1 and 2 above both have the extensio $\mathsf{Cn}(\{p \wedge q\})$. But in 2, p is less entrenched than q, and will disappear if a revision whic demands that one of p and q goes. Thus, we stipulate:

> *Sentences later in the list are more entrenched than those earlier.*

State 3 has the extension $\mathsf{Cn}(\{\neg p \wedge q\})$. This is because $\neg p$, which is more entrenche than $p \wedge q$, overrides the p component of $p \wedge q$. But the q component is not overridde Thus,

> *Sentences later in the list have the effect of overriding those earlier, in the case of conflict.*

It should now come as no surprise to find that

> *Revision of OTPs is in general effected by appending the revising sentence to the end of the sequence.*

Thus, the three belief states mentioned above can be revised by $\neg p \vee \neg q$, yielding

$1'$. $[p \wedge q, \neg p \vee \neg q]$ $2'$. $[p, q, \neg p \vee \neg q]$ $3'$. $[p \wedge q, \neg p, \neg p \vee \neg q]$.

The extension of state $1'$ is $\mathsf{Cn}(\{p \leftrightarrow \neg q\})$, which was the outcome of the corresponding example for full meet revision described above. But state $2'$ has as extension $\mathsf{Cn}(\{q\})$. Since 1 and 2 had the same extension and $1'$ and $2'$ do not, it should be clear that *there is more to an OTP than its extension.*

Belief state $3'$ has the extension $\mathsf{Cn}(\{\neg p \wedge q\})$, which is the same as it had before the revision. This is because the revising sentence was consistent with the belief state it revised.

Let us note some important facts about OTP revision.

1. OTPs have *memory.* If $?$ is an OTP, then the extension of $? * p * q * \neg(p \wedge q)$ includes $q \wedge \neg p$. This is because the theory was more recently revised with q than with p, so q is more entrenched. Older information is discarded more readily than newer.

2. But, information is never wantonly discarded.

3. The more you revise an ordered presentation, the more complicated (= longer) it gets. That is because ordered presentations are nothing more than *revision histories.*

The semantics of linear OTPs is of course just the special case of the semantics for arbitrary OTPs given in chapter 2. The crucial definition is that of $\sqsubseteq_\phi$ (§2.2.4), which is an ordering on interpretations which measures how nearly an interpretation satisfies ϕ. For the purposes of this chapter, we may slightly extend proposition 3.24 to obtain the following characterisation of $\sqsubseteq^\Gamma$:

Proposition 4.3

1. $M \sqsubseteq^{[]} N$ always; and

2. $M \sqsubseteq^{\Gamma * \phi} N$ if $M \sqsubset_\phi N$ or ($M \sqsubseteq_\phi N$ and $M \sqsubseteq^\Gamma N$).

This brings out the compositional nature of linear OTPs. $? * \phi$ is $?$ with ϕ appended.

Lemma 4.4 $M \sqsubset^{\Gamma * \phi} N$ if $M \sqsubset_\phi N$ or ($M \sqsubseteq_\phi N$ and $M \sqsubset^\Gamma N$).

4.5 The AGM axioms

As stated, we intend to use these ordered theory presentations as representations of belief states in order to model belief revision. The obvious way to do this is to let

belief states = ordered theory presentations

and define $? * \phi$ to be $?$ with ϕ appended (as in definition 3.20, but in the new notation). Of course we have been assuming this definition so far in the chapter. Note that under this arrangement there are no contradictory theories (proposition 3.18).

In this setting, we can investigate the truth or falsity of the abstract K axioms given in section 4.3 (page 64). We obtain the following.

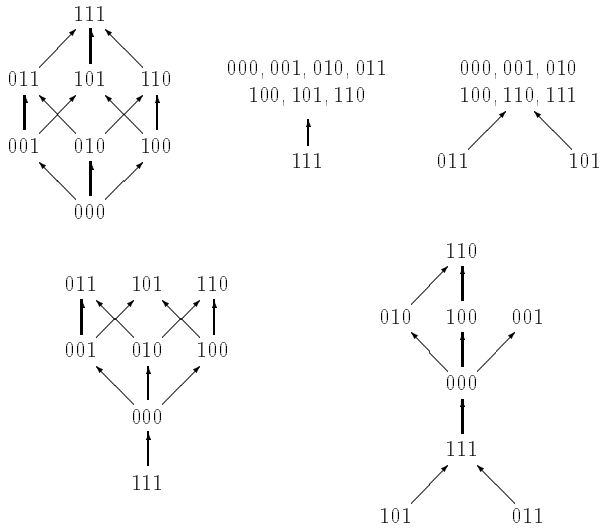


Figure 4.1: The counterexample to $\mathcal{K}4$ (see text)

$\mathcal{K}$**1** $? * \phi$ is a belief state.

This is true. If $?$ is an OTP then so is $? * \phi$.

$\mathcal{K}$**2** $\phi \in |? * \phi|$.

This is false. For example, $\bot \notin |[\,] * \bot|$; for, as one can check, $|[\,] * \bot| = \mathsf{Cn}(\emptyset$ However, $\mathcal{K}2$ is true if $\phi \neq \bot$, by proposition 2.27.

$\mathcal{K}$**3** $|? * \phi| \subseteq |? \,| + \phi$.

True. We need to show that $M \Vdash ?$ and $M \Vdash \phi$ imply $M \Vdash ? * \phi$. Suppo not, i.e. suppose $M \sqsubset^{\Gamma * \phi} N$ for some N. By lemma 4.4, either $M \sqsubset_\phi N$, whi contradicts $M \Vdash \phi$ (proposition 2.54) or $M \sqsubset^\Gamma N$, which contradicts $M \Vdash$ (definition 2.24).

$\mathcal{K}$**4** If $\neg\phi \notin |? \,|$ then $|? \,| + \phi \subseteq |? * \phi|$

This is false. Let $\phi_1 = p \wedge q \wedge r$, $\phi_2 = \neg p \vee \neg q \vee \neg r$ and $\phi_3 = (p \leftrightarrow q) \vee \neg r$. Th counterexample is obtained by setting: $? = [\phi_1, \phi_2]$ and $\phi = \phi_3$. To see this, v should first examine the orderings for each of ϕ_1, ϕ_2 and ϕ_3. They are shown the top half of figure 4.1. Applying proposition 4.3, the orderings $\sqsubseteq^\Gamma$ and $\sqsubseteq^\Gamma$ (i.e. $\sqsubseteq^{[\phi_1,\phi_2]}$ and $\sqsubseteq^{[\phi_1,\phi_2,\phi_3]}$ respectively) are as shown in the bottom half of th figure. We can check the following:

- $\neg\phi_3 \notin |[\phi_1, \phi_2]|$, that is to say, there is a model M such that M is $\sqsubseteq^{[\phi_1,\phi_2}$ maximal and $M \nVdash \neg\phi_3$. Such an M is 110. Thus, the antecedent of K holds.

– But the consequent is false. For we can find ψ such that $? \models \psi$ but $? * \phi \not\models \psi$, namely $\psi = (\neg p \wedge q \wedge r) \vee (p \wedge \neg q \wedge r) \vee (p \wedge q \wedge \neg r)$. We can see this by inspecting the diagrams. Every model of $?$ is a model of ψ. But there is a model of $? * \phi$ which is not a model of ψ, namely 001.

$\mathcal{K}5$ $? * \phi$ is contradictory implies $\phi = \bot$.

This is vacuously true since there are no contradictory belief states.

$\mathcal{K}6$ If $\models \phi \leftrightarrow \psi$ then $|? * \phi| = |? * \psi|$.

True. This follows from proposition 2.2.4.

$\mathcal{K}7$ $|? * (\phi \wedge \psi)| \subseteq |? * \phi| + \psi$.

True. We need to show that if $M \Vdash ? * \phi$ and $M \Vdash \psi$ then $M \Vdash ? * (\phi \wedge \psi)$. If $\phi = \bot$ then $? * \phi \equiv ? * (\phi \wedge \psi)$, and we are done. So suppose $\phi \neq \bot$, and $M \Vdash ? * \phi$ and $M \Vdash \psi$, but $M \sqsubset^{\Gamma * (\phi \wedge \psi)} N$ for some N. Since $M \Vdash ? * \phi$ and $\phi \neq \bot$, we have $M \Vdash \phi$ by proposition 2.27. Therefore, $M \Vdash \phi \wedge \psi$. By lemma 4.4, either $M \sqsubset_{\phi \wedge \psi} N$, which contradicts $M \Vdash \phi \wedge \psi$, or $M \sqsubset^{\Gamma} N$. But this also leads to a contradiction, for then, since $M \sqsubseteq_{\phi} N$, we obtain $M \sqsubset^{\Gamma * \phi} N$ by lemma 4.4, contradicting $M \Vdash ? * \phi$.

$\mathcal{K}8$ If $\neg \psi \notin |? * \phi|$ then $|? * \phi| + \psi \subseteq |? * (\phi \wedge \psi)|$

False. The counterexample given for $\mathcal{K}4$ holds here too. Set $? = [p \wedge q \wedge r]$, $\phi = \neg p \vee \neg q \vee \neg r$ and $\psi = (p \leftrightarrow q) \vee \neg r$.

On this way of using OTPs as belief states, we have shown that $\mathcal{K}1$, $\mathcal{K}3$, $\mathcal{K}5$, $\mathcal{K}6$ and $\mathcal{K}7$ are valid; that $\mathcal{K}2$ is valid under the proviso that $\phi \neq \bot$; and that $\mathcal{K}4$ and $\mathcal{K}8$ are not valid.

It is worth pointing out that the lack of contradictory belief states and the partial failure of $\mathcal{K}2$ are easily solved, by adding a new belief state to represent the contradictory belief state and modifying the definition of revision. Thus,

$$\text{belief states} = \text{ordered theory presentations} \cup \{\bot\}.$$

Revision on these belief states is defined as follows:

$$? \breve{*} \phi = \begin{cases} \bot & \text{if } \phi = \bot \\ [\phi] & \text{if } \phi \neq \bot \text{ and } ? = \bot \\ ? * \phi & \text{otherwise} \end{cases}$$

This emulates what the AGM axioms intend for $\bot$, in that

1. There is a unique contradictory belief state.
2. Revising any state with the contradictory sentence results in the contradictory state (K2).
3. The contradictory state can only be obtained in this way (K5), so in particular
4. Revising the contradictory state with a non-contradictory sentence will *not* result in the contradictory state.

For the psychological plausibility of these stipulations, or otherwise, see [23]. Especial the first one is debatable! Our point is simply that if we take this definition of $? \breve{*}$ on board, we obtain that $\mathcal{K}1$, $\mathcal{K}2$, $\mathcal{K}3$, $\mathcal{K}5$, $\mathcal{K}6$, and $\mathcal{K}7$ are satisfied, and $\mathcal{K}5$ is satisfie in a more satisfying manner. $\mathcal{K}4$ and $\mathcal{K}8$ are still false for the same reasons.

4.5.1 The AGM axioms $\mathcal{K}4$ and $\mathcal{K}8$

$\mathcal{K}4$ and $\mathcal{K}8$ are serious violations of the AGM axioms, and there is no easy way making them satisfied in the framework of OTPs. One must face the question: a they desirable axioms for belief revision? We believe the answer is no.

Consider the diagrams given in figure 4.1. As far as our counterexample is co cerned, the question of the validity of $\mathcal{K}4$ hinges on whether $001 \sqsubset_{\phi_1} 110$ or not. If th was so, then we would also have $001 \sqsubset^{[\phi_1, \phi_2, \phi_3]} 110$ and $[\phi_1, \phi_2, \phi_3]$ would have only t model 110. Therefore, $\mathcal{K}4$ (and $\mathcal{K}8$) would hold.

Should $001 \sqsubset_{p \wedge q \wedge r} 110$ be the case? At first sight it seems clear that 110 is bett at satisfying $p \wedge q \wedge r$ than 001 is, for 110 satisfies two of the atomic propositions whi 001 satisfies only one. But this kind of cardinality argument is flawed. Why is it bett to satisfy $p \wedge q$ rather than r? Perhaps r itself expresses a conjunction of facts. A two oranges better than one apple?

The AGM book does not provide any argument in favour of $\mathcal{K}4$ and $\mathcal{K}8$. Consid the following story. I am expecting a friend called John to arrive. He can come by ca bike, or train. I am doubtful about whether he will arrive or not, however, becau I believe that his car and bike are both at the repairers; and also, the trains are n working today (for a change). Let:

- p mean that his car is unavailable for use
- q his bike is unavailable
- r the trains are unavailable

Initially I believe

$$p \wedge q \wedge r.$$

Now suppose John actually arrives. I have no reason to doubt that he came by o of the usual means of transport (for example, he didn't ask me for money for a taxi Therefore I revise my beliefs by

$$\neg p \vee \neg q \vee \neg r.$$

In the course of conversation it turns out that the repairer phoned him this mornir to say that both his car and his bike were available for collection. I reason as follow If the trains are still not working, he may have asked Richard for a lift to the repaire His bike fits in the back of Richard's car, so then they could have collected both item But, Richard may have been unavailable or unwilling. Either way, he will have collecte both items or neither, so I revise with:

$$r \rightarrow (p \leftrightarrow q)$$

If the trains are working $(\neg r)$ I cannot draw the conclusion $p \leftrightarrow q$, since he may ha gone by train to pick up either the car or the bike, or neither, or he may still ha asked Richard and got both.

The question now is: have I got enough information to conclude which means of transport were available for John to use? I believe no.

To see why the answer is no, we use exactly the argument given in example 1.12, page 20. Suppose r, that is, the trains are still not working. I have already reasoned that this implies $p \leftrightarrow q$, and since John is actually here (so $\neg p \vee \neg q \vee \neg r$), it must be that $\neg p \wedge \neg q$. Therefore, $\neg p \wedge \neg q \wedge r$. On the other hand, suppose $\neg r$, i.e. that the trains *are* working. This tells me nothing about p and q. But since I started with the belief that $p \wedge q$ and John's arrival (by train, presumably) is consistent with these, I retain them. Therefore, $p \wedge q \wedge \neg r$. So I conclude $(\neg p \wedge \neg q \wedge r) \vee (p \wedge q \wedge \neg r)$, or, equivalently, $(p \leftrightarrow q) \wedge (p \leftrightarrow \neg r)$.

We have argued that it is not rational to conclude $\neg r$ in this case. We have also noted that the theory of belief revision outlined in this chapter does not conclude $\neg r$. Indeed, we have argued that it concludes precisely what it is rational to conclude. It should be pointed out in fairness to the AGM theory that it does not insist on $\neg r$ either. To see this, consider what happens if the revision function specified in proposition 4.2 is applied to the revision history in question. We get

$$\begin{aligned} \mathsf{Cn}\{p, q, r\} * (\neg p \vee \neg q \vee \neg r) * (r \rightarrow (p \leftrightarrow q)) &= \mathsf{Cn}\{\neg p \vee \neg q \vee \neg r\} * (r \rightarrow (p \leftrightarrow q)) \\ &= \mathsf{Cn}\{(p \wedge q) \vee \neg r\} \end{aligned}$$

$\neg r$ is not derivable from this theory.

What we have shown is that if we augment the system of OTPs for belief revision so as to obtain $\mathcal{K}4$ and $\mathcal{K}8$, then we would have a system which concluded $\neg r$ in this case, which is undesirable.

4.6 Examples

Here we list some facts about linear OTPs, together with some references to examples in the literature to which the facts seem relevant.

$$\begin{aligned} |[p]| &= \mathsf{Cn}(\{p\}) \\ |[p, q]| &= \mathsf{Cn}(\{p, q\}) \\ |[p, q, \neg q]| &= \mathsf{Cn}(\{p, \neg q\}) \\ |[p, q, \neg p]| &= \mathsf{Cn}(\{\neg p, q\}) \\ |[p \wedge q, \neg p]| &= \mathsf{Cn}(\{\neg p, q\}) \\ |[p \wedge q, \neg p \vee \neg q]| &= \mathsf{Cn}(\{p \leftrightarrow \neg q\}) \\ |[p, q, \neg p \vee \neg q]| &= \mathsf{Cn}(\{\neg p, q\}) \\ |[p \vee q, \neg q]| &= \mathsf{Cn}(\{p, \neg q\}) \end{aligned}$$

We also have that

$$s \rightarrow p \in |[s, s \rightarrow p, s \rightarrow q, \neg q, \neg p]|$$

(cf. Hansson [33, page 7:12]), and, for example,

$$\begin{aligned} p \leftrightarrow q \in |[p, q]|, \quad \text{but} \quad p \leftrightarrow q \notin |[p, q, \neg p]| \\ p \leftrightarrow q \in |[p, p \leftrightarrow q]| \quad \text{and} \quad p \leftrightarrow q \in |[p, p \leftrightarrow q, \neg p]| \end{aligned}$$

(cf. [33, page 4:3]).

Chapter 5

Default Reasoning

In this chapter, existing frameworks for default reasoning are examined and compare with the theory presented in this thesis. We establish a set of criteria by which compare and contrast them, which includes how they handle two famous examples default reasoning in the literature. We also look at some formal properties of defau systems. Finally, we consider related frameworks.

5.1 Introduction

Classical logics allow us to draw incontestable conclusions from sets of premises. Th is very well when we have complete information about a situation. But usually we ha only partial information, and we choose to augment it with prejudices or presumptio or presuppositions in order to be able to reason effectively. Such presumptions, presu positions or prejudices we will call *defaults*. The conclusions we draw with the aid these defaults are not as certain as the ones we might have drawn had we had comple information; instead, they are *defeasible*—they can be defeated by the acquisition more information, which might override some of the defaults we had.

