Technical Report

UCAM-CL-TR-246
ISSN 1476-2986

Number 246

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

# A relevance-based utterance processing system

Victor Poznański

February 1992

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted December 1990 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Girton College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

*http://www.cl.cam.ac.uk/techreports/*

ISSN 1476-2986

## Summary

This thesis presents a computational interpretation of Sperber and Wilson's Relevance Theory, based on the use of a non-monotonic logic supported by a reason maintenance system, and shows how the theory, when given a specific form in this way, can provide a unique and interesting account of discourse processing.

Relevance Theory is a radical theory of natural language pragmatics which attempts to explain the whole of human cognition using a single maxim: the Principle of Optimal Relevance. The theory is seen by its originators as a computationally more adequate alternative to Gricean pragmatics. Much as it claims to offer the advantage of a unified approach to utterance comprehension, Relevance Theory is hard to evaluate because Sperber and Wilson only provide vague, high-level descriptions of vital aspects of their theory. For example, the fundamental idea behind the whole theory is that, in trying to understand an utterance, we attempt to maximise significant new information obtained from the utterance whilst consuming as little cognitive effort as possible. However, Sperber and Wilson do not make the nature of information and effort sufficiently clear.

Relevance Theory is attractive as a general theory of human language communication and as a potential framework for computational language processing systems. The thesis seeks to clarify and flesh out the problem areas in order to develop a computational implementation which is used to evaluate the theory.

The early chapters examine and criticise the important aspects of the theory, emerging with a schema for an ideal Relevance-based system. Crystal, a computational implementation of an utterance processing system based on this schema, is then described. Crystal performs certain types of utterance disambiguation and reference resolution, and computes implicatures according to Relevance Theory.

An adequate reasoning apparatus is a key component of a Relevance-based discourse processor, so a suitable knowledge representation and inference engine are required. Various candidate formalisms are considered, and a knowledge representation and inference engine based on autoepistemic logic is found to be the most suitable. It is then shown how this representation can be used to meet particular discourse processing requirements, and how it provides a convenient interface to a separate abduction system that supplies non-demonstrative inferences according to Relevance Theory. Crystal's powers are illustrated with examples, and the thesis shows how the design not only implements the less precise areas of Sperber and Wilson's theory, but overcomes problems with the theory itself.

Crystal uses rather crude heuristics to model notions such as salience and degrees of belief. The thesis therefore presents a proposal and outline for a new kind of reason maintenance system that supports a non-monotonic logic whose formulæ are labelled with upper/lower probability ranges intended to represent strength of belief. This system should facilitate measurements of change in semantic information and shed some light on notions such as expected utility and salience.

The thesis concludes that the design and implementation of Crystal provide evidence that Relevance Theory, as a generic theory of language processing, is a viable alternative theory of pragmatics. It therefore merits a greater level of investigation than has been applied to it to date.

## Acknowledgements

I am grateful to Karen Sparck Jones for all the help and encouragement she gave me, and for her meticulous and discerning scrutiny of my thesis.

I am indebted to my father and Daniel Mack for their patient reading and rereading of several drafts of this dissertation, and to Chris Hadley for checking my first draft, for his artistic direction, for helping produce several of the diagrams in this text and for occasionally throwing his long-suffering pet, Dog, at me when our office became too cramped.

I would like to thank Julia Galliers, Ann Copestake and Versha Mukhi for reading and commenting on my thesis, for providing a listening ear, and for providing moral support.

Thanks also to Dick Crouch, Dave Frost, Diana Hutchison, Margaret Levitt, Bob Moore, Steve Pulman, Keith Van Rijsbergen, Graham Titmus and everyone else who gave me their advice and help or had to put up with me over the last few years.

Last, but by no means least, thanks to my parents for their ever-present support, encouragement and love.


---

Except where otherwise stated, this dissertation is the result of my own work and is not the outcome of any work done in collaboration.

I hereby state that this dissertation is not substantially the same as any I have submitted for a degree, diploma or other qualification at any other university. Further, no part of this dissertation has already been or is being concurrently submitted for any such degree, diploma or other qualification.

# Contents

# Chapter 1

# Introduction

Relevance is a theory of *radical pragmatics*: it claims that a single principle governs the whole of human language understanding.

In 1980, Dan Sperber and Deirdre Wilson (SW), the originators of Relevance Theory (RT), introduced their ideas to a group of colleagues at the Colloquium on Mutual Knowledge held at the University of Surrey. This is a sample of the exchanges that ensued, appearing in the consequent book, Mutual Knowledge (Smith, 1982).

From Gerald Gazdar and David Good:

> ...the notion of Relevance gives rise to a number of apparently intractable problems. (pp. 88)

> Not only does S&W's theory seem implausible when we consider the hearer's task, it also has unpalatable consequences for our conception of the speaker. (pp. 96)

From SW's reply:

> ...if Gazdar and Good had spent as much time...on raising and pursuing [ *more pertinent questions* ] as they did on formulating their straw man,...constructing objections to it, anticipating quite implausible replies to these objections...their paper might have been as valuable as it was provocative. (pp. 109)

From Terence Moore:

> ...nowhere in their paper is there any suggestion as to how expressions such as "amount of processing effort" or "measurement of relevance" are to be given empirical substance... (pp. 111)

> It is not at all clear that the principle of relevance is yet sufficiently well delineated to throw light on the subtle and still largely mysterious processes involved in the comprehension of language. (pp. 112)

From Yorick Wilks:

> Reading all this was like being whisked back on a time machine to an easier, more relaxed and agreeable age...SW really must flip through a few years of issues of *Cognitive Science*...they will find a host of articles on these issues, all far beyond the level of "my first thoughts on psychological or computational access to relevant information". (pp. 116)

From SW's reply to Wilks:

> Wilks' comments are an excellent illustration of the sort of obsessive misinterpretation mentioned in our paper...It does not seem to have occurred to him that to achieve these implications, he has had to engage in massive misconstrual of our text. (pp. 118)
>
> ...by proposing an integrated, explanatory theory of utterance-interpretation, we are doing what some AI specialists have tried to do, and failed. (pp. 122)

Since its provocative introduction, RT has remained controversial, yet gained a substantial following in linguistic circles. Few linguistics conferences go by without several papers giving a Relevance-theoretic account of various language phenomena.

But what is it that has made the theory so controversial? What are the "apparently intractable" problems associated with RT? Can any of them be overcome? What relation does RT have with the field of Computational Linguistics, and what, if any, are its repercussions for AI researchers? These are the questions which I have addressed in the course of this thesis. More specifically:

- I have built a computer-based understanding system that calculates some of the pragmatic phenomena in the spirit of RT. This is a novel approach, which has required considerable rationalisation of the available material.
- A non-monotonic logic has been used as the basis of a knowledge representation language. This has made it possible to be precise about some of SW's more hazy notions. Additionally, the relationship and interaction of deductive, defeasible and non-demonstrative reasoning processes can be explicated.
- It is possible to use this system to give meaning to certain natural-language constructs which are best understood using the Relevance paradigm. To my knowledge, no other similar systems can handle these constructs in as elegant a manner.
- A new non-monotonic logic which supports degrees of belief will be proposed which can give SW's notions of "strength" some theoretical support. This logic has other useful properties that enhance the power of a reasoning system.
- RT will be related to some similar AI programs that tackle similar problem areas.

## 1.1 A Computational Approach To RT

One good test of any theory is to express it in such a way that it is possible to implement it on a computer. If the theory is poorly specified, ambiguous or inconsistent, it will not be possible to specify or build a complete computational model of it. In such an eventuality, it may be possible to alter or refine the original theory in order to make a computational model of it. Of course, such an implementation does not confirm or necessarily falsify the theory (Nagel, 1961), but it makes it possible to evaluate its relation and contribution to current computer-based comprehension systems.

In their book "Relevance" (Sperber and Wilson, 1986), SW produced an extended account of RT which they hoped would allay the doubts of their critics and provide it with a firm theoretical foundation. Using these ideas, I have built a simple computer-based story comprehension program called Crystal[1]. Unfortunately, the state-of-the-art

[1] To be precise, Crystal processes logical representations of fragments of childrens' stories.

in Computational Linguistics is not sufficiently advanced for a complete implementation of RT to be feasible. However, the building of Crystal has helped identify some of the critical problems of the theory, and in some cases, to solve them.

Thus, the aim of this research is not to justify RT as an alternative theory of linguistics. Rather, it is to investigate whether it is possible to provide a computational "instantiation" of the theory that can perform certain kinds of pragmatic processing, and if so, to evaluate its impact on current computer-based understanding systems and uncover its relation to them. Over the course of this thesis, I hope to convince the reader that Crystal is indeed an implementation of a Relevance-based system that draws on current AI technology in order to implement some of SW's ideas. Furthermore, it does have similarities to some prevailing language processing systems, but has a fundamentally different and arguably better theoretical foundation.

In order to explain the design of Crystal, I first provide a critical account of RT, highlighting the essential problems of the theory. Next, I develop a schema for an ideal Relevance-based system from which the basic design of Crystal is extracted. At this point, a basic architecture for the system will have been developed. The remainder of the thesis will consider the various components of this architecture in detail. The interaction of the reasoning and context control mechanisms are described, followed by details of the knowledge representation and non-demonstrative reasoning components of the system. Next, Crystal is compared with related systems that tackle similar problems and explain the benefits of RT. Finally, I consider ways in which Crystal can be extended to increase its coverage of pragmatic phenomena.

During the remainder of this chapter, I first circumscribe Crystal's domain of coverage. Next, I provide a basic background to the field of pragmatics, from what I understand to be SW's viewpoint. I then explain the goals of Crystal.

## 1.2 Pragmatics

RT chiefly addresses the field of *pragmatics*, which, informally, is the study of language usage. Pragmatics is a relatively new discipline in the field of linguistics. As a consequence, there is considerable disagreement even as to its exact domain of coverage (Lyons, 1977). It is therefore worth examining the subject further, explaining which aspects of it I will cover. In section 1.3, I will explain the RT view of pragmatics.

A common view of pragmatics, expounded in Gazdar (1979), is that it covers all areas of meaning that are not truth-conditional or:

$$\text{Pragmatics} = \text{Meaning} - \text{Truth Conditions}$$

Truth-conditional meaning or *propositional content* is the information normally extracted by compositionally constructing a semantic representation of the sentence in a syntax-driven way (eg. Dowty et al. (1981)).

Gazdar's definition does not explain the extent to which the truth conditions cover meaning and therefore exactly which phenomena are covered by pragmatics. However, pragmatic phenomena are often associated with notions of context which will be further discussed in section 1.3.

In order to give the reader an idea of the kinds of phenomena that RT covers, I will enumerate some of the phenomena that usually fall under the domain of pragmatics

(Levinson, 1983)[2]:

- Referring Expressions
- Presupposition
- Implicature
- Speech acts
- Discourse Structure

Much as RT addresses all of these issues, Crystal only handles referring expressions, implicature and, to some extent, discourse structure.

### 1.2.1 Referring Expressions

These are the natural language expressions that can only be understood by assuming that they relate to some spatio-temporal point in the discourse or the world.

There are syntactic and semantic constraints that can be placed on referring expressions (Reinhart, 1983), as when "I" and "myself" co-refer in the same sentence. However, there are many expressions which are best considered pragmatically. These expressions include *deictic* expressions that refer to objects not mentioned in the discourse (eg. the present time) and expressions that refer to previous discourse objects. The latter may require common-sense reasoning to be successfully resolved. Consider:

Olga took the bucket to her car and washed it. (1)

Analyses of referring expressions that employ syntactic, semantic and discourse constraints usually claim that the bucket is made mentally prominent (*focussed*) in this sentence. Consequently, it is often assumed that the bucket is the referent of "it" in (1). It is the employing of our common-sense knowledge that people are unlikely to wash buckets under normal circumstances that helps us to understand this sentence.

I will only look at discourse references, especially those that involve common-sense reasoning.

### 1.2.2 Implicatures

According to Levinson (1983), implicatures are inferences derived by consideration of the non-truth-conditional attributes of a sentence. They should be distinguished from *entailments* which are the logical consequences of the propositional content alone. For example:

Some Poles have long, unpronounceable names. (2)

entails:

Some Poles have unpronounceable names. (3)

However, an implicature which is not entailed by (2) is:

[2]This is sometimes even used as a definition of pragmatics.

Not all Poles have long, unpronounceable names. (4)

Implicatures can be *conventional* or *conversational*. Conventional implicatures are associated with the forms from which they originate by some convention. Conversational implicatures are made by considering the semantic form of the sentence and the circumstances under which it was uttered.

### Conventional Implicature

There are no general principles that can derive conventional implicatures. They vary from language to language and are often associated with certain words. They are also independent of the context in which they were uttered and therefore not *defeasible* or cancelable. As an example,

Vlad seems quite affable, but he does have the unfortunate
habit of impaling people. (5)

In this sentence, "and" could be substituted for "but" without changing the propositional content of the sentence. "but", however, has the conventional implicature associated with it that gives an indication that what follows it contrasts with what precedes it. Similarly, words like "sir" conventionally imply that the addressee is of socially higher status.

### Conversational Implicature

Conversational implicatures, on the other hand, are context dependent and defeasible. They are *detachable*, since changing the syntactic representation of the sentence while maintaining the same semantic form should not change them. They are also *calculable*, meaning that there is a demonstrative argument that can be constructed to generate such an inference from general pragmatic principles. Conversational implicatures should also be *non-conventional*: they are not due to the conventional meaning of the expressions from which they were derived. Consider the following sentences which are syntactically different but semantically equivalent:

Mary slapped John. (6)

John was slapped by Mary. (7)

They both share the conversational implicature:

Mary did not kill John by slapping him. (8)

This can be cancelled, though:

Mary slapped John.
Unfortunately, since she was a mountain gorilla,
that killed him. (9)

Conversational implicatures can be based solely on the semantic representations of forms from which they were derived as is (8) in which case they are *general*, or they can be based on the particular context, in which case they are *particular*. Particular implicatures are less easy to compute. There may be many different alternatives in a given situation:

Would you like some more newt eyes? (10)

My cold is getting better. (11)

The answer (11) might be implicating an affirmative answer or a negative answer. Given, say, that the hearer believes that more newt eyes will hasten their recuperation, the answer may be "yes". If the hearer believes that since the cold is getting better and there is no further need for any newt, the answer implicated may be "no". There is no "correct" answer to question (10); it depends purely on the circumstances.

### 1.2.3 Discourse Structure

This attempts to explain how sentences are organised into larger units. Levinson (1983) identifies two approaches:

1. Discourse Analysis.

2. Conversation Analysis.

*Discourse Analysis* attempts to use established linguistic devices for analysing texts, often in the form of a grammar of speech acts. *Conversation analysis* is a more informal, psychological approach that tries to explain basic *moves* in a conversation by observation and questioning individuals as to why they responded in a particular manner.

I will only consider the organisation of very short discourse segments with simple, linear structure, so what I have to say about discourse structure is limited.

## 1.3 The Role of Context in Communication

We can now move on to the RT view of pragmatics. I should like to stress at this point that the following is not intended as a comprehensive account of the areas of discussion, but more as a background in which to understand my elaboration of RT in chapter 2.

It is apparent that most pragmatic phenomena are irrevocably associated with some notion of *context*. This covers the temporal and spatial co-ordinates of the objects explicitly or implicitly involved in a discourse as well as the beliefs, desires and intentions of the participants. A common view of the context is as a means for establishing the truth-conditional meaning of an utterance. However, by the end of this section, it will be seen that RT does not adhere to this view; the interaction between utterances and contexts is a bidirectional process.

A unidirectional view of context regards it as a component that is necessary to establish the truth-conditional meaning of *deictic* and *indexical* words. Typical examples are pronouns such as "I" and "you", temporal adverbials such as "yesterday" and demonstratives such as "here". The context is also used to decide which pragmatic phenomena are occurring, for example distinguishing between direct and indirect speech acts. Many sentences are under-specified or elliptical, their completed forms being recoverable from the context. The defeasible nature of presuppositions and implicatures is also due to cancellation by its elements.

One problem with this view of context is that it does not help shed any light on the semantics/pragmatics division. Researchers such as Montague have tried to extend semantics to include a context of appropriate co-ordinates. The term *utterance* is used to refer to a sentence uttered in its context[3]. Meaning is then a function $\mathcal{M}$:

$$\mathcal{M} : \text{utterance} \rightarrow \text{proposition}$$

This is known as *indexical semantics.* This technique cannot disambiguate ambiguous candidate referents. For example:

Shut the door. (12)

will not distinguish between any of the candidate doors. In order to make this distinction, the psychological phenomenon of *salience* or cognitive accessibility can be useful (Lewis, 1972). However, this seems to contradict the spirit of formal semantics, which is intended to be a competence model, whereas salience is often regarded as an aspect of performance. Thus, it appears that operations that require the use of context should be considered to be pragmatic.

As further evidence, it has been observed that interactions between sentences and contexts are not unidirectional. Not only can a context be used to disambiguate sentences, but sentences may contribute to the determination of the context. This may be by altering the salience of objects in the context or by adding new non-truth-conditional information to the context. An example due to Katz is the use of "dog" or "doggie". Both have the same semantic meaning, but the latter is usually used in the presence of children. A better definition of $\mathcal{M}$ might be:

$$\mathcal{M} : \text{utterance} \rightarrow \text{context}$$

By this definition, valid sentences are those that successfully modify a given context. The new context may well include the propositional content of the utterance.

This brings us to the RT view of pragmatics. Typically, sentences uttered by people are fragmentary representations of their propositional content. The role of pragmatics is to enhance the context to such a point that it is possible to flesh out the utterance enough to obtain its meaning. This will be explained in more detail in chapter 2.

## 1.4 Code and Inference Approaches to Meaning

SW place great emphasis on the fact that there may be non-determinacy in communication; there may not be an exact proposition that someone is trying to communicate.

Thus, SW try to avoid what they call *code-based* approaches to communication. These approaches are based on the information theory of Shannon and Weaver (1949). Here, information from a *source* is encoded, transmitted across a *channel*, and decoded at the other other end known as the *destination.* In the case of language, the source and destination are the communicator and perceiver, and the channel is the communication medium, for example air, paper or telephone lines. In order for communication to be successful using this approach, the communicator and the perceiver will need to share common coding/decoding rules, which for language are known as *mutual knowledge.*

[3]Furthermore, an utterance can be *any* information used for communication, like writing, sign language, Morse code etc. When appropriate, I will also the use the terms *speaker* and *hearer* to mean the source of an utterance and its perceiver, respectively.

On the other hand, inferential approaches do not assume that communication can be perfect. Much as some information is shared, still other information is not. Understanding involves constructing hypotheses or *non-demonstrative inferences* about the meaning of an utterance and using deductive inference to test these hypotheses. People may opt for the wrong meaning, in which case the communicator is misunderstood. It may even be the case that different people legitimately understand the same utterance in different ways. The role of pragmatics here is to provide general rules for selecting between possible interpretations.

### 1.4.1 Problems with Code-Based Approaches

Aside from the fact that SW believe utterances may be inherently non-deterministic, they identify several problems with the notion of mutual knowledge:

#### Implausibility of Mutual Knowledge

It turns out that it is not sufficient for communicating parties merely to hold the same first-order beliefs concerning the germane discourse objects, they must each be totally aware that the other recognises this fact, as illustrated by Clark and Marshall (1981); communication can fail when both the communicator, $C$, and the speaker, $S$, both believe proposition, $P$, but do not realise that the other recognises this.

Thus, $S$ and $H$ mutually believe $P$[4] iff:

1. S believes P
2. H believes P
3. S believes H believes P
4. H believes S believes P
5. S believes H believes S believes P
6. H believes S believes H believes P

$\vdots$

*ad infinitum*

SW claim that the establishment of mutual belief is impossible because it would require that the communicators consciously enumerate the necessarily infinite number of beliefs.

Clark and Marshall argue that we use induction procedures based on co-presence of various factors to establish mutual knowledge.In order of reliability, these factors are:

1. Physical co-presence: the communicators can both perceive the same object and are aware of this (due to the proximity of the object and the communicators).
2. Linguistic co-presence: the communicators are speaking to each other, or both in close proximity when some utterance occurs.
3. Community membership: the information is known to be shared by some culture or sub-group of a culture of which the communicators are members.

Since these procedures are not totally reliable, mutual knowledge cannot be guaranteed, thus deterministic communication cannot be assured.

[4]Mutual beliefs are often used synonymously with *mutual knowledge*.

### Mutual Knowledge is Insufficient and Unnecessary

Mutual knowledge does not necessarily provide enough information to resolve references and disambiguate sentences. It is possible that several candidate referents are mutually known by some individuals. Consider:

Bring me the aardvark, please. (13)

There may be several candidate aardvarks, all of which are mutually known. The notion of mutual belief is not sufficient to distinguish them.

Consider $S$ uttering the following to $H$:

Vlad has invited us to his barbecue. (14)

This utterance may have several meanings, among which could be:

- $S$ wants to see the reaction of $H$ to (14).
- $S$ wants to inform $H$ that they have been invited.
- $S$ wants to know if $H$ wants to go.
- $S$ wants them both to go and to prevent $H$ making any alternative arrangements.

The meaning could even be a conjunction of some combination of the above.

These statements increase in specificity. Using mutual knowledge, it may be possible for $H$ to work out which of these was intended. However, even in the case where exactly what was intended cannot be worked out, $H$ can respond to the first of the alternatives and successfully communicate with $S$. Thus, mutual knowledge is not necessary.

## 1.5 The Co-operative Principle

The principal and most influential theory of pragmatics to date is that of Grice (1975). This theory accounts for utterance meaning by assuming that the parties involved are co-operating and adhering to some fundamental rules known as the *conversational maxims*. These are partitioned into four categories as follows:

*Maxims of Quantity*

1. Say no less than is required.
2. Say no more than is required.

*Maxims of Quality*

1. Do not say what you believe to be false.
2. Do not say that for which you lack adequate evidence.

*Maxim of Relevance*

1. Make only relevant contributions.

*Maxims of Manner*

1. Avoid obscurity.
2. Avoid ambiguity.
3. Be brief.
4. Be orderly.

Grice claimed that these jointly form a *co-operative principle* which states:

> Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged. (Grice, 1975)

Is the co-operative principle a code-based or inference-based approach? The answer is either. Some advocates of the maxims appeal to the use of mutual knowledge to assist the understanding process, others do not. However, SW claim that Grice's early thoughts about meaning[5] suggest an inferential approach becauses understanding involves guessing the communicators' intentions (Grice, 1957).

### 1.5.1 Gricean Implicature

Grice first used the notion of implicature in order to explain communication according to his co-operative principle. The maxims constrain what counts as a valid implicature by disallowing those that do not adhere to the rules.

At the same time, implicatures do not just constitute what directly follows the fact that an utterance conforms to the maxims, they include the assumptions that need to be made in order that the utterance itself conforms to the maxims. Consider:

Q: Can you pick my brother up from the station? (15)

A: I'm going on a barrel-making course today. (16)

This apparently violates the relevance and quantity maxims. However, an argument can be constructed along the lines: "I am going on a barrel-making course today. The course may well be given at the time your brother is arriving at the station. If he is coming then, I cannot pick him up. If not, I will pick him up". This argument includes several implicatures which ensure that (16) obeys the relevant maxims. An example of an implicature that maintains relevance is:

If your brother arrives when I am on the course, I cannot pick him up. (17)

[5]This was what Grice called non-natural meaning (meaning-nn) to distinguish it from the natural language use of "$A$ meant $B$" which approximates to "$A$ caused/resulted in $B$".