Examples of such defaults at work are ubiquitous, and we could not function e fectively as human beings without using them. We constantly enter into stereotype situations where hundreds of assumptions are made about our and other people's b haviour, and quite often a small proportion of them are proved wrong. When we ent a restaurant we assume the man approaching us will show us to a table; we assun that the items on the menu have been cooked and will be served in portions suitab for one person. The waiter assumes we will order food, that we will want a main cour before a desert and that we have enough money to pay the bill. Any of these defaul can be overridden.

There are a variety of frameworks for reasoning about these stereotyped situation some of which are of a *logical* nature and others *non-logical*. Perhaps the best-know non-logical example is R. Schank's *scripts* [64]. A script is a parametrised represent tion of a stereotyped story (such as the restaurant). The parameters can be set for t particular story at hand; they may include the name of the restaurant, the number persons dining, the particular dishes ordered, the amount paid, and so on. The scrip represent the norms of restaurant behaviour; the values of the parameters just fill the details.

This chapter is concerned with the logical approaches to defaults. One of the best-known examples of a default in the logic literature is the information that *birds can fly*. We can use this to deduce about any bird that it can fly, unless there is information available to the contrary. As J. McCarthy says [50]: "If I hire you to build me a bird cage and you don't put a top on it, I can get out of paying for it even if you tell the judge that I never said my bird could fly. However, I complain that you wasted money by putting a top on a cage I intended for a penguin, the judge will agree with you that if the bird couldn't fly I should have said so."

Logics for expressing and manipulating defaults were first proposed in the early 1980s in a special issue of *Artificial Intelligence* [1]. Since then there has been an abundance of new proposals and variations on existing proposals, and quite a few issues have emerged. An important summary of the state-of-the-art as it was in 1987 is contained in M. Ginsberg's Introduction to a collection of influential papers [25].

This chapter concerns logical formalisms used to represent and reason with defaults. In the literature the terms 'default logics' and 'non-monotonic logics' have been taken as synonymous and used to describe such formalisms. A non-monotonic logic is a logic which fails the property of *monotonicity*:

$$\frac{\Phi \mid\!\sim \psi}{\Phi, \phi \mid\!\sim \psi}.$$

This property says that adding a premise can never inhibit a conclusion.

I prefer the term 'default logic' to 'non-monotonic logic' because the latter term includes any logic which happens to fail the monotonicity property. This property merely states that the set of conclusions grows monotonically with the set of premises. A logic may fail this property and have nothing to do with the representation of defaults; examples include linear logic [26] and relevance logics [2]. It happens that default logics are necessarily non-monotonic, but the converse is not true.

But the term 'default logic' is not ideal either, because some formalisms for default reasoning such as circumscription and model-minimisation are motivated as an alternative *way of using* classical logic rather than an alternative logic. McCarthy makes this point in [50], D. Poole in [53], and indeed we have motivated OTPs in this way too. Therefore 'default reasoning system' seems to be a better term than 'default logic'. We will use the term 'default system' as a convenient abbreviation.

5.2 Criteria for classifying default systems

We will not attempt to summarise the huge variety of formalisms for defaults which have been proposed. Such surveys already exist elsewhere [25, 56, 47]. Instead we will look at a small number of existing logics and classify them according to the following themes:

1. **Representation.** How are defaults represented? We will see examples of default systems which represent defaults by rules of inference; by sets of predicates; and by ordinary sentences.

2. **Method.** Given some way of representing defaults, how should the logic b defined? Existing default systems split into at least two cases, the *proof-theoret* and the *semantics* based.

3. **Conflicting defaults.** How does a formalism deal with conflicting default This is the crucial element in assessing default systems. All of the 'problem mentioned in the literature (such as the two famous ones described below) ha to do with conflicting defaults. We may distinguish between two principal ways resolving conflicts, which we will call the *explicit exception* way and the *extern heuristic* way. In the former, exceptions to defaults are coded up in the theor either as part of the defaults or separately from them. In the latter, no exceptio are mentioned. Instead, an heuristic such as the specificity principle mentione in §1.2.1 is employed within the logic to resolve the conflict. This distinction w become clearer with the examples of default systems below.

   Related to the question of conflicting defaults is the question of whether or can express relative priorities between defaults, to determine which one tak precedence in the event of a conflict.

4. **Application area.** Some non-monotonic systems have been developed for pa ticular applications only, not for arbitrary defaults.

5. **Formal properties.** Makinson [47] describes several properties such as wea cut, weak monotonicity and reflexivity which classify default systems accordin to their underlying consequence relation.

5.3 Two examples of default reasoning

As well as the criteria described above, we will also make use of the following tw examples of default reasoning to classify the various existing systems (and our own To the reader acquainted with default systems they will be very familiar. Althoug hackneyed, they are excellent examples for showing the key differences between th formalisms.

The examples we chose concern inheritance and persistence, which are undoubted the principal uses of defaults to be found in the literature. There are others, howeve V. Lifschitz [44] distinguishes between five types of default reasoning and cites n less than 32 different examples. Inevitably, therefore, the analysis we shall give incomplete.

5.3.1 Inheritance defaults

If every object in a class has a certain property, then every object in any subcla also has it; that is to say, properties of a class are inherited by any subclass. But, already remarked in the Introduction, this is not true of default properties. When w are interested in whether defaults about classes are inherited by subclasses, we will ca them 'inheritance defaults'.

We will consider the well-known example concerning birds and penguins an whether they can fly. The class of penguins is a subclass of the class of birds. B

the property of being able to fly, which holds of birds by default, is not inherited by penguins. In the usual formulation of this example, we have the following *factual* premises

> Penguins are birds;

together with the *defaults*

> Birds can fly, and
>
> Penguins cannot fly.

We want the following results:

1. If Fred is stated to be a bird (whether he is also a penguin or not is not stated), we want to conclude that he can fly.
2. But if it is stated that he is a penguin, we want to conclude that he cannot fly.

The reason this example is interesting is that there are two defaults which compete in certain circumstances. It is easy to get result 1 correctly, but it is in the case of result 2 that the defaults conflict. Our intuition that the second of the two defaults should have priority and block the application of the first is based on the specificity principle mentioned in chapter 1:

> Defaults about a specific class of objects take priority over defaults about a more general class.

Some default systems have this principle 'built in', while in others we have to express the desired priority between the defaults as part of the theory. In the latter case, we will be interested in whether the means of expressing this priority always works.

5.3.2 Persistence defaults

Another kind of default widely discussed in the literature concerns the effects of actions. An action is usually described by stating what changes come about when the action takes place. For example, we may say that the action of putting block A on top of block B will result in block A being on top of block B. By this description we intend that everything else, such as the position of block C, remains the same. More precisely, we intend that unless it can be shown from the axioms of the situation at hand that the action affects the position of block C, we should be able to deduce that it does not affect it.

The problem of having to specify, for each action, the fluents[1] which are not changed by it is called the *frame problem*. In general, a given action leaves most fluents unchanged. The problem of specifying this may be solved by employing defaults which say that actions have no effect on fluents; these defaults are overridden by the axioms which say what effects actions do have. Since these defaults express the fact that the values of fluents persist through the occurrence of actions, they are called *persistence defaults*.

[1] We suppose that the state of the system is specified by the values of certain variables; these variables are called fluents.

There is a massive literature on this subject, and the reader is assumed to b familiar at least with the general ideas; otherwise our description here will probab be too terse. Introductory material is contained in [11, 24, 32, 36, 68].

The most famous example of persistence defaults is called the Yale Shootin Problem[2], and was proposed by S. Hanks and D. McDermott [32]. It is well-known b cause none of the then-available default systems could (starting from what was thoug of as the intuitively correct coding) obtain the intuitively correct answer. It is an e ample we will use in our comparison of default systems.

We have a gun and a man. The gun can be loaded or unloaded, the man can b alive or dead. Imagine 3 situations, which we will call 1, 2 and 3. 1 is the initi situation, in which the gun is loaded and the man is alive. Situation 2 results fro waiting an indeterminate period after situation 1. Situation 3 is the result of firing th gun in situation 2. We have the following premises

> The gun is loaded in 1;
>
> The man is alive in 1; and
>
> If the gun is loaded in 2, then the man is not alive in 3.

together with the *defaults*

> If the man is alive in i then he is alive in $i+1 \quad (i \in \{1,2\})$; and
>
> If the gun is loaded in i then it is loaded in $i+1 \quad (i \in \{1,2\})$.

We want the following result:

> The man is not alive in 3.

Again, we have competing defaults. Intuitively, nothing happens between 1 and Therefore the gun is loaded in 2, and the man is alive in 2. Since the gun is loade in 2, the man is dead in 3.

The reason that this example is famous is that all the formalisms for default reaso ing available at the time it was introduced allow there to be another possible outcom It is that the gun should miraculously become unloaded during the wait action betwe 1 and 2. Then, when it is fired in 2, we cannot conclude that the man dies.

Even before considering any particular formalism, we can see how the second sc nario comes about. Let A be the scenario which we expect, in which the man dies. L B be the one in which the gun becomes unloaded, and the man lives.

- A can be obtained by starting with the factual premises, and using the defaul to show that situation 2 is identical with 1. Since the gun is loaded in 2, the ma must be dead in 3.
- B is also obtained by starting with the factual premises. We use the first defau to conclude that the man is alive in 2, and then use it again to show that he alive in 3. If he is alive in 3, it must be that the gun was not loaded in 2.

The second scenario may seem a bit less natural than the first, because to obtain involves reasoning from later situations to earlier ones. But that fact does not stop th logical conclusions. Note that

[2] Our description here is slightly (but immaterially) simplified from the original.

- A is obtained by using each default once (to get from 1 to 2) and by overriding the first default once (to get from 2 to 3).
- B is obtained by using the first and overriding the second default (to get from 1 to 2) and by using the first again (to get from 2 to 3).

The important point is that one cannot chose A on the grounds that it employs more defaults or violates fewer defaults than B. Each scenario uses two instances of the defaults and violates one.

Much of the literature about this example focusses on the idea, due to Y. Shoham [68], that defaults relating to earlier states of the system should take priority over defaults relating to later states. In the example, this successfully avoids scenario B. Thus, we may stipulate a principle for persistence defaults, analogous to the specificity principle for inheritance defaults. The *chronology principle* states that:

> Defaults about an earlier state take priority over defaults about a later state.

(It is important to note that this principle is appropriate when using defaults to predict the outcome of action sequences; that is, for so-called 'prediction problems'. There are other examples of uses of persistence defaults, for example in 'explanation problems' where it is desired to account for a known outcome, in which this principle manifestly gets the wrong answer. An example of this is H. Kautz' 'stolen car problem' [36].)

As for specificity, some default systems have this principle 'built in' (such as Shoham's logic of *chronological ignorance*), while in others we have to express the desired priority between the defaults as part of the theory. But in the latter case, the method of expressing this priority often fails to have the desired effect, as we will see.

5.4 Default systems

We now consider some default reasoning formalisms in the light of the criteria and examples described in the last two sections.

5.4.1 Reiter's 'Default Logic'

In Reiter's 'Default Logic' [55] defaults are **represented** as rules of inference which have a consistency-check side condition. In Reiter's system one would encode the first default about birds as

$$\frac{b(x) : f(x)}{f(x)}$$

which is read as: if x is a bird and it is consistent to conclude that x can fly, then x can fly. In general, a default rule is an expression of the form:

$$\frac{\alpha : \beta}{\gamma},$$

The formula α is the precondition to the rule, β is the clause that is checked for consistency with the database and γ is added to the database if β is consistent. A

rule such as the one above about birds, where $\beta = \gamma$, is called a normal default rul
If β implies γ the rule is semi-normal; otherwise it is non-normal. In general, defau
rules are preferred to be semi-normal or normal, as non-normal rules have peculi
properties.

The **method** for reasoning with default rules is as follows. A default theory
Reiter's formalism is a set of sentences S together with a set of default rules D. A
extension of this default theory is a logical theory such that

1. None of the rules can consistently be applied to obtain a conclusion not alread
in the extension.
2. Subject to this condition, the extension is minimal.

Consider D as an operator on a logical theory T, returning a new logical theory $D(T$
which is the result of applying zero or one rules in D to T. Then $T \subseteq D(T)$. A
extension E is a least fixed point of this operator.

Reiter's logic can deal with some examples of **conflicting defaults**, but not other
It will work for the inheritance example (§5.3.1), but not the persistence examp
(§5.3.2).

The inheritance example. One may consider coding the example into the defau
theory

$$b(\text{Fred}) \qquad \forall x.\,\big(p(x) \rightarrow b(x)\big) \qquad \frac{b(x) : f(x)}{f(x)} \qquad \frac{p(x) : \neg f(x)}{\neg f(x)}$$

This corresponds to case 1 of the example. There is only one extension of the theor
which contains $f(\text{Fred})$. Thus, result 1 is satisfied. Now suppose we replace $b(\text{Fre}$
with $p(\text{Fred})$, for case 2 of the example. It is easy to check that there are two extension
one containing $f(\text{Fred})$ and the other with $\neg f(\text{Fred})$. There are two because there a
two ways of obtaining the operator D, one for each order in which we can apply t
rules.

To obtain result 2 correctly we have to state the first default in a more guarde
fashion, namely:

$$\frac{b(x) : \neg p(x) \wedge f(x)}{f(x)}$$

This says that birds which are not known to be penguins (that is, it is consistent wi
current information that they are not penguins) can fly. Replacing the former rule
this one yields a theory with a single extension in both cases 1 and 2, which contai
the right answer in both cases.

Thus, this logic falls into the category of logics which employ *explicit exceptions* f
resolving the conflicts between defaults. The fact that penguins are exceptions to t
default about birds is explicitly indicated in the rule.

The persistence example. Again in this example it is a question of giving great
priority to some defaults than others; in this case the second default should be preferre
The facts are that

$$\ell_1 \qquad a_1 \qquad \ell_2 \rightarrow \neg a_3$$

where ℓ_i and a_i mean respectively that the gun is loaded and the man is alive in state i. Learning from the previous example, we should write the defaults as

$$\frac{a_1 : a_2}{a_2} \qquad \frac{a_2 : \neg\ell_2 \wedge a_3}{a_3} \qquad \frac{\ell_1 : \ell_2}{\ell_2}$$

The first default simply states that the property of aliveness persists from situations 1 to 2. The second default says the same about situations 2 and 3, but we have coded in the fact that ℓ_2 is an exception to this, in the hope of making this rule yield priority to the persistence of the loaded property. The third rule expresses this persistence from states 1 to 2. (Since we are not bothered about the value of ℓ_3 we have not bothered about the persistence of ℓ from 2 to 3.)

This is not the coding of the example given in Reiter's logic by Hanks and McDermott in the usual paper. We have simplified rather dramatically by using a propositional language and making explicit the identities of the states. This simplification is justified since the same problem occurs in this simpler setting as occurred in Hanks and McDermott's, but the simpler setting is rather easier to understand. However, I accept that the simpler setting may not do justice to some of the subtler solutions to the problem which have appeared in the literature. As these are not the main interest of this chapter, I feel this is not a significant loss.

Returning to the example, we find that there are still two extensions. They are obtained in the way already described above (§5.3.2).

- Starting with the facts $\{\ell_1, a_1, \ell_2 \rightarrow \neg a_3\}$, apply the first default to give $\{\ell_1, a_1, a_2, \ell_2 \rightarrow \neg a_3\}$, then the third default to give $\{\ell_1, \ell_2, a_1, a_2, \neg a_3\}$. The second default cannot be applied since we have $\neg a_3$. We conclude that the man is dead in state 3. This is scenario A.

- For scenario B, again start with $\{\ell_1, a_1, \ell_2 \rightarrow \neg a_3\}$ and apply the first default to give $\{\ell_1, a_1, a_2, \ell_2 \rightarrow \neg a_3\}$. Now apply the second default to give $\{\ell_1, \neg\ell_2, a_1, a_2, a_3\}$. The third default cannot be applied since we have $\neg\ell_2$. We conclude that the man is alive in state 3.

Solutions to this problem using Reiter's logic have been proposed by Morris which employ non-normal defaults.

5.4.2 Circumscription

In McCarthy's circumscription ([49, 50, 43] and others) defaults are **represented** as ordinary first order sentences. Their status as defaults results from the fact that they contain predicates which are *minimised* in the logic, in a way which will become clear. The simplest way of coding the default that birds can fly is as

$$b(x) \wedge \neg \mathsf{ab}_b(x) \rightarrow f(x)$$

This is read as: if x is a bird and x is not *abnormal* then x can fly. The predicate ab_b is called an abnormality predicate. The subscript reflects the fact that there may be several such predicates; this one corresponds to abnormal *birds*. (In general, the

predicate being minimised need not be called 'ab' or represent abnormality; this simply a useful idiom.)

The **method** for reasoning with defaults in circumscription is the following. *Inste of considering all models of a circumscriptive theory, only models in which the extensi of the abnormality predicates is minimal are considered.* This means, in effect, that augment a circumscriptive theory with the information that the abnormality predicat are to be minimised.

This is best illustrated with the examples. We will find, again, that circumscriptio works well for the inheritance example, but not for the persistence example.

The inheritance example. The correct way of coding case 1 of this example is t following:

$$\begin{aligned}
&b(\text{Fred})\\
&\forall x.\ \Big(p(x) \rightarrow b(x)\Big)\\
&\forall x.\ \Big(b(x) \wedge \neg \mathsf{ab}_b(x) \rightarrow f(x)\Big)\\
&\forall x.\ \Big(p(x) \wedge \neg \mathsf{ab}_p(x) \rightarrow \neg f(x)\Big)\\
&\forall x.\ \Big(p(x) \rightarrow \mathsf{ab}_b(x)\Big)
\end{aligned}$$

The last sentence in this set can be thought of as the particular way of coding circumscription the fact that the default about penguins takes priority over the defau about birds. It says, in effect, that penguins are exceptions to the birds default becau they are abnormal birds. Like Reiter's logic, circumscription also employs explic exceptions to resolve the conflict between competing defaults.

As stated, we consider only models which are minimal in the ab predicates. E inspection of the theory, we can see that this means that in such models ab_p and a shall have empty extensions. The *circumscription* of this theory with respect to ab_b, a contains the five axioms above, and also

$$\begin{aligned}
&\forall x.\ \Big(\neg \mathsf{ab}_b(x)\Big)\\
&\forall x.\ \Big(\neg \mathsf{ab}_p(x)\Big)
\end{aligned}$$

We have $\neg\mathsf{ab}_b(\text{Fred})$, and so by the birds default we conclude $f(\text{Fred})$.

Now consider the five axioms, but with the first one replaced by

$$p(\text{Fred})$$

The extension of ab_p is still empty, but the fifth of the axioms means that at lea Fred must be in the extension of ab_b. The circumscription of the new five axioms wi respect to $\mathsf{ab}_b, \mathsf{ab}_p$ contains the new five axioms, and also

$$\begin{aligned}
&\forall x.\ \Big(\mathsf{ab}_b(x) \leftrightarrow (x = \text{Fred})\Big)\\
&\forall x.\ \Big(\neg \mathsf{ab}_p(x)\Big)
\end{aligned}$$

We conclude $\neg f(\text{Fred})$.

We thus conclude the correct answer in both cases.

The persistence example. We will code this as a propositional example again. (For the original predicate coding, see [32].) The theory to be circumscribed is formed from the sentences

$$\ell_1$$
$$a_1$$
$$\ell_2 \rightarrow \neg a_3$$
$$a_1 \wedge \neg \mathsf{ab}_{a_1} \rightarrow a_2$$
$$a_2 \wedge \neg \mathsf{ab}_{a_2} \rightarrow a_3$$
$$\ell_1 \wedge \neg \mathsf{ab}_{\ell_1} \rightarrow \ell_2$$
$$\ell_2 \rightarrow \mathsf{ab}_{a_2}$$

Again, we wish to minimise the abnormality propositions. This means making them false when possible. However, as the reader may by now expect, there is competition between them about which ones can be made false.

- ab_{a_1} and ab_{ℓ_1} can be made false, but the resulting theory then contains ab_{a_2}. It also contains $\neg a_3$. This is scenario A.
- ab_{a_1} and ab_{a_2} can be made false, but the resulting theory then contains ab_{ℓ_1}, and also contains a_3. This is scenario B.

Experts on the Yale Shooting Problem may be frustrated by this propositional version which leaves out much of the latitude for solutions provided by the original coding. For example, it is not clear how the state-based minimisation of Baker [3] should work in this setting. Perhaps it cannot. But this is of no significance for the emphasis of this chapter, which is the representation of defaults and their priorities.

My view is that the Yale Shooting problem can be solved by making explicit the fact that the persistence of loadedness between states 1 and 2 takes priority over the persistence of aliveness between 2 and 3. I claim that this was implicit in the original codings by the fact that loaded-in-2 is stated as an exception to the persistence of alive between 2 and 3. But the early formulations of the problem failed because this method of stipulating the priorities between the defaults failed. All I have to add to the debate is that the semantics given to default priorities in this thesis do not fail in this respect. Proposals for the Yale Shooting Problem which address more general problems in temporal reasoning (such as Baker's mentioned above) are orthogonal to the discussion of default priorities.

5.4.3 Veltman's Update Semantics

Veltman's Update Semantics [74] is a much more recent approach to defaults, and is part of an emerging school in Amsterdam focussing on the 'dynamics' of logic. According to that school, the meaning of a sentence is given not by its models but by the change it brings about in the information state of the agent which understands it. Thus, sentences are functions between information states. (As was seen in chapter 4, theories of belief revision can be seen in this way too.)

Whereas circumscription and Reiter's default logic are about any kind of default Update Semantics was designed specifically for inheritance defaults. It represents d faults simply by sentences in the language, with a special connective $\rightsquigarrow$ for defau implication. Case 1 of the inheritance example of birds and penguins becomes:

$$b(\text{Fred})$$
$$\forall x.\ \big(p(x) \rightarrow b(x)\big)$$
$$\forall x.\ \big(b(x) \rightsquigarrow f(x)\big)$$
$$\forall x.\ \big(p(x) \rightsquigarrow \neg f(x)\big)$$

This is the simplest representation we have seen so far. No explicit exceptions are me tioned, and no artificial predicates like the abnormality predicates of circumscriptio need be employed.

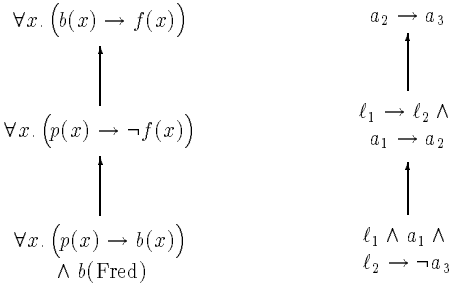
The method by which Update Semantics works is complicated, and the read should see Veltman's paper for full details. Here is an outline. As stated, sentenc denote functions between information states. An information state is a collection models (representing the ways the world might be, given the current informatio together with a family of pre-orders on the models. These pre-orders are called 'e pectation patterns', and represent the defaults with which the agent is acquainted; other words, they represent his expectations about the world. There is an expect tion pattern for each subset of the models in an information state, with 'coherenc conditions relating them.

By virtue of the fact that it is designed for inheritance defaults, Update Semanti gets the correct answer for the theory above, and also for the theory with $p(\text{Fre}$ ($f(\text{Fred})$ and $\neg f(\text{Fred})$ respectively).

5.4.4 Ordered theory presentations

OTPs represent defaults by sentences in the language. They obtain their status defaults by their position in the ordering. Sentences minimal in the ordering have t status of facts, and there are as many levels of defaults as may be needed by considerin OTPs of arbitrary depth. The mechanism of OTPs was given in chapter 2. Conflictin defaults may be resolved by rearranging the ordering.

The ordered presentations corresponding to case 1 of the inheritance example a the persistence example are the following.



They respectively prove $f(\text{Fred})$ and $\neg a_3$ as required. The OTP for case 2 of the inheritance example has $p(\text{Fred})$ instead of $b(\text{Fred})$, and proves $\neg f(\text{Fred})$.

We do not intend to conclude from this analysis that the logic of ordered theory presentations is superior to all the other default systems because it obtains the correct answer to the Yale Shooting Problem. Such a conclusion would be terribly naïve for many reasons. For one, our solution depends on ordering the persistence defaults according to the precedence of the state in which they apply. In many formalisms this would mean decomposing a persistence default into lots of instances, which is at best inelegant; at worst it is impossible. Another reason is that our solution is a crude application of the chronology principle, but, as already seen, this is not appropriate for all examples of reasoning about actions. What we have shown is that the theory of OTP given in this thesis does correctly implement prioritisation of defaults in cases (such as the Yale shooting problem) where other logics fail. We also hope that we have shown that the representation of defaults, and interacting defaults in particular, is clearer in the theory of OTPs than in many of its rivals.

5.4.5 Other systems with ordered defaults

There are other default systems in which hierarchies of defaults may be represented; in this section we mention the similarities and the differences with OTPs. The two systems we will discuss D. Vermeir's Ordered Logic [75, 40] and G. Brewka's preferred subtheories [7].

Vermeir's motivation is to generalise logic programming by introducing an ordering among the rules in a logic program. To this end he considers partially ordered sets of 'rules'; a rule is a clause $Q_0 \leftarrow Q_1, Q_2, \ldots, Q_n$. Each Q_i may be negated, and n may be 0. The intended meaning of such an 'ordered program' is similar to the meaning we give to the corresponding ordered theory presentation, except that the semantics of the connectives $\leftarrow$ and negation is not the classical one; the framework is restricted to the language mentioned; and there is no 'partial' satisfaction of sentences such as the one we describe in this thesis.

G. Brewka's preferred subtheories is presented as an extension of Reiter's default logic (§5.4.1) and of Poole's default logic [53]. The motivations of this work are similar to those of this chapter, namely to give a system in which hierarchies of defaults may be expressed. Compared with this work, there are both limitations and advantages of Brewka's approach. Among the limitations are (i) the restriction to linear orderings among defaults; (ii) a restricted syntax and the restriction to that particular syntax; and (iii) no ability to handle partial satisfaction (that is, to adopt part of a default when the whole would lead to inconsistency). However, his semantics are simpler than the semantics presented in this thesis.

More work comparing these systems to ours is in hand.

5.5 Formal properties of default systems

The study of default systems has, I believe, been transformed by a new concern, namely the formal properties of the underlying consequence relation. The first default systems introduced in the 1980 special issue of *Artificial Intelligence* [1] did not even have well-

defined consequence relations. D. Gabbay [22] and M. Clark [10] first observed tha instead of focussing on the *negative* properties of such consequence relations, that i their *non*-monotonicity, one should instead ask what properties they do have. The gave the name 'cautious monotonicity' to the property

$$\frac{\Phi \mid\!\sim \phi \quad \Phi \mid\!\sim \psi}{\Phi, \phi \mid\!\sim \psi}.$$

This property, which is weaker than full monotonicity, has become widely accepted a desirable property for default systems.

The story of the properties of default consequence relations has been pursued the work of S. Kraus, D. Lehmann and M. Magidor [38, 42] and also by D. Makinso [46, 47]. Makinson's [47] is, in my opinion, the most authoritative and systemat study to date. He describes and motivates a set of conditions on a default consequenc relation and analyses existing systems according to whether they have the condition In this section we outline his principal conditions and check the theory of OTPs of th thesis against them.

5.5.1 Makinson's conditions

Makinson describes a set of conditions on a default consequence relation $\mid\!\sim$, or, equi alently, a default consequence *operation* C. As usual, consequence relations and ope ations are interchangeable:

$$\Phi \mid\!\sim \psi \text{ iff } \psi \in \mathsf{C}(\Phi).$$

As elsewhere in this thesis, Φ, Ψ, ... are sets of sentences, while ϕ, ψ, χ, ... are sing sentences.

The expression $\Phi \mid\!\sim \psi$ (or $\psi \in \mathsf{C}(\Phi)$) should be read as: ψ follows from Φ *in t context of an understood set of defaults*. It is unfortunate (and detracts slightly fro the systematic study) that these defaults are nowhere made explicit. Consequently, t behaviour of the consequence relation under variations of the defaults—and for tha matter, questions of default representation—are not examined at all in his work.

Makinson's conditions also refer to classical consequence, written $\mid\!\!-$ as a relation Cn as an operation. $\Phi \mid\!\!- \psi$ is to be read as ψ follows from Φ without using the default The understood set of defaults can be thought of as augmenting classical consequenc to default consequence. Therefore, the first property we may expect is

Supraclassicality

$$\frac{\Phi \mid\!\!- \psi}{\Phi \mid\!\sim \psi}$$

or, in the language of consequence operations, $\mathsf{Cn}(\Phi) \subseteq \mathsf{C}(\Phi)$.

It says that anything which can be derived without the defaults can also be derive with them.

The next three conditions are together called 'cumulativity'. They are weak form of Tarski's conditions on standard consequence relations (described in proposition 2.9 These weak forms have already been proved for natural consequence (proposition 2.4 and, in a certain context, for OTPs (proposition 3.28).

Inclusion: If $\psi \in \Phi$ then $\Phi \mathrel{|\!\sim} \psi$.

Cautious monotonicity:

$$\frac{\Phi \mathrel{|\!\sim} \phi, \text{ for all } \phi \in \Psi \quad \Phi \mathrel{|\!\sim} \psi}{\Phi, \Psi \mathrel{|\!\sim} \psi}$$

Weak cut:

$$\frac{\Phi \mathrel{|\!\sim} \phi, \text{ for all } \phi \in \Psi \quad \Phi, \Psi \mathrel{|\!\sim} \psi}{\Phi \mathrel{|\!\sim} \psi}$$

They are jointly (but not quite individually) equivalent to the following conditions on C:

- Inclusion: $\Phi \subseteq \mathsf{C}(\Phi)$.
- Cautious monotonicity$^-$: $\Phi \subseteq \Psi \subseteq \mathsf{C}(\Phi)$ implies $\mathsf{C}(\Psi) \subseteq \mathsf{C}(\Phi)$.
- Weak cut$^-$: $\Phi \subseteq \Psi \subseteq \mathsf{C}(\Phi)$ implies $\mathsf{C}(\Phi) \subseteq \mathsf{C}(\Psi)$.

(The $^-$ signs represent the fact that these C versions of cautious monotonicity and weak cut are implied by the $\mathrel{|\!\sim}$ versions, but imply them only in the presence of inclusion. In other words, they are equivalent in the presence of inclusion but slightly weaker otherwise.)

An inference relation is said to satisfy **cumulativity** if it satisfies cautious monotonicity and weak cut.

For the justification of these principles in intuitive terms, we cannot do better than quote Makinson. "Cut may be seen as expressing a determination not to allow the length, intricacy or manner of a derivation of a conclusion to reduce the freedom with which it is used in further inference. There is no 'diminution of usability' with respect to distance from origins. Once inferred, a proposition may be called upon in conjunction with the original information, unless genuinely new (i.e. uninferable) information is also added. Cautious monotonicity, on the other hand, may be seen as expressing a certain irreversibility in the drawing of conclusions. Once inferred, a proposition may be retained irrespective of what other inferred propositions are added to the stock of usable information. We need never go back unless, once more, genuinely new information is brought in" [47].

The next condition we will consider is

Distributivity: If Φ and Ψ are Cn-closed sets of sentences (that is, $\Phi = \mathsf{Cn}(\Phi)$ and $\Psi = \mathsf{Cn}(\Psi)$) then

$$\frac{\Phi \mathrel{|\!\sim} \phi \quad \Psi \mathrel{|\!\sim} \phi}{\Phi \cap \Psi \mathrel{|\!\sim} \phi},$$

or, in the language of C: if Φ and Ψ are Cn-closed then $\mathsf{C}(\Phi) \cap \mathsf{C}(\Psi) \subseteq \mathsf{C}(\Phi \cap \Psi)$.

Finally, the following condition has had attention in the literature [47, 38]

Rationality:

$$\frac{\Phi \mathrel{|\!\sim} \phi \quad \Phi \not\mathrel{|\!\sim} \psi}{\Phi, \psi \mathrel{|\!\sim} \phi}$$

This is again a weak form of monotonicity, which says that premises may be added an argument if their negations are not derivable from the original set. One interestin feature of this rule is the negated $\mathrel{|\!\sim}$ relation above; Makinson describes such condition as 'non-Horn' (because, when expressed as clauses, they are not Horn clauses).

Makinson considers other conditions, but these are the principal ones. Before v turn to the question of which of these conditions are satisfied by ordered theory prese tations (and before we make that question precise), we will introduce some terminolog of Makinson's, together with a result, which will make the job easier.

First, some background. As has been pointed out already, the technique of orderin interpretations which we use so extensively in chapter 2 is not new. It originat in McCarthy's first circumscription paper [49] in a rather narrow context which w broadened first by Shoham [67, 8], and independently by P. Besnard and P. Sieg [4] and Kraus/Lehmann/Magidor [38]. It is also used in Veltman's Update Semanti [74], from which we drew inspiration. In all of those papers, the ordering works in th opposite way to the one we have used in this thesis, that is, $M < N$ means M is bett than N; and therefore, one is interested in minimal models[3]. Makinson has examine constraints on such 'preferential model structures', as he calls them, and has relate these constraints to the conditions on $\mathrel{|\!\sim}$ described above.

In brief, he defines a preferential model structure to be a triple $\langle \mathbf{M}, \Vdash, < \rangle$ whe $\mathbf{M}$ is an arbitrary set, $\Vdash$ is an arbitrary relation between $\mathbf{M}$ and the sentences in th language and $<$ is an arbitrary relation on $\mathbf{M}$. If $M \in \mathbf{M}$, then M satisfies ϕ if $M \Vdash$ holds; and M preferentially satisfies ϕ, written $M \Vdash_< \phi$, if $M \Vdash \phi$ and for all $N < M$ $N \not\Vdash \phi$. We also define $M \Vdash \Phi$ for a set of sentences Φ if $M \Vdash \phi$ for all $\phi \in \Phi$, an $M \Vdash_< \Phi$ if $M \Vdash \Phi$ and for all $N < M$, $N \not\Vdash \Phi$.

A preferential model structure defines a *preferential inference relation* $\mathrel{|\!\sim}$ in th following way:

$$\Phi \mathrel{|\!\sim} \psi \text{ iff } \forall M \in \mathbf{M}.\, M \Vdash_< \Phi \text{ implies } M \Vdash \psi,$$

that is, every 'minimal' model of Φ satisfies ψ.