### 1.5.2 Flouting

Grice also introduced the concept of ostensive *flouting* of the maxims. This is the deliberate disobeying of the maxims for the purposes of communication. Consider:

Could you speak a bit louder, please? (18)

In the context of someone shouting, this would violate the maxim of quality. But since this is obvious, it is assumed that the statement is ironical and that in order to conform to the maxim of relevance, the intention is quite the opposite. Typically, it is the tropes that involve the flouting of the maxims.

### 1.5.3 Calculability

As already mentioned, conversational implicatures have to be calculable; they must be derivable by some demonstrative argument. Grice claims that this amounts to conformance to the following schema:

1. The speaker S has said the proposition $p$.
2. There is no reason to believe that $S$ is not obeying the co-operative principle.
3. $S$ could not be being co-operative unless he believes $q$.
4. $S$ and the listener $H$ mutually believe that $q$ is required for co-operation.
5. $S$ has done nothing to suggest $\neg q$ to $H$.
6. $S$ therefore intends $H$ to believe $q$.
7. $p$ has implicated $q$ in this context.

As an example, considering utterance (16), $S$ in the schema would be speaker $A$ and $H$ in the schema would be $Q$. The proposition $p$ might be:

$A$ says $A$ is going on a barrel-making course
on the 24th of June, 1989. (19)

and $q$ might be:

If the brother of $Q$ arrives when $A$ is on the course, then
$A$ cannot pick him up. (20)

There is no explanation of how $q$ was obtained here. Clearly, the argument that included the implicature (20) is not the only argument that could be made to ensure that utterance (16) conforms to the co-operative principle. It is therefore difficult to conceive of an effective procedure that could compute $q$.

Additionally, mutual belief is being used in the process of communication; SW take this to be a degeneration towards a code approach to meaning. Their response is to provide an alternative framework in which to describe these phenomena.

## 1.6 What is Relevance Theory?

In a nutshell, RT assumes that a single maxim, the maxim of relevance, subsumes all of the other maxims. This cannot be flouted. Furthermore, its application is governed by a single principle, known as the *Principle of Relevance.*

Relevance also seeks to unite the code and inference approaches using a psychologically plausible model of language. In order to do this, the notion of mutual knowledge is weakened to *mutual manifestness* and a Relevance measure defined as a comparative quantity. SW also attempt to give a unified account of essential deductive and non-demonstrative inferences which they call *contextual effects.*

A more detailed description will be provided in chapter 2.

## 1.7 The Psychology of Meaning

SW claim that a theory of human communication needs to have psychological plausibility. Thus, they draw on the notion of the *modular mind* to explain how the meaning of an utterance is established.

### 1.7.1 The Modular Mind

Chomsky (1957) suggests that we understand language by building representations of the linguistic structures that we perceive using specialised mental organs. Grammar, for example, builds syntactic representations of the relationships between lexical tokens. These representations in themselves cannot be regarded propositionally, they can only be checked for well-formedness. Some part of the mind can be considered to be functionally devoted to building grammatical representations from tokens.

Fodor (1983) claims that Chomsky's notion of a specialised mental organ fits into a much more general scheme. He argues that at least some of the mind can be partitioned into three functional types: transducers and independent systems of two types, known as *input systems* and *central systems*. Transducers take data from the sensory organs and convert it into information suitable for the input processes. The input systems produce representations of *states of affairs* in the "real world" from their sense data. These representations are not formally true, although the perceiver believes that they are adequate representations of reality. Thus, Chomsky can be considered to be postulating a grammatical input module. Central systems integrate knowledge and perform inferences on information issued by the input systems.

The input systems are informationally encapsulated modules. Each module handles a unique representation. For example, the vision system handles only distal layouts or visual information. It is assumed that the input systems can convert their internal representations into a modality-neutral representation to be fed as input to the central processes.

The central systems are not informationally encapsulated. They take representations as input and integrate them, possibly drawing inferences from them. The meaning of these representations must therefore be common to all of them.

It is the task of the input systems to convert their local representations into common representations. This conversion may not be deductive or deterministic so the input processes might hypothesise several alternatives. The central processes only deal with the formal properties of these representations, i.e. they draw inferences from them solely on the basis of their syntactic properties.

### 1.7.2 The Semantics-Pragmatics Division

RT uses Fodor's notions to explain utterance interpretation in the following manner. The input processes deliver modality-neutral *logical forms*[6] to the central processes incrementally. These are uninstantiated propositions, possibly being fragmentary, vague, with ambiguous sense and reference. As processing progresses, the central processes provide limited feedback until a proposition is finally formed.

The exact division between semantics and pragmatics is a matter of debate. From a RT viewpoint, communication involves the following stages (Blakemore, 1987):

1. Conversion of utterances into logical forms by the input processes.
2. Instantiation and disambiguation of the logical form yielding its propositional content.
3. Assignment of truth-conditions to the propositions.

Blakemore points out that the assignment of truth-conditions should not be part of a theory of cognition because their very representation in our central processes would result in the propositional forms no longer representing states of affairs in the external world.

The first two items above represent her distinction between semantics and pragmatics, which amounts to a distinction between linguistic and non-linguistic knowledge. Hence, it is the second item which is the subject of this thesis and what I will refer to as pragmatics.

## 1.8 A Computational Account of Relevance

### 1.8.1 Goals of a Computational Account

Historically, computer-based language understanding programs have not been motivated by linguistic theories. As a consequence, a potentially rich source of extant knowledge has been denied to AI researchers. A parallel argument applies to linguists; it will become apparent that the Relevance theorists are equally as ignorant of the work that has been produced by AI researchers. In this thesis, I hope to bring together some of the current work that has been produced by the AI and linguistics disciplines. For linguists, the necessary concreteness of an implementation has uncovered problems and novelties hitherto unrealised. For AI researchers, there is the promise of a sound, theoretical approach.

### 1.8.2 What Should Crystal Do?

Ideally, Crystal should understand everything that humans could communicate. Unfortunately, this is a rather tall order given the current state of our knowledge in the appropriate disciplines and the somewhat limited scope of a Ph.D. project.

The more practical approach I have taken is to build Crystal so that it processes logical representations of fragments of children's stories using some of the ideas of RT. With this approach, Crystal:

- Disambiguates certain utterances.
- Resolves certain referring expressions.

[6] SW's usage of this term is not the common one. However, I adopt their usage of the term throughout this thesis.

- Produces certain inferences that would normally be classified under the umbrella of Gricean implicature.

A justifiable criticism of many computer understanding systems is that their approach is ad-hoc, and there is no underlying consistent theory. I have therefore taken great pains to ensure that the majority of the tools that I have used have well-defined semantics.

In the next two chapters, I will develop a design for an ideal Relevance-based language processing system. I will then present the design for Crystal, and show its relation to the ideal design. Given that this relationship holds, the above abilities of Crystal can be said to be based on RT.

### 1.8.3 An Example of Crystal's Processing

Consider the following three utterances, that might form a fragment of a discourse:

Nutkin felt hungry. (21)

He found a nut. (22)

Nutkin is a penguin. (23)

The reader may find it quite difficult to make sense of these utterances, yet Crystal will produce an interpretation as follows:

(21) generates the generalised conversational implicature that most dedicated childrens' story readers should draw: animals called Nutkin are squirrels. Being a squirrel, Nutkin likes to eat nuts. A guess is made that the pronominal reference "he" in (22) refers to Nutkin. There are two senses of "nut" that might have been intended in (22): the fruit and the object associated with bolts. Crystal assumes the first because it is coheres better with (21). In particular, Crystal generates the particularised conversational implicature that Nutkin may eat the nut to assuage his hunger. Now, (23) does not connect to (22) in the same way that (22) connects to (21). Its purpose is to counter our predictions. Nutkin is not a squirrel after all, he is a non-flying bird. Furthermore, Nutkin, being a penguin, eats only fish and therefore cannot eat the nut. Thus, the two conversational implicatures mentioned so far are cancelled. As is the case for many absorbing stories, they engage us because they are *not* always predictable[7].

Much as some of the above capabilities have been implemented in existing story-processing systems, Crystal differs from them in that it operates according to the unitary Principle of Relevance. The concepts that I have used to describe the pragmatic phenomena are, in fact, calculated using the alternative notions that are described in chapter 2.

## 1.9 Thesis Structure

The remainder of this thesis is as follows.

- Chapter 2 provides the essentials of RT and explains its controversial nature and some of its problems that affect a computational implementation.

[7] I should, perhaps, stress that I do not personally find the above discourse riveting.

- Chapter 3 sketches an ideal language processing system and then a design for a greatly restricted version, Crystal.
- Chapter 4 gives a detailed description and justification of Crystal's deductive inference component and its representation of the context.
- Chapter 5 considers the effect that RT has on knowledge representation. A suitable knowledge representation is built based on first-order logic.
- Chapter 6 considers how Relevance should be calculated and used.
- Chapter 7 explains how the deductive component of Crystal and its knowledge representation can be enhanced to support stereotypical and defeasible reasoning using *non-monotonic logic.* A novel use of the logic for natural language semantics will be demonstrated.
- Chapter 8 explains how some non-demonstrative inferences necessary for language understanding can be made by a straightforward enhancement to the inference engine that permits a simple kind of abduction. Since, at this point, Crystal will have been been fully described, it is related to some current computer language processing systems.
- Chapter 9 suggests ways of enhancing Crystal to cope with phenomena such as *degrees of belief* and utility by enhancing the non-monotonic logic described in chapter 7 to handle probability ranges.
- Chapter 10 is a conclusion. It begins by explaining the nature of the tests carried out with Crystal and then goes on to evaluate RT and its implications for natural-language processing.
- Appendix A describes Crystal and its user-interface.
- Appendix B gives a detailed example of some Crystal's workings and output.
- Appendix C gives a glossary of terms concerning Crystal's *reason maintenance system*, which supports its non-monotonic logic. This includes some implementation details.
- Appendix D provides a typical set of meta-rules used by Crystal's inference engine.
- Appendix E provides some short examples processed by Crystal in order to illustrate its capabilities.

# Chapter 2

# Relevance

Before I give an overview of this chapter, let me refresh the readers' memory about my goals. It is not my intention to evaluate RT with respect to other candidate pragmatic theories, but rather to check its internal consistency by building a computational account of it. The performance and potential abilities of an implementation can then be evaluated and compared with other *computational* systems to see if it is providing any useful or different insights. Thus, during the development of Crystal, I have tried to make as few alterations to RT as possible.

The aim of this chapter is to provide an overview of RT, placing special emphasis on its aspects that are significant when considering a computational implementation. In the course of this chapter, it will be seen that various aspects of RT are problematic and some vital areas are underspecified.

There are problems with the endeavour of describing RT. Firstly, the theory is constantly evolving, so it is hard to provide a single authoritative account. Secondly, almost as if to illustrate SW's notion of the non-determinism of utterances, I have often found it difficult to obtain a concrete interpretation of their writings. Consequently, the account presented here represents a consistent amalgamation of the information contained in the following sources: Sperber and Wilson (1982a), Sperber and Wilson (1986), Wilson and Sperber (1986), Wilson and Sperber (1987), Blakemore (1987) and Wilson and Sperber (1988). In places, I introduce my own material to illustrate my understanding of RT.

The structure of this chapter is as follows:

1. The RT view of *meaning-nn* and communication is given. Since this has very little bearing on a computational implementation, this section is presented with minimal detail.

2. The Principle of Relevance is explained incrementally, using a series of increasingly specific definitions.

3. SW's conjectures concerning mental architecture are discussed. The way an utterance interacts with a context is described by considering the way the memory and inference components of the central systems interact according to RT. In this section in particular, the problematic aspects of RT from a computational viewpoint will be raised. I will also make various suppositions in order to flesh out some of the underspecified areas. This section is sub-divided into the following areas:

   (a) Some constraints on the structure of logical forms are described.

   (b) The *conceptual memory*, the long-term storage component, is discussed.
   (c) The *deductive device*, the critical inferential process, is described.
   (d) The context is briefly discussed.

4. The significant problems in developing a computational account are listed.

## 2.1 Communication

In this section, I describe SW's alternative notion of meaning-nn which they call *ostensive-inferential* communication.

### 2.1.1 Problems with Gricean Meaning

Various attempts have been made to formalise Grice's notions of meaning-nn and communication, the most notable of which have been by Strawson (1971) and Schiffer (1972). For them, meaning involves the hearer recognising the speaker's intention to inform the hearer of something. More formally, the hearer must recognise the *communicative intention*:

*Informative Intention:* The intention to inform the audience of something.

*Communicative Intention:* The intention to inform the audience of one's informative intention.

Sperber and Wilson (1986, pp. 30–31) construct an example that suggests that the recognition of the communicative intention is not sufficient for successful communication. For guaranteed success, either an infinite number of meta-communicative intentions need to be recognised or the informative intention needs to be mutually known. Both are seen as psychologically implausible. In response, SW invent the notion of *manifestness* in a *cognitive environment* to explain communication.

### 2.1.2 Redefining Meaning

#### Manifestness and Cognitive Environments

SW begin a redefinition of communication by weakening the notion of mutual knowledge so that it is psychologically plausible in their opinion.

To begin with, they define the notions of *manifestness* and a *cognitive environment*:

*Manifest:* A logical form, $p$, is manifest to S at time $t$ iff S can mentally entertain $p$ and accept it as true with some degree of belief at that time.

*Cognitive Environment:* The set of all logical forms manifest to an individual.

Thus, manifest information is that which is present in the central processes at a given time. It may originate from the input processes, the memory or by inference. An individual's cognitive environment at a given time consists of the propositions that are accessible to that individual. This is intended as an alternative to the notion of context prevalent in the literature.

However, manifestness is not merely a quantitative concept, it can be used comparatively. The *degree of manifestness* of a proposition is dependent on its cognitive prominence. The notion of manifestness is distinct from the notion of strength of belief. A proposition can be strongly believed but weakly manifest. Consider:

Margaret Thatcher will never play cricket for Albania. (24)

This hypothesis is almost certainly true, but was probably not very manifest in the reader's cognitive environment at the time of reading. It appears that SW believe that manifestness decreases with depth of inferencing[1]. So, for example:

Zapata believes that agrarianism is important. (25)

is more manifest than the higher-level belief:

Zapata believes that Zapata believes that Zapata believes that agrarianism is important. (26)

even though (26) might be deduced from (25). At some depth of nesting of belief, a proposition may not be manifest at all.

### Mutual Manifestness

SW define information which is *mutually manifest* to a group. This is the weaker version of mutual knowledge:

*Shared Cognitive Environment:* The intersection of two or more individual's cognitive environments.

*Mutual Cognitive Environment:* A shared cognitive environment in which it is manifest which people share it.

*Mutually Manifest:* A proposition, $p$, which is a member of a mutual cognitive environment is mutually manifest to all those who share that environment.

A mutual cognitive environment is established using the same inductive procedures that were mentioned in section 1.4.1.

Why is mutual manifestness weaker than mutual knowledge? Because for a proposition to be mutually manifest, the parties involved need only to believe that each other share certain beliefs, including this belief itself. Various information may support this, but it cannot involve an infinite regress, because only beliefs that are cognitively plausible can be manifest. Thus, infinitely high-levels of belief cannot form part of a cognitive environment.

Mutual cognitive environments describe the common ground that exists between individuals. Often, the purpose of communication is to enlarge this ground. This can be contrasted with the normal usage of mutual knowledge which is to describe which objects can meaningfully be mentioned in a discourse. Consider the situation where a person, $S$, is being troubled by a ferret in the presence of a friend, $H$:

$S$: I wish ferrets weren't so difficult to remove. (27)

[1]If they are not saying that manifestness decreases with depth of inferencing, then they are at least saying that it decreases with depth of nesting of beliefs. They do not mention either option explicitly, but presumably when reasoning about common knowledge, at some point, the higher-level knowledge must become less manifest. If this were not the case, the notion of "mutual manifestness" which will be introduced shortly, would be equally as psychologically implausible as mutual belief.

$S$ may not believe $H$ believes that $S$ believes that the creature present is a ferret, and that $H$ believes $S$ believes this etc. Indeed, the speaker may not even believe that $H$ realises that the creature is a ferret. She merely needs to have the confidence that $H$ is able to make the necessary assumptions to understand the utterance.

There is a small problem with the notion of mutually manifest information. A shared cognitive environment is the intersection of all manifest information. However, since manifestness is a comparative concept, the shared cognitive environment must somehow combine the degrees of manifestness of each propositionally equivalent pair of beliefs. If, alternatively, a shared cognitive environment only treats manifest propositions qualitatively, it is impossible to distinguish the following situations:

1. The proposition $p$ is mutually manifest to $S$ and $H$. $p$ is very strongly manifest to $S$ and very weakly manifest to $H$, or vice versa.

2. The proposition $p$ is mutually manifest to $S$ and $H$ and equally as manifest to each of them individually.

This could cause problems for SW's definition of communication below.

### The Informative and Communicative Intentions

SW redefine the communicative intention as follows:

*Informative Intention:* To make manifest or more manifest to the audience a set of assumptions.

*Communicative Intention:* To make it mutually manifest to the audience and the communicator that the communicator has a particular informative intention.

The informative intention makes a certain set of information more manifest to the hearer. Its definition uses "manifest" rather than "mutually manifest" to allow for the fact that the speaker can communicate a lie.

The set of information made manifest by the informative intention may also be vague. Several conflicting possibilities may be weakly manifest. Consider:

*Caesar*: Let's leave this orgy. (28)

*Calpurnia*: We've only just come. (29)

Calpurnia may be implicating "no", "soon" or "yes". It is easy to envisage many contexts where the "no" and "soon" readings are equally manifest to Caesar.

The same non-determinacy may occur with deictic references:

*Calpurnia*: Listen to that lyre. (30)

Here, it is not clear exactly which aspect of the sounds emanating from the lyre Calpurnia wishes to be heard. It could be the harmony, rhythm or timbre of the music. In certain circumstances, exactly what she wishes to communicate may not matter.

The definition of communicative intention is such that both parties must be aware that communication is taking place; communication cannot take place unwittingly.

### 2.1.3 Ostensive-Inferential Communication

How is the speaker's informative intention made mutually manifest to the speaker and the hearer? This takes three basic steps:

1. The speaker produces some stimulus which pre-empts the hearer's attention. The hearer recognises that this stimulus is deliberate.

2. When possible, the speaker decodes the stimulus using the conventional meaning of its constituents.

3. By consideration of this conventional meaning, the hearer infers the speaker's informative intention.

Each of these will be examined in turn.

#### The Attention Pre-empting Stimulus

The stimulus can be anything that pre-empts attention, for example rapid movements, loud noises, or speech. Of course, some of these stimuli do not normally have any conventional meaning, in which case the hearer can immediately skip to step three.

It is vitally important that the stimulus appears deliberate. SW claim that the speaker effects this either by using exaggerated, unnatural actions or by employing a medium that is normally used only for communication. Such stimuli are said to be *ostensive*.

Consider a situation where $S$ and $H$ are present at a seminar on Crystal given by the author. $S$ suddenly makes a yawning sound, much louder and longer than both $S$ and $H$ consider normal. This both pre-empts $H$'s attention and appears deliberate. This is therefore an ostensive stimulus.

#### Decoding The Conventional Meaning

Many stimuli have a conventional meaning associated with them. This is not to say that it is the "actual" or "correct" meaning. Since this meaning is "conventional", it is shared by the participants in the communicative act. Thus, they will all be able to obtain the stimulus' conventional import by a decoding process. According to SW, this is the only role of decoding in the comprehension process; the rest of the meaning must be inferred.

Speech and language are special cases. According to Fodor (1983), we have an input module dedicated to the decoding of language. Consequently, the decoding of language is a reflex action; the decoding of most other stimuli is voluntary.

To continue with the "yawning" example given above, $H$ will recognise that the conventional meaning of a yawn is that the yawner, $S$, is tired.

#### Inferring the Informative Intention

The hearer now uses the Principle of Relevance to work out what the speaker meant to convey. This is the subject of the next section and the purpose of Crystal.

From the "yawning" example, $H$ must now attempt to work out what $S$ actually meant. Perhaps $S$ is telling $H$ that the seminar is enough to make him want to sleep.

It is worth noting that, generally, a *set* of information is communicated. This will also include many peripheral facts that are merely highlighted. For example, in the yawning example, it may be weakly manifest that $S$ is a woman, that $S$ has opened her mouth, etc. When utterances are vague, all of the putative interpretations will be weakly manifest within this set.

### Defining Ostensive-Inferential Communication

SW define communication thus:

*Ostensive-inferential communication:* The communicator produces a stimulus which makes it mutually manifest to communicator and audience that the communicator intends, by means of this stimulus, to make manifest or more manifest a set of assumptions.

## 2.2 The Principle of Relevance

In order to explain the Principle of Relevance, I first describe the problems with the Gricean approaches that it claims to overcome. Subsequently, an initial definition of a Relevance measure in a fixed cognitive environment, *Relevance to an Individual*, is provided. This definition is extended to handle a varying cognitive environment as *Contextual Relevance*. Finally, the Principle of Relevance is defined in terms of the notion of *Optimal Contextual Relevance*.

### 2.2.1 Problems with Grice's Maxims

For RT to be of any use, it must offer some advantages over Grice's approach. Therefore, I will briefly describe SW's criticisms of the co-operative principle which they claim RT overcomes.

#### The Maxims are Too Vague

Many of Grice's maxims are difficult to formalise. Take, for example, "Avoid obscurity" or "Make only relevant contributions". There are no known computable tests for obscurity and relevance. The reason for this is that these notions are not precise. However, if the maxims are only used descriptively, it is impossible to fulfill the calculability constraint.

As a corollary to this, there are no computational systems that have attempted to implement all of the Gricean maxims. There have, however, been implementations of small subsets, for example those that deal with generalised scalar implicature (Gazdar, 1979; Hirschberg, 1985).

Many other computational systems that involve the co-operative principle merely make intuitively justified claims that they are adhering to one or more of the maxims, for example Carberry (1986).

#### The Maxims are not Obviously Complete

Partially due to the fact that the maxims are underspecified, they may not be enough to explain co-operative dialogue. As evidence for this, some researchers have argued that the maxims require modification. For example, Harnish (1979) has further sub-categorised the maxims of manner and Joshi et al. (1984) have argued that the maxim of quality needs to be revised.

### Problems with Calculability

Wilson and Sperber (1986) argue that there are problems with calculability which stem from the third stage of Grice's "working out" schema presented in section 1.5.3:

3. $S$ could not be being co-operative unless he believes $q$

It is not explained how $q$ has been derived. It appears that it is a hypothesis or *non-demonstrative* inference. If so, it is difficult to see how $q$ can be mutually believed by $S$ and $H$, since, by its very nature, it cannot be deduced and is defeasible.