Makinson then considers the following constraints on preferential model structure The structure $\langle \mathbf{M}, \Vdash, < \rangle$ is

- *stoppered,* if for all interpretations M and sets of sentences Φ with $M \Vdash \Phi$, the is an $N \leqslant M$ such that $N \Vdash_< \Phi$. Intuitively, this means that any model of a s of sentences can be improved into a minimal model.

[3] The reader may wonder why we chose to fly in the face of this well-established convention, choosing to order interpretations in the opposite sense and therefore to seek $\sqsubseteq^\Gamma$-*maximal* interpret tions. There are two reasons. The first is that the fact that other workers order models in the oppos way is for the historic reason that in circumscription one wants to minimise abnormality predicat this reason does not apply in the more abstract setting of this thesis. On the contrary, it is mo intuitive to move *upwards* in an ordering when one is moving to better and better models. The seco reason is that one typically looks at *ascending chains* and *maximal elements* in domain theory a information systems theory, with which we see links with our work. Cf. lemmas 3.15 and 3.16.

- *classical,* if $\Vdash$ behaves in the classical way with respect to the logical connectives (that is, the conditions on $\Vdash$ given in example 2.4 on page 25 hold).
- *transitive,* if $<$ is transitive.

It turns out that different combinations of these constraints give inference relations satisfying various conditions of the ones described. We will just quote one result, which will be relevant for the next subsection.

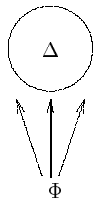
Proposition 5.1 (Makinson.) The inference relation of a classical and stoppered preferential model structure satisfies Supraclassicality, inclusion, cumulativity and distributivity.

5.5.2 Makinson's conditions and OTPs

We have already noted that Makinson's conditions make no reference to the set of defaults which are implicit in the relation $\mid\!\sim$ (or the operation C). On the other hand, one of the attractive features of the framework of Ordered Theory Presentations as a default system is that there is *no difference* between defaults on one hand and 'sure rules' or facts on the other, except the priority they are given in the ordering. We view this as a desirable feature since we believe that, philosophically, the so-called sure rules and the defaults have the same provenance. They should all form part of the theory, or database, from which we make deductions. A sentence does not have the status of a default in isolation, but only in relation to other sentences; to be precise, it is a default relative to those sentences which can override it.

Nevertheless, we can go quite some way in examining Makinson's conditions in the context of ordered theory presentations over classical logic $\langle L, \mathcal{M}, \Vdash\rangle$. In order to emulate variation of the facts with a fixed set of defaults, we can consider the consequences of the ordered presentation $\Delta * \Phi$ with Δ finite and fixed and Φ varying[4]. We can think of this OTP as a way of representing that which in other default formalisms might be called 'the theory Φ with defaults Δ'. Notice that Δ is itself an OTP; that is, we are still allowing defaults with different priorities.

Recall that graphically $\Delta * \Phi$ may be represented as



[4]Strictly, we should write $\Delta * \bigwedge \Phi$, not $\Delta * \Phi$. We will use the latter as an abbreviation for the former for this section. In fact, it would not be hard to redefine ordered theory presentations such that the points were labelled by sets of sentences instead of just sentences, which would remove the need for this abbreviation, and for the assumption that Φ is finite which its use implies. All the definitions and results of chapters 2 and 3 would go through.

(For the exact definition, see definition 3.20.)

Using this idea we can define a consequence relation $\mid\!\sim$ which embodies the default as in Makinson's work. The obvious thing to do is to let $\Phi \mid\!\sim \psi$ mean $\Delta * \Phi \models$ However, we know from proposition 3.18 that $\bot$ does not have its classical behaviou in the context of OTPs. We can get improved results by setting:

Definition 5.2 $\Phi \mid\!\sim \psi$ if $\bigwedge \Phi = \bot$ or $\Delta * \Phi \models \psi$.

That is to say, if Φ is contradictory then it entails everything; otherwise, it entai just what the illustrated OTP entails.

Lemma 5.3 $\mid\!\sim$ is the inference relation corresponding to the preferential model stru ture $\langle \mathcal{M}, \Vdash, \sqsupseteq^{\Delta}\rangle$.

Proof We have to show:

$$\Delta * \Phi \models \psi \text{ or } \bigwedge \Phi = \bot \quad \text{iff} \quad \forall M.\, M \Vdash_{\sqsupseteq^{\Delta}} \Phi \text{ implies } M \Vdash \psi$$

If $\bigwedge \Phi = \bot$ then both sides are true; the left-hand because the second disjunct is tru and the right-hand is vacuously true. If $\bigwedge \Phi \neq \bot$ then by definitions 2.25 and 5.2, is sufficient to show that $M \Vdash_{\sqsupseteq^{\Delta}} \Phi$ iff $M \Vdash \Delta * \Phi$. Since $\bigwedge \Phi \neq \bot$, this follows fro proposition 3.27.

Proposition 5.4 The preferential model structure $\langle \mathcal{M}, \Vdash, \sqsupseteq^{\Delta}\rangle$ is classical and sto pered.

Proof Classicality follows from the fact that $\langle L, \mathcal{M}, \Vdash\rangle$ is classical logic. We sho that it is stoppered as follows. Suppose $M \Vdash \Phi$. We seek $N \sqsupseteq^{\Delta} M$ with $N \Vdash_{\sqsupseteq^{\Delta}}$ By lemma 3.16 pick N such that $M \sqsubseteq^{\Delta * \Phi} N$ and $N \Vdash \Delta * \Phi$. (Recall that classic logic is compact, and we assumed Δ was finite.) Then, by proposition 2.27, $N \Vdash \Phi$. remains to prove:

1. $N \sqsupseteq^{\Delta} M$, i.e. $M \sqsubseteq^{\Delta} N$. Since $M \equiv_{\Phi} N$, this follows from proposition 3.24.
2. $N \Vdash_{\sqsupseteq^{\Delta}} \Phi$. We already have that $N \Vdash \Phi$. Suppose $N' \Vdash \Phi$ with $N \sqsubset^{\Delta} N'$. Sin $N \sqsubseteq_{\bigwedge \Phi} N'$, by corollary 3.25 we have $N \sqsubset^{\Delta * \Phi} N'$, which contradicts $N \Vdash \Delta *$

Corollary 5.5 $\mid\!\sim$ satisfies supraclassicality, inclusion, cumulativity, and distributivit

Proof From proposition 5.1.

We have shown that OTPs over classical logic can yield a default inference relatio in the sense of Makinson, with good formal properties.

Chapter 6

Applications in Software Engineering

This chapter represents the beginnings of applications of the ordered theory presentations described in this thesis to topics in software engineering. We start by describing some of those topics in §6.1 and §6.2, and then we consider how our formal account of *defaults* and *revisions* may be applied in specification theory (§6.3 and 6.4). In §6.5 we make these ideas more concrete by working out an example in a particular logic called MAL. In §6.6 related work is compared and finally, conclusions are drawn in §6.8.

Some of the material in this chapter has been published as [60].

6.1 Introduction

Software engineering is concerned with the design and development of software and software systems. A *software system* is a system of one kind or another which is driven by software; examples include lift systems, nuclear reactors, washing machines and so on. Software engineering includes the study of the *software process*—the process by which software is obtained from informally stipulated *requirements*—as well as issues of *software correctness*, *specification theory*, *modularity*, *re-use* and other topics. All of these will be discussed in one way or another in this chapter.

One of the most important concepts in software engineering is the *specification*. A specification is a formal description of a piece of software or a software system. The specification stands between the informally stipulated initial requirements and the final implementation (see figure 6.2 on page 91); it is against it that correctness may be measured. The connection between software engineering and logic is the fact that the specification of a system denotes a *theory presentation* in a logic. As already seen in §2.1, a theory presentation is a finite collection of sentences; they are the axioms of the specification.

It should come as no surprise to the reader that the principal idea of this chapter is that better results can be achieved by giving the semantics of specifications in terms of *ordered* theory presentations. This will enable us to include *defaults* in specifications, and formally to describe *specification revision*.

An important notion in specification theory is that of *structure*. Large systems should be split into small components and specified independently, in order to en-

hance readability, writability, and to improve the chances of being able to demonstra correctness. The components into which a specification is split are variously calle modules, objects and agents. The structure of a specification is conveniently illu trated in diagrams like those of figure 6.1. Diagram (a) shows how an n-floor li system is composed of a lift and n floors; a floor is itself composed of two buttons, o for going up and one for going down. The lift is made of a panel of n lift buttons a a set of doors and the lift's position. Part (b) of the figure shows how structuring also used to represent the *provenance* of specification components; it may be in tern of aggregations, as in (a), or specialisations and revisions, as in (b), which shows se eral versions of a specification of the behaviour of a UNIX-like command shell. The examples are considered more fully later in the chapter.

In logical terms, the 'objects' in structure diagrams represent a pair consisting the *language* used to describe the component in question together with the *axiom* which express the behaviour. The language of an object is often called its signatur The axioms form a theory presentation over the language. The 'arrows' are theoren preserving maps between these theory presentations. That is to say, an arrow betwe two objects is in the first instance a map between the languages, satisfying certa syntactic criteria such as preserving sorts. In addition, the map can be extended mapping sentences in an obvious way, and should be such that any consequence of t axioms of the first object is mapped to a consequence of the axioms of the second objec In specification terms, this means that the second object *inherits* the language of t first object (modulo possible renaming) and also inherits its behaviour or characte All this will be stated formally later in the chapter (§6.5).

The main ideas in this chapter are the following:

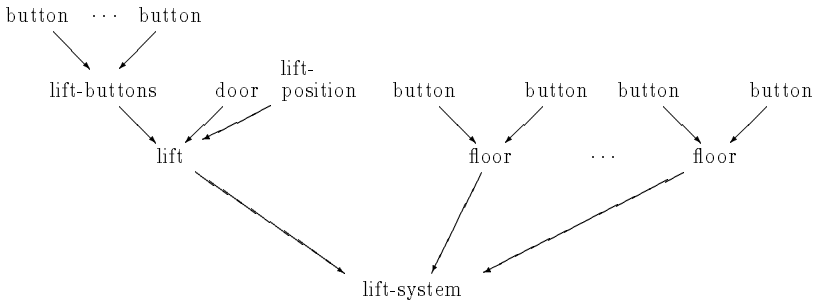
1. Ordered theory presentations are the right tool for giving semantics to specific tions with defaults and specification revision. Thus, a specification should deno an OTP rather than an ordinary theory presentation.

2. Moreover, the structure of the OTP representing a specification with defaul comes from the structure of the specification.

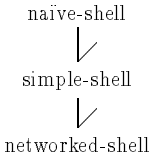
3. And the structure of the OTP representing a specification with a revision histo comes from the process by which the specification was obtained.

4. Finally, in an integrated framework for structured specifications, these ideas ma be combined to obtain the semantics of a specification by an OTP whose structu comes both from the specification's structure and the process by which it w obtained.

This chapter represents work of a more speculative nature than the main body the thesis, and is also the subject of ongoing research. Much of the outstanding researc pertinent to OTPs in general will be of use here; for example, the development of proof theory is perhaps the biggest outstanding problem. There are also technical issu which are of particular relevance to this chapter; for example, making the concept ordered presentations properly *institution independent* would mean wider applicabili (this point will be expanded upon in §6.6.2). There is also some work in demonstratin that the techniques advocated in this chapter are of value to software engineers. Son objections to the ideas are raised and, I hope, quashed at the end of the chapter (§6.7



(a) The structure of the lift system



(b) The structure of a networked-shell

Figure 6.1: Structures for specifications

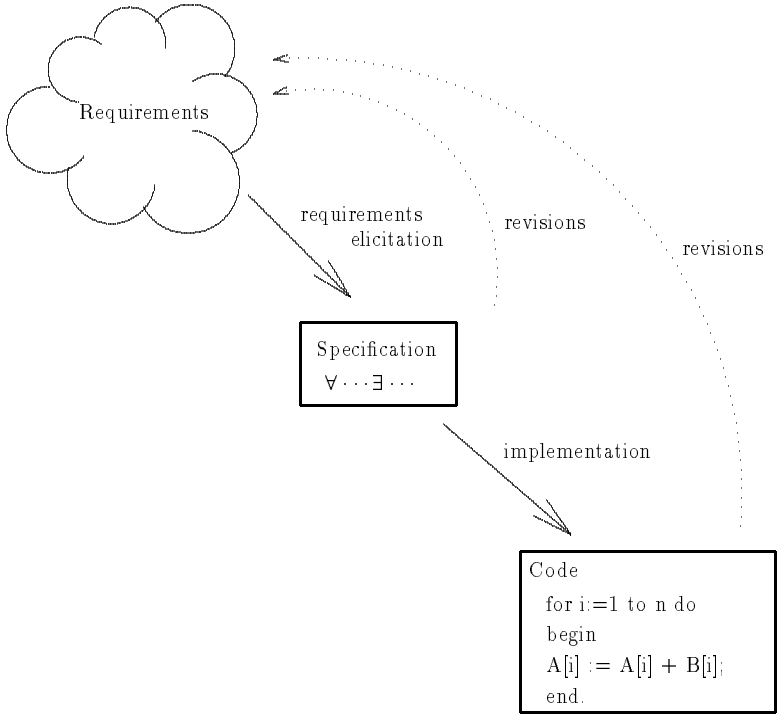


Figure 6.2: The software process

6.2 The software process

The 'software process' is the means by which software systems are produced, startin from a loose specification of requirements dictated by a 'customer'. The idealise picture of how this takes place is shown in figure 6.2 (one should for the moment igno the dotted arrows). There are three persons involved. The customer has the inform requirements in his or her head. The specifier has the job of eliciting these requiremen and writing them down in a formal specification. The programmer implements t specification by writing a program which meets it.

Of course, it is widely recognised that this *never happens* (see eg. [52]). T reasons for this are mostly that revisions to the informal requirements take place parallel with the processes of elicitation and implementation. These are represente by the dotted arrows. Some of these revisions have 'external' causes; the custom is responding to demands from, say, his or her organisation. But some are inhere to the process of *formalisation*. The requirements elicitation process typically caus

the customer to realise that there are gaps in his requirements, or inconsistencies or undesirable consequences which cause him to change the requirements along the way. He did not realise for example (until it was pointed out to him) that asking for x and y meant that he would have to have z as well. So he changes things as he goes along. Some of the undesirable consequences in his requirements may not turn up until the testing stage after the implementation has been carried out. For further ways in which the formalisation and implementation can bring about changes in the initial requirements, see eg. Lehman [41].

The fact that the process of elicitation leads to changes in the requirements is one of the benefits of formalisation, and should not be eschewed. The whole point is that it is necessary to flush out the inconsistencies and undesirable consequences as early in the software process as possible. But still, the model shown in the diagram is unattractive—every time the informal requirements change, the process of formalising the specification has to start again from scratch. The question then arises, is there a way of revising or changing a specification en-route? This question has already had some attention in Finkelstein [19], where a low-level mark-up language for stipulating revisions to specifications is proposed. In this chapter, we will advocate applying the results of chapter 4 to this situation, which will yield a high-level method of revising specifications. The specifier can say, in effect: I want this specification, or as much of it as I can have, given that I also want this property.

Another aspect of the idea of revising specifications, which again shows how the concept is intrinsic to the software process, is so-called 'incremental specification'. What often happens in describing requirements is that the full story is not given all at once. Rather, some broad generalisations are made in the first instance, and later on these are qualified and modified by more detailed statements. For example, consider the following 'specification':

> The admission charge is £2. Students, old age pensioners and unemployed persons get 20% reduction; but old-age pensioners resident in Westbourne Lodge get 30%. There is a 10% surcharge at weekends (this applies whether the price is discounted or not). Parties of over 10 persons are admitted at £1·50 per person.

In this specification, later sentences fill in details of (and thereby contradict) the generalities of earlier sentences. Thus, the specification is acquired incrementally and the theory of belief revision will be of use in modelling this formally. The elicitation process—the process of obtaining a formal specification from informal requirements—is the most difficult aspect of software engineering. Coding these kinds of generalisations as defaults in the specification will make it easier.

We call these generalisations *explanatory defaults*. For a more computer-flavoured example, consider the process by which the behaviour of a command shell (in UNIX, say) is explained. Typically, initial explanations will include statements like "`rm` *file* removes the named file", but these statements should be regarded as defaults because they only hold *most* of the time. Such explanations are quickly followed by provisos, like "you must be in the same directory as *file*", "you must be the owner of *file*", and so on. These are the exceptions to the default. On small systems the list of such exceptions may be small enough to enumerate, but systems which interact in wider contexts need more and more exceptions to be catered for. The file system must be

mounted *read-write*, for example; the network must have the right authorisations, ar so on. All we can really say in the last analysis is that `rm` *file* tends to remove *file*, ar does so only if a multitude of other conditions are satisfied.

6.3 Specifications with defaults

The examples of revision and defaults in the last section are intrinsic to the softwa process; they arise naturally and must be dealt with (whether formally or not) if one to have an acceptable theory. In this section we consider defaults for themselves—on without which we could do, but which make the specification task easier. Defaults specifications occur when a component has a certain "normal" behaviour which ma be overridden when the component is incorporated into another.

Aggregation

Imagine specifying a lift system. There is a lift, with buttons and indicator lights insid and there are doors. There are n floors, again with buttons and indicator lights. Th indicator lights switch on and off in response to button pressings, and the lift goes fro floor to floor depending on the state of the lights. Sometimes it opens its doors to l people in and out.

Here are some statements which might be included in the customer's informal r quirements.

1. If the lift is at the ith floor and it goes down by one floor, then it is at the $i \perp 1$t floor ($2 \leqslant i \leqslant n$).
2. Pressing and releasing the alarm button causes the alarm to sound.
3. The lift will not move up or down unless the doors are closed.
4. When the lift is at the ith floor, the indicator light for the ith floor is off.
5. Pressing and releasing a button for a floor causes the corresponding indicat light to come on.
6. Pressing and releasing a button for a floor causes the lift to arrive at that floo

Not all of these statements are true all of the time about lifts; and in particular, son of them contradict others. The statements are in increasing order of *violability*. Th first is always true, for it simply says what it is for the lift to go down. The secon and third are *nearly* always true; only things like power failures can cause them n to happen. Number 4 is more routinely violated, for example by holding down the *i* button. The fifth statement has yet more common exceptions; for example, if the lift already at that floor the indicator light will not come on. (Nevertheless, it is the *nor* for the light to come on when the button is pressed, and an *exception* when this fails Statement 6, as people who live or work in large blocks will know, is best described hopeful.

An 'ideal' lift should satisfy all of these statements, insofar as they are consiste with one another and the other statements in the specification. Even when there a

inconsistencies, we may want to retain 'as much' of the sentences as possible. We may want one sentence partially to override another. Thus, for example, sentences 4 and 5 are inconsistent, given certain other likely assumptions[1]. We may reject 5 for the special case of the lift being at the floor for which the button is pressed, but we want to retain it for all the other cases.

Many questions arise from the above discussion.

1. Can we handle these kinds of defaults in specifications by using OTPs?
2. Where does the ordering in an OTP come from?
3. Does the way in which sentences partially override each other in OTPs match with the requirement that sentence 4 partially overrides 5 in the example?

We cannot give definitive answers. We *can* say the following.

Not all the statements made in the informal requirements stipulations are appropriate for inclusion in the specification, or at least, not as they stand. Sentence 6, for example, is more the kind of thing one would want to check as a consequence of the specification than code in directly. It might be coded in implicitly, by a lot of axioms constraining the behaviour of the lift in a more 'local' way. On the other hand, sentences 1 and 2 are precisely the kinds of sentences one would expect to find in a specification. So are sentences 4 and 5, apart from the fact that, as we have observed, they conflict.

The conflict between 4 and 5 can be resolved by appealing to the *specificity principle*. It states that

> *Default statements about a more specific class of objects override those about a bigger class when there is conflict.*

This principle (which was already introduced in §1.2.1) is well-known in artificial intelligence [13, 69]. It applies in this case because statement 4 is about lifts, while statement 5 is about buttons. The structure of the lift specification (figure 6.1(a)) is that the lift object (or module) incorporates (n copies of) the button object. Therefore, the class of lifts is more specific than that of buttons. The specificity principle says then that statements about lifts override those about buttons, so 4 overrides 5.

Our provisional answer to Question 1 is that we will in the main restrict ourselves to defaults to which the specificity principle is applicable. This may be too restrictive, but widening the class is left as a matter for further research. Even with the restriction, it should become clear that these defaults form a huge class. This means we can already answer Question 2:

> The structure of the OTP for a specification with defaults comes from the structure of the specification.

The precise way in which this works will become clear in §6.5.4, where the example of the lift and sentences 4 and 5 is worked out in full.

[1] The additional assumptions required are the 'locality axioms' to be described in §6.5.3. In this case they say simply that the press-and-release action does not affect the directly affect the floor the lift is at.

As far as Question 3 is concerned, the answer is surely 'yes'. The examples §1.3 and §4.6 should be enough to persuade the reader in the 'static' case witho actions. For the case with actions, there is a danger of phenomena like the so-calle Yale shooting problem [32] to appear; this is discussed elsewhere (chapter 5).

Specialisation

The lift-button example has to do with *aggregation*, that is, to do with putting sma objects (like buttons) together to form larger ones (like lifts). This can easily be se by looking at the full structure of the lift specification, given in figure 6.1 (page 90 Defaults about the aggregated objects override those of the components.

But *specialisation* is another specification construct which has to do with speci cation structure, and the specificity principle applies here too. A specialisation of a object is another object of the same kind (loosely speaking) which contains all th features of the first object and more besides. Consider, for example, the user-interfa of an auto-teller (cash dispenser). As an object in its own right, it has actions su as the pressings of keys, and state variables which describe the message on its scree Its usual behaviour is to echo characters typed on its keyboard on its screen. Now o may consider a specialisation of this object, which has the same features as before b the additional feature of a 'password mode', in which it does *not* echo characters o its screen.

The proposed way to handle this situation is to stipulate that the echoing behavio is a default which the specialisation overrides. The specificity principle sees to it th the default of not echoing (the default of the specialised interface) overrides the echoi default, because the specialised interfaces form a more specific class. The key poi is that the behaviours of these interfaces differ from one another on certain action although, of course, the behaviour of the specialised interface is *mostly* the same that of the original interface; that is why it is appropriate to speak of inheritanc albeit with exceptions.

The key idea in such examples is that axioms or defaults in wider contexts ca override defaults in smaller ones. A wider context may be created from a smaller o by aggregation or specialisation, as in the above examples.

Explanatory defaults again

It turns out that explanatory defaults can be viewed as defaults arising from specia isation, and are thus also amenable to analysis by our method. Consider again th example of the UNIX `rm` command. The first stage of the explanation, in which th axiom "`rm` *file* removes *file*" is given, should be thought of as the specification of th 'naïve shell'. Ultimately, after many elementary exceptions and specifications of va ant behaviour have been given, we may arrive at the specification of the 'simple shel It specifies the way shells used to work, in the good old days before networks, and it a specialisation of the naïve shell in which some of the defaults have been overridde Then, dozens of further exceptions and variations are given, until a supposedly exa description of the behaviour of unix shells in a networked setting is obtained. This turn is a specialisation of the simple shell. The structure diagram is then the one give in figure 6.1(b) (page 90). Thus, explanatory defaults can be viewed as specialisatio

defaults.

6.4 Design by difference, or specification revision

The sections above described using defaults in specifications, with the resulting OTP having an order which came from the structure of the specification. But there is another way in which the ordering of an OTP can arise during the software process, which is by specification revision. This idea is still a matter for further research, but it is of great importance if one is to get full value from specified components. We mention it briefly here as a placeholder for the (yet to be developed) full story.

The idea is to apply the methods of chapter 4 to specifications. This has both small-scale and large-scale applications.

- In the small, one can consider re-using components from a library of standard objects. If a component doesn't quite fit the application because it has unwanted properties, revise it with the desired properties.
- In the large, whole specifications may be constructed in this way. For example, the recent Rover TV advertisement showed how the Metro motor-car was conceived as a Mini with certain properties added. These properties conflicted with the old ones, which means the *revision* is not merely a matter of *refinement* or *enrichment*. Thus, the Metro is specified by stipulating its *differences* from a Mini.

In practical terms one may envisage a software engineering environment (implemented on a computer) which allows one to explore a 'design space' by both small-scale and large-scale revisions of the type described here.

The obvious difference in the case of *specification revision* as against specifications with defaults is that the ordering in the resulting OTP comes not from the structure of the specification, as it did for defaults, but from the process by which the specification was obtained (the revision history). But in fact, these genealogies are not so different. One can think of a revision history as showing the structure (through refinement) of a component; for example, one can think of the structure diagram for the networked-shell (figure 6.1(b)) as a revision history or one can view the earlier objects as the components of which the networked-shell is made. On the other hand, a non-linear structure diagram such as that of figure 6.1(a) represents a revision history in more than one dimension. For example, the manufacturer's intention is that the button's light illuminates when the button is pressed. This is encoded in the button's specification. But the specification was revised for incorporation in the lift, since in that context it is to have the property that it does *not* light when pressed in certain circumstances; namely, when the lift is in a state in which the request made by the user by pressing the button is inappropriate. The revision is implemented via a complicated interface between the components which may not even be part of the specification—that is why defaults are needed.

6.5 Structured specifications and modal actio logic

In this section the 'classical' theory of structured specifications in modal action logi is described formally (§6.5.1 to §6.5.3). Then the formal changes needed to use OTI for the semantics for specifications is given (§6.5.4).

6.5.1 MAL, its syntax and semantics

Modal action logics (also known as *dynamic logics* or *multi-modal logics*), have for ov a decade been used to specify state-based software systems. The basic idea of mod action logics is to model actions moving the system from one state to another. Such logic has a family of modal operators, one for each action that the system can underg and its semantics is given by a set of states and a family of relations on the states, o interpreting each modality. For example, the fact that the action a if executed in state satisfying condition ϕ results in a state satisfying ψ is expressed by the axiom

$$\phi \rightarrow [a]\psi.$$

There are many accounts of modal action logics [17, 15, 29, 62]. We describe a simp version which we refer to as MAL below. This logic satisfies the conditions of §2.1, the semantics of OTPs in it is defined in chapter 2.

We have mentioned how a component within a specification is, in logical terms, *signature* together with a *theory presentation* over the signature. A MAL signature a set of action symbols and a set of proposition symbols; the action symbols are us to describe the actions which the system may perform, and the proposition symbo are used to represent the state of the system. Thus, actions update the values of t propositions.

A MAL signature $S = \langle A, P \rangle$ consists, then, of two sets; a set of actions A and a s of propositions P. For example, here are the signatures for some of the objects of t lift system (figure 6.1(a)):

button has the signature $\langle \{\text{press}, \text{cancel}\}, \{\text{lit}\} \rangle$. The button may be pressed or ca celled, and has a light which may be on or off.

door has the signature $\langle \{\text{open}, \text{close}\}, \{\text{doors-open}\} \rangle$.

lift-position has the signature $\langle \{\text{up}, \text{down}\}, \{\text{floor}_1, \ldots, \text{floor}_n\} \rangle$. floor_i represen whether the lift is at the ith floor or not.

lift has the signature $\langle \{\text{press}_1, \ldots, \text{press}_n, \text{cancel}_1, \ldots, \text{cancel}_n, \text{open}, \text{close}, \text{up}, \text{down} \{\text{lit}_1, \ldots, \text{lit}_n, \text{floor}_1, \ldots, \text{floor}_n, \text{doors-open}\} \rangle$. Notice the renaming of the pre actions.

lift-system has, in addition to the signature elements of the lift, the action pres alarm and the attribute alarm.

Given a signature, atomic propositions are composed to form more complex se tences using the usual boolean operators $\wedge$, $\vee$, $\rightarrow$, $\neg$ etc. There is also the constru

$[\cdot]\cdot$ which, as already mentioned, is used to describe the effects of actions. If a is an action symbol and ϕ a sentence (which may also contain action terms) then $[a]\phi$ expresses the fact that ϕ holds after a has taken place. The syntax of formulas is therefore as follows:

- If $p \in P$ then p is a sentence.
- If ϕ and ψ are sentences and $a \in A$ then $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, $\phi \leftrightarrow \psi$, and $[a]\phi$ are sentences.

To illustrate this syntax, here again are the first five of the six statements about the lift given on 93.

1. $\text{floor}_i \rightarrow [\text{down}]\text{floor}_{i-1}$ (for $2 \leqslant i \leqslant n$)
2. [press-alarm]alarm
3. doors-open $\rightarrow$ ([up]$\bot \wedge$ [down]$\bot$)
4. $\text{floor}_i \rightarrow \neg\text{lit}_i$
5. $[\text{press}_i]\text{lit}_i$

An interpretation M for a signature is a function which takes states to an assignment of truth values to the atomic propositions. States are represented by *traces*. A trace is a finite sequence of actions in the signature, and denotes the state which results by performing the actions in the initial state. Thus, If σ is a trace and p an atomic proposition, then $M(\sigma)(p)$ is a truth value which says whether p is true or false in the state resulting from performing the actions in σ in the initial state.

Satisfaction in states is defined in the following way:

$$\begin{array}{rll} M(\sigma) \Vdash p & \text{if} & M(\sigma)(p) = \mathbf{t} \\ M(\sigma) \Vdash \neg\phi & \text{if} & M(\sigma) \nVdash \phi \\ M(\sigma) \Vdash \phi \wedge \psi & \text{if} & M(\sigma) \Vdash \phi \text{ and } M(\sigma) \Vdash \psi \\ M(\sigma) \Vdash \phi \vee \psi & \text{if} & M(\sigma) \Vdash \phi \text{ or } M(\sigma) \Vdash \psi \\ M(\sigma) \Vdash \phi \rightarrow \psi & \text{if} & M(\sigma) \Vdash \phi \text{ implies } M(\sigma) \Vdash \psi \\ M(\sigma) \Vdash \phi \leftrightarrow \psi & \text{if} & (M(\sigma) \Vdash \phi \text{ iff } M(\sigma) \Vdash \psi) \\ M(\sigma) \Vdash [a]\phi & \text{if} & M(\sigma \circ a) \Vdash \phi \end{array}$$

(In the last clause, $\sigma \circ a$ is σ with a appended.) This is a rather naïve way of handling actions, which means that the logic cannot support concurrent actions. It has the advantage of being simple, however, which suits the purposes of this chapter.

Satisfaction in *interpretations* is then defined as follows:

$$M \Vdash \phi \text{ if for each } \sigma,\ M(\sigma) \Vdash \phi.$$

This means that a sentence is true overall in an interpretation iff it is true in every state of the interpretation.

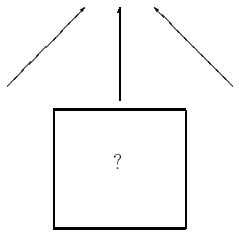
If ? is a set of sentences and ϕ a sentence, $? \models \phi$ holds if for every M, if $M \Vdash \psi$ for each $\psi \in ?$ then $M \Vdash \phi$. $? \models \phi$ is read ? entails ϕ. If ? is the set of axioms of a specification and $? \models \phi$, then ϕ is a consequence of the specification.

6.5.2 The frame problem

The frame problem is the problem of having to specify the action-attribute pairs whi are such that the action does *not* affect the attribute. For example, the 'open' actio in the lift specification does not affect the 'floor_1' proposition. This is of course t case for the majority of such pairs. The number of frame axioms needed to do th grows rapidly with the size of the signature, and specifications therefore quickly becon cluttered with such axioms.

This problem is widely known in AI, where the solution is to employ a default *fra rule* which says for every action a and attribute p that (unless there is proof to t contrary) a does not affect p. With OTPs one is of course in an excellent position follow this route; if ? is the OTP encoding the axioms of the specification[2] in questio one might simply add the relevant default:

$$\bigwedge_{\substack{a \in A \\ p \in P}} (p \leftrightarrow [a]p)$$



Such an approach must be augmented by an explicit priorisation of competing defaul if problems like those of Hanks and McDermott [32] are to be avoided. These problem and this remark are expanded upon in chapter 5, but other than for making th remark we have not investigated the defaults approach. This is because the structurin mechanism mentioned earlier provides an alternative solution to the frame proble which has been widely used in specification theory.

6.5.3 The structuring principle

The structuring principle mentioned in §6.1 (see figure 6.1) is important in specificatio not only because it enhances readability and verifiability but also because it overcom the frame problem which is a characteristic of action-based logics. This works becau the structure of a specification affords a way of telling, in the majority of cases, wheth an action can affect an attribute or not. There are several ways of arranging thi including the following:

- The principle may dictate that an action can only affect the state-variabl (propositions) in the signature in which the action is declared. Thus, for exar ple, the 'press' action can only affect the 'lit' proposition in the lift specificatio since 'press' is declared in **button** and 'lit' is the only proposition declared **button**. This is often called 'object-orientation'.

[2] The way in which Γ is obtained has not yet been described. This is done in §6.5.4.

- The dual of this approach would be to stipulate that a state-variable can only be changed by actions in the signature in which the state-variable is declared.
- Or, one may take a mixed approach (dubbed 'agent-orientation' in [62]), in which annotations to actions and state-variables control exactly the scopes in which they can update and be updated.

The object-oriented approach (the first one) is the most popular.

Care must be taken in framing the 'locality axioms' which these principles give rise to. For example, if one takes the object-oriented approach it is easy to be too restrictive. The axiom

$$\text{floor}_3 \leftrightarrow [\text{press}_5]\text{floor}_3$$

says that the press_5 action does not affect the floor_3 proposition, as wanted, but it means further that the floor_3 proposition can never be changed by an occurrence of press_5. This is too strong if we want to allow concurrent actions, for a press_5 may occur concurrently with an up, in which case floor_3 may change. However, our simple semantics has already ruled out concurrent actions.

As well as controlling locality, the structuring principle is about making large specifications out of small ones. We have seen how the lift system specification is composed out of smaller specifications and ultimately out of atomic ones. Each node in the lift-system structure diagram represents the specification of a component, and the arrows are maps between the specifications in the following way. If $A \xrightarrow{f} B$ is a map f between A and B, then

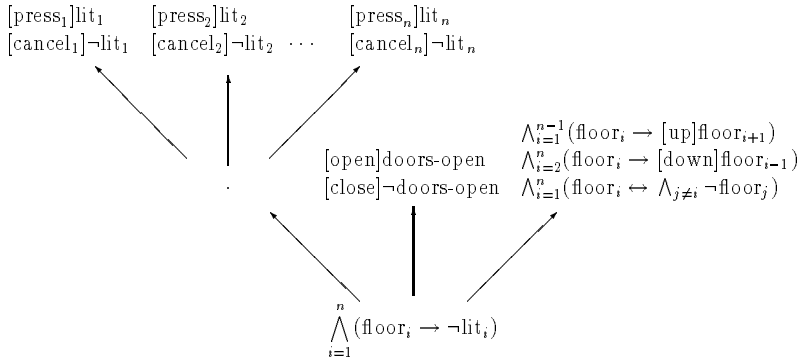
- f is a map between the signatures of A and B; that is, it maps the actions of A to actions of B, and also A-propositions to B-propositions. (In a more general setting, types and sorts also have to be preserved.) f can be extended to a map from sentences in the signature of A to sentences in the signature of B in the obvious compositional way: $f(\phi \wedge \psi) = f(\phi) \wedge f(\psi)$, $f([a]\phi) = [f(a)]f(\phi)$, etc.
- f preserves the properties of A. That is, if $A \models \phi$ (the axioms of A entail a sentence ϕ) then $B \models f(\phi)$

This is essentially the categorical framework of Goguen and Burstall [9].