### Problems with Flouting

Levinson (1983) suggests that when normative rules are provided for language usage, meaning can also be conveyed by flouting these rules. However, in the case of the Gricean maxims, it is not clear how one can detect when the rules are being flouted. When $S$ has apparently contravened the maxims, should $H$ attempt to make implicatures that may obviate this, or assume that the flouting is deliberate and subsequently attempt to infer the actual meaning?

## 2.2.2 Overcoming the Problems of the Maxims

SW attempt to solve the above problems as follows:

- They assume that there is a single maxim which everybody obeys and attempt to define this maxim precisely. This should overcome the vagueness and completeness problems.
- They claim to provide a psychologically plausible, computational account of the production of both non-demonstrative and deductive inferences. This should overcome problems of calculability.
- The maxim of Relevance is not normative and therefore cannot be flouted.

## 2.2.3 Defining Relevance

The purpose of this section is to lead up to a definition of the *Principle of Relevance*, SW's alternative to the co-operative principle. The idea is that all acts of ostensive-inferential communication obey this principle.

### What is Relevance?

The term "Relevance" used in RT is not identical to the natural language concept[2]. Informally, a proposition is Relevant if it connects into the hearer's cognitive environment in a manner that provides a relatively high information yield for relatively low overheads in terms of cognitive effort.

Consider the following three utterances that I assume are not Relevant in the reader's current cognitive environment, although semantically well-formed:

Xhosa is the language of the Bantu people of Cape Province. (31)

[2] I will generally show this distinction by writing the non-natural usage with a capital "R".

This sentence is an example. (32)

The author of this thesis is a gerbil. (33)

They are intuitively not Relevant because (31) cannot be connected into the reader's cognitive environment, (32) is not making anything more manifest, and (33) is false, thus results in no significant changes to the cognitive environment.

This provides an indication of how incoming information might be judged Relevant. It might:

1. Introduce new information that connects into the hearer's cognitive environment.
2. Reinforce an existing connection by making the hearer believe it with more strength.
3. Cause the hearer to modify parts of her existing cognitive environment.

These modifications to the environment are known as *contextual effects*. Presumably, a contextual effect results in the proposition involved becoming more manifest.

It is not sufficient for an utterance merely to produce contextual effects to be Relevant. It must also do this in an economic fashion. If Relevance were to be considered as a comparative quantity, then:

Tarski is Polish. (34)

should be more Relevant than

Tarski is Polish and Tarski is Polish. (35)

under the circumstances that (34) yields any contextual effects in the hearer's cognitive environment. This is because both (34) and (35) yield identical contextual effects but (35) requires more parsing effort.

SW generalise examples such as the above so that one utterance is considered more Relevant than another if it consumes less processing effort but yields (at least) the same contextual effects.

Thus, SW arrive at a definition of Relevance as follows (Sperber and Wilson, 1986, pp. 145):

> (43) *Relevance to an individual* (comparative)
> *Extent condition 1:* an assumption is relevant to an individual to the extent that the contextual effects achieved when it is optimally processed are large.
> *Extent condition 2:* an assumption is relevant to an individual to the extent that the effort required to process it optimally is small.

The definition mentions "Relevance to an individual" to emphasise the fact that Relevance can only be evaluated with respect to a particular person's cognitive environment.

SW's only useful observations about effort are that inferencing cannot be said to consume effort. If it did, no utterance would ever be Relevant. For similar reasons, they claim that monitoring Relevance does not consume effort.

Even though SW have avoided giving some exact quantitative measure of Relevance, it will be seen that subsequent definitions essentially require such a measure.

### The Contents of the Context

The definition of Relevance to an Individual does not incorporate the notion of a changing context. In order to include it, the interaction of an utterance with the central processes is outlined, then the notion of *Contextual Relevance to an Individual* is defined.

SW postulate the existence of particular central processes: some memory modules and a deductive inference engine which they call the *deductive device.* The cognitive environment consists of the contents of the memory of the deductive device, possibly enhanced by adding information from various other memory components and the input processes.

The central processes work with symbolic representations of propositional attitudes, which SW call *factual assumptions.* These are composed of basic atomic units called *concepts.* Information about these concepts is stored in a long-term storage module called the *conceptual memory.*

The deductive device can operate on the factual assumptions in its memory, which I shall call the *deductor memory*, finding their logical consequences, and merging these consequences back.

### Processing an Utterance

The initial contents of the deductor memory contain information processed from previous utterances. This constitutes the *initial context.* The input processes will provide the central processes with a representation of an incoming utterance which denotes the utterance's conventional meaning.

The representation of the utterance is added to the deductor memory. Deductive inferencing then proceeds. Presumably, non-demonstrative inferencing can occur during this process as well. The quantitative Relevance of the utterance is then assessed. If it is not sufficiently high, a series of context expansions followed by inferencing is repeated until it is. If the utterance is not Relevant enough in any of the expanded contexts, the comprehension process fails.

### Expanding the Context

There are three ways to expand the context:

1. Further information can be requested from the input processes. This is effectively making a closer examination of the external environment. Progressive expansions involve closer and closer scrutiny.

2. Information derived from previous utterances can be added. I will assume that old information is eventually rejected from the deductor memory and placed in some medium-term memory which I shall call the *context buffer.* Including information from previous utterances amounts to introducing information from this buffer. Further expansions involve progressively adding more information from the buffer, in reverse temporal order of its removal from the deductor memory. The context buffer differs from the conceptual memory in that information it contains concerns previous processing contexts. This information is indexed using its chronological proximity to the current time.

3. The representation of the utterance can be used to obtain information from the conceptual memory. SW are vague about this, but note that the simplest way of doing this would be to copy the information about each concept mentioned in the utterance from the conceptual memory to the deductor memory. For subsequent expansions in this manner, progressively more of the factual assumptions present in the deductor memory can be used for indexing into the conceptual memory.

### The Order of Context Expansions

If, at some processing point, the Relevance of the utterance is not sufficiently high, the context can be expanded to an arbitrary degree in any one of the three mentioned directions, followed by a bout of inferencing. Each possible expansion will contain the previous context as a subset[3].

SW do not give any further information on the ordering of context expansion. They do, however, give more information about how an utterance can fail or succeed to be Relevant.

The representation, $U$, of an utterance, can fail to be Relevant, because:

1. $U$ is already believed in the smallest context as strongly as possible[4]. An example of this is (32).

2. $U$ is not already believed in any context and results in no contextual effects. Examples of this are (31) and (33). It is unlikely that (31) will ever be brought into the context during expansion, and its introduction as an utterance does not result in contextual effects. (33) is believed to be false, and is therefore never introduced into the context at any stage.

SW only enumerate typical cases where an utterance succeeds in being Relevant:

1. $U$ is believed in the initial context at less than maximal strength. Expansion is justified as long as Relevance is at least maintained (ie. the information yield outweighs the effort required).

2. $U$ is not believed in any context[5], but yields contextual effects in the initial context. Expansion is justified while Relevance is at least maintained.

3. $U$ is not believed in any context, but yields contextual effects in an enlarged context. The context must be expanded until Relevance is achieved, and may be further expanded while it is at least maintained.

4. $U$ is believed in an enlarged context, but yields contextual effects in the contexts that do not contain it. $U$ is being forcibly introduced into smaller contexts to minimise effort by preempting some context expansion.

---

[3]This is not strictly true if defeasible inferencing is allowed. However, a partial order can be placed on contexts that describes whether one is an expansion of another. This can be used as an alternative definition of "bigger".

[4]This implicitly makes the arguable claim that a belief which is certain cannot be made any more manifest.

[5]SW are somewhat vague as to what aspects of the central processes constitute the context. I discuss this further in chapters 3 and 4.

An example of case 1 might be my repeating a definition given some time ago which the reader did not completely remember.

Case 2 is perhaps the most usual case. Consider:

I have a Rottweiler. His name is Buttercup. (36)

The second sentence yields the immediate information that "His" refers to the dog, that the dog is male, and that the dog's name is Buttercup.

Case 3 is typical of utterances that we have to "think about" to understand. For example:

Do you like my new ball gown? (37)

I'm a vegan. (38)

(38) can be understood as a reply to (37) given that certain assumptions are made accessible through the visual input systems and conceptual memory. The intention of the speaker of (38) might be to make manifest a set of assumptions that includes:

1. The ball gown is made of silk.
2. Silk is produced through the exploitation of silkworms.
3. Silkworms are animals.
4. An object made of a substance can be said to be involve that substance.
5. The ball gown involves the exploitation of animals.
6. Vegans disapprove of the exploitation of animals.
7. If someone disapproves of something, they prefer not to form objective opinions about something which involves it.
8. The speaker prefers not to provide an objective opinion about the ball gown.

It is unlikely that the above steps are available in the initial context. The utterance was an efficient way of communicating why the vegan would prefer not to answer the question.

The fourth case explains how information that would ultimately have become accessible is made more accessible by mentioning it explicitly. For example, if the vegan in (38) was unsure about the degree of accessibility of the information necessary to understand her reply, she could make some of it more explicit by uttering:

I'd prefer not to give an opinion because I'm a vegan
and your dress is made of silk. (39)

### Contextual Relevance

SW appear to redefine Relevance to an Individual in terms of their existing notion of Relevance to an Individual. To rationalise this redefinition, I shall call it *Contextual Relevance.*

*Contextual Relevance to an Individual:* An assumption is contextually relevant to an individual at time $t$ iff it is sufficiently relevant to that individual in one of the accessible contexts at that time.

This definition is even weaker than the descriptions of Relevance in a context informally described above, but is intended as their full definition of Relevance[6].

### The Presumption of Optimal Relevance

When communication takes place, SW claim the hearer attempts to communicate a certain set of assumptions as efficiently as possible. The hearer assumes the speaker is doing this and attempts to retrieve the information using the notion of Relevance. More formally:

*Presumption of Optimal Relevance:*

1. The set of assumptions that are intended to be made manifest or more manifest to the perceiver are contextually Relevant enough to justify their provision of processing effort and attention.
2. The ostensive stimulus is the most contextually Relevant one that could have been used for communicating this set.

Thus, any act of ostensive-inferential communication should be optimally Relevant:

*Principle of Relevance:* Every act of ostensive communication communicates the assumption of its own optimal Contextual Relevance.

The hearer of an utterance uses non-demonstrative inference, deduction and context expansion controlled by considering its Contextual Relevance in order to obtain its associated meaning. The consequence of the Presumption of Optimal Relevance is that the first Relevant reading obtained is assumed to be the intended one. If this reading is not obtained, the speaker has failed to communicate successfully because her estimations of the mutual cognitive environment and the hearer's cognitive environment are wrong. Consider:

Have you seen the bulb in the garden? (40)

Most people would assume this refers to an organic bulb, because this information is made more manifest by the use of the word "garden". If the other reading was intended, the utterance should be re-phrased as:

Have you seen the electric light bulb in the garden? (41)

[6]Where there is no ambiguity, I shall refer to Contextual Relevance to an Individual as Contextual Relevance or just Relevance.

## 2.3 Logical Form

It is now possible to examine the function of the pertinent central processes in more detail. It will be seen that there are many problems and unresolved issues associated with SW's extant descriptions. Where possible, I attempt to fill in details in a way compatible with RT.

Fodor calls the symbolic representations common to all of the central processes *logical forms*. They play an essential role in RT because:

1. They represent the utterance and the context.
2. Their syntactic properties partially determine contextual Relevance. This will become apparent over the next few chapters.
3. Their associated semantic property of *strength* can also contribute to contextual Relevance.

Thus, a description of their manipulation by the central processes will allow a more concrete definition of contextual effects to be provided.

SW claim that all information communicated between the central processes can be expressed in terms of the propositional attitudes of *belief* and *desire*. Beliefs and desires are stored in distinct areas where their appropriate modality is implicit. *Factual assumptions* are the representations of beliefs. Thus the factual assumption[7]:

My squid cannot play the sitar. (42)

implicitly represents:

I believe that my squid cannot play the sitar. (43)

Despite the fact that these forms are so important, SW do not think it necessary to provide much specific information about their form and semantics. In this section, I will attempt as full an explanation of them as is possible from SW's various texts.

### 2.3.1 What Must Factual Assumptions Represent?

Since SW claim that factual assumptions represent individuals' beliefs and therefore their cognitive environment, I assume that assumptions must be able to represent the following, which might be components of the context (according to Lyons):

1. Knowledge of social role, role in the speech event, and relative social standing.
2. Knowledge of relevant spatio-temporal co-ordinates.
3. Knowledge of the level of formality.
4. Knowledge of the channel of communication.
5. Knowledge of the subject matter.
6. Knowledge of the domain that determines the register of the language.

Some of these aspects include linguistic knowledge.

[7]I will use SW's convention of representing factual assumptions in English. In chapter 5, I introduce a more formal representation.

### 2.3.2 Concepts

All factual assumptions are composed of concepts. Information about these concepts is available in the conceptual memory. It will be seen that these concepts are not semantic primitives, but are defined in terms of each other.

### 2.3.3 Assumption Schemas

#### Instantiation

The representation of the utterance delivered by the input processes is rarely a factual assumption because it is not fully instantiated. For example, it may contain a pronoun. SW note that even uninstantiated assumptions can sometimes take part in inference. For example, from:

He is dead. (44)

it is possible to infer

He is not alive. (45)

Thus, the central processes must also manipulate uninstantiated assumptions which SW call *assumption schemas.*

#### Uses of Assumption Schemas

The notion of assumption schemas is very important to RT. However, SW do not explain in any detail how they are instantiated or exactly what a schema means. I will provide some examples of the apparent uses here, and suggest a semantics for the schemas in chapter 5.

The most obvious use of an assumption schema is to represent an utterance. For example, the utterance:

He is a Rottweiler. (46)

might be represented by:

$\boxed{\phantom{xx}}$ is male and a Rottweiler. (47)

The box will be filled by the symbol denoting the appropriate male Rottweiler. The instantiation of this schema will be a non-demonstrative inference: it may not prove correct.

Schemas can also be used to represent prototypical or stereotypical information:

$\boxed{\phantom{xx}}$ is a car with four wheels. (48)

This is instantiated by some car, and will produce the non-demonstrative inference that it will have four wheels.

Schemas may also be used to postulate the existence of some object, the extension, name or value of which is unknown at this point. For example:

The outside temperature is $\boxed{\phantom{xx}}{}^\circ$ Centigrade. (49)

Even without knowledge of the value of the outside temperature, the schema might be used for inferencing.

Finally, schemas can be used to express information that is true of all members of a category:

The axolotl $\boxed{\phantom{xxx}}$ is an amphibian. (50)

This is true if instantiated by a symbol denoting an axolotl.

Note that the first two uses of the schemas are non-deductive and the final two uses are deductive[8].

The instantiation of assumption schemas is implicitly an inferential process. For example, (50) should not be instantiated by *any* object, it should be instantiated by an object that is an axolotl. Indeed, (50) might be considered to be a rule that takes any object and infers that it is an amphibian *if* it is an axolotl.

### 2.3.4 Strength of Belief

#### Strength and Probability

SW associate a *strength* with every factual assumption and assumption schema. They are not very precise about what this represents.

They deny that strength can be regarded as a subjective probability value. This is because they assume that such values are independent of the form of an assumption and can only be changed in the light of new evidence.

They go on to provide an example (Sperber and Wilson, 1986, pp. 77–78) where a person holds certain incompatible beliefs and subsequently revises them. SW believe that the fact that it is possible to believe an assumption less strongly due to the revision process means that it cannot be represented using probability theory. They further support this with the observation that strengths may be dependent on processing history:

> ...these variations in strength are neither the object nor the output of a special logical computation. They arise, rather, as by products of various cognitive processes, deductive and non-deductive. (pp. 77).

In chapter 9, I will show that SW's supposition that strength cannot be modelled using probabilities is wrong due to the fact that they have not sufficiently distinguished degree of belief, deduction and belief revision.

#### Representing Strength

SW suggest that we can explicitly represent beliefs about the strengths of other assumptions. In such cases, the strengths are represented using some gross scale, such as *weak, strong, very strong, certain.* They see these as degrees of confirmation, but note that the categories are "fuzzy". They also note that strengths cannot always be compared. For example:

I am more certain that the Taj Mahal was completed in 1648
than I am that my coffee is cold. (51)

When the strengths of the two propositions expressed in (51) are of similar magnitude, the comparison is meaningless.

[8]SW do not demonstrate all of these uses. However, I am making reasonable assumptions from the appropriate texts in order to be a little more specific.

## 2.4 Conceptual Memory

Conceptual memory is a long-term storage module which contains the extra information about concepts that can be used during utterance interpretation to enhance the context. SW assume it has some structure.

### 2.4.1 The Structure of the Conceptual Memory

The memory consists of "chunks" of information that can be indexed by a particular concept. Each chunk provides information about its associated concept. This information does not contain semantic primitives such as those described in Katz (1972), Wilks (1975) and Schank and Riesbeck (1981). Rather, this information is defined in terms of the other concepts that can be indexed in the memory. This seems quite reasonable. Pulman (1983) argues that semantic primitives have no adequate semantics themselves and are merely thinly-veiled representations of words[9]. Dowty (1979) goes as far as dismissing these primitives as conjunctions of predicates that entail the word meaning; these conjunctions, he argues, are not expressive enough to constitute an adequate meaning representation.

Do conceptual entries provide a necessary and sufficient definition of a word? Pulman (1983, Chapter 2) argues that it is rarely possible to provide a complete definition for a word, but it is normally possible to provide the necessary truths that follow from it. Therefore, much as concepts are defined in terms of each other, they cannot be completely defined in this way.

SW do not provide concrete examples of any logical forms. However, for the purposes of illustration, I will consider a logical form expressed in first-order predicate calculus. The utterance:

Buttercup is a black Rottweiler. (52)

might translate into the following:

$$\text{Rottweiler}(\textit{buttercup}) \wedge \text{Black}(\textit{buttercup}) \tag{2.1}$$

The concepts that constitute form (2.1) are: Rottweiler, Black, *buttercup* and $\wedge$. Each of these could be used to access a chunk of information, called a *conceptual entry* (see figure 2.1).

These conceptual entries provide information about the meaning of the concept, defining it in terms of other concepts (shown in the diagram as black dots). In the diagram, it can be seen that the entry for "Black" makes references to the entry for "Colour".

### 2.4.2 The Structure of Conceptual Entries

I assume each entry is indexed by a unique key: its associated concept. Entries are partitioned into three, possibly empty sections:

1. The lexical section.
2. The encyclopædic section.
3. The logical section.

I examine each of these in turn.

[9] I am assuming that concepts are the semantic entities denoted by words.

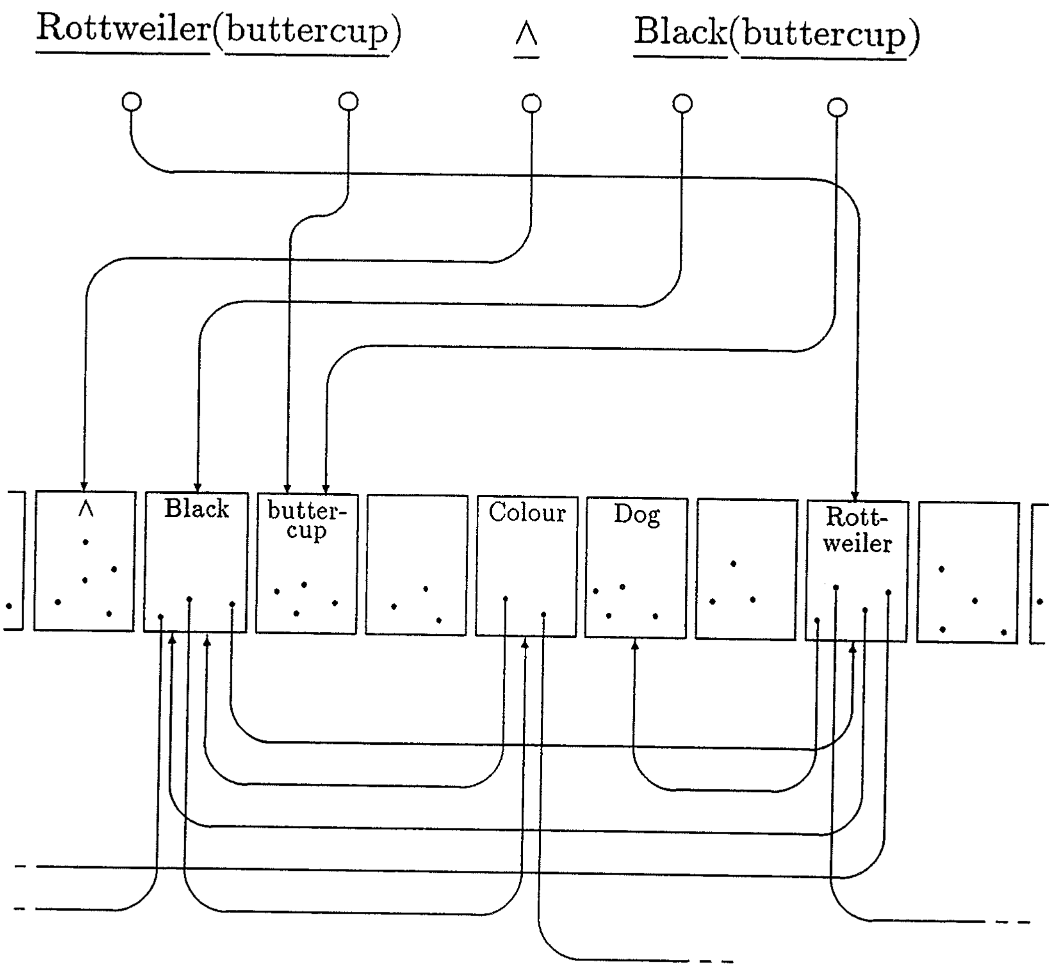



Figure 2.1: Concepts are defined in terms of other concepts in the memory.

### The Lexical Section

This contains information concerning the vocalisation of the concept, if one exists. This will include phonetic and lexical information about the word or words that correspond to that concept. This data may be used in the interpretation and generation of speech.

### The Encyclopædic Section

This contains information about the denotation and extension of the concept. This may include stereotypical information expressed by assumption schemas or some other means and information about particular objects that are somehow involved with this concept.

### Logical Rules

These are rules used by the deductive device. They operate on encyclopædic information in the deductor memory to deduce new information. A typical example from Sperber and Wilson (1986) is:

> (51) *Salt-elimination rule*
> *Input:* $X$ – salt – $Y$
> *Output:* $X$ – substance of a certain kind – $Y$

This rule will take any assumption containing the concept "salt", and deduce the same assumption with "salt" replaced by "substance of a certain kind". SW intend that these rules should be used in place of meaning postulates (Fodor et al., 1980). In AI terms, logical rules are *meta-rules* (Nilsson, 1982; Weyhrauch, 1985); these are declarative statements of knowledge concerning the control of some system. In sections 2.5.3 and 2.5.4, some constraints on the structure of these rules will be given.

SW note that logical and encyclopædic information may be interchangeable. Nevertheless, they suggest that the distinction between the philosophical notions of analytic and synthetic truths may be the distinction between logical and encyclopædic entries. The logical entry provides knowledge without which the concept cannot be said to be properly understood. Lacking encyclopædic information is more like a lapse of memory. However, the very fact that the two kinds of knowledge can be interchangeable seems to invalidate this analytic/synthetic bifurcation.