6.5.4 Specifications and OTPs

The idea is that a structured specification denotes an OTP in which the ordering comes from the structure of the specification. Thus, conflict between sentences in different components is resolved by the specificity principle. For example, part of the OTP corresponding to the lift system showing how the conflict between sentences 4 and 5 is resolved is given below. Only the 'lift' branch of the tree in figure 6.1(a) is given, and

only the axioms relevant to the discussion are shown:



6.6 Related work

There has been similar work done by S. Brass and U. Lipeck in [5]. Those authors ar I are currently working on a unification of our ideas [6]. In this section we descril other, less directly related work.

6.6.1 Deontic MAL

In the fully-fledged version of MAL presented in [62], there are also *deontic predicat* written *per* and *obl* which apply to action terms. See also [16, 35, 37]. These deont predicates are used to express the fact that certain actions are (or aren't) permitted obliged in certain states. Deontic predicates provide a more elegant way of expressi sentence 3 of the lift specification, for example:

3. doors-open $\rightarrow \neg per(\text{up}) \wedge \neg per(\text{down})$

In the earlier encoding of this sentence (§6.5.1) it was a logical impossibility for t lift to go up or down with the doors open. The ordering of the theory presentatio meant that we could violate this, but still the encoding with deontic predicates seen neater. Deontic predicates add much to expressibility. For example, it is now possib to express sentence 6:

6. $\text{floor}_i \wedge i \leqslant j \rightarrow [\text{press}_j]\,obl(\text{up}^{j-i})\text{floor}_i \wedge i \geqslant j \rightarrow [\text{press}_j]\,obl(\text{down}^{j-i})$

The first of these says that if you are at a certain floor and you press the button f a superior floor the lift is obliged to move upwards by the appropriate amount; t second is the opposite case. Of course this is rather fanciful, for we have not said wh up^{j-i} means and still less what $obl(\text{up}^{j-i})$ means; also, the lift will not in general ob the request at once, but may interleave it with others. However, our aim here is mere to motivate some of the issues which deontic MAL is attempting to address, and particular to point out the nature of the increased expressive power.

The semantics of deontic predicates may be simplified[3] as follows. It was already seen (§6.5.1) that MAL interpretations are traces, i.e. sequences of action terms. From a trace we can obtain the state which would result if the actions of the trace were carried out, in order, starting with the initial state. We add to the interpretations two functions P and O which, given a trace, return the sets of actions which are permitted and obliged respectively. A trace is said to be normative if every action which takes place in a state is permitted in that state (according to the function P); and every obliged action (according to O) in a state is eventually performed in some later state.

The question of crucial importance as far as this chapter is concerned is how the specification of norms with deontic predicates relates to its expression with defaults, or with OTPs in particular. This question is of course open. It seems to be the case that the two modes of expression are complementary. For example, it was already noted that the expression of sentence 3 is more elegant in the deontic framework[4]. But the deontic framework has no hope of expressing norms like 4, for they are entirely static and the deontic predicates act only on actions. If it is true that both ways of expressing norms are desirable, one might ask how they interact together. In short, what are the properties of ordered presentations of *deontic* theories? That is one line of research I hope to undertake after my Ph.D.

6.6.2 Institutions

The 'proper' way of arranging matters when using logic for specification is to use institutions, because they provide an up-front way of interfacing components with different signatures. The structure diagrams are in fact diagrams in a category of specifications in which the morphisms are theorem-preserving maps between the signatures. How, then, does this relate to the use of OTPs? Again, this is a matter for further work and I can only hint at the solution:

The theory of institutions may be generalised to the so-called π-institutions of [18]. The difference is that π-institutions work for any logic satisfying the structural properties of proposition 2.9 (page 27), not just those defined in terms of models and satisfaction. Also, in π-institutions the theory *presentation* is the crucial unit of information, not the *theory*. This suits our purposes. Morphisms exist, then, between theory presentations and it is proved that for the definition of morphism mentioned at the end of §6.5.3 it is sufficient to require (in terms of what was said there) that f map A-axioms to B-axioms: if $\phi \in A$ then $f(\phi) \in B$.

This leads naturally to the idea of morphisms between OTPs, and therefore to institutions handling defaults, which we dub 'd-institutions'. It is obvious from §6.5.4 that we require that A-axioms be mapped to B-axioms, but they may be further up the ordering in B than in A. Thus,

Proposal 6.1 Ordered theory presentations are now paired with their signatures. Let $? = \langle L, X, \leqslant_X, F\rangle$ and $?' = \langle L', X', \leqslant_{X'}, F'\rangle$ be theory presentations with signatures L and L'. Let $f : L \to L'$ be a map between their signatures. f is a morphism from

[3] See [16] for the full story.

[4] Although, it has yet to be shown that the deontic framework alone can enable reasoning after norms have been violated.

$?$ to $?'$ if there's a morphism of partial orders $g : X \to X'$ such that for each $x \in X$ $f(F(x)) = F(g(x))$.

In other words, $?$ is mapped into $?'$ in a way that preserves its ordering. The are options, such as: can two x's in X collapse to the same one in X'? (The propos says yes, but we could change it.) What about the other way around? etc. Mo investigation is needed.

6.6.3 Other default logics

The idea of using default information in specifications was motivated in sectio 6.1 to 6.4 as a way of giving a formal account of many issues in software enginee ing hitherto treated informally. But why should one use OTPs, given the plethora arguably better established default logics on the market? A full examination of the alternative logics for defaults is given in chapter 5. In short, the reasons for which t framework of OTPs scores highly over rival non-monotonic logics are:

1. Defaults are represented by ordinary sentences in the language. The alternativ taken by other default formalisms, for example in representing defaults by rul of inference or sets of predicates to be minimised, would mean that the specifi had to expend more effort in coding up the defaults required. (More detail this point can be found in chapter 5.)

2. The interaction between defaults of different priorities is simple to express OTPs, and much harder in other formalisms.

3. The specificity principle yields the appropriate ordering of defaults in cases i volving inheritance (like the button/lift example). This means that we do n need to enumerate the exceptions to defaults, as is necessary in most other defau logics. (This point is amplified in chapter 5.)

4. The ability to handle defaults not just in classical logic but in modal and mult modal logics is available with OTPs. (It is hoped that future work will exte the theory of OTPs to arbitrary institutions).

5. Ordered theory presentations exhibit the relationship between theory revisio and default theories. They would therefore be a suitable theory to form the bas of a software engineers' tool which supports both of these ideas.

6.7 Objections

It was pointed out at the beginning of this chapter that the ideas presented here a of a more speculative nature than elsewhere in the thesis. The reader may dislike t idea of the 'loose' specifications motivated here. This section is devoted to presentin objections to defaults and revision in specifications and, I hope, to allaying them.

The most common objection raised is that specifications are by nature exact, and goes against the grain to introduce the slack which comes with defaults and revision I have much sympathy with this view, but I believe that the benefits gained fro

defaults and revisions outweigh the disadvantages. Among the benefits are the ability to represent normative behaviour when it really is a characteristic of the object being specified; the ability to explore a design space; the *improvement* in modularisation which can be obtained (see below); and freedom from the chore of filling in every little detail, instead being able to allow conflicts to resolve automatically. Furthermore, from a methodological point of view, we narrow the gap between the informal requirements and the specification in figure 6.2; without, I believe, the price of widening the gap on the other side, between specification and code. This is because the specifier has an improved medium for expressing the intuitions and intentions behind his or her specifications.

The improvement in modularisation referred to above can be seen by considering the effect of coding in the exception to sentence 5 of the lift specification (§6.5.1). Sentence 5 expresses the fact that the buttons light when pressed, and is an axiom of the button object. The exception noted is when the lift is already at the relevant floor, so taking account of this the axiom would become:

$$\neg \mathrm{floor}_i \rightarrow [\mathrm{press}_i]\mathrm{lit}_i.$$

But this cannot now be an axiom of the button object after all, but must be an axiom of the complete lift system. This is because the vocabulary it uses is not available in the button signature. Thus the motivation for structuring (that is, dividing the specification into constituent objects and axiomatising them individually) in the first place is foiled: every axiom has to be part of the biggest object in order to list all the exceptions.

It might be objected that if some axioms are allowed to override others, we may quickly get into a mess in which we do not know which axioms are being affected by which others. To counter this objection, it should be possible to check at any stage whether a certain axiom expressing a norm is being overridden or not, by checking whether it is a consequence of the specification. And again, the *advantage* is that one can explore the design space by changing the order around until the desired effect is achieved. This gives great flexibility to the specifier. Of course, the ability to do these things assumes a sophisticated interactive software environment which supports OTPs; such a thing is yet to be developed.

Another technical objection is that not all axioms express behaviour which may be overridden. For example, we may wish to keep locality axioms inviolable. This would be prudent, for if we override such axioms we may lose our intuitive understanding of the specification. There are other axioms which should never be overridden too; for example, we already noted that sentence 1 of the lift is true 'by definition'. A purely technical manoeuvre will accommodate this, we can stipulate that a specification denotes a pair $\langle \Delta, ? \rangle$ consisting of an ordinary theory presentation Δ (the inviolable axioms) and an ordered theory presentation ? (the norms). Models of the pair $\langle \Delta, ? \rangle$ are defined as the $\sqsubseteq^{\Gamma}$-maximal models of Δ.

6.8 Conclusions

Much work remains to be done, both technically and motivationally. The technical work includes the development of a proof theory for OTPs and making them properly

institution-independent. The motivational work is to give more elaborate exampl
which are more fully worked out in order to convince practitioners of the value of t
ideas. Of course, these two areas of work go hand-in-hand; technical developments w
enable the motivational ones, which in turn give direction to the technical ones. T
ultimate word on this subject is still a long way off, but I hope that this chapter h
at least introduced the story.

Chapter 7

Conclusions and further work

In this chapter, we describe unfinished work, further work, related work and then recap on the main points of the thesis. The *unfinished work* we discuss is the topic of verisimilitude, introduced in chapter 1. This is done in §7.1. A variety of topics come under the heading of *further work*, and are dealt with in §7.2. Much *related work* has already been discussed in chapters 4, 5 and 6, but an important example has been left until this chapter, described in §7.3. Final remarks are given in §7.4.

7.1 Unfinished work: verisimilitude

The topic of verisimilitude concerns the measurement of theories with respect to the truth. Its origins are in the philosophy of science, and it attempts to give a formal account to the idea, for example, that Einstein's relativistic physics (while perhaps not completely true) is genuinely *closer to the truth* than Newton's classical physics; and the latter, in turn, is closer than Aristotle's physics.

As far as a formal account is concerned, the subject is still a long way from being able to account for the improvements in scientific theories described above. One reason is the so-called *incommensurability* of these theories (T. Kuhn [39]); this means that the language of (say) Newtonian physics cannot be translated into the language of relativistic physics, because the latter deals with entirely different concepts to the former. The formal accounts of verisimilitude currently available not only assume inter-translatability; they assume that the two theories are expressed in exactly the same language.

From the point of view of computer science, the philosophical demands are not so great, and the benefits of a formal account of verisimilitude are more tangible. We have already given the example of predictions in the economy in §1.2.4; this kind of application is relevant for expert systems and in artificial intelligence more generally. In software engineering, one may view specifications and implementations as logical theories, as explained in chapter 6, and the ability to order implementations which do not fully satisfy a specification according to how nearly they do has obvious benefits.

In the literature on verisimilitude (our main source has been T. Kuipers' [12]) the 'truth' is taken to be a logical theory which is complete[1]. However, many of the

[1]Recall that a theory is a consequence-closed set of sentences. A theory T is complete if for all ψ, $T \vdash \psi$ or $T \vdash \neg\psi$.

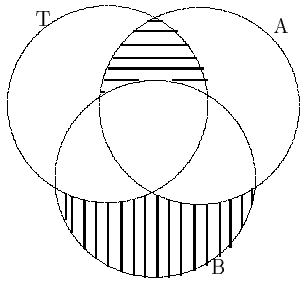


Figure 7.1: The shaded parts are empty iff $B \triangle T \subseteq A \triangle T$

formalisations of verisimilitude in fact define a ternary relation on *arbitrary* theories

$$A \leqslant_T B \text{ if } B \text{ is as close to } T \text{ as } A \text{ is.}$$

The first formal definition of this relation is due to Popper [54]:

Definition 7.1 $A \leqslant_T^{(P)} B$ if $B \triangle T \subseteq A \triangle T$.

If A, B and T are sets (here they are sets of sentences), the condition $B \triangle T$ $A \triangle T$ is illustrated in figure 7.1. The shaded areas are empty if the condition satisfied. This can be restated as the following two conditions:

$$A \cap T \subseteq B, \qquad \text{and} \qquad B \perp T \subseteq A$$

Since T contains only *true* sentences, the first of these can be thought of as saying th B has all the true sentences that A has. If T is complete then its complement consis entirely of *false* sentences, in which case the second condition means that B has more false sentences in it than A has. If T is not complete then the second conditio is not so intuitive.

Another definition of the same relation, due to D. Miller and T. Kuipers (see va Benthem [72]) is

Definition 7.2 $A \leqslant_T^{(K)} B$ if $[\![B]\!] \triangle [\![T]\!] \subseteq [\![A]\!] \triangle [\![T]\!]$.

Recall that $[\![A]\!]$ is the set of models of A. The same diagram and the equivale conditions still hold, with appropriate substitutions ($[\![A]\!]$ for A, etc.). We can par phrase the two conditions as: any A-model which might have been the true situatio must also be a model of B (so B doesn't loose any models); and any models of B whi couldn't have been the true situation must be A-models (so B doesn't introduce a bad models).

We can also show that

$$A \leqslant_T^{(P)} B \quad \text{implies} \quad A \leqslant_T^{(K)} B$$

but the converse implication is false.

These are the principal definitions discussed in the literature. It turns out, however, that both have undesirable consequences. The following observations are apparently due to P. Tichý:

Proposition 7.3

1. If $\leqslant$ is $\leqslant^{(P)}$, then $A <_T B$ implies $B \subseteq T$.
2. If $\leqslant$ is $\leqslant^{(K)}$, then $A \leqslant_T B$ if $[\![A]\!] \cap [\![T]\!] = [\![B]\!] = \emptyset$.

(As usual, $A < T$ if $A \leqslant T$ and $T \not\leqslant A$.)

The first means that $\leqslant^{(P)}$ cannot strictly order "false" theories (that is, theories with at least one false sentence in them). From the point of view of comparing scientific theories, this is obviously inadequate, because although Newton's and Aristotle's theories of physics are both known to be false, the former is closer to the truth than the latter. The second point says that the contradictory theory B (with no models) is an improvement on any theory A which shares no models with T. It is counterintuitive that the contradictory theory should be an improvement on anything.

A proof of the first is given in [66, page 49]; the second is trivial to demonstrate. It should be noted that the second item is not seen as grounds for complete rejection of $\leqslant^{(K)}$; it is still widely discussed.

Neither of the two definitions considered have the maximality property mentioned in §1.2.4, that

$$A \leqslant_T B \quad \text{if} \quad T \subseteq B.$$

Indeed, this property is not even mentioned by van Benthem [72] who considers a variety of constraints of this kind on notions of verisimilitude. I find this surprising. This condition holds trivially if T really is 'the truth', for then T is a complete theory and $T \subseteq B$ implies $T = B$ for any theory B. It is hard to think of examples when T is incomplete in the philosophy of science (perhaps there are some in quantum physics?), but in the computer science examples mentioned earlier, examples abound. In artificial intelligence, we may have partial knowledge about a particular domain against which we wish to measure theories; and in software engineering, the specifications against which we measure implementations are typically incomplete

It turns out that the definitions introduced in chapter 2 easily yield a notion of verisimilitude which has good intuitive properties, does not have the problems of proposition 7.3, and satisfies our maximality property. Moreover, the Miranda program code of appendix A which implements the definitions of chapter 2 can easily be extended to compute verisimilitude orderings for propositional theories, and a few of these are given in appendix B.

The definition of verisimilitude is essentially the ordering $\sqsubseteq_\phi$ of §2.2.4. First, we can define this ordering in terms of a theory T rather than a sentence ϕ:

Definition 7.4

1. $p \in T^{+(-)}$ if T can be presented with only positive (negative) occurrences.
2. $T \models\!\!\!\sim \psi$ if $T \models \psi$ and $T^{\pm} \subseteq \phi^{\pm}$.

3. $M \sqsubseteq_T N$ if $T \models\!\!\!\sim \psi$ implies ($M \Vdash \psi$ implies $N \Vdash \psi$).

Compare definitions 2.34, 2.38, and 2.45.

Since theories can be thought of as sets of models (namely, those that they satisfy we need to raise this definition from being the definition of an ordering on *points sets of points*. We do this in the Egli-Milner way.