## 2.4.3 Local and Global Processes

SW's conceptual memory structure does not fully adhere to Fodor's idea of the modular mind.

Fodor contrasts *local* modules with *global* systems. The input processes are local in nature. They are informationally encapsulated and have very limited interaction with the central processes. Each of them may therefore employ a unique representation. By contrast, the central processes share each other's forms, thus they require a common symbolic representation.

The deductive device is unusual as a central process because it has some local characteristics. In particular, the logical rules that will be loaded into it are not general logical forms, but specific meta-level instructions. Thus, the deductive device has some limited modularity.

The lexical and logical entries contain specific information which is not normally accessed by the central processes. Consequently, there is no reason to suppose that this information is expressed as a logical form. Thus, the lexical information used by the input processes would be better considered to be part of the germane local modules rather than part of the conceptual memory. Similarly, the logical rules are better considered to be encapsulated in the deductive device.

## 2.5 The Deductive Device

Two modes of inference are used during utterance comprehension:

1. Deductive inferences. These are inferences drawn from the symbolic form of an assumption. A deductive inference must be a logical consequence of its antecedents (Genesereth and Nilsson, 1987).

2. Non-demonstrative inferences. These are inferences that are not semantically entailed by their antecedents. Typically, they are characterised by abduction or induction.

According to SW, deduction is required even during non-demonstrative inferencing, in which case it is used to test a hypothesis for logical consistency in a context. The only means of obtaining non-demonstrative inferences explicitly mentioned by SW are assumption schemas. Somehow, the non-demonstrative inference processes must interact with the deductor memory.

The deductive device is responsible for all aspects of mental deductive inference. Furthermore, the contents of the deductor memory form at least part of the context. Thus, the notion of contextual effects will be defined in terms of the behaviour of the deductive device.

### 2.5.1 Structure of The Deductive Device

The only aspects of the deductive device that SW mention are the logical rules and the deductor memory. However, it is reasonable to assume that the logical rules must be stored in a *logical memory* and that there is some overall *control process* which governs the module.

### 2.5.2 Forward-Chaining in the Deductive Device

The control process applies rules in a *data-driven* manner, as opposed to a *goal-driven* manner. Rather than trying to establish the fact that a particular proposition follows from the context by working backwards from that proposition to the facts in the context, the controller establishes all the consequences it can from the context by applying the logical rules to them. This is known as *forward-chaining* (see Moore (1982) for more details).

More formally, the inference function of the deductive device takes the *transitive closure* of the forward-inferences[10]. Specifically, all of the consequences that can be obtained by applying the logical rules to the deductor memory are obtained. These consequences are

[10]This is my own formal interpretation, but it follows straightforwardly from SW's descriptions.

merged with the deductor memory and the inference function called recursively on the result. The procedure terminates when no more inferences can be made. In this case, a *fixed point* of the inference function has been found: if the function is called on the contents of the deductor memory, it merely returns them unchanged.

### 2.5.3 Constraints on the Logical Rules

The inference function will only terminate when no more inferences can be derived. However, problems can arise if any of the logical rules are *introduction rules.*

An introduction rule is a rule that deduces a consequent which contains every concept found in its antecedent plus at least one more concept. SW claim that such rules are not used by the deductive device because:

1. The inference procedure might never terminate of its own accord.

2. Such rules are cognitively implausible for spontaneous inferencing.

Consider, for example, an $\wedge$-introduction rule which infers $A \wedge A$ from $A$. Such inferences might be useful for mathematical reasoning, such as natural deduction, but SW claim they are not the sorts of inferences that are useful from a common-sense point of view. Furthermore, its use will result in an infinite conjunction of $A$s being derived.

Therefore, logical rules are constrained to be *elimination rules.* These are rules that have at least one less concept in their consequent than in their antecedent. SW claim all cognitively useful rules can be expressed as elimination rules. The inference function is guaranteed to terminate with a finite fixed point when the logical rules[11] are exclusively of this form and the encyclopædic information are propositions[12].

### 2.5.4 Contextual Implication

SW attempt to define how new information might be said to link up with the existing context. Non-trivial inferences that connect with the context are known as *contextual implications.* I will gradually build up their definition in this section.

#### Non-Trivial Implications

Non-trivial implications are those that SW envisage to be cognitively plausible. More precisely:

*Non-trivial implication:* A set of assumptions, $P$, logically and non-trivially implies an assumption $q$ iff there exists a derivation whose last step is $q$ and all of the steps in the derivation consist either of premises taken from $P$ or of inferences made from previous steps using only elimination rules.

[11] In fact, the rules need only be non-introduction rules to guarantee termination.

[12] If predicate calculus with terms that include functions are allowed at the encyclopædic level, then termination is no longer guaranteed (see section 4.2).

### Analytic and Synthetic Rules

Logical rules can be considered to be functions from assumptions to assumptions. SW posit that they will have one of the following signatures:

$$\mathcal{A} : assumption \longrightarrow assumption \tag{2.2}$$

$$\mathcal{S} : assumption \times assumption \longrightarrow assumption \tag{2.3}$$

Rules of type (2.2) that rewrite a single input assumption are known as *analytic rules.* Rules of type (2.3) that rewrite two input assumptions to yield a single assumption are known as *synthetic rules.* Multiple input synthetic rules can be built by cascading two-input synthetic rules.

A typical non-trivial analytic rule is *and-elimination*, which might be expressed by the two logical rules:

*And-elimination (a):*
**Given:** $p \wedge q$
**Yield:** $p$

and

*And-elimination (b):*
**Given:** $p \wedge q$
**Yield:** $q$

The two forms of the rule are necessary because a rule can only yield one result at a time yet $p \wedge q$ non-trivially implies both $p$ and $q$.

A typical synthetic rule is *modus ponens*:

*Modus Ponendo Ponens*:
**Given:** $p \supset q$
**And:** $p$
**Yield:** $q$

SW claim that analytic rules are both sufficient and necessary for understanding a concept[13], whereas synthetic rules involve exploiting its consequences in some way. It is the synthetic rules which may involve the integration of information into a context since they can take one argument from the incoming form and the other from the context.

### Analytic and Synthetic Implications

The notion of being "analytic" or "synthetic" can also be applied to non-trivial implications:

*Analytic implication:* A set of assumptions $P$ analytically implies assumption $q$ iff there is a derivation using premises from $P$ which has $q$ as a the final step and all inference steps are the results of applying analytic rules to previous steps.

[13]However, I have noted in section 2.4.2 that such a complete definition cannot generally be provided.

*Synthetic implication:* A set of assumptions $P$ synthetically implies assumption $q$ iff there is a derivation using premises from $P$ which has $q$ as a the final step and $q$ is not an analytic implication.

Analytic rules break down or generalise an assumption, yielding analytic implications. A synthetic implication must use information from at least two sources in order to derive the new information; some of the inference steps may apply analytic rules, but at least one step must apply a synthetic rule.

SW have arranged things so that the meta-rules which control the deductive device are not fixed. In normal logical systems, the meta-rules would have expressed logical truths. SW suggest that, in their system, the relationship between analytic and synthetic implications is closely linked to the relationship between semantic truths (i.e. those that follow from meaning postulates) to logical truths (i.e. those that follow from the logical connectives, and therefore, SW claim, have "wider import").

### Defining Contextual Implications

It is now possible to be precise about "connecting into the context". Information connects into the context if it is a contextual implication:

Let $\vdash$ mean "non-trivially implies" and $\not\vdash$ mean "does not trivially imply"[14], then:

*Contextual implication:* A set of incoming assumptions $P$ contextually implies an assumption $q$ in context $C$ iff:

1. $P \cup C \vdash q$
2. $P \not\vdash q$
3. $C \not\vdash q$

Clearly, a contextual implication must be a synthetic implication. It is a trivial task for the deductive device to compute whether an inference is synthetic or analytic by noting which sort of rule has been applied to obtain it. Additionally, if the source of an assumption is somehow available, then it is equally as trivial to compute whether an inference is a contextual implication by checking the sources of the premises from which it was derived.

### Problems with the Definition

It will be shown in chapter 4 that, as SW suggested, it is possible to convert logical rules into encyclopædic information. From this, it will also be shown that the distinction between analytic and synthetic implications is arbitrary and the notion of contextual implication is meaningless.

This problem can be overcome by having a fixed set of logical rules in the deductive device. However, SW's comments about analytic/synthetic and logical/semantic truths will no longer be valid.

[14] I will assume this convention for the remainder of the thesis.

### 2.5.5 Strength and Inference

The deductive device must also manipulate the strengths associated with assumptions in its memory. SW provide some simple rules for manipulating strengths, but it is impossible to evaluate their integrity because of the lack of an underlying semantics. I attempt to provide such a theory in chapter 9. In this section, I will explain what may cause strengths to change.

There are three kinds of strengthening:

1. Dependent strengthening.
2. Independent strengthening.
3. Retroactive strengthening.

If an assumption is strengthened by any of these means, it is said to be *contextually strengthened.*

#### Dependent Strengthening

When a conclusion is deductively drawn from some premises, its initial strength will be derived from them. The magnitude of the strength depends on the type of logical rule used to derive it and the strength of the premises. This type of strengthening is known as *dependent strengthening.*

#### Independent Strengthening

If an inference strand implies an existing assumption that is held at less than full strength, it may be strengthened. Again, simple rules may be applied to determine the resultant value. This is known as *independent strengthening.*

#### Retroactive Strengthening

This does not occur as a by-product of deductive inferencing. SW suggest that if a non-demonstrative inference predicts an interpretation that subsequently turns out to be Relevant, this prediction is further strengthened. This is known as *retroactive strengthening.* Retroactive strengthening is intended to support the Presumption of Optimal Relevance by making the Relevant predictions the ones that will be most certain.

#### Some Problems With Strength

Ignoring the lack of semantic theory, both independent and dependent strengthening assume that the strengths derived for each inference strand are solely dependent on their premises. This is extremely unlikely, as research in statistical inference has shown (see, for example, Pearl (1988) and Genesereth and Nilsson (1987)).

### 2.5.6 Handling Contradictions

It is possible that an inconsistency is detected in the deductive device. This can arise because of an incorrect hypothesis, because of faulty encyclopædic information, or because the incoming utterance is incompatible with the hearer's cognitive environment. All of these cases are handled in the same manner.

If $p$ and $q$ are found to be contradictory, a belief revision function is invoked. It executes the following steps:

1. Given $p$ and $q$, the weakest believed assumption $w$ is located.
2. The set of assumptions, $A$, that analytically imply $w$ are found.
3. The set of assumptions $S$, consisting of the the weakest of each pair of assumptions that synthetically imply $w$, is found.
4. $w$ is erased.
5. The inferences dependent on $w$ are updated[15].
6. The revision function is called recursively on each member of $A \cup S$.

Thus, the deductive device must contain a module that can effect belief revision. I shall call this the *reason maintenance process*.

### 2.5.7 Contextual Effects

It is now possible to formalise the notion of contextual effects given in section 2.2.3. In the same order as they were enumerated, they can be defined as:

1. Contextual Implications.
2. Contextual Strengthening.
3. Erasures.

SW do not explain how these are to be quantified. The simplest approach would be to merely count each occurrence of an effect as a unit of information.

Note that contextual implications are a syntactic notion, strengthening is a semantic notion, and erasures are a meta-logical notion based on an assumptions semantics.

## 2.6 Context

SW do not make it clear what criteria are used for deciding which assumptions to reject into the context buffer. All they say is that the context can be expanded to include processing information from previous utterances in reverse temporal order.

Presumably, the deductor memory is finite and bounded. Thus, once the memory fills up, it remains full because there is no reason to empty it. New information will be constantly input, and less useful information rejected, into the context buffer[16].

SW also posit other memories that could be used for context switching. These would be used to preserve a particular context while another is being considered. Typical examples of this are situations where several conversational strands are being interleaved.

Although the handling of multiple and extended contexts are significant problems, I will not address them in this thesis. I will avoid them by only considering short discourse fragments that concern a particular domain.

[15] This step is not mentioned by SW, but is clearly necessary.

[16] If the rejection criteria are not exclusively based on the time the assumption was input to the deductive device, temporal information would have to be explicitly associated with an assumption for expansion purposes.

## 2.7 Does RT Fulfill its Goals?

The fundamental concepts of RT have been described. The interaction of the central modules when constrained by RT will be described in chapter 3. However, enough information has already been provided for an evaluation of contextual Relevance.

RT tackles the problems associated with Gricean pragmatics. But can SW claim to have achieved the goals outlined in section 2.2.2?

In the course of this chapter, I have identified several problems with RT. These will be re-iterated for convenience. I will then endeavour to answer the above question.

### 2.7.1 Some Problems with RT

#### Manifestness

Manifestness plays an implicit role in the central processes. Communicated information becomes manifest to an individual to some degree, but SW give no details of its practical implementation in the central processes.

Aside from its associations with salience, manifestness also appears to be related to strength (assuming strength is related to truth):

> [ *The communicator's* ] mutually manifest intention is merely to make manifest some [ *appropriate, non-determinate* ] assumptions. Hence, she does not make any of these assumptions more than weakly manifest. She does not guarantee their truth as strongly as she guarantees the truth of [ *the more determinate assumptions* ] (Sperber and Wilson, 1986, pp. 198).

#### Relevance to an Individual

This is the fundamental definition of Relevance on which the rest of RT is based. If statements like "Relevant enough" are to be taken seriously, Relevance must be defined as a calculable quantitative measure. This is not the case; the definition uses the somewhat vague extent conditions.

Furthermore, the notion of "cognitive effort" is not sufficiently explicated. Without some enumeration of the processes that consume effort and an explanation of their degree of consumption, RT is vacuously correct; any appropriate effort value could be argued to fit a particular case.

#### Contextual Relevance to an Individual

Aside from the fact that it is based on the underspecified notion of Relevance to an Individual, the definition of contextual Relevance does not provide enough detail about how and when the context should be expanded or when this expansion process should stop. Furthermore, insufficient information is provided concerning the contents of the "chunks" that constitute conceptual entries.

#### Logical Form

The vital notion of contextual implication is irrevocably bound up with logical form. However, no syntax or semantics are provided for them. The same is true for assumption schemas. Presumably, part of the reason for this is the largely unconstrained nature of the logical rules.

### Non-demonstrative Inference

The non-demonstrative inference processes essential to RT are barely sketched. The only such process mentioned involves the instantiation of assumption schemas. However, the interaction of this process with the other non-demonstrative processes and the deduction process is not explained.

### Strength

The description of strength is informal. The associated manipulation rules are *ad-hoc* and given no theoretical justification.

### Conceptual Memory

The structure of conceptual memory is not in keeping with Fodor's theories. Logical rules should be local to the deductive device; linguistic information should be local to the appropriate input processes.

### The Epistemological Status of Logical and Encyclopædic Information

SW's descriptions of logical and encyclopædic information are such that the notions of analytic and synthetic implications, and consequently contextual implications, are vacuous.

### Reason Maintenance

SW have provided an incorrect and oversimplified model of the process of belief revision.

### Contextual Effects

As they stand, contextual effects appear to be unrelated to each other. Since they all contribute to the information measure necessary to compute Relevance, it would be convenient to have a unified account of them.

## 2.7.2 The Controversy over RT

The Principle of Relevance is not normative and cannot be flouted. I have seen no objections to the theory that question this.

The main problems with RT arise from the faults mentioned above, many of which have not previously been voiced by its critics. Given these objections, it is not surprising that RT is seen by some as inadequate. It certainly lacks the precision necessary to provide a computational account that satisfies the calculability criterion. The assertion that the Principle of Relevance is not normative and cannot be flouted can only be evaluated after such an account has been developed. This is one of the functions of this thesis.

# Chapter 3

# Crystal

The design of Crystal is intended to be a simplified version of an ideal Relevance-based processing system. In order to outline such a system, I first describe how the processes mentioned in chapter 2 interact during spoken ostensive-inferential communication. Next, an ideal Relevance-based processing system is described and a similar schema developed for Crystal. The correspondence between this ideal system and Crystal is then presented.

## 3.1 Ostensive-Inferential Communication

In order to explain ostensive-inferential communication, I first introduce some more of SW's concepts. Next, these concepts are used to describe the inferences that must be drawn by someone hearing an utterance. Subsequently, details of the linguistic input processes are provided. I will then have provided enough information to sketch the modules, data and control paths that must be present in an ideal system in the next section.

Throughout this section, I will draw on the following example:

> Brünnhilde enters the fortress Valhalla, in which she cohabits with Wotan. She hears angry muttering from the bathroom and approaches it. Wotan, who is inside the Jacuzzi in the bathroom, recognises her footsteps and says:
>
> The water is too frothy. (53)
>
> Brünnhilde believes Wotan is requesting her to make the water less frothy by manipulating the controls nearby.

### 3.1.1 Terminology

#### Enrichment

SW identify three aspects of utterance instantiation:

1. Syntactic and semantic disambiguation.
2. Reference resolution.
3. Reifying underspecified concepts.

These latter two processes processes *enrich* an utterance[1]. The process of enrichment results in successive *developments* of an utterance; each progressive development is more specific than its predecessor.

(54) might represent a fully developed version of (53):

$$\begin{array}{l} \text{Wotan [said to Brünnhilde at time } t]_1 \text{ that} \\ \text{[the water in the Jacuzzi at time } t]_2 \text{ is} \\ \text{[too frothy for Wotan at time } t]_3. \end{array} \tag{54}$$

The square parentheses represent added information; the numbered subscripts are just intended for reference purposes.

Sub-clause [1] is enriched so that the addressee is added. Sub-clause [2] locates the referent of "the water". Sub-clause [3] reifies "too frothy", specifying its object of comparison. All of the sub-clauses now include the speech time.

The propositional attitude of an utterance is developed in the same way that the above sub-clauses are. SW claim that developments are largely obtained through the instantiation of assumption schemas. SW assert that there is a limit to which an utterance can be enriched. This seems unlikely because arbitrary amounts of information can be used to enhance the logical form of the utterance; I will further argue this in chapter 8.

### Implicature and Explicature

SW differentiate two types of inference:

1. *Explicatures:* inferences that are developments of the logical form of an utterance.

2. *Implicatures:* inferences drawn from explicatures and the context.

Implicatures correspond to both conversational and non-conversational Gricean implicatures. Explicatures are not usually considered in the Gricean framework.

Explicatures can be *explicit* to varying degrees, depending on how close they are to the original logical from of the utterance:

*Explicitness:* An assumption communicated by utterance $U$ is explicit to the extent that it is a development of a logical form encoded by $U$.

Thus, there is not a sharp divide between the categories of implicature and explicature.

Implicatures can be divided into two classes:

1. *Implicated premises:* the non-deductive premises used by the inference process. These will arise by instantiation of assumption schemas, by non-demonstrative inference, or by introduction from the input processes.

2. *Implicated conclusions:* information that follows deductively from some implicated premises.

SW do not seem to allow explicatures to be similarly classified. Since they often have some properties of implicatures, perhaps the above terminology can be applied to them as well.

[1] SW do not appear to regard reference resolution as enrichment, but this is of no consequence here.

### 3.1.2 An Example of Ostensive-Inferential Communication

Using the above concepts, I describe the typical inferences and assumptions a hearer must make during the communication of a spoken utterance. *How* this happens will be explained in more detail in section 3.3.1.

#### The Mutual Cognitive Environment

Consider the point at which Brünnhilde hears Wotan's utterance, (53). On hearing Wotan's utterance, the following are strongly manifest to Brünnhilde[2]:

1. Wotan is in the bathroom.
2. Wotan is making splashing noises.
3. Wotan is speaking to someone.
4. Wotan sounds angry.
5. Wotan has said that the water is too frothy.

All of these are implicated premises, supplied by the input modules. Item 5 was decoded from the utterance by the linguistic input processes. It is reasonable to assume that items 1–5 are mutually manifest to Wotan and Brünnhilde.

Brünnhilde may find these items Relevant enough and not process this information any further. For example, she may have been trying to locate Wotan. However, under normal circumstances, the context is expanded enough to warrant the following, less strongly manifest inferences:

6. Wotan is aware of Brünnhilde's presence.
7. Wotan's utterance is addressed to Brünnhilde.
8. Wotan has said to someone that the water somewhere is too frothy for something.
9. Wotan has said to Brünnhilde that the water somewhere is too frothy for something.
10. Wotan's utterance is optimally Relevant to Brünnhilde.

Item 6 is a non-demonstrative inference based on Brünnhilde's knowledge that she has loud, characteristic footsteps and that she is the normally the only person other than Wotan in Valhalla.

Item 7 is another non-demonstrative inference drawn from item 6 and the fact that Wotan must be addressing either her or himself and that it is more usual to talk to someone else.

Item 8 is an explicature supplied by the input processes.

Item 9 is an explicature, deductively drawn from item 7 and item 8. If it were an implicature, it would be regarded as an implicated conclusion. However, SW have not made it clear whether this terminology can be applied to explicatures. This highlights some of the incongruities with SW's classification of inferences.

[2]Since no details of logical form have been given by SW, I adopt their convention here of representing them in English.

Item 10 is an implicated premise present due to the Principle of Relevance.

Items 1–10 form the essential part of the mutual cognitive environment that enables Brünnhilde to establish Wotan's communicative intent and thereby establish his informative intention. Inference is required even at this stage, since Brünnhilde must establish that Wotan is talking to her.

### Establishing the Informative Intention

Brünnhilde proceeds to establish Wotan's informative intention by expanding her cognitive environment to include implicated premises about "water", "frothy" and "too", then finding some resulting non-trivial implications. The following inferences may become manifest to Brünnhilde:

a. If somebody says something, they typically believe it.

b. Wotan believes that the water in the Jacuzzi is too frothy for Wotan.

c. If $X$ believes that something is too $P$ for them, they desire that it be made less $P$ for them by $Y$.

d. Wotan desires that the water in the Jacuzzi be made less frothy for him by someone.

The utterance may not yet be Relevant enough to Brünnhilde, in which case further context expansion/inferencing might take place. The expanded cognitive environment may include information about "Jacuzzi" and "near"[3]. After inferencing, the following may be manifest to Brünnhilde:

e. The water in a Jacuzzi can be made less frothy by the Jacuzzi controls.

f. Wotan would like somebody to make the Jacuzzi less frothy using the Jacuzzi controls.

g. Brünnhilde is nearer to the Jacuzzi controls than Wotan.

h. If $X$ is nearer to something than $Y$ then it is typically more accessible to $Y$.

i. If $X$ desires $P$ and $Y$ can more easily achieve $P$, then $X$ typically desires that $Y$ achieve $P$.

j. Wotan would like Brünnhilde to make the water in the Jacuzzi less frothy using the Jacuzzi controls.

Item j can be considered to be an explicature which has enriched the propositional attitude of the original utterance representation.

Not all of the information necessary to produce these inferences has been noted here. For example, item b relies on the implicated premise that the "someone" that the water is too frothy for is "Wotan".

It is apparent that Brünnhilde may have several interpretations of (53) depending on how far she feels she should expand the context and which information is available at

[3] I assume that information about beliefs, desires and plans is always available because it is so often used. This is not a view endorsed by SW.

particular expansion levels. All SW say about this is that the context will be expanded while Relevance has not been achieved; the particular degree of Relevance depends on the individual situation.

Many of the assumptions provided by the second expansion/inferencing phase are less manifest and probably of weaker strength than those of the first phase: much as Brünnhilde can be sure that Wotan believes the water in the Jacuzzi is too frothy for him, she is less sure that Wotan wants her to change this situation.

#### Classifying the Inferences

If SW's classifications are applied to items a–j, it is apparent that much of the important work in establishing the informative intention is achieved by implicated premises:
The implicated premises are: a, c, e, g, h, i.
The implicated conclusions are: b, d, f.
The following may be regarded as explicatures: b, d, f, j.

SW claim that instantiation of assumption schemas explains much of the non-demonstrative reasoning activity occurring here. However, they do not describe this process in enough detail. Consider:

b. Wotan believes that the water in the Jacuzzi is too frothy for Wotan.

This instantiates the schema provided by item 9. An important question that must be raised is "How does instantiation take place?". Why was the slot filled with "Wotan" rather than "Brünnhilde", or "any living creature"? Presumably, the first available instantiation that is Relevant is selected; unfortunately, SW have not explained how these possible instantiations are generated, so there is no way to evaluate this assertion.

#### Determinacy

Incidentally, Brünnhilde may draw several inferences which Wotan may not have predicted when he produced his utterance. For example:

The filter in the Jacuzzi needs cleaning. (55)

Wotan should see a doctor about his bad eye. (56)

In general, less manifest information is less likely to be an intended part of the hearer's cognitive environment. The onus is on the speaker to ensure that information is prominent enough for the hearer to be sure of what the speaker meant. SW argue that poetic effects often result from an utterance producing much weakly manifest information.

### 3.1.3 The Language Modules

The procedure of spoken utterance interpretation depends on the operation of the input modules that deal with language.

According to SW, the language modules generally output some kind of assumption schema that needs enrichment. The schema will include information such as the speech time and uninstantiated argument places that need to be filled. A complete schema representing Wotan's utterance (53) might be:

Wotan [said to $\square$ at time $t$] that
The water [ $\square$ ] is too frothy
[for $\square$ ] [at time $\square$ ] (57)

However, the logical form is not delivered as a monolithic unit, rather it is progressively enriched around the point of *focal stress* of the utterance. This is the point which is most vocally stressed; it typically falls at the end of an English utterance.

The chain of developments provide specific semantics for increasingly more of the grammatical constituents of the utterance; the constituents yet to be developed are given only vague semantics.

Thus, in simplified form, developments of (53) will appear as follows, assuming the point of focal stress is "frothy":

Wotan said *something* (58)

Wotan said the water *is something* (59)

Wotan said the water is too *something* (60)

Wotan said the water is too frothy. (61)

If a lexical or syntactic ambiguity occurs, each possible interpretation is output. So, for example, "frothy" can have two senses: "covered with froth" or "without serious content". Thus, (61) would not be output as a single form, but rather as:

Wotan said the water is too $\text{frothy}_1$.
Wotan said the water is too $\text{frothy}_2$. (62)

If some linguistic ambiguity occurs at an early stage in incremental processing, the number of candidate forms may be manifold. Thus, SW postulate the presence of an *inhibitor* line, that can inform the linguistic modules when a particular constituent has been disambiguated. When the module is so informed, it only outputs the disambiguated form in future developments[4].

SW claim that the advantages of incremental processing are:

1. It is psychologically plausible.
2. It may help cut down the cost of processing ambiguities (see section 3.3.1).
3. It is used for the recovery of presuppositions.

### 3.1.4 What is Communicated by an Utterance?

For the sake of simplicity, I will make the assumption that an utterance has been understood when the speaker's communicative intention is realised. According to RT, this will occur when the hearer's cognitive environment has been suitably modified. How is such a modification realised? The relatively short-term memory components of the hearer whose contents constitute the context are altered. These memory components should contain all of the significant explicatures and implicatures that follow from the utterance according to the Principle of Optimal Relevance.

Thus, the interpretation of an utterance is the contents of the deductor memory and any other relevant memory components.

[4] I assume that the inhibitor line can relax its constraints on the language module in the case where it makes incorrect guesses concerning some selection between ambiguous representations; if not, processing would have to start again from scratch.

## 3.2 An Ideal Relevance-Based System

Enough detail has now been provided in order to make some concrete statements about the processes that must be present in the mind that take part in utterance interpretation. First, a basic schema for these processes will be outlined. Next, the interaction of these processes is illustrated using an example. Finally, the suitability of the Principle of Optimal Relevance for making linguistic choices during incremental interpretation is questioned and a slightly modified version suggested.

## 3.3 Schema for An Ideal Relevance-Based System

One possible arrangement of the components of a modular mind that processes utterances according to RT as discussed in this chapter and chapter 2 can be seen in figure 3.1.

I will provide brief details of each component:

General input systems: These are informationally encapsulated modules that provide information about the external environment in the form of assumption schemas. The only relevant path from them connects to the deductor memory. Presumably, information is being issued from them continuously, but can be selectively filtered before being input to the deductive device.

Language module(s): This is the input module which incrementally produces semantic fragments representing the meaning of a spoken utterance. An inhibitor line issuing from the deductive device (control section) disables redundant output.

Central systems: These are the less informationally encapsulated modules responsible for storing and reasoning with logical forms.

Conceptual memory: This is a long-term storage module. It is divided into chunks which are uniquely indexed by the (name of) a concept. Each chunk is further divided into logical, encyclopædic and lexical sections. These, in turn, contain meta-level information, object-level information, and information for the input modules. There must be paths which feed lexical information to the language modules (not illustrated), logical information to the logical memory of the deductive device, and encyclopædic information to the deductor memory. Information may be able to flow back from the deductor memory to the conceptual memory, but this is not covered by RT.

During context expansion, concept names are fed via a special line to the conceptual memory which in turn passes information from the associated conceptual entries to the deductive device.

Deductive device: This is the component of the central processes responsible for deductive reasoning. It may also be responsible for the instantiation of assumption schemas. It must consist of a *control section*, a logical memory, and the deductor memory.

Control section: This is responsible for co-ordinating the deductive device with the other germane central processes during communication. It consists of inference control, reason maintenance and context control sections. These latter two sections may be better considered to be distinct from the deductive device, since they perform extra-logical functions.

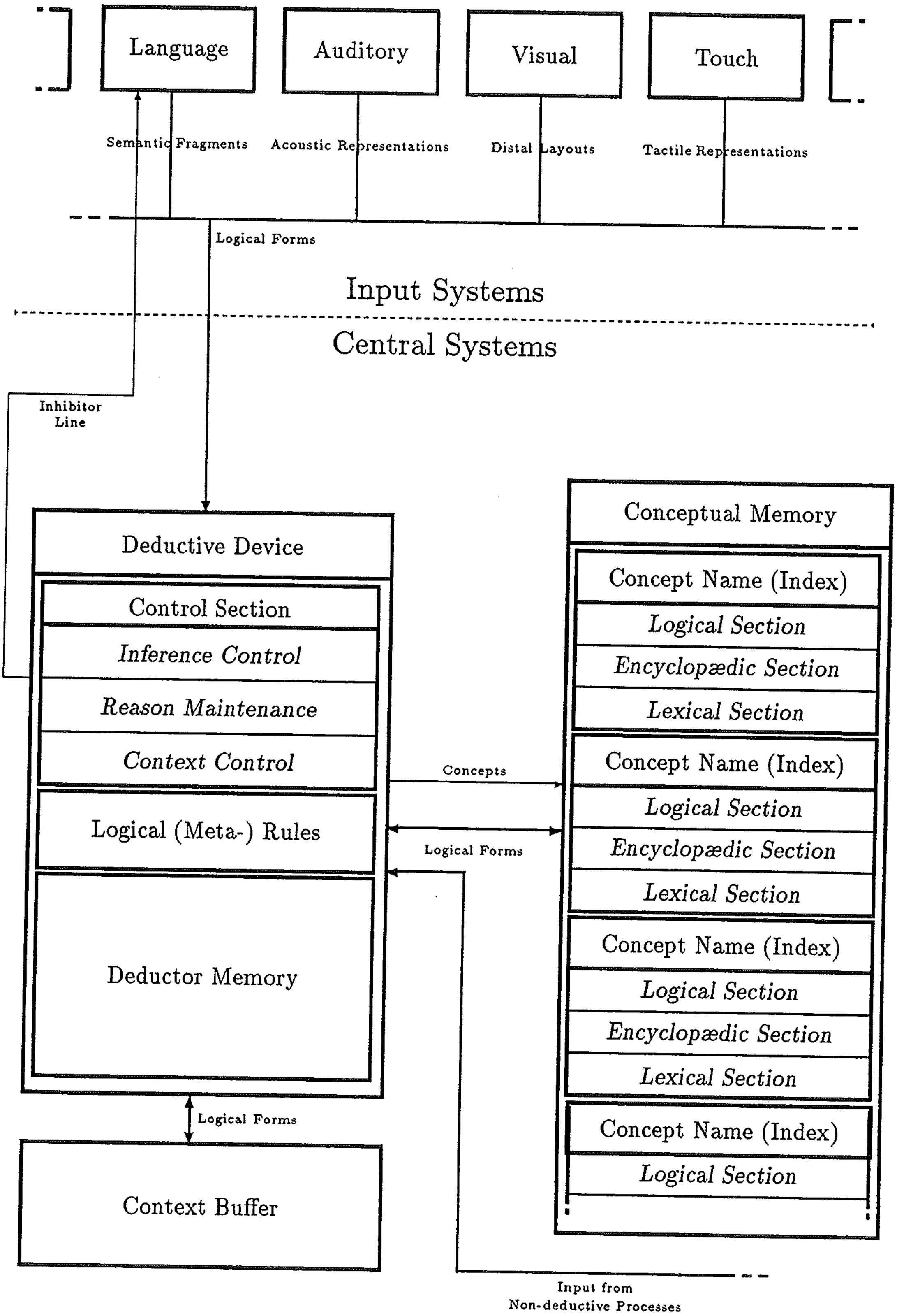



Figure 3.1: Schema for an Ideal System

Inference control section: This is responsible for taking the closure of forward-inferences in the deductive device; it does this by applying the logical rules to the encyclopædic information in the deductor memory. It must also be responsible for dependent and independent strengthening of this information. Finally, it must be able to calculate whether an inference constitutes a contextual implication, contextual strengthening or neither.

Reason maintenance section: This is responsible for maintaining consistency of the information within the deductor memory. The reason maintenance system must also keep count of the contextual effects due to erasures. This module is also responsible for recovering from incorrect guesses.

Context control section: This collects information about the contextual effects and effort expended during inferencing and calculates the overall Relevance. If this is not enough, it may use some of the concepts in the deductor memory to further index into the conceptual memory. It may also request more information from the context buffer or the input processes. Another round of processing can then proceed.

Logical rules: These are the meta-rules fed from conceptual memory and stored in the logical memory.

Deductor memory: This contains encyclopædic information, input from the conceptual memory, input processes or the inference controller. The non-trivial implications of its contents are added by the inference process. Assumptions within the deductor memory must contain associated information about their source and inferential status (i.e. whether or not they are contextual effects). Separate non-demonstrative inference processes may operate on this information and add hypotheses back in. Old information may be rejected from this memory to the context buffer, or drawn back during context expansion.

Context buffer: This consists of one or more memories that can be used to store the results of previous utterance processing. The control section can pass information to and from these memories, as dictated by the context control section. The same memories may be used to support context switching.

### 3.3.1 The Process of Ostensive-Inferential Communication

With reference to schema 3.1, it is possible to explain how the various central processes co-operate during communication of spoken language.

Spoken utterances are incrementally processed due to the operation of the language modules. Each development is processed by inferencing and by asking Relevant questions about its underdeveloped aspects.

Returning to example (53), the first schema that will be produced by the language:

Wotan said *something* (58)

The control section instructs the input processes to route this into the deductor memory. The context control section judges that the results are not Relevant enough, so it further requests information about "Wotan" and "saying", which is subsequently added from conceptual memory. The inference control section is then instructed to produce some inferences. During this process, Brünnhilde will ask herself what information may complete the utterance, for example:

What did Wotan say? (63)

Presumably, Brünnhilde answers this using *backwards* or goal-directed reasoning; SW have contradicted their own claims that only forward-reasoning is used during communication. At this stage of utterance processing, it is unlikely that she can answer this question. However, she will have brought in contextual premises about "Wotan" and "Say", eg. item b above.

Next, the controller requests a further development from the language module:

Wotan said the water *is something* (59)

which results in Brünnhilde asking herself the question:

What did Wotan say about the water? (64)

Context control adds information about "Water" to the context at this point and inferencing is repeated. Brünnhilde may use the backwards-chaining question to identify what "the water" is referring to. Under the circumstances that there are several potential water sources in Valhalla, Brünnhilde must try to select the most appropriate option. She will choose the first Relevant referent; if none can be found, the next development is processed. When she considers the water in the Jacuzzi, Brünnhilde will find it Relevant because she knows Wotan is in the bathroom and that he is making splashing noises. The water in the Jacuzzi will be most manifest because it is the only source that could make the magnitude of splashing noises that Brünnhilde hears. If Brünnhilde opted for, say, the water in the sink, she would later have to change her mind when she discovered that the water being referred to was frothy. This would be the responsibility of the reason maintenance process.

Gradually, an interpretation is built up from each development. When a linguistic ambiguity is presented to the deductive device, a tentative decision is made as to which is the best interpretation. Other interpretations in later developments are then disallowed through the use of the inhibitor line. If the interpretation is later proved to be inconsistent, the system must backtrack back to contexts where it made the fundamental choices of referent, disambiguated linguistic forms or introduced contextual premises. This is the job of the reason maintenance process[5].

(60) is similarly introduced, resulting in the loading of information about "too" followed by further non-trivial inferencing. Now, Brünnhilde asks the question:

What is excessive about the water in the Jacuzzi (and for whom)? (65)

Undesirable things about Jacuzzi water might include, "too hot", "too cold", "not frothy enough", "too frothy", "too dirty", etc. Thus, the options are narrowed to several equally manifest alternatives. If the first Relevant one is "too hot" it will be selected but overridden when (61) is loaded in.

Finally, each of the forms (62) are tried out in the context. The first Relevant interpretation is selected. Presumably, $\text{frothy}_1$, "having froth" is preferred over the "lacking serious content" because it results in contextual implications about Jacuzzis, which $\text{frothy}_2$ does not.

---

[5] This is entirely my own invention. However, SW do acknowledge that the hearer can make the wrong choices and that they must somehow recover from this.

By now, the developed representation of (53) might have been derived. Even so, the context expansion process may decide to further expand the context at this point because the utterance is not yet Relevant enough. This may result in the inference that, not only does Wotan want the water to be made less frothy, he wishes Brünnhilde would do this. In this way, an indirect speech act has effectively taken place, without any recourse to explicit principles of speech act theory.

#### Processing Problems

Some problems can now be identified with this processing scheme.

- What questions should be asked? In the question-answer phase of processing, SW provide no explanation of how the questions are derived.
- How are the questions answered? SW imply that contextual premises may be required to answer them. If this is so, the mechanism of question answering is complex and requires further description.
- How is consistency efficiently maintained? Since it is possible to make wrong choices throughout processing, consistency must be restored when necessary by some kind of backtracking process, which SW have not even mentioned.
- How Relevant is "Relevant enough"? Processing terminates when the utterance is Relevant enough, but no details are given as to the magnitude of this quantity. Furthermore, SW claim it can vary with circumstances. Unless, they are more precise about how it varies, there is a danger that any utterance can be judged Relevant vacuously.

### 3.3.2 Optimal Processing using Relevance

There are two similar strategies that can be employed when processing the representation of an utterance. The second of these is not endorsed by SW because, at face value, it appears less efficient. However, there are arguments as to why this second strategy might be more useful in some cases.

Consider the first strategy:

1. A search is made for the first Relevant putative interpretation. Heuristics can be used to guide this search so that certain interpretations are chosen over others[6]. For example, the most salient candidate referent could be chosen, rather than checking some unordered set for the first Relevant utterance. Since these are only heuristics, they may result in unnecessary processing effort under some circumstances.

The first strategy operates according to the Principle of Optimal Relevance; this is reasonable when selecting between candidate referents because, invariably, exactly one candidate will be the most salient. However, there is not a similar counterpart of salience for selecting between linguistic ambiguities in a partially processed utterance[7]. Assuming that the first Relevant word sense chosen is correct seems unwarranted optimism.

Consider SW's own example:

[6] SW do not explicitly mention this, but I feel it is a reasonable assumption.

[7] However, if the full utterance is considered, it is possible to use other words to bias the sense selection process by considering which word is most "related".

Jennifer admitted stealing. (66)

This will have a development which will be something like:

Jennifer admitted *something.* (67)

"Admitted" has two senses. $\text{Admitted}_1$ is to state the truth of something. $\text{Admitted}_2$ is to permit to enter.

Consider a context where Jennifer is in her mother's house, about to confess to stealing, when a policeman rings the doorbell. She is not sure whether she should let him into the house or first explain the situation to her mother. There are two, equally plausible continuations of (67):

Jennifer [$\text{admitted}_1$ stealing] [to her mother] (68)

Jennifer [$\text{admitted}_2$ the policeman] [into the house] (69)

Both of these are Relevant and, possibly, equally manifest continuations. According to SW, if a rational hearer has any problems deciding on the sense of "admit", this is a mistake on the part of the speaker. The speaker should have phrased the utterance in a less ambiguous manner. This seems acceptable only in cases where a single interpretation is the most manifest. At early stages of incremental processing, such an interpretation is less likely to occur. (66) is intuitively a reasonably phrased utterance for either continuation — we do not seem to have any problems in understanding it.

It is quite possible that, from development (67), one continuation may be more Relevant than the other, but there is no guarantee that it will be selected; the *first* Relevant interpretation will be chosen. Now, in the case of multiple candidate referents, there are obvious heuristics for selecting the most Relevant possibility. However, there are no such heuristics for word senses. If the most common sense is tried first, it would mean that the less common senses should never be used when all senses are Relevant. So, for example, (69) should not be a viable continuation because it is Relevant yet uses a less common word sense. However, it appears to be pragmatically well-formed; we do not have problems understanding it.

Another problem with opting for the most common Relevant word sense is that it may result in unnecessary processing effort. This effort is incurred when attempting to correct mistaken predictions of word sense caused by taking the first Relevant interpretation.

An alternative means of deciding on word selections is to select the *most* rather than the *first* Relevant interpretation. There is thus another, more fail-safe processing route for linguistic ambiguities, which is:

2. Process each putative interpretation in the context; select the most Relevant one. If several candidates are maximally Relevant, do not make a choice at this point.

One reason for incremental semantic interpretation is to cut down the number of possible interpretations. Consider an utterance with three lexically ambiguous words. This first word has $L$ senses, the second $M$, and the third $N$. If all possible word sense permutations were substituted into the whole form, $L.M.N$ possible interpretations would be processed.

Using method 1 above, there would usually be a maximum of $L + M + N + (\mathit{length} - 3)$ interpretations, where *length* is the length of the utterance. This would be the case when only the last of each the alternative senses proved Relevant[8]. In the best case, there would be only *length* interpretations because for each word the first sense found would be Relevant.

Using method 2 above, a minimum of $L + M + N + (length - 3)$ interpretations will be processed.

These figures would suggest that method 1 should always be employed. However, they do not take into account the fact that method 1 may require huge amounts of processing effort to undo the effects of the bad choices it made earlier on.

## 3.4 The Design and Operation of Crystal

The design of Crystal is a very simplified and slightly modified version of the ideal system presented in section 3.1. In this section, I justify the differences between the two systems. A schema for Crystal is then presented and explained. Its correspondence to the ideal system will be made explicit.

### 3.4.1 The Domain and Coverage of the System

#### Language

Much as RT is intended to cover all cases of ostensive-inferential communication, it is essentially a linguistic theory. Most of SW's descriptions are of language comprehension. Thus, Crystal will only deal with language. Furthermore, Crystal only processes forms derived from English sentences, although in principle it could process logical representations of any language (if the conceptual memory entries were appropriately changed).

#### Input Modules

Crystal does not implement any input modules. Complete assumption schemas representing utterances are delivered directly to the deductive device. Assumption schemas representing discourses are input to the the system using special editors (see appendix A).

Since RT only concerns itself with the functionality of the language modules, there is no need to implement the other modules. The only important aspects of the language modules are:

- The incremental nature of their output.
- The fact that they deliver assumption schemas to the central processes.

Since assumption schemas are delivered to the central processes in Crystal, I need only justify my omission of incremental interpretation.

The incremental aspect of the language modules is important for:

- The recovery of presuppositions.
- Efficiency of processing ambiguities.
- Cognitive plausibility.

[8]There is, in fact, an unlikely worst case where no decisions can be made for the word senses until the last word of the utterance is enriched because none of previous developments prove Relevant enough to make any lexical choices.

- Correct processing order for certain constructs (see chapter 7).

The idea behind Crystal's design was to experiment with some of the ideas of RT in a small processing system, not to build a practical language processor; hence, efficiency was not an important design criterion.

Since presuppositions are recovered via incremental processing, it is useful to have a cognitive justification for this processing strategy. Crystal does not attempt to recover presuppositions so cognitive plausibility is no longer an issue.

Processing order is critical for explaining certain linguistic phenomena. Crystal simulates the appropriate incremental processing by using a left to right inferencing strategy. This is appropriate for all English sentences where the focal stress is on the last word.

Thus, the input modules can be omitted from Crystal because:

1. Presuppositions are ignored.

2. Efficiency is unimportant.

3. The assumption schemas input to the central processes can be argued to be similar those that might be output by the language modules.

4. Only utterances with focal stress on their last word are considered, and Crystal processes their representations in left to right order. Re-ordering the logical representation of the utterance can simulate processing forms with foci in different positions.

### Context Maintenance

Although SW do not explain in any depth how information moves between the deductor memory and the context buffer, it is a vital part of discourse processing.

If discourse segments are not linear or are of reasonable length, it becomes necessary to shunt information between the two memories. The same is true for intertwined discourse threads. Thus, to avoid the use of a context buffer, Crystal only processes short linear discourse segments.

### Domain

Crystal's domain is that of simple children's stories. This is for the following reasons:

1. The domain is simple enough to apply existing technology.

2. Related computational systems are available for comparison.

Current knowledge representation and inference techniques are nowhere near a level of sophistication that could cope with general language. Thus, the choice of children's stories has been popular with AI researchers for several years (see, for example, Charniak (1972), Wilensky (1978), Schank and Riesbeck (1981)). Even though current technology can barely cope with even these domains, children's story understanding is a relatively tractable problem because of the relatively straightforward plans and goals of the individuals described by the discourse.

There are already several good story comprehension systems in existence, which are compared and contrasted with Crystal in chapters 8 and 9.

#### Linguistic Coverage

Crystal is largely limited to the understanding of simple, declarative sentences. It has little or no coverage of other speech acts, presuppositions, tropes etc.

### 3.4.2 Features of Crystal

#### Theoretical Support

It will be seen in chapter 7 that Crystal can effectively process certain language phenomena because it adheres to RT. In principle, it can be extended to handle many more using exactly the same principles.