Definition 7.5 $A \leqslant_T^{(R)} B$ if $\forall M \Vdash A.\, \exists N \Vdash B.\, M \sqsubseteq_T N$ and $\forall N \Vdash B.\, \exists M \Vdash A.\, M \sqsubseteq N$

This says that for every model of A we can find a model of B which more near satisfies T; and also, every model of B more nearly satisfies T than some model of A

It is easy to check that neither points of proposition 7.3 holds if $\leqslant$ is $\leqslant^{(R)}$, b inspecting the diagrams in appendix B.

A full comparison of this definition with the existing work on verisimilitude has n been carried out; that is why this section is called 'unfinished work'.

7.2 Further work

There are many ways in which the work described in the thesis can be extended an improved; some of these have already been mentioned in earlier chapters. Some of the are about improving the match between the theory of OTPs and related subjects (suc as default reasoning and belief revision); others aim to further the theory of OTI itself. I mention two of the latter kind here which strike me as important.

7.2.1 Institution independence

Ordered theories have been defined for logics that are specified in terms of models an satisfaction and which have an appropriate notion of positive and negative occurrenc of symbols in sentences. This level of abstraction is close to the notion of *institutio* introduced by R. Burstall and J. Goguen [9, 27]. An institution is roughly a log specified in terms of models and satisfaction, but also with additional emphasis o modularity and composability of languages and theories. We would like to generali the definitions for OTPs so they work with an arbitrary institution. The main m tivation for the additional modularity which institutions provide is from specificatio theory.

A fundamental notion in the topic of institutions is that of *signature*, introduced chapter 6. A signature is a set of non-logical symbols which are used to form sentence By 'non-logical' symbols we mean those other than the operators which are built in the logic like $\wedge$, $\rightarrow$ and so on. A signature is a collection of proposition symbol predicate symbols, function symbols, sort symbols, etc. A signature morphism is map between signatures which preserves signature structure—for example, it ma predicates of a certain arity to predicates of the same arity, it preserves sorts, and on. The precise requirements on a signature morphism depend on the signatures question.

Informally, an institution consists of a collection of signatures and signature mo phisms, together with for each signature Σ

- a collection of Σ-sentences,
- a collection of Σ-interpretations, and
- a Σ-satisfaction relation between Σ-interpretations and Σ-sentences

such that a certain condition called the *satisfaction condition* holds. It says that when you change signatures (with a signature morphism), the satisfaction relation between sentences and models changes consistently [28].

A theory in an institution is a signature Σ together with a consequence closed set of Σ-sentences. A morphism between theories is a morphism between their signatures which preserves satisfaction; that is, every model of the sentence translated by the morphism can be reverse-translated into a model of the sentence. In this way, a complex object is specified by a diagram of smaller objects, and its overall behaviour is given by the *colimit* of the diagram.

The institution concept allows intertranslatability between theories and the operation of putting theories together to form bigger ones with the possibility of identifying signature elements.

I have not looked in detail at how the notion of extension of extra-logical symbols in an interpretation which is crucial to the definitions of OTPs may be derived from an institution. That is why it is a matter of further work. I hope that the substantial structure that institutions provide—particularly the morphisms between interpretations of the same signature—will provide the necessary hooks. If they do not, it will be necessary to extend the institution concept.

As well as making OTPs more general, this exercise may improve their definition by making the notion of extension of extra-logical symbols in an interpretation more primitive than the definition of positive and negative occurrences from which it is presently derived.

7.2.2 Proof theory

In his thesis, Y. Shoham [68] argues that proof theory does not make sense for default logics.

> Many of the notions that are quite clear in monotonic logic, such as complete axiomatisation, cease to make sense in the context of non-monotonic logic. The whole motivation behind non-monotonic logics is the desire to be able to jump to conclusions, inferring new facts not only from what is already known but also from what is not known. This seems to imply that traditional inference rules, which are rules for deriving new sentences from old ones, are inadequate. ... Rules that demand checking consistency no longer have the computational advantages of traditional inference rules. Perhaps something else is possible, along the lines of what are known as systems for truth maintenance, in which the entities manipulated by programs are not sentences, but rather beliefs and records of justifications for each belief.

Plainly we must read 'default logic' for 'non-monotonic logic', for the fact that a logic is 'non-monotonic' (i.e. a logic failing the monotonicity property) is not enough to

prevent it having a proof theory presented in a perfectly respectable way; witne
linear logic's sequent calculus. Moreover, simply the loss of computational properti
is not a sufficient reason for concluding that there can be no proof theory, since ma
proof theories are undecidable. However, I argue that Shoham's intuition is correc
and that the reasons go beyond computational questions.

The distinction between proof theory and model theory is blurred, and there a
many borderline cases. I propose a characterisation of these concepts which I think
intuitive, but has some surprising cases. For example, according to it, Reiter's defau
logic (§5.4.1) is a semantics based definition, in spite of the apparent 'rules of inference

A proof theory is a system which yields proofs. Thus, given a presentation $?$ ar
a sentence ϕ, if $? \mathrel{|\!\sim} \phi$ then we should be able to find a positive demonstration of th
fact, namely a proof of ϕ from $?$. On the other hand, if $? \not\mathrel{|\!\sim} \phi$ then this is very hard
show in proof theory. We have to show of all the 'potential' proofs that none of the
are proofs of ϕ from $?$. As there are infinitely many 'potential' proofs this is har
Thus, in proof theory we may convincingly show that $? \mathrel{|\!\sim} \phi$, but we cannot easi
show that $? \not\mathrel{|\!\sim} \phi$.

Whereas proof theory deals in proofs, model theory deals in models. To show tha
$? \models \phi$, we need to show that each of the possibly infinite collection of models of $?$ is
model of ϕ. This is difficult. To show that $? \not\models \phi$, on the other hand, is much easie
We simply exhibit a single model of $?$ which is not a model of ϕ. In conclusion, t
primitive notion of proof theory is $\mathrel{|\!\sim}$, whereas the primitive notion in model theory
$\not\models$.

This idea also goes some way towards explaining why soundness proofs are in gener
much easier than completeness proofs. To show soundness, we show for all $?$ and ϕ tha
$? \mathrel{|\!\sim} \phi$ implies $? \models \phi$. Expressing this in terms of the 'primitive' notions, soundne
becomes:

$$\text{not}\Big(? \mathrel{|\!\sim} \phi \quad \text{and} \quad ? \not\models \phi\Big)$$

We might expect this to be relatively easy to do because we just show that both $? \mathrel{|\!\sim}$
and $? \not\models \phi$ cannot hold at once. Completeness proofs, on the other hand, involv
showing that (in terms of the primitives):

$$? \mathrel{|\!\sim} \phi \quad \text{or} \quad ? \not\models \phi$$

This is more difficult because it is a disjunction; which of $? \mathrel{|\!\sim} \phi$ or $? \not\models \phi$ we sho
depends on the particular $?$ and ϕ.

If one accepts these characterisations of proof theory and model theory, one
led to the conclusion that all the usual default formalisms are model-theoretic; th
perhaps supporting Shoham's claim. For example, Reiter's 'default logic' cannot b
proof theoretic because it does not yield proofs. To show that $? \mathrel{|\!\sim} \phi$ in his syster
one has to show that all extensions of $?$ contain ϕ. These extensions are really model
so this fits squarely with our model theory characterisation, not the proof theory on
Thus, we can show that ϕ does not follow from $?$ by exhibiting one extension of
which doesn't have ϕ; but to show that it does follow is more difficult.

As Shoham points out, the reason that it is hard to get a proof theoretic accou
of default logics is because in any such logic there must be some kind of consisten
check before a default can be used. This may appear in a disguised form, for examp

in the form of the model orderings present in circumscription and in this thesis, but it is there nevertheless.

As far as OTPs are concerned, we may be able to go some way towards a proof theory before encountering the problems associated with this consistency check. Specifically, it is possible that the relation of natural consequence (definition 2.38) can be given a proof theory. Showing weak structural properties (proposition 2.44) is some way towards this, and one idea which I have not yet had time to explore is a connection between natural consequence and linear logic. For example, the natural consequence relation exemplified on page 39 does not identify logical-$\wedge$ and meet, nor $\vee$ and join, which the classical Lindenbaum algebra does. This means we get two 'conjunctions' and two 'disjunctions'. Distributivity rules seem to fail however; but the connections, if any, have yet to be established. A connection with linear logic would, of course, answer the question of proof theory for natural consequence.

Two people have suggested algorithms for the special case of linear propositional OTPs, namely Dov Gabbay and Pierre-Yves Schobbens (private communications). For reasons already discussed, such an algorithm necessarily involves a consistency check. The task of comparing these algorithms with each other and with the semantics of OTPs remains to be done.

7.3 Related work: 'the living database'

Dov Gabbay's 'living database' is an ambitious research programme whose ultimate aim is to incorporate all of the examples of practical reasoning mentioned in the introduction, and many more besides. A living database is a database—it represents some facet of the world—but also has built in to it its own behaviour under updates and revisions, changes of priorities between units of information, temporal changes, and so on. It has structure which encodes some dynamic aspects of the database as well as just facts about the domain in question. Any unit information in the database comes with some 'meta-information', such as:

- its *provenance*; this is perhaps the agent which asserts it, or its justification on terms of other sentences.
- the *time* at which it is true.
- some measure of its *reliability*.
- information to do with how it interacts with other sentences currently in the database or sentences which may appear in the database as a result of some update.

The list is potentially endless, and any particular piece of information can have any combination of these annexes.

It is obvious that the ordered theory presentations of this thesis are a move in this direction, in which the additional meta-information which accompanies each sentence is its location in the partial order. As described in other chapters, this information may represent provenance, time or reliability depending on whether one views the partial order as arising from the structure of a specification, a revision history, or the stipulated

interaction of defaults or evidence. Much remains to be done to make this truly livin however. For example, the revision strategy which is the subject of chapter 4 on works by giving the revising sentence maximum priority; we would like a more refine way of updating with sentences whose priority can be expressed in terms meaningf to the database. Also, continual revision in the way of chapter 4 yields rather string 'databases'; a truly living database is constantly reformatting itself as it sorts o conflicts and works through deductions—rather like human brains.

The living database programme embraces a range of particular theories of whi this thesis represents one with a model-theoretic flavour. Gabbay's own main exampl are databases expressed within a *labelled deductive system* (LDS) [21] which has a pro theoretic flavour. In LDS each sentence is explicitly paired with a label and each pro rule of the system has side conditions applying to the labels which determine wheth the rule can be applied or not; and if so, how its conclusion will be labelled. Take, f example, the familiar rules of $\rightarrow$ introduction and elimination in natural deduction:

$$\frac{\begin{array}{c}[\phi]\\ \vdots\\ \psi\end{array}}{\phi \rightarrow \psi} \qquad \frac{\phi \quad \phi \rightarrow \psi}{\psi}.$$

The first rule says that if ψ can be deduced from ϕ then $\phi \rightarrow \psi$ can be deduce and moreover the conclusion $\phi \rightarrow \psi$ doesn't depend on ϕ, which can therefore b 'discharged' (as represented by the square brackets). The second rule says that fro ϕ and $\phi \rightarrow \psi$ one may deduce ψ. Gabbay gives an example of LDS in which the lab of a sentence is a set of nodes upon which it depends; for example, it may be a set sentences in another theory. The rules become:

$$\frac{\begin{array}{c}[\phi_a]\\ \vdots\\ \psi_b\end{array}}{(\phi \rightarrow \psi)_{b-a}}\ a \subseteq b \qquad \frac{\phi_a \quad (\phi \rightarrow \psi)_b}{\psi_{a \cup b}}.$$

Thus, $\rightarrow$-elimination accumulates dependencies; ψ is dependent on anything that ϕ $\phi \rightarrow \psi$ was. But this is not true for $\rightarrow$-introduction, for $\phi \rightarrow \psi$ does not depend o things on which ϕ depended. The side-condition $a \subseteq b$ must hold for the rule to b applicable.

Other rules may combine the labels in different ways. In this example the labe were just unstructured sets, but more generally they may have a complex algebra their own. Indeed, in many examples the labels themselves form a logic; so one ca consider what logic arises from, say, classical logic with labels from intuitionistic logi For more details, see Gabbay's forthcoming book [21].

7.4 Recap and final remarks

This thesis is about the framework of ordered theory presentations as a means unifying many aspects of practical reasoning, in artificial intelligence and in softwa engineering.

As we have mentioned in the proceeding sections, the work of this thesis is on going. I hope that improvements to the OTP definitions may be obtained by connecting with the framework of institutions, in such a way that much of the theory of OTPs can remain in place. To this end, I have emphasised where appropriate the modularity of these definitions; in particular, properties of the definition of $\sqsubseteq^{\Gamma}$ in terms of $\sqsubseteq_{\phi}$ do not depend on the definition of $\sqsubseteq_{\phi}$ except insofar as it is required to satisfy assumption 2.16.

Notwithstanding this further work, I hope that OTPs as they stand are seen as a direct way of linking at least the topics of belief revision and default reasoning. I have shown that they have good properties in the terms of those topics. I believe that they provide links with other aspects of practical reasoning, such as verisimilitude and prioritised evidence, as has been indicated.

Appendix A

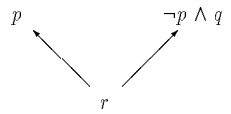
A Miranda program for propositional OTPs

The Miranda code for ordered theory presentations in propositional logic is close
based on the mathematical definitions given in chapter 2. One of the virtues of Miranc
is that this is possible.

A formula is defined to be either an atom P, Q, R,..., or $\top$, or $\bot$, or it is t
negation, conjunction, disjunction, implication or bi-implication of other formulas.

```
formula ::= P | Q | R | S |
            TOP | BOT |
            N formula |
            formula $A formula | formula $O formula |
            formula $I formula | formula $J formula
```

First we specify ? by means of a set of `points`, and a set of pairs `pt_ord` used
generate the ordering. The function `sent` maps `points` to sentences. For example,
specify the OTP specifies



we define

```
points = [1,2,3]
pt_ord = [(1,2),(1,3)]
sent 1 = R
sent 2 = P
sent 3 = (N P) $A Q
```

Given such an OTP, we work out the names of the atoms actually used.

```
atoms = sort (mkset (used (map sent points)))
        where
        used ((x $I y):t) = used (x:[]) ++ used (y:t)
```

```
used ((x $J y):t) = used (x:[]) ++ used (y:t)
used ((x $A y):t) = used (x:[]) ++ used (y:t)
used ((x $O y):t) = used (x:[]) ++ used (y:t)
used ((N x):t) = used (x:t)
used (x:t) = [x] ++ used t
used [] = []
```

Now generate the set of interpretations of the language used. An interpretation is a map from the atoms to $\{\mathbf{t}, \mathbf{f}\}$; we represent them by sequences of 0's and 1's.

```
interps = mods (#atoms)
          where
          mods n
          = [('0':m)|m<-p]++[('1':m)|m<-p], if n>0
          = [[]], otherwise
             where p=mods(n-1)
```

`leqX` is the ordering on `points`. It is the reflexive transitive closure of `pt_ord` viewed as a relation.

```
leqX x y
    = x=y \/ or [leqX z y | z<-points; member pt_ord (x,z)]
```

Now we define a function `sat` which takes an interpretation and a sentence and evaluates whether the sentence is satisfied or not in the interpretation.

```
sat m (N s) = ~sat m s
sat m (s1 $A s2) = sat m s1 & sat m s2
sat m (s1 $O s2) = sat m s1 \/ sat m s2
sat m (s1 $I s2) = ~sat m s1 \/ sat m s2
sat m (s1 $J s2) = sat m (s1 $I s2) & sat m (s2 $I s1)
sat m TOP = True
sat m BOT = False
sat m p = m!(idx p atoms)='1'
          where
          idx x (x:y) = 0
          idx x [] = error"can't idx empty list"
          idx x (y:z) = 1+idx x z
```

The models of a sentence are the interpretations which satisfy it. (This kind of definition makes one glad one is using Miranda!)

```
models sent = [m | m<-interps; sat m sent]
```

We represent formulas by the sets of their models. Therefore, the set of formulas is the power-set of the set of interpretations.


```
formulas = powerset interps
           where
           powerset [] = [[]]
           powerset (x:y) = (map (f x) (powerset y)) ++ (powerset y)
                            where f a b = (a:b)
```

The positive monotonicities of a sentence are the atoms with the property that their extension is increased in a model of the sentence, the result is also a model of th sentence. Negative monotonicities are defined similarly.

```
monoP phi = [p | p<-atoms; subset (map (inc p) phi) phi]
            where inc p m = subst '1' (idx p atoms) m
monoN phi = [p | p<-atoms; subset (map (dec p) phi) phi]
            where dec p m = subst '0' (idx p atoms) m
```

The natural consequences of a sentence are the consequences which preserve th monotonicities.

```
natcons phi = [psi | psi<-formulas; subset phi psi;
                 subset (monoP phi) (monoP psi);
                 subset (monoN phi) (monoN psi)]
```

We have $M \sqsubseteq_\phi N$ if for all ψ such that $\phi \models \psi$, $M \not\Vdash \psi$ or $N \Vdash \psi$.

```
leq phi m n
= and [~member psi m \/ member psi n | psi<-natcons phi]
```

For convenience, we define $\sqsubseteq_x$ and $\sqsubset_x$ too.

```
lep x m n = leq (models(sent x)) m n
ltp x m n = lep x m n & ~lep x n m
```

$M \sqsubseteq^\Gamma N$ if any point x which has the misfortune of having the property tha $M \not\sqsubseteq_x N$ is at least good in that there is a $y \leqslant x$ with $M \sqsubset_y N$.

```
leG m n = and (map good [x | x<-points; ~(lep x m n)])
          where good x = or [ltp y m n | y<-points; leqX y x]
```

Also, $M \sqsubset^\Gamma N$ if $M \sqsubseteq^\Gamma N$ and $N \not\sqsubseteq^\Gamma M$.

```
ltG m n = leG m n & ~leG n m
```

The maximal models are those which have nothing above them.

```
maxmods = [m | m<-interps; ~or [ltG m n | n<-interps]]
```

We also used `subset` (it checks whether its first argument is a subset of its second and `subst` (which substitutes a token in a list at a specified position).

```
subset [] l = True
subset (x:y) l = member l x & subset y l

subst tok 0 (h:t) = tok:t
subst tok n (h:t) = h:subst tok (n-1) t
subst tok n [] = error"string too short in function subst"
```

This code is sufficient to compute the models of a propositional ordered theory presentation. I have written other functions which display orderings among interpretations and sentences, but it is not reproduced here. It is surprising that so little code is needed (hardly more than a page, without the comments).