#### Clarity of Operation

Crystal generally draws on technology that has already been given theoretical underpinnings, such as non-monotonic inference engines and reason maintenance systems, which are theoretically supported by non-monotonic logics. The processing algorithm is fixed, working according to the Principle of Relevance; the user can only "tune" the system by changing the data (in the form of conceptual knowledge and short discourse fragment representations) in order to modify the behaviour of the system.

#### State-of-the-art Technology

Some of the most recent developments in knowledge representation and inferencing are used in Crystal. This enables these representations to be evaluated in a practical system. Novel uses for some of these representations and mechanisms are demonstrated. In particular, I show several uses of non-monotonic logic in language processing.

#### Contribution to RT

Many of the problems concerning RT uncovered in chapter 2 and this chapter have been overcome in the implementation of Crystal. To my knowledge, Crystal is the first computational system to be based on RT.

### 3.4.3 Schema of Modules within Crystal

Crystal's design shown in figure 3.2 is based on schema 3.1. A brief description of each module follows.

Input Systems: The sole input to the system is derived from an *utterance editor*, which feeds the deductive device with logical forms via the interface.

Utterance Editor: An utterance editor stores a sequence of logical forms representing a simple discourse. Each form is, in principle, derivable using compositional semantics from a syntactic representation of its associated utterance. Before a form is input to the central processes, it is converted into a canonical representation suitable for the deductive device. I call this conversion process *normalisation*.

Central Processes: I have rationalised the organisation of the central processes shown in schema 3.1. The central processes now consist of four major components:

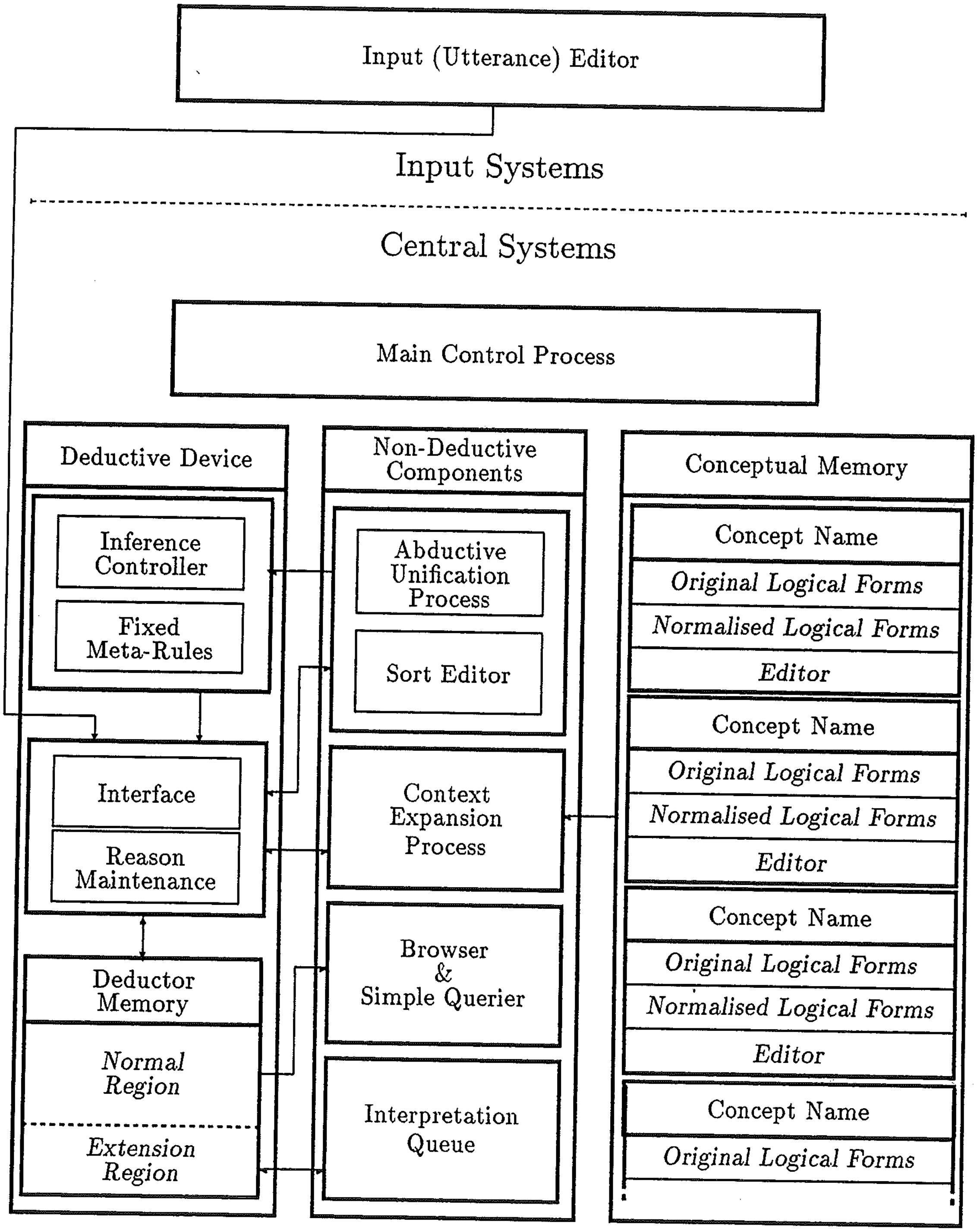



Figure 3.2: System Perspective of Crystal

1. The *main control process*. This coordinates the other modules according to the Principle of Relevance (control connections are not illustrated.).
2. The *deductive device*. This is responsible for deductive inferencing. Certain kinds of defeasible inferencing are included here.
3. The *non-deductive components*. These include a non-demonstrative inference generator, and various other essential components.
4. The *conceptual memory*. This serves the same function as it does for the ideal system.

Inference controller: This controls the application of the logical rules to the information in the deductor memory. However, it draws on a fixed set of logical (meta-) rules contained within the deductive device. The meta-rules are in the deductive device for reasons of modularity outlined in chapter 2. The reason for having a fixed set is explained in chapter 4. Information is fed to the controller via the interface and the abductive unification process.

Reason Maintenance: This maintains the consistency of information within the deductor memory using a *reason maintenance system*. Its operation will be discussed in chapters 7, 8 and 9.

Interface: The interface is a conglomeration of features from several modules in the ideal schema. All transactions with the deductor memory must go via the interface. It may simplify the data before merging it with the memory. The interface maintains all data structures within the deductor memory. Since all contextual implications, strengthening and erasures must be effected via the interface, it can keep count of the information and effort change in the system. It will be further discussed in chapters 4 and 7. When merging information into the deductor memory, the interface works in conjunction with the abductive unifier to carry out the questioning phase of utterance interpretation.

Deductor Memory: This is as for the ideal system. Information is stored within it in the form of special nodes for the reason maintenance process. The memory capacity can be temporarily expanded during context expansion; this is the purpose of the extension region.

Abductive Unification Process: This serves two different roles. On the one hand it is a simple pattern matcher used by the inference controller. On the other hand, it is a non-demonstrative inference generator with access to the deductor memory, via the interface. The latter component is used to form non-demonstrative hypotheses necessary to provide an answer for the questioning phase of utterance interpretation. Further details are provided in chapter 8.

Sort Editor: This allows for the creation of *sorts*, similar to a typing mechanism, for controlling the inference process. All logical forms within the system are sorted using sorts drawn from a hierarchy constructed using this editor. Only the unifier makes use of these sorts. More detail is provided in chapter 5.

Context Expansion Process: If the main control process decides that the context needs to be expanded, it passes some logical forms from the deductor memory (via the interface) to the context expansion process. This finds the set of concepts contained in these forms and instructs the conceptual memory to add information about them to the deductor memory (via the interface). This is described in more detail in chapter 4.

Interpretation Queue: This is a medium-term memory capable of storing previous contexts. It is used when attempting to find the most Relevant interpretation of an ambiguous utterance. It can also be used for interpreting certain linguistic constructs, such as "but" (see chapter 7). It is described more fully in chapter 4. There is no direct correspondence with any component of the ideal schema, but it is a necessary component of a Relevance-based system (see chapters 4 and 7.

Conceptual Memory: This only holds encyclopædic information, and is thus also known as the *encyclopædic memory.* For the sake of modularity, linguistic information would be stored with the input processes and logical information stored in the deductive device (see chapter 2). The conceptual memory contains entries which can be retrieved by using their associated concept name as primary key.

Each conceptual memory entry has an associated editor, which allows input of encyclopædic information concerning the concept name. The editor normalises this information, maintaining the unconverted original data for reference purposes. Details of the encyclopædic knowledge representation are found in chapter 5.

During processing, the conceptual memory will take a list of concepts from the interface and pass back their associated encyclopædic entries.

Browser and Querier: This is not part of the understanding system, but is a passive inspector and querier of the deductor memory.

### 3.4.4 Crystal's Processing Strategy

A major claim of RT is that the most Relevant interpretation of an utterance is the correct one. SW augment this by claiming that the first interpretation found should be the most Relevant one.

The first claim can be tested by processing several different interpretations of an utterance and checking if the most Relevant one is intuitively correct. As a practical means of processing, this has some obvious disadvantages, but it enables the notion of Relevance to be evaluated.

The second claim is effectively arguing that the means for selecting a Relevant interpretation are more or less deterministic. A practical implementation of RT would employ appropriate interpretation selection rules or heuristics[9]. In section 3.3.2, I argued that it is difficult to find suitable ordering heuristics for eliminating lexical/grammatical ambiguities, and that trying out all of the possibilities in these cases may not be too inefficient. Employing a similar strategy for pragmatic ambiguities is not so practicable because there are many more ways an utterance can be enriched from a particular context, and notions of manifestness seem more appropriate here.

Therefore, Crystal's processing strategy is a follows:

- Given a linguistically ambiguous utterance, all possible realisations are tried in the available context and the most Relevant interpretation selected. If this strategy

[9]SW have not explicitly pointed this out, but it follows from the Principle of Optimal Relevance.

was implemented incrementally, it might result in better efficiency than that of its alternatives.

- For reference resolution and other kinds of enrichment, the first consistent Relevant interpretation is selected using various heuristics for evaluating manifestness.

In what way can Crystal be said to have understood anything? In the ideal system, the context is maintained by the short-term memory components of the system. It is the contents of these memories which constitute understanding. In Crystal, the only such memory component is the deductor memory (as will be explained in more detail in chapter 4). The contents of this memory are therefore what has been communicated by one or more utterances. What of significance will be in this memory? The enriched, disambiguated utterance and various implicatures and explicatures that it induces. The less obviously important remainder of the information will be that which was already present or that which was accessed during context expansion but not used. In order to evaluate Crystal's capabilities, the processed utterance and all of its demonstrative and non-demonstrative implications must seem reasonable to a human observer. Reasonable word senses must be chosen, correct reference resolution must occur, and hypotheses must be plausible.

### 3.4.5 A Brief Example

Let us consider how Crystal might process a discourse fragment. Reconsidering the discourse provided in chapter 1, we can examine how Gricean-style inferences can be produced without appeals to the maxims. The exact mechanisms by which these inferences are produced are the subject of the remaining chapters.

Nutkin felt hungry. (70)

He found a nut. (71)

Let us assume that (70) has already been processed and its non-trivial implications are present in the deductive device, including the fact that:

Nutkin wants something to eat. (72)

and

Nutkin is a squirrel. (73)

The latter is a defeasible inference drawn from stereotypical information indexed by the name "Nutkin".

(71) would be processed as follows: "nut" is ambiguous; "$\text{Nut}_1$" is the food with a kernel and shell, "$\text{Nut}_2$" is the metallic object often used for fastening objects.

Thus, two forms will be separately processed in the context invoked by (70):

He found a $\text{nut}_1$. (74)

He found a $\text{nut}_2$. (75)

(74) is processed first. The normalised representation of the form would be loaded into the deductor memory. Next, information about "Male", "Find", and "$Nut_1$" would be loaded from encyclopædic memory by the context expansion process. Inferencing would then begin, derived from (74) by processing it in left-right order.

One of the first inferences drawn would be the analytic implication of (71) that "he" is a male, referring expression. The abductive unification process guesses that "he" refers to Nutkin. Thus it is also inferred that Nutkin is male. This defeasible guess is asserted into the deductive device, and inferencing continues. Inferences about Nutkin finding things will be made next, because finding is the next word in the utterance. For example:

If Nutkin has found something, then he knows where it is. (76)

Eventually, the "$nut_1$" part of the utterance will be processed. Amongst other implications, it will be inferred that:

The $nut_1$ is food. (77)

Now, the interface and abduction system ask the question[10]:

What is done with the food? (78)

The abductive unifier guesses that it is the food that Nutkin wants to eat. When all of the non-trivial inferences have been found, the process stops. In this case, the main controller judges the result to be Relevant enough.

The contents of the deductor memory are preserved on the interpretation queue, and the context invoked by (70) is restored. Now (75) is loaded into the deductive device, along with encyclopædic information about "Male", "Find" and "$Nut_2$". Inferencing is then repeated. This time, less contextual effects occur because Nutkin cannot eat the nut as it is not food.

The Relevance of this interpretation is compared to the Relevance of the "$Nut_1$" interpretation at the top of the interpretation queue. The Relevance of the "$Nut_1$" interpretation is found to be higher so the context restored from the queue.

To illustrate the function of the reason maintenance process, consider the effects that the following has on the contents of the deductor memory after processing (71):

Nutkin is a penguin. (79)

The defeasible inference (73), and everything that depends on it, is immediately cancelled. Information about Penguins includes the fact that they only eat fish, which is inconsistent with the guess that Nutkin will eat the nut. The reason maintenance process is invoked, and decides that the best recovery plan is to remove the guess. The deductor memory is then updated. In this case, it is the fact that (79) overrides an expectation resulting in contextual erasures that makes it Relevant.

A full transcription of the above example can be found in appendix B.

---

[10]The question asking is automatically carried out every time the inference controller attempts to draw an inference or the interface merges an assumption with the deductor memory. The mechanics of this process are described in detail in chapter 8.

### 3.4.6 Summary

Much as a formal correspondence cannot be proved, it should be apparent that Crystal works using a scaled-down version of RT. I have argued that my omissions will not affect the fundamental functionality of the ideal system and that the reconfigurations are theoretically justified.

The remaining chapters detail the precise interaction and design of the modules depicted in schema 3.2. The next chapter will describe and justify the basic control algorithm which drives the deductive device and co-ordinates it with the other central processes. After that, the encyclopædic knowledge representation is described with particular attention to problems and constraints that bear on it due to RT. It will then be possible to discuss the measurement and utilisation of Relevance (chapter 6), and subsequently consider the other system processes in more detail.

# Chapter 4

# The Deductive Device and the Context

This chapter details the operation of the deductive device, the context expansion process and the main control process which co-ordinates them.

The deductive device performs the following functions:

1. It can form the closure of forward deductions in the deductor memory.
2. It maintains the strength of assumptions in the deductor memory.
3. It instantiates assumption schemas.
4. Its memory constitutes the context.
5. It makes some quantitative evaluation of Relevance.
6. When necessary, it performs belief revision.

In this chapter, I discuss items 1–4 in detail. I will discuss the remaining items in chapters 6 and 7.

The context expansion process is responsible for feeding further conceptual information into the deductive device when an utterance representation is not yet Relevant. The main control process coordinates the deductive device and context expansion processes on the basis of some quantitative Relevance measure.

The structure of this chapter is as follows. The logical rules which express how the deductive device can rewrite encyclopædic information are discussed. It is concluded that it is possible, provided some modifications are made to SW's account of the rules, to formally describe the inferential behaviour of the deductive device. I then show that, if the logic employed by the deductive device is as powerful as FOPC, it is necessary to place extra constraints on the deductive device or the knowledge representation in order to ensure that the inference algorithm terminates. After this, I enumerate some of the critical issues for the inference and context expansion processes before I provide a detailed discussion of their algorithms.

Crystal's forward-chaining inference algorithm is portrayed and justified to some extent. The functionality of the interface, the process that forms a boundary between the deductor memory and the other processes, is then described. Next, the context expansion process is discussed and explained. Finally, the simple algorithm by which the main control process co-ordinates all of these processes is depicted.

The components of the central systems that I describe in this chapter are not, on the whole, constructed using novel technology. It is their co-ordination by the main control process using the Principle of Relevance which is of interest. It will be seen that many vital details, left vague by SW, have been specified in order to implement these processes.

## 4.1 Logical Rules

In this section, I will discuss SW's notion of logical rules. It will be seen that numerous problems that are incurred if their ideas are implemented directly. However, these problems do not occur if their logical rules are just taken to be the inference rules of a traditional logic. An important property of many logics, including propositional calculus and FOPC, is *monotonicity*: adding premises to a theory can never result in a decrease in the number of resulting theorems. In contrast, logics of defeasible reasoning can be *non-monotonic* (see chapter 7). The logic I employ in the deductive device is non-monotonic, supporting defeasible reasoning. However, in this chapter I will only refer to the monotonic aspects of the inference engine. This component is not altered by the addition of non-monotonicity, which is supported by the separate reason maintenance process.

The employment of a conventional logic in the deductive device contravenes SW's informal definitions of logical and encyclopædic information. However, the decision to utilise such a logic was made in order to overcome some fundamental flaws in SW's categorisations. Logical rules determine what follows from encyclopædic information in the deductor memory, and therefore what can constitute a contextual effect[1]. SW's logical and encyclopædic information do not form a logic in the traditional sense because the logical rules present in the deductive device are allowed to vary (Goldblatt, 1987) and can even be treated as encyclopædic information. Their inference system is more akin to Markov algorithms, which have the general computing power of Turing machines. Thus, without further constraints on the form and function of logical and encyclopædic information, all SW have said about the deductive device is that it is an arbitrary computing machine that can produce arbitrary contextual effects depending on its configuration.

In this section, I will describe various problems inherent in the way SW carve up logical and encyclopædic knowledge. These are basically the following:

- Logical rules can be converted into encyclopædic information. A consequence of this is that the definition of contextual effects is inadequate and therefore Relevance is a vacuous measure.

- Since the logical rules are allowed to vary, the behaviour of the deductive device is unpredictable.

The notions of analytic and synthetic truths have posed problems for philosophers for centuries, and I have already indicated in section 2.4.2 that SW's speculations concerning them appear to be misguided. Furthermore, they make no use of these concepts in their theory. I will show in sections 4.1.1 and 4.1.2 that SW's conjectures concerning the nature and distribution of logical and encyclopædic knowledge are inadequate, and therefore all that they can be said to have uncontroversially said about the deductive device is that it is a general computing engine. These deficiencies are overcome by making the deductive

[1] I therefore refer to logical rules as being at a higher "level" to encyclopædic rules because they operate on them.

device an inference engine, ideally supporting a logic of human knowledge (but necessarily rather restricted in Crystal). The logical rules become the inference rules of the logic, and everything else is considered to be encyclopædic information. This new knowledge structure no longer endeavours to address the irrelevant problem of analytic and synthetic truths. An outline of the argument is:

1. There must be some way of determining where conceptual information should be placed.

2. Logical and encyclopædic information should be treated as separate, non-interchangeable types.

3. The set of logical rules should be of finite, bounded size.

4. The previous three claims along with the fact that the deductive device should be performing *deduction* suggest that logical rules should be treated as the inference rules of some logic, and encyclopædic information should be treated as the premises for a theory in this logic.

The first point speaks for itself. The remaining points will be individually argued in the forthcoming sections.

### 4.1.1 Interchangeability of Conceptual Information Types

Since SW provide only the vaguest description of what constitutes logical or encyclopædic information (see section 2.4.2), there is no precise way of judging the level at which a particular datum should be put. Furthermore, they claim it is possible to treat logical rules as encyclopædic information. In this section, I will show that this is indeed the case, but if allowed, it results in catastrophic problems for RT. The solution will be to make logical and encyclopædic information distinct categories.

In order to illustrate the problem, I will show that when a logical rule that licenses analytic inferences is converted into an encyclopædic assumption, it licenses synthetic inferences. Thus an analytic rule that cannot possibly yield a contextual implication can be converted to an encyclopædic rule that can. It follows that it is possible to organise the conceptual information in the deductive device in such a way that almost *any* utterance can be made Relevant by distributing the rules to appropriate levels in order to make them yield contextual effects when triggered.

Let us assume that the logical rules support a subset of FOPC. Let the contents of the deductor memory be $\Delta$. The non-trivial implications of $\Delta$ will be every $\phi$, such that $\Delta \vdash \phi$. The logical rules determine the behaviour of $\vdash$. Using these definitions, I will show that, at least for rewriting systems equivalent to FOPC, logical rules can be expressed as encyclopædic information. From this it will follow that analytic implications can be expressed as synthetic implications, therefore the notion of contextual implication is arbitrary and Relevance void.

The *deduction theorem* of FOPC states that, if $\Delta \cup \{\psi\} \vdash \phi$, then $\Delta \vdash (\psi \supset \phi)$. Informally, if $\psi$ is added to the deductor memory, and $\phi$ follows from it by the application of some logical rules, then we could rephrase this inference in terms of material implication at the encyclopædic level. Thus, assuming a finite domain, any logical rule could be rewritten at the encyclopædic level by rewriting all of its instantiations in terms of material implication.

As a consequence, meta-rules that yield analytic implications, yield synthetic implications if migrated to the encyclopædic level and vice versa. The deduction theorem allows analytic rules to be rewritten in terms of an encyclopædic datum; this datum will yield inferences when used as the major premise input to the synthetic rule of modus ponens. For example, consider the utterance representation:

$$\text{Valkyrie}(\textit{Brünnhilde}) \tag{4.1}$$

There may be a logical "Valkyrie-elimination" rule that would yield the assumption:

$$\text{Female}(\textit{Brünnhilde}) \tag{4.2}$$

In this case, (4.2) is not a contextual implication since it was derived solely from the utterance.

On the other hand, consider the encyclopædic fact that constitutes the migrated Valkyrie-elimination rule:

$$\forall x.\, \text{Valkyrie}(x) \supset \text{Female}(x) \tag{4.3}$$

Now, the synthetic logical rule of modus ponens can be applied to (4.1) and (4.3) to yield (4.2). This will be a contextual implication, since it was derived from the utterance and the context.

Thus, the notion of contextual implications is imprecise. It is arbitrary as to whether an inference is a contextual effect or not. Consequently, the measure of Relevance is not necessarily very informative. The crux of this problem is that inference rules and object level information can be interchanged. If logical and encyclopædic information are made distinct, unalterable entities, the problem no longer exists because the deduction theorem cannot be applied. Therefore, meta- and object-level information should not be interchangeable[2].

### 4.1.2 Using a Fixed Set of Meta-Rules

I have already explained in section 2.4.3 that logical rules should be associated with the deductive device rather than the conceptual memory. In this section, I argue that the set of logical rules should be fixed for the following reasons.

#### Semantics

The fact that logical rules are not defined by SW, and in any case are allowed to vary, means that the reasoning system has not been provided with any semantics. In fact, the behaviour of the deductive device will depend on its processing history. A theory which places such weight on the notion of information must surely be precise as to what actually constitutes information.

Because they can vary, logical rules can be arranged so that any datum is a contextual implication. If this is allowed, any example can be argued to provide the necessary effects and the theory of Relevance becomes totally vacuous.

Even assuming that every logical entry was given semantics and the logical memory allowed infinite capacity, there is no guarantee that the combined logical rules are mutually consistent.

[2]Alternatively, some very restrictive conditions under which meta- and object-level information are allowed to move might be formulated.

Many AI researchers attempting to provide a logic of human reasoning have felt the need to abandon FOPC in order to handle modal concepts like belief, time, action, possibility and defaults. All of these logics are not necessarily compatible: it is not always possible to just merge every meta-rule to yield a consistent theory. However, this is exactly what SW are suggesting. Each conceptual entry is an individual theory with its own meta-rules. It seems unwarranted optimism to believe that the union of all of these rules are consistent. Although it is conceivable that human beings may have conflicting encyclopædic beliefs, surely they cannot simultaneously use incompatible rules for actually understanding concepts.

### Avoiding Redundancy

Rules can be expressed at the encyclopædic or logical level. Duplication of the rules at both levels might result in misleading calculations of contextual effects. One method of avoiding such redundancy is to fix the set of logical rules. If this is done, it is clear which rules need not ever be expressed as encyclopædic information.

### Boundedness of Logical Memory

There is a consequence of SW's approach that they do not discuss at all: if the deductor memory is of bounded, finite size then presumably the memory that contains the meta-rules (the logical memory) is also of finite and bounded size. As a result, it must be subject to exactly the same rejection principles as the deductor memory.

SW point out that ignorance of encyclopædic information amounts to a lapse of memory; ignorance of logical rules amounts to a lack of *understanding*. Much as it is reasonable to reject encyclopædic information about a concept as it becomes less manifest, it is much less reasonable to claim we simultaneously lose our understanding of it.

If useful logical rules were rejected from a cognitive environment due to space constraints, vital inferences may not be made. As an example, consider some logical rules for understanding 'in':

*In-transitivity:*
<u>**Given:**</u> In$(A, B)$
<u>**And:**</u> In$(B, C)$
<u>**Yield:**</u> In$(A, C)$

and

*In-Containment:*
<u>**Given:**</u> In$(A, B)$
<u>**Yield:**</u> Contains$(B, A)$

It is easy to envisage sentences that cannot be understood if even one of these rules were not present. Consider the following:

Salome carefully placed the head in a box,
then put the box in a polythene bag. (80)

She put the bag into a freezer. (81)

In order to understand (81) given a context of (80), it is necessary to understand that, if something that contains an object is put in a freezer, the contained object will be preserved. Both the *In-Containment* and the *In-Transitivity* rules are necessary to deduce

The polythene bag contains the head. (82)

from utterance (80).

If (81) resulted in any rule concerning "in" being rejected, it would not be correctly understood.

Thus, logical rules should be fixed and non-rejectable. A good test of whether a rule should be expressed at the object or meta-levels is whether it can be sensibly be omitted from the deductive device. Typically, the rules concerning the concepts common to most forms, like "$\wedge$" and "$\supset$", cannot be rejected.

This approach has been widely adopted by AI researchers, especially in the realm of expert systems (e.g. Davis et al. (1977), Davis and Buchanan (1977), Buchanan and Duda (1982), Hayes-Roth et al. (1983), Weyhrauch (1985)).

### 4.1.3 Summary: Employing a Logic

The removal of the traditional notions of logical and encyclopædic rules only results in unnecessary complications: the semantics of the system are inadequate; it is impossible to define contextual implication satisfactorily; and it is undesirable to reject meta-rules.

The solution to this problem is to provide the deductive device with a predetermined, invariant set of logical rules. However, what should these rules be? The only suitable candidate is the inference rules of some logic, which describe the meaning of the concepts common to most assumptions. Thus, the logic should handle logical connectives, beliefs, desires, intentions, time, events etc. More specific data, including meaning postulates, stereotypical information, and information about particular instantiations, belongs to the encyclopædic entries. The issue of what should be treated as logical rules is again examined in chapter 5.

This partitioning of conceptual information has been implemented in Crystal and appears to be adequate.

### 4.1.4 Meta-Rule Structure

In chapter 9, I will argue that the inference process can be bounded without restricting logical rules to be elimination rules by using notions of confidence. In the current version of Crystal, I have still utilised elimination rules at the meta-level.

The meta-rules express a subset of the inference rules of FOPC. FOPC was selected as the logic because:

- Most established linguistic theories of semantics utilise FOPC or some superset of it.
- Its model and proof theoretic properties are well-understood.

FOPC is not an ideal logic of common-sense reasoning, and in chapters 5 and 7 the deductive device is augmented to handle a stronger reasoning system which copes better with common-sense reasoning. However, this will not affect the meta-rules described below.

As explained in section 2.4.2, logical rules can be either analytic or synthetic elimination rules. Analytic rules accept a single piece of encyclopædic information and return another; synthetic rules accept two pieces of encyclopædic information and return another. It is only through synthetic rules that contextual implications can ever be established.

Since the deductive device expresses a traditional logic, SW's appeals to philosophical notions of analytic/synthetic truths and logical/semantic truths are no longer appropriate. In Crystal, logical rules provide meaning for the logical connectives. Everything else is expressed at the encyclopædic level. Analytic implications generally serve to break down encyclopædic knowledge; synthetic implications serve to combine it.

The meta-rules provide the typical axioms of FOPC plus an additional rule for detecting inconsistencies.

### Analytic Rules

Analytic rules consist of three fields: an identifier, a trigger and an inference field. The identifier field is used by the reason maintenance process, but does not take part in the inference process. If the inference controller can find an assumption that matches the trigger, it can draw the deduction described by the inference field.

A typical analytic rule is "and-elimination", expressed in the system as:

**Identifier:** And Elimination (a)
**Trigger:** $!A \wedge !B$
**Inference:** $!A$

where lexical items preceded by "!" are variables which can match an arbitrary expression.

The above expresses the fact that the rule identified by the string "And Elimination (a)" can be applied to any assumption or schema of the form $!A \wedge !B$ and will yield $!A$ (since $!A \wedge B \models A$[3]). For example, the and-elimination (a) applied to:

$$\text{Mammoth}(tina) \wedge \text{Woolly}(tina) \tag{4.4}$$

yields:

$$\text{Mammoth}(tina) \tag{4.5}$$

The process of matching variables to logical forms uses the *unification* algorithm common to most logical inference systems[4] (Robinson, 1965; Nilsson, 1982).

Because the inference controller is not aware of commutative operators (such as $\wedge$ and $\vee$), two versions of the rules that involve these operators must be supplied, one for each operand [5].

Thus, for symmetry, there must be another and-elimination rule that can yield the other conjunct:

**Identifier:** And Elimination (b)
**Trigger:** $!A \wedge !B$
**Inference:** $!B$

[3]'$\models$' can be read as "semantically entails".

[4]In fact, Crystal's unifier has been modified to handle sorted assumptions (see section 5.4) and abduction (see chapter 8).

[5]It would be possible to allow the inference controller to commute such operators automatically or allow certain rules to yield several results. However, these do not change the fundamental properties of the deductive device, and would only be useful for increasing efficiency.

When and-elimination (b) is applied to (4.4) it yields:

$$\text{Woolly}(tina) \tag{4.6}$$

#### Synthetic Rules

Synthetic rules are similar to analytic rules except that they have two triggers. Both triggers must be matched to fire a rule. Consider:

**Identifier:** Disjunctive Modus Ponens (b)
**Trigger 1:** $(!A \vee !B) \supset !C$
**Trigger 2:** $!B$
**Inference:** $!C$

Given:

$$\text{Curried}(!X) \vee \text{Chillied}(!X) \supset \text{Hot}(!X) \tag{4.7}$$

and:

$$\text{Chillied}(vindaloo2) \tag{4.8}$$

the inference

$$\text{Hot}(vindaloo2) \tag{4.9}$$

can be made with the assistance of the unification process.

#### Detecting Inconsistencies

Inconsistency is detected by a meta-rule. This meta-rule passes a special type of assumption, labelled "INCONSISTENT", to the interface which then hands control over to the reason maintenance process to resolve the problem. The rule to detect inconsistencies is:

**Identifier:** Inconsistency
**Trigger 1:** $\neg !A$
**Trigger 2:** $!A$
**Inference:** INCONSISTENT$(!A, \neg !A)$

### 4.1.5 A Logic-Based Description of the Deductive Device

The deductive device can now be described formally. The contents of the deductor memory can be regarded as a set of premises of a theory that describes the current context of a particular individual. The logical memory contains a fixed set of logical rules that define the meanings of the logical particles. The theorems that constitute the context are derived by taking the closure of all inferences that can be obtained by applying the meta-rules to the deductor memory.

Because the meta-rules are elimination rules, all inferences are sound with respect to FOPC but not complete. That is, given a theory $T$ and an assumption $a$, it will always be the case that $(T \models a)$ if $(T \vdash a)$ but not necessarily true that $(T \vdash a)$ if $(T \models a)$. This has important consequences. It is not possible to make every valid inference. There is a possibility that a crucial inference is not made. In particular, it will not always be possible to detect when the theory expressed by the deductive device is inconsistent.

However, this is not unreasonable behaviour. It is certainly the case that human beings often hold inconsistent beliefs that they are not aware of (until it is pointed out to them by an external agent). Furthermore, any particular inference can be made by adding extra encyclopædic information. It is only the general case that cannot be solved.

## 4.2 Encyclopædic Information

The detailed syntax and semantics of encyclopædic information is discussed in chapter 5. In this section, I shall discuss the constraints that must be placed on any representation due to the operation of the deductive device.

SW assert that logical rules must be elimination rules because this should guarantee termination of the inference process. However, if the logic employed by the deductive device is as powerful as FOPC with functions, this is not sufficient. Consider the following two pieces of encyclopædic knowledge, expressed in FOPC:

$$\mathrm{Coypu}(!x) \supset \mathrm{Coypu}(\mathrm{Mother.Of}(!x)) \tag{4.10}$$

$$\mathrm{Coypu}(\mathit{cedric}) \tag{4.11}$$

Using the logical rule of modus ponens, these two pieces of information can be combined to infer:

$$\mathrm{Coypu}(\mathrm{Mother.Of}(\mathit{cedric}))$$

$$\mathrm{Coypu}(\mathrm{Mother.Of}(\mathrm{Mother.Of}(\mathit{cedric})))$$

and so on, ad infinitum.

Some of the possible solutions to this problem are:

1. Force information that can be treated as rules (like material implication) to be "elimination" rules. For example, for material implication, the RHS must eliminate a concept from the LHS.

2. Prevent the interface from entering assumptions into the deductor memory that are too deeply nested (see section 8.4.5 for a detailed example using the "=" relation).

The second approach was used in Crystal.

## 4.3 Processing an Utterance: Some Issues

It is now possible to describe the operation of the deductive device as induced by the logical and encyclopædic information that it contains. From the account of RT developed in chapter 2, the following describes how an utterance representation is processed.

The utterance representation is loaded into the deductor memory after production by the input processes. At this point, the context may be optionally expanded to an arbitrary degree. In the case of Crystal, the context is only expanded from the conceptual memory and previous utterances. Next, inferencing takes place. The inference controller forms the closure of forward inferences in the deductor memory. SW claim that further context expansion/inferencing is *justified* so long as the utterance is Relevant in the current context. This is very non-committal; it leaves the following issues unanswered:

1. Should the inference process stop immediately if the utterance is found to be Relevant?

2. When should context expansion begin?

3. To what extent should the context be expanded?

Detailed consideration of the expansion and inference process itself leads to further questions. When the context is expanded, is the information copied into the deductor memory, or does it otherwise augment it? In section 2.5.4, a declarative definition of contextual implication was presented. Procedurally, SW note that contextual effects must be realised by tagging the contents of the deductor memory. What should these tags be? Having tagged an assumption, does it maintain this status indefinitely, or does it change? How are new assumptions added to the deductor memory, and at which points is Relevance measured? How do all of the pertinent processes interact?

The remainder of this chapter will address these issues. Some of the above points will be reexamined in greater detail in later chapters.

## 4.4 The Inference Controller

The inference controller takes the contents of the deductor memory as one input, the meta-rules as the other, and produces a transformed deductor memory. This section provides a detailed account of its operation.

First, I consider and decide upon the conditions under which the inference process should terminate. I then discuss and describe the inference algorithm itself. Finally, I consider how contextual implications can be identified using SW's idea of tagging assumptions in the deductor memory. I show that it is necessary to increase the number of different categories of tags from those suggested by SW in order to achieve the required effects.

### 4.4.1 Terminating the Inference Process

The only reasonably detailed descriptions of the inference process in the Relevance literature can be found in Sperber and Wilson (1986). Unfortunately, SW do not provide sufficient details of its termination conditions. They are only prepared to say that termination may occur for any one of several reasons outlined in chapter 2. However, these conditions only specify which level of context expansion is appropriate to make an utterance Relevant, they do not specify at which point the inference engine should stop; should inferencing stop as soon as incoming utterance is considered Relevant, or does it go on to take the closure of forward inferences in the current context?

Crystal's inference controller always completes the closure for the following reasons.

#### The Possibility of Further Contextual Effects

If inferencing for the current utterance and context stopped when Relevance was achieved, then there may be many more inferences that could be made. When a new utterance is processed, it may in itself have little connection with the context, yet the inferences that could be made from the last utterances still result in the utterance being considered Relevant. This is clearly undesirable.

It may be argued that the inferences of the previous utterance that are yet to be made may no longer have the status of contextual effects since the original utterance has become part of the context[6]. However, these inferences may well be providing intuitively significant new information. Moreover, if they are not to be regarded as contextual effects, then they are causing unnecessary expenditure of effort by the system in doing all of this inferencing.

If inferencing continues until all possible inferences have been made then no further processing is required in ensuing utterances to process the original context.

### Detection of Inconsistencies

Human reasoning is not perfect; sometimes, the theory contained in the deductive device will be inconsistent. It is desirable to detect inconsistencies as rapidly as possible.

If inferencing is truncated prematurely, it is possible that the contents of the deductive device are inconsistent, yet this has not been discovered. This discovery may not occur until the next utterance is processed. Consequently, many inferences will have been made unnecessarily, since they are based on faulty premises. The erasure of these inferences will result in the unnecessary expenditure of effort.

It may be argued that if inconsistencies should be detected as early as possible, then the context should be expanded as much as possible to ensure this. However, this would also require unwarranted effort and, since the inference system is incomplete, even indefinite expansion will not guarantee detection of any inconsistency.

It is appropriate to make as many inferences as possible in the initial context since most information (including inconsistencies) should be communicated in it.

### Declarative Nature of the Deductive Device

A property of FOPC and many associated logics is that the closure of all inferences results in a single *extension* or set of theorems. Therefore, the particular implementation used for forming the closure is irrelevant. However, if the full closure is not taken then the resulting set depends on the order of inferencing, thus destroying the declarative nature of the deductive device.

### Relevance as a "Magic Number"

Assuming that there is some way of measuring information and effort in a cognitive system, some measure of Relevance must be computable[7]. The only property this factor must have is that its magnitude can be compared with other Relevance factors.

SW claim that RT is such that it should be possible to make gross comparisons of the Relevance of utterance/context pairs. There is no arbitrary Relevance factor whose value determines whether an utterance has been understood or not[8]. If such a factor existed, a value could be provided for it which indicates exactly when inferencing should stop for

---

[6]Recall that contextual implications must be drawn from the utterance *and* the context.

[7]It is not important at this juncture whether it is a scalar or even numerical quantity.

[8]There may, however, be a value of Relevance below which the measure itself may be considered meaningless.

sufficient understanding[9].

In Crystal, the closure algorithm is run uninterrupted. Relevance is checked only at the end of the computation, at which time decisions are made as to how to progress.

### 4.4.2 The Closure Algorithm

#### The Order of Inferencing

The order of inferencing is significant in this particular implementation for three reasons.

1. The left-right inference strategy simulates incremental interpretation (see chapters 3 and 7).

2. Crystal uses a non-monotonic logic (see chapter 7) which can have multiple extensions. The particular extension selected depends on the order of application of certain rules, which in turn depends on the order of inferencing[10].

3. For reasons discussed above, inconsistencies should be detected as soon as possible. The inference algorithm should facilitate this[11].

The candidate inference algorithms are all restricted Markov algorithms known as *incremental inference procedures* (Genesereth and Nilsson, 1987). These algorithms take a database and enhance it in stages by adding new information derived from the previous stage[12]. The two principal strategies are *breadth-first* and *depth-first* inferencing.

Procedures that work in a breadth-first manner work in several rounds. On each round, all of the conclusions that can be derived from the current database using the meta-rules are derived. The following round derives all of the inferences that it can from the conclusions and the database. This goes on until no more inferences can be derived.

A depth-first inference strategy also works in rounds. The first round forms the set of conclusions that can be derived by applying the meta-rules to the the first element of the database plus any other element of the rest. The successor database is formed by appending the conclusions to the front of the previous database[13]. The process then repeats until no more conclusions can be derived.

Crystal uses a depth-first strategy for the following reasons:

- Depth-first inferencing better simulates incrementality; all the consequences of a particular assumption are explored before the other assumptions are considered.

- The ordering of information in the deductor memory will reflect the transition from older to newer information (see section 4.6.4).

[9]SW's claim that Relevance can only be used comparatively contradicts their assertion that the context can be expanded if an utterance is not "Relevant enough". I will discuss this further in chapter 6.

[10]This may appear to destroy the declarative nature of the deductive device, but this dependence on the order of inferencing could be avoided by placing a priority relation on assumptions.

[11]The process of consistency restoration can be viewed declaratively as an ordering on alternative consistent extensions.

[12]One property of these systems is *monotonic growth* of the database: inferences are never retracted. Crystal allows for the retraction of inferences but these are performed by the separate reason maintenance process. I will examine this separately.

[13]The algorithm that implements this below works in a slightly different manner and appends new information to the end of the database.

- Crystal makes hypotheses which it adds to the deductor memory during inferencing. These hypotheses may be inappropriate, resulting in inconsistencies. The depth-first inferencing strategy explores every consequence of a particular assumption before considering the rest. This means that inconsistencies due to bad assumptions will be detected as rapidly as possible.

- The depth-first strategy seems to be a more appropriate model of human reasoning. People tend to explore a single strand of thought at a time.

#### The Depth-First Inference Algorithm

The inference algorithm used is similar to that described in Genesereth and Nilsson (1987). The inference function takes a set of premises and returns a set of conclusions that is a fixed-point.

The only significant point of divergence between Crystal's inference algorithm and Nilsson's occurs because Crystal allows assertions of equality (see section 8.4.5). When such assertions are made, the agenda of clauses yet to be processed is reset to be the entire deductor memory since it is possible that new inferences can be made from assumptions that have already been removed from it.

### 4.4.3 The Contextual Status of an Assumption

Recall SW's definition of contextual implication given in section 2.5.4:

*Contextual implication:* A set of incoming assumptions $P$ contextually implies an assumption $q$ in context $C$ iff:

1. $P \cup C \vdash q$
2. $P \nvdash q$
3. $C \nvdash q$

In chapter 2, it was noted that the source of an inference needs to be noted in order that the inference controller be aware of when it is making contextual implications. The above definition does not mention this source. Therefore, I will develop a more suitable one.

#### The Sources of An Assumption

The definition above only considers two possible sources: incoming assumptions and a context. However, in practice, there are four possible sources of an assumption:

1. The assumption is either a representation of the utterance or an analytic implication derived from the utterance representation, i.e. fundamentally considered part of the utterance. For example, if the following is an utterance:

   The yeti stood up and spoke in abominable Yiddish. (83)

   Then its analytic implications, such as (84), can be considered to be part of the utterance.

   The yeti stood up. (84)

2. The assumption was introduced from conceptual memory or is an analytic implication of such an assumption, i.e. fundamentally considered part of the encyclopædic memory. For example, the following information may be introduced from conceptual memory when (83) is processed.

   Yetis typically live on Tibetan mountains. (85)

3. The assumption was already present in the context, i.e. derived from previous utterances. The following may be part of the context in which (84) is uttered:

   Ymir the yeti is learning to talk. (86)

   This may have originally been derived from, say, a previous utterance, but is now considered to be just part of the previous context in which a new utterance is processed.

4. The assumption is a contextual implication introduced by the inference process. For example, from (84) and (85), the following contextual implication might be inferred:

   Ymir lives on a Tibetan mountain. (87)

Let these different sources be abbreviated to *Utterance, Memory, Context* and *Contextual.Implication.*

Can these four categories be conflated into the two that SW suggest? If this were so, it would not be possible to distinguish between assumptions of source Memory, Context, and Contextual.Implication. However, this is not the case.

Consider an inference drawn from an encyclopædic entry loaded by a previous utterance and some new encyclopædic information associated with a fresh utterance. This is surely a contextual implication, but it would not be if the categories of Context and Memory were not distinguished.

Erasing assumptions of source Context is usually regarded as a contextual effect. Reconsider the utterance:

The author of this thesis is a gerbil. (33)

If this were to be added to the deductor memory, it should result in an inconsistency. Since most people do not believe that gerbils can write[14], the inconsistency is normally resolved by erasing (33). This utterance is not informative because it does not add any new information, neither does it result in a modification of the hearer's currently salient beliefs (i.e. those of source Context)[15].

### Implementation

When an assumption is entered into the deductor memory via the interface, a node is constructed to represent it. This node is a structure consisting of several fields which include its *content*, and *contextual status*. The former is the propositional content of the assumption, the latter is a member of {Utterance, Memory, Context, Contextual.Implication}.

A revised, more implementation dependent definition of contextual implication is:

[14]To be more precise, most people believe that gerbils cannot write more strongly than they believe that the author of this thesis is a gerbil.

[15]Erasure of information of source Memory is also significant, but is another issue: it is the realisation that one's understanding of a concept might be wrong.

*Contextual implication (2):* A set of assumptions $P$, all of identical status, contextually implies an assumption $q$ in the context of a set of assumptions $S$, all of identical status, iff:

1. $P \cup S \vdash q$
2. $P \nvdash q$
3. $S \nvdash q$
4. Either Status($P$) = Status($S$) = Contextual.Implication or Status($P$) $\neq$ Status($S$)

where Status is a function from sets of assumptions of identical status which returns that status.

A consequence of this definition is that information derived solely from two sets of status Memory can never yield new information. This is obviously correct if the two sets originate from an encyclopædic entry, but more contentious if they originate from different entries.

### Contextual Status of Non-demonstrative Inferences

All non-demonstrative inferences made by the system are considered to be of type Contextual.Implication because, intuitively, they are providing new information.

## 4.5 The Interface

During the inference process, information is added to the deductor memory via the interface. The interface is responsible for constructing nodes suitable for handling by the RMS and querying system. It also keeps track of the number of contextual effects and the effort expended by the system.

A running count of the calculated Relevance is maintained from this but, by the argument given in section 4.4.1, it need not be examined until termination of the inference process. There are six situations in which passing an assumption through the interface does not merely result in the construction of a new node:

1. The assumption results in redundancy of encyclopædic information in the deductor memory, for some reason.
2. The assumption refers to some kind of non-demonstrative inference.
3. The assumption refers to inconsistencies within the deductor memory.
4. The assumption asserts the equality of two entities.
5. The assumption asserts some kind of definite reference.
6. The assumption can be simplified.

Each is discussed in turn.