Using these definitions and the example OTP given, `maxmods` evaluates to `["011","111"]`, which is $q \wedge r$ (example 1.7).

Appendix B

Theory comparison diagrams

For a variety of theories T over the language $\{p, q\}$, we give the ordering $\leqslant_T$ whic
orders other theories in the language according to their closeness to T. See §7.1 f
details of the definition.

In each diagram, the formula ϕ appears as an abbreviation for the theory $\mathsf{Cn}(\{\phi\}$

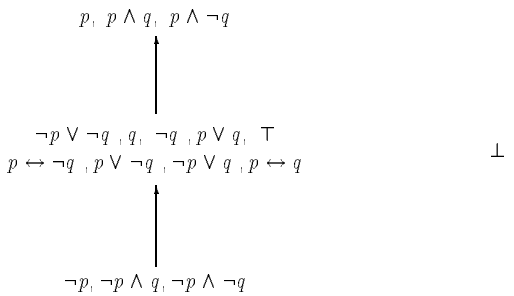


Figure B.1: The ordering for $\mathsf{Cn}(\{p\})$

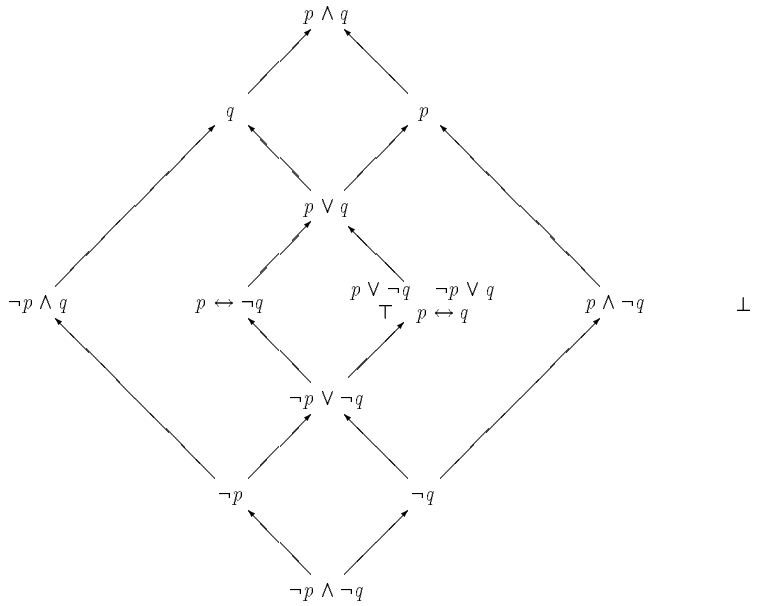


Figure B.2: The ordering for $\mathsf{Cn}(\{p, q\})$

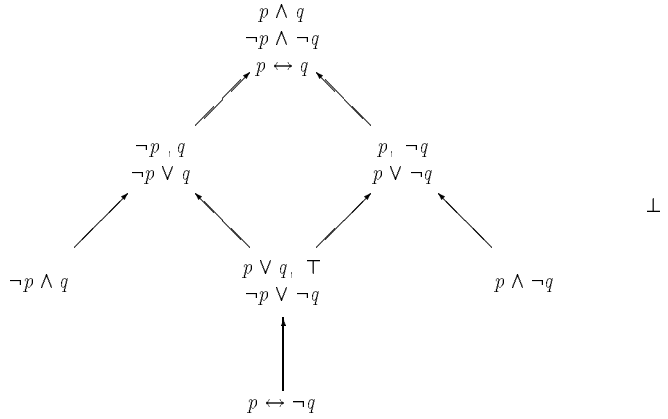


Figure B.3: The ordering for $\mathsf{Cn}(\{p \leftrightarrow q\})$

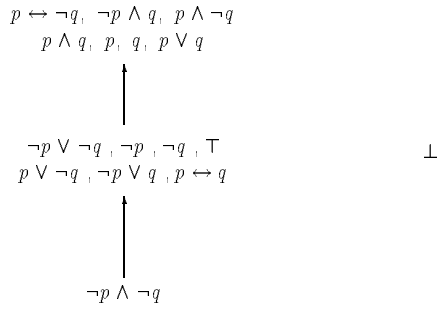


Figure B.4: The ordering for $\mathsf{Cn}(\{p \vee q\})$

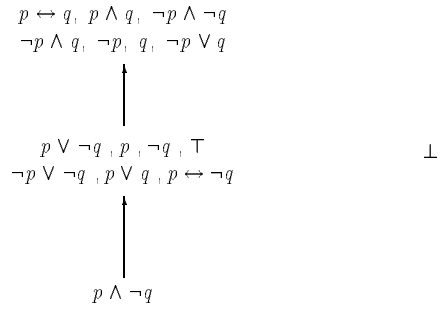


Figure B.5: The ordering for $\mathsf{Cn}(\{p \rightarrow q\})$

Bibliography

[1] Artificial intelligence. Special Issue on Non-Monotonic Logic, volume 13, 1980.

[2] A. R. Anderson and N. D. Belnap. *Entailment*, volume 1. Princeton University Press, 1975.

[3] A. B. Baker. Nonmonotonic reasoning in the framework of situation calculus. *Artificial Intelligence*, 49:5–23, 1991.

[4] Philippe Besnard. The preferential-models approach to non-monotonic logics. In P. Smets, A. Mamdani, D. Dubois, and H. Prade, editors, *Non-standard Logics for Automated Reasoning*. Academic Press, 1988.

[5] S. Brass and U. W. Lipeck. Semantics of inheritance in logical object specifications. In C. Delobel, M. Kifer, and Y. Masunaga, editors, *2nd Int. Conf. on Deductive and Object-Oriented Databases (DOOD'91)*, pages 411–430. Lecture Notes in Computer Science 566, Springer-Verlag, 1991.

[6] S. Brass, M. Ryan, and U. Lipeck. Hierarchical defaults in specification. To appear, 1992?

[7] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *Proc. International Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1043–1048. Morgan Kaufmann, 1989.

[8] A. L. Brown and Y. Shoham. New results on semantical non-monotonic reasoning. In M. Reinfrank, J. de Kleer, and M. L. Ginsberg, editors, *Non-monotonic Reasoning*. Lecture Notes in Artificial Intelligence 346, Springer-Verlag, 1988.

[9] R. Burstall and J. Goguen. Putting theories together to make specifications. In *Proc. Fifth International Joint Conference on Artificial Intelligence*, pages 1045–1058. Morgan Kaufmann, 1977.

[10] M. R. B. Clarke and D. M. Gabbay. An intuitionistic basis for non-monotonic logic. In P. Smets, A. Mamdani, D. Dubois, and H. Prade, editors, *Non-standard Logics for Automated Reasoning*. Academic Press, 1988.

[11] F. Brown (editor). *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop*. Morgan Kaufmann, Los Altos, CA, 1987.

[12] T. Kuipers (editor). *What is Closer-to-the-Truth*. Rodopi, Amsterdam, 1987.

[13] D. Etherington and R. Reiter. On inheritance hierarchies with exceptions. I *Proc. Third National Conference on Artificial Intelligence*, pages 104–108, 1983

[14] R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in database In *Proc. 2nd ACM SIGACT-SIGMOD Symp. on Principles of Database System* pages 352–365, 1983.

[15] J. Fiadeiro and T. Maibaum. Describing, structuring and implementing object In *Proc. REX Workshop on Foundations of Object-Oriented Languages*. Springe Verlag, 1991.

[16] J. Fiadeiro and T. Maibaum. Temporal reasoning over deontic specification *Journal of Logic and Computation*, 1(3):357–395, 1991.

[17] J. Fiadeiro and T. Maibaum. Towards object calculi. Technical report, Deprtme of Computing, Imperial College, London, 1992.

[18] J. Fiadeiro and A. Sernadas. Structuring theories on consequence. In D. Sanel and A. Tarlecki, editors, *Recent Trends in Data Type Specification, LNCS 33* Springer Verlag, 1988.

[19] A. Finkelstein. Reviewing and correcting specifications. In *Proc. Computers a Writing IV*, pages 219–237. Kluwer, 1991.

[20] A. Fuhrmann. Theory contraction through base contraction. *Journal of Phil sophical Logic*, 20:175–203, 1991.

[21] D. M. Gabbay. Labelled deductive systems. Manuscript in preparation, 1991.

[22] D. M. Gabbay. Theoretical foundations for non-monotonic reasoning. part Structured non-monotonic theories. In *Proc. Third Scandanavian Conference Artificial Intelligence (SCAI'91)*, 1991.

[23] P. Gärdenfors. *Knowledge in Flux: Modelling the Dynamics of Epistemic State* MIT Press, 1988.

[24] M. R. Genesereth and N. J. Nilson. *Logical Foundations of Artificial Intelligenc* Morgan Kaufmann, Los Altos, CA, 1987.

[25] M. Ginsberg. Introduction. In M. Ginsberg, editor, *Readings in Non-monoton Logic*. Morgan Kaufmann, 1988.

[26] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50, 1987.

[27] J. A. Goguen and R. M. Burstall. Introducing institutions. In E. Clarke a D. Kozen, editors, *Proc. Workshop on Logics of Programming*. Lecture Notes Computer Science 164, Springer-Verlag, 1984.

[28] J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for co puter science. Manuscript, 1985.

[29] R. Goldblatt. *Logics of Time and Computation*. CSLI Lecture Notes, 1987.

[30] Ian Hacking. What is logic. *Journal of Philosophy*, 76:285–318, 1979.

[31] A. G. Hamilton. *Logic for Mathematicians*. Cambridge University Press, 1978.

[32] S. Hanks and D. McDermott. Default reasoning, non-monotonic logics and the frame problem. In *Proc. Fifth National Conference on Artificial Intelligence (AAAI)*, pages 328–333, 1986.

[33] S. O. Hansson. *Belief Base Dynamics*. PhD thesis, Department of Philosophy, Uppsala University, 1991.

[34] S. O. Hansson. From logical atoms to basic beliefs. Submitted for publication, 1992.

[35] A. J. I. Jones and M.J. Sergot. On the role of deontic logic in the characterisation of normative systems. In *First International Conference on Deontic Logic in Computer Science*, 1991.

[36] H. Kautz. The logic of persistence. In *Proc. Fifth National Conference on Artificial Intelligence*, pages 401–405, 1986.

[37] S. Khosla and T. S. E. Maibaum. The prescription and description of state based systems. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, *Temporal Logic in Specification*. Lecture Notes in Computer Science 398, Springer-Verlag, 1989.

[38] S. Kraus, D. Lehmann, and M. Magidor. Non-monotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[39] T. S. Kuhn. *The Structure of Scientific Revolutions*. Univeristy of Chicago Press, 1970.

[40] E. Laenens and D. Vermeir. A fixpoint semantics for ordered logic. *Journal of Logic and Computation*, 1(2):159–185, 1990.

[41] M. M. Lehman. Software engineering, the software process and their support. *Software Engineering Journal*, 6(5):243–258, 1991.

[42] D. Lehmann. What does a conditional knowledge base entail? In *Proc. First International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*. Morgan Kaufmann, 1989. Toronto.

[43] V. Lifschitz. Computing circumscription. In *Ninth International Joint Conference on Artificial Intelligence*, pages 121–127, 1985.

[44] V. Lifschitz. Benchmark problems for formal non-monotonic reasoning, version 2.00. In M. Reinfrank, J. de Kleer, and M. L. Ginsberg, editors, *Non-monotonic Reasoning*. Lecture Notes in Artificial Intelligence 346, Springer-Verlag, 1988.

[45] S. Lindström and W. Rabinowicz. Epistemic entrenchment with incomparabilities and relational belief revision. In A. Fuhrmann and M. Morreau, editors, *The Logic of Theory Change*. Lecture Notes in Artificial Intelligence 465, Springer Verlag, 1991.

[46] D. Makinson. General theory of cumulative inference. In M. Reinfrank, J. de Klee and M. L. Ginsberg, editors, *Non-monotonic Reasoning*. Lecture Notes in Artifici Intelligence 346, Springer-Verlag, 1988.

[47] D. Makinson. General patterns in non-monotonic reasoning. In D. Gabbay, C. Ho ger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence*. Oxfo University Press, 1992. Forthcoming.

[48] D. Makinson and P. Gärdenfors. Relations between the logic of theory change ar non-monotonic logic. To appear.

[49] J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artifici Intelligence*, 13:27–39, 1980.

[50] J. McCarthy. Applications of circumscription to formalising common-sense know edge. *Artificial Intelligence*, 28:89–116, 1986.

[51] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*. Lectu Notes in Artificial Intelligence 422, Springer Verlag, 1990.

[52] D. L. Parnas and P. C. Clements. A rational design process: How and why fake it. In *IEEE Transactions on Software Engineering*, volume 2, pages 251–25 1986.

[53] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:2 47, 1988.

[54] K. R. Popper. *Conjectures and Refutations*. Routledge and Kegan Paul, Londo 1963.

[55] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[56] R. Reiter. Non-monotonic reasoning. *Annual Reviews of Computer Science*, 198

[57] H. Rott. Preferential belief change using generalised epistemic entrenchment. Ko stanzer Berichte zur Logik und Wissenschaftstheorie 15.

[58] H. Rott. Two methods of constructing contractions and revisions of knowled systems. *Journal of Philosophical Logic*, 20:149–173, 1991.

[59] M. D. Ryan. Belief revision and ordered theory presentations. In P. Dekker ar M. Stokhof, editors, *Proc. Eighth Amsterdam Colloquium on Logic*, 1991. appear.

[60] M. D. Ryan. Defaults and normativity in specifications. In J. J. Ch. Mey and R. Wieringa, editors, *Proc 1st International Conference in Deontic Logic Computer Science (DEON'91)*, 1991. To appear.

[61] M. D. Ryan. Defaults and revision in structured theories. In *IEEE Symposium Logic in Computer Science (LICS)*, pages 362–373, 1991.

[62] M. D. Ryan, J. Fiadeiro, and T. Maibaum. Sharing actions and attributes in modal action logic. In T. Ito and A. Meyer, editors, *Theoretical Aspects of Computer Software*, pages 569–593. Lecture Notes in Computer Science 526, Springer Verlag, 1991.

[63] M. D. Ryan and M. R. Sadler. Valuation systems and consequence relations. In D. Gabbay S. Abramsky and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 1. Oxford University Press, 1992. Forthcoming.

[64] R. C. Schank and R. P. Abelson. *Scripts, Plans, Goals and Understanding*. Erlbaum, Hillsdale, N.J., 1977.

[65] K. Schlechta. Some results on theory revision. In A. Fuhrmann and M. Morreau, editors, *The Logic of Theory Change*. Lecture Notes in Artificial Intelligence 465, Springer Verlag, 1991.

[66] G. Schurz and P. Weingertner. Verisimilitude defined by relevant consequence elements. In T. Kuipers, editor, *What is Closer-to-the-Truth*, pages 47–78. Rodopi, Amsterdam, 1987.

[67] Y. Shoham. A semantical approach to nonmonotonic logics. In *Proc. 10th International Conf. on Artificial Intelligence (IJCAI)*, pages 388–392. Morgan Kaufmann, 1987.

[68] Y. Shoham. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*. MIT Press, 1988.

[69] D. Touretzky. Implicit ordering of defaults in inheritance systems. In *Proc. Fifth National Conference on Artificial Intelligence*, pages 332–325, 1984.

[70] J. van Benthem. Partiality and non-monotonicity in classical logic. Technical report, CSLI, 1984.

[71] J. van Benthem. *A Manual of Intensional Logic*. CSLI Lecture Notes, 1985.

[72] J. van Benthem. Verisimilitude and conditionals. In T. Kuipers, editor, *What is Closer-to-the-Truth*, pages 103–128. Rodopi, Amsterdam, 1987.

[73] J. van Benthem and K. Doets. Higher order logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 1. Dordrecht: D. Reidel, 1983.

[74] Frank Veltman. Defaults in update semantics I. In Hans Kamp, editor, *Conditionals, Defaults and Belief Revision*, pages 28–64. Center for Cognitive Science, Edinburgh, 1990. DYANA deliverable R2.5A.

[75] D. Vermeir, P. Geerts, and D. Nute. A logic for defeasible perspectives. In *Proc. Tübingen Workshop on Semantic Networks and Non-monotonic Reasoning*, volume 1, 1989.