### 4.5.1 Redundancy from a New Assumption

#### Reasons for Redundancy

New assumptions can result in redundancy because:

- The new assumption is identical to an existing assumption.
- The new assumption subsumes an existing assumption.
- The new assumption is subsumed by an existing assumption.

An assumption $a_1$ *subsumes* another assumption $a_2$ iff $a_1$ entails $a_2$ but $a_2$ does not entail $a_1$. For example, the sentence:

All aardvarks eat termites (88)

subsumes the sentence:

Aardvarks called Abelard eat termites (89)

These sentences approximate to the following logical forms[16]:

$$\text{Aardvark}(!a) \supset \text{Eats}(!a, \mathit{Termites}) \tag{4.12}$$

and

$$\text{Aardvark}(\mathit{abelard}) \supset \text{Eats}(\mathit{abelard}, \mathit{Termites}) \tag{4.13}$$

It can be seen that (4.12) has the same form as (4.13) except that the variable $!a$ is replaced by *abelard*. Since $!a$ represents any constant, (4.12) can be considered more general. Subsumption is detected by a simple matching process, similar to that given in Nilsson (1982, pp. 174).

In chapter 5 it will seen that objects in Crystal are *sorted* — they are categorised as belonging to particular classes. A subsumption mechanism for sorts is also provided which ensures that sorted entities are always classed by the most specific sort.

The reason why I am describing this well-known phenomenon in such detail is that I will modify the following subsumption-checker to partially handle the question/answer phase of interpretation in chapter 8. This is because it is one of the few points in the system when it can be argued that some form of backward inferencing (i.e. querying) is occurring.

#### Implementation Details

The interface constructs deductor memory nodes from an assumption and information concerning its derivation. This information is incorporated into the node as a *justification*. For present purposes, the sole uses of justifications are:

1. To check if a particular derivation has been produced before and, if so, to prevent it being repeated.

[16] This is a simplification of my own representation.

2. To contribute to a crude measure of strength/manifestness.

It will be seen in chapter 7 that these same justifications are necessary to support the reason maintenance process.

Deductor memory nodes have a field known as the *justification set.* If the contents of an existing node subsume an incoming node, the new node is not added. Rather, a *justification* that depicts the sources and method of its derivation, is added to the justification set. Thus, the justification set contains possible reasons for believing the node. A new justification is only added to a justification set if it is not already a member: a node is only justified once for a particular reason.

When a new assumption is added to the deductor memory via the interface, it first scans the deductor memory in reverse temporal order to check if the assumption subsumes, is equal to, or is subsumed by any existing clause. This scanning can be considered to be a kind of backward-reasoning; in chapter 8 it will be seen that this phase interacts with a non-demonstrative inference generator to enrich forms in a manner similar to the question-answer phase of interpretation described in chapter 3.

When it is decided that one assumption is equal to, or subsumes another, the existing node is updated to reflect the extra knowledge in the following ways:

1. If the new assumption, $A_n$, is identical to, or subsumed by an existing assumption $A_e$, it is not added to the memory.
   (a) If the justification of $A_n$ is already present in the existing assumption's justification set, no further action is taken.
   (b) If the justification is not present, it is added to the justification set of $A_e$. If the contextual status of the new assumption is Contextual.Implication, then this becomes the status of the existing node, and the information count is updated accordingly.
2. If the new assumption subsumes an existing assumption, it replaces it. The justification set of the node for $A_n$ will be the union of the justification set of $A_e$ and the justification for the $A_n$. The contextual status is Contextual.Implication if any of $A_n$ and $A_e$ was a contextual implication or that of $A_n$ otherwise.

If the status of an existing node is changed to Contextual.Implication then the contextual statuses of all of the inferences made from it are changed to Contextual.Implication and the information count updated. Hence, it is possible that reinforcing an already existing assumption might change it and its dependent inferences into contextual implications.

### Strength and Manifestness

In chapter 2, it was explained that strengthening an assumption makes it more manifest. Making an assumption in the context more manifest produces contextual effects. It is not clear from SW's writings whether an assumption held at full strength can ever become more manifest.

In Crystal, I have modelled strength and manifestness in a rather crude manner. I assume that every fresh justification for an utterance strengthens it. Thus, every new justification for a contextual implication counts as another contextual effect.

Much as this is a crude heuristic, it does have some justification; it has already been shown in chapter 2 that a strongly held belief can be made more manifest. Thus, the number of justifications in the justification set might be better considered to be a measure of manifestness rather than of strength[17]. I will suggest theoretically-motivated enhancements to this scheme in chapter 9.

### 4.5.2 Default Assumptions

Default assumptions have special forms which are directives to the interface to add their argument only if this is consistent with the beliefs in the deductor memory (see chapter 7). It is impossible to guarantee consistency since the inference rules are incomplete. However, the interface can check whether the negation of the assumption to be added has not yet been added to the deductor memory. Unlike the scan during subsumption checking, this scan for the negation does not invoke the non-demonstrative inference generator.

Thus, when a directive to add a default assumption, $P$, is passed to the interface, a search is made for its negation. If $\neg P$ is present, the directive is added to the deductor memory and no further action is taken; if $\neg P$ is not present, then it is consistent to believe $P$, so the directive and the default assumption are asserted. If, at some later stage, $\neg P$ is asserted, the reason maintenance process is invoked to update the assumptions dependent on $P$. A complete explanation of this system is provided in chapter 7.

### 4.5.3 Contradictions

If the inference controller detects an inconsistency, it asserts a node of the form "INCONSISTENT($P, \neg P$)" (see section 4.1.4). On detecting this, the interface passes control to the reason maintenance process to revise the beliefs of the system and restore consistency. This is effected by tracing back through the inconsistency's justifications to find the possible causes of the problem. One of these is then erased from the deductor memory[18].

### 4.5.4 Equality

Crystal has a special system, known as the equality sub-system, to deal with assertions of equality (see chapter 8). When an assertion of the form:

$$A = B \tag{4.14}$$

is made, $A$, $B$ and the justification for this assumption is passed to the equality sub-system. The node is then asserted as per normal.

### 4.5.5 Definite References

Assumptions of the form REF($x$) state that $x$ is a definite reference to an entity mentioned in the discourse. Similarly, if an assumption is of the form IT($x$), it states that $x$ is a pronominal reference. The interface asserts the former in the normal way after its argument is inserted onto a "REF list". The same is done for the latter, except that its

[17]Another reason for considering every addition of a new justification to be a contextual effect is that Gricean implicatures are *reinforceable*: an implicature can be conjoined with an explicit mention of its content without a sense of anomalous redundancy. This assumes SW's implicatures are close enough to the Gricean case to bear such comparison.

[18]In reality, it is not physically removed, but tagged as "not believed". This is true of all erasures (see chapter 7).

argument is also added onto an "IT list". The purpose of these lists is explained in section 8.3.2.

### 4.5.6 Simplification

Before an assumption is asserted in the deductor memory, an attempt is made to simplify its form using various heuristics. For example, assumptions of the form, $A \wedge A$ would be simplified to just $A$, and those of the form $A \vee \neg A$ would be reduced to true.

Assumptions that are just "true" are not input to the deductive device, because they are of no use. Thus tautologies are not at all manifest and do not form part of the cognitive environment.

Assumptions that are "false" are considered to be contradictions. The reason maintenance process is invoked to restore consistency in the same way as described in section 4.5.3.

## 4.6 The Context

SW do not describe exactly what constitutes the context and how it changes with successive utterances and context expansions. These issues will now be examined. In particular, I will consider:

1. Which memory components can be said to hold the context.
2. How the initial context is determined.
3. How the context varies with expansions and successive utterances.
4. How previous contexts can be incorporated into the deductive device.

Each is discussed in turn.

### 4.6.1 The Location of the Context

SW state that the contents of the deductor memory constitute an initial context, however they do not make it clear whether an enlarged context is also contained wholly within the memory.

The issue can be resolved by considering the purpose of the deductor memory. Its function must be to provide a region on which the inference controller can operate. If this were not the case, it would be serving no purpose at all. Now, if the context were to be expanded by adding encyclopædic information, this data could either copied into the deductor memory or included for consideration by the inference controller in some other way. However, the latter could not be possible because it would make the function of the deductor memory void. Thus, it is *only* the deductor memory contents which constitute the context.

Since it will generally be full, adding inferences or expanding the context will result in the rejection of some information to the context buffer. This not considered by Crystal, which contains an indefinitely large *expansion region* for the deductor memory, depicted in figure 3.2. Thus, plausible and computationally tractable results will only eventuate from short discourses.

### 4.6.2 The Initial Context

As explained in chapter 2, the contents of the deductor memory form an initial context for processing an utterance. SW claim that the context need not immediately be expanded when the representation of the utterance is input to the deductive device; I argue the converse.

There are two fundamental situations which can arise:

1. Either the deductor memory is empty or it does not contain encyclopædic information about any concepts in the utterance representation.

2. The context contains information about some or all concepts in the utterance representation.

Consider the first case. Usually, only analytical implications follow from the utterance representation. The utterance might yield synthetic implications if there are inferences that can be drawn from its very presence in the deductive device. Such an example was provided in chapter 3, where Brünnhilde is looking for Wotan, so his utterance is Relevant in itself because it locates him. However, according to Fodor, the decoding of an utterance is a reflex process that cannot be restrained. Thus, effort is always expended by the automatic decoding process. This effort could be counterbalanced by information that *must* follow from Wotan's utterance by the Principle of Relevance. If Brünnhilde does not attempt to access this information, she is not adhering to the presumption of optimal Relevance. In other words, if Brünnhilde wishes to make the utterance as Relevant as possible, she should not stop when it is Relevant, but when she believes it is *maximally* Relevant; this should be after at least one expansion with appropriate conceptual information. According to RT, the information derived from the processing of an utterance in such a way should always be enough to offset the effort in expanding the context etc.

Consider the second case. At least part of the utterance might be understood using encyclopædic information already present in the deductor memory. However, it would be a strange theory of language that claimed that an utterance need only be partially processed if this is Relevant. Consider a situation where Van Helsing staggers into his office. It becomes apparent that he has a bite on his neck. He sits down, and says:

I've just been bitten by a rabid bat. (90)

Under these circumstances, it is reasonable to assume that even a fragment of this utterance whose concepts might already be present in the context will be Relevant, for example:

I've just been bitten *by something.* (91)

Clearly, the utterance might be Relevant at this stage, but it cannot be said to have been understood. The fact that the bat was rabid is likely to be extremely significant. It may make the utterance more like a request for help than some sort of explanation. Thus, it is necessary to expand the context with the meaning of all of the concepts in the utterance representation and process the entire utterance representation even if a fragment of this utterance alone is Relevant.

Hence, in Crystal, the utterance representation is immediately passed to the context expansion process. This picks out the concept names from the utterance and uses them to index into the encyclopædic memory. The appropriate conceptual information is consequently loaded into the deductor memory. The assumptions that are added are given a status of Memory. Thus, it is the utterance representation and associated encyclopædic information that form an initial context. The inference process is then initiated and the contextual implications made are added to the end of the deductor memory[19]. After this, the deductive device will contain assumptions of various statuses.

### 4.6.3 Context and Contextual Status

The definition of contextual implication does not explain whether the contextual status of the sources is unwavering once it is has been initially determined or whether it varies with time.

In fact, there are two situations where it may be necessary to alter the contextual status of an assumption in order to account for various phenomena:

1. When a new utterance is processed.
2. When the context is expanded.

Each is discussed in turn.

#### Processing Successive Utterances

The contextual status of all of the assumptions in the deductor memory must be set to Context. If not, inferences drawn from successive utterances would not be regarded as contextual implications. Consider the following:

> They're auctioning either a lot of De Koonings
> or a lot of Pollocks. (92)

when followed by

> It's a lot of Pollocks. (93)

yields the inference

> They are not auctioning a lot of De Koonings. (94)

If (92) and (93) were two successive utterances, the inference (94) could be drawn from the two of them alone. Surely this should be treated as a contextual effect. Indeed, it may be the most manifest belief communicated by (93).

In order to explain the fact that inferences drawn from information in the deductor memory and a new utterance representation are contextual implications, it must be the case that beliefs in the deductor memory change their contextual status. The simplest explanation is that all of the information in the deductive device has become part of the background context derived from a previous utterance, i.e. they have a contextual status of Context.

Any inferences drawn from the background context and another assumption type are regarded as contextual implications (by the formula for determining contextual implications given in section 4.4.3). This means that examples such as (94) would be regarded as contextual implications because (92) has status Context whereas (93) has status Utterance.

---

[19] In practice, the deductor memory is stored in reverse so that adding new information is easier, as are reverse chronological searches.

### Context Expansion

My definition of contextual implication does not regard inferences derived from two encyclopædic entries to be a contextual effect. However, should this also be true of inferences derived from two encyclopædic entries after context expansion?

Consider the following utterance:

Giuseppe painted his wife's portrait. (95)

Let us imagine a situation where, for some reason, (95) is not Relevant enough in the current context. The context will be expanded with information about the concepts in the utterance. Assume that the following assumptions are amongst those loaded into the deductive device:

Giuseppe enjoys art. (96)

People called Giuseppe are typically Italian. (97)

People who enjoy art typically know of artists famous in their region. (98)

These will all be of contextual status Memory. Consequently, the following inference, drawn from these assumptions, is also of type Memory and not a contextual implication:

Giuseppe knows of artists famous in Italy. (99)

If the overall Relevance is still too low, the context will again be expanded. This time, expansion involves using all of the currently salient concepts to index into the conceptual memory. Thus, the following assumptions may be among those brought into the deductor memory with contextual status Memory:

Italy is part of Europe. (100)

John Flaxman is famous in Europe for his drawings. (101)

From (99) and assumptions (100) and (101), it is possible to infer:

Giusseppe knows of John Flaxman's drawings. (102)

Should (102) be regarded as a contextual implication? Since it is initially very weakly manifest, it is difficult to appeal to intuitions about what normally constitutes significant new information. It is certainly difficult to conceive of a situation where somebody utters (95) with the specific intention of conveying only (102) to the hearer. Rather, this is a situation where one of many weakly manifest assumptions are made a little more manifest (see section (2.1.2)).

If (102) is to be regarded as a contextual implication (and I think it should), either the definition of such implications is inadequate or the contextual status of items in the context changes before context expansion. Assuming the former is not the case, one explanation for the latter is that, before context expansion, everything in the deductor memory is marked as having a contextual status of Context. Thus, context expansion would have the same effect on the deductor memory as would a fresh incoming utterance representation.

In chapter 6, I revise the notion of Relevance so that context expansion occurs under more restrictive circumstances. The consequence of this is that it is unlikely that (95) would be processed in the way I have just described. Consequently, I have not investigated the issue of how contextual status changes during context expansion because, in practice, such situations should not arise.

### 4.6.4 Previous Contexts: Reference Resolution

SW suggest that the context can be expanded by including information from previous utterances. In this section, I argue that part of this function can be modelled without the need for a context buffer.

There are two obvious reasons for including previous processing information when expanding the context:

1. To allow more implicated conclusions to be drawn.

2. To pick up candidate referents in a plausible order.

The first point is not dependent on the chronological ordering of the information, and is handled in Crystal by having an indefinitely large extension region.

The second point is handled by the ordering of information in the context buffer. The left-right depth-first inferencing strategy results in information in the deductor memory being ordered in a manner related to the way it was derived through the use of incremental processing; considering the memory to be a list, successive elements represent newer information in the sense that it either arose from a more recent utterance or from a segment of the utterance closer to its point of focal stress.

SW do not explain how references are resolved, but it is reasonable to assume that resolution is attempted during the question-answer phase of interpretation. Presumably, the deductor memory is first scanned for a candidate referent. If it is not present, increasing amounts of the context buffer are then added to the deductor memory and the search repeated until either a referent is located or further expansion is not possible (due to physical or Relevance constraints). This is simulated by Crystal, which attempts to answer questions addressed to the deductor memory by scanning it backwards for an answer. This scan constitutes a reverse chronological search. Thus, not only is the context automatically expanded, but putative referents are picked up in the correct order.

Although this argument only concerns Crystal's structure, it equally applies to the ideal system; as long as the assumptions in the deductor memory are chronologically ordered, there is no need to keep the deductor memory and the context buffer explicitly separate[20].

### 4.6.5 The Context Expansion Process

This process takes as input a set of assumptions, $\{A_1, \ldots A_n\}$ and returns encyclopædic information about all of the concepts named in them.

The process works as follows:

1. For each assumption, $A_i$, extract the set of concept names $N_i$[21].

2. Form the set $S$ such that:

$$S = \bigcup_{i=1}^{n} N_i$$

[20] However, alternative memories are still needed for context switching.

[21] Concepts like "$\wedge$" are filtered out at this stage since they are defined by the meta-rules. The same is true of anonymous constants discussed in chapter 5 because they do not have encyclopædic entries.

3. Let Enc($cn$) be a function that returns the encyclopædic information for concept name $cn$. If each element of the set $S$ is $s_j$, return the list:

$$[\mathrm{Enc}(s_1), \mathrm{Enc}(s_2), \ldots, \mathrm{Enc}(s_j), \ldots]$$

Enc($s_i$) is not added to the list if it has already been loaded into the deductor memory. In order to support this, the context expansion process maintains a list of all concepts whose encyclopædic entries have been downloaded into the deductive device. All of the assumptions of every Enc($s_i$) in the list are assigned a contextual status of Memory.

The initial context is formed by appending the contents of the deductor memory, $D$, the utterance representation, $U$, and the results of expanding the context from the singleton set $\{U\}$.

Expanded contexts are formed by appending the contents of the deductor memory *at expansion time*, $D$, to the encyclopædic information derived from $D$.

## 4.7 Control of the Deductive Device

The main control process is responsible for the synchronisation of the various processes in the system. In particular, it is responsible for:

1. The flow of logical forms into the deductive device.
2. Instructing the context expansion process to expand the context with encyclopædic information.
3. Instructing the deductive device to commence inferencing.
4. The examination of Relevance measures.

These points will be discussed in this section.

### 4.7.1 Processing a Single Utterance Representation

The processing of a single utterance representation can be summarised as follows.

The utterance representation is loaded into the deductor memory. It is also passed as a singleton set to the context expansion process. The controller appends the returned clauses to the deductor memory via the interface. The result is that the deductor memory will contain the utterance and the assumptions associated with its constituent concepts.

The inference process is then started and allowed to run to completion. During this process, the interface is informed of the number of contextual effects and the effort expended in order to achieve them. Relevance is calculated as a function of these two parameters[22] (see chapter 6).

If the utterance is not Relevant enough, the contents of the deductor memory are marked with contextual status Context, then the context is expanded and the inference

[22]In fact, it is simply the amount of information divided by the amount of effort.

process restarted[23]. This will require more effort. If the utterance is not more Relevant, the process aborts: the utterance cannot be understood.

The context is expanded further by passing the set of all assumptions in the deductor memory to the context expansion process, and loading back the results. Concepts that have already been used as indices are ignored and no effort is assumed to have been expended in reconsidering them.

The context expansion/inference process is repeated until the utterance becomes Relevant or a processing stage aborts.

It would be useful if there was some way of selectively expanding the context using only portions of the deductor memory. Such a system might be based on notions of prominence or salience; a guide for doing this will be proposed in chapter 9.

In order to illustrate a simple interpretation cycle, I will assume that a representation of the following utterance has been passed to the deductive device:

Nutkin felt hungry. (103)

Let us assume its representation mentions only the concepts "Nutkin", "feel" and "hungry". The encyclopædic entry for "Nutkin" may be as follows:

Nutkin is a person. (104)

People denoted by Nutkin are typically squirrels (105)

Squirrels are small, furry animals with furry tails. (106)

Squirrels are red or grey mammals that like nuts. (107)

Note that this entry contains the culture dependent information that Nutkin is usually the name of a squirrel. Furthermore, in the domain of children's stories, animals are people.

The encyclopædic entry for "feel" might contain the following:

If $A$ feels $B$ then $A$ is a person and $B$ is a feeling. (108)

If $A$ feels $B$ then they have the feeling of $B$. (109)

The encyclopædic entry for "hunger" might contain:

If something has hunger, then it is a person. (110)

If something has hunger, then it wants some food. (111)

If something has hunger and has some food that it wants, it eats it. (112)

If something eats while it is hungry, then it is often not hungry after this. (113)

[23]It will be seen in chapter 6 that an utterance is not Relevant enough when its Relevance is below a minimal threshold or it has not been sufficiently developed.

The utterance representation is placed in the deductor memory with contextual status Utterance. After this, the encyclopædic assumptions (104) to (113) are loaded with contextual status Memory. Inferencing then takes place, and the deductor memory is augmented with all of the non-trivial inferences that follow from the assumptions in the deductor memory. Some of these inferences will be derived from the encyclopædic entries alone, for example:

Nutkin is a small, furry animal with a furry tail. (114)

is derived from (105) and (106). It consequently has contextual status Memory and is not a contextual effect. Other inferences will be derived from the utterance representation and assumptions that originated in encyclopædic entries, for example:

Nutkin wants some food. (115)

is derived from (103), (109) and (111). It is therefore a contextual implication and contributes to the Relevance of the utterance in the context.

### 4.7.2 The Utterance Processing Algorithm

An algorithm for processing one of a possible stream of utterances will now be sketched (see figure 4.1). The main procedure is Process.Utterance which processes the next utterance representation, passed to it as a parameter. The function Relevant takes a calculated Relevance Factor from the interface and returns a boolean indicating whether this exceeds a predetermined minimum Relevance threshold[24]. Processing terminates successfully when an Utterance that has just been processed is Relevant. If it is not initially Relevant, the procedure Expand.Context is called. This takes the Relevance of the utterance so far and uses it to determine whether it is worth expanding the context any further.

### 4.7.3 Handling Short Dialogues containing Ambiguities

Short dialogues without ambiguity could be handled as follows. First, the deductor memory is flushed and any state variables reset. Then the first utterance is processed using Process.Utterance (as illustrated in 4.1). Further utterances can then be processed just by further calls of Process.Utterance, passing each utterance to it as an argument. As mentioned in chapter 3, a special Utterance browser provides facilities for the sequential processing of such streams.

When lexical or syntactic ambiguities are present in a dialogue, the above algorithm can be adapted to form part of a more general understanding procedure.

The disambiguation procedure called 'Handle.Ambiguous' is passed a list of putative literal meanings of an utterance. This list is generated automatically for the case of lexical ambiguity and manually for structural ambiguity. A sketch of the algorithm can be seen in figure 4.2.

The algorithm uses the interpretation queue to store the most Relevant interpretation during processing and the background context. The queue holds states of the deductive device, including the contents of the deductor memory and various state variables used for the calculation of Relevance[25].

[24]The determination of the threshold will be explained in chapter 6.

[25]Much as the interpretation queue is utilised here as a stack, random access to arbitrary elements is allowed. The use of this will be explained in chapter 7.

```
PROCEDURE Process.Utterance(Utterance)
BEGIN
    Change Status of all Nodes to Context
    Append Utterance to Deductor Memory
    Expand context using Concepts in Utterance
    Form closure of forward inferences
    Calculate Relevance.Factor

    IF NOT Relevant(Relevance.Factor)
    THEN
        Expand.Context(Relevance.Factor)
    END IF
END Process.Utterance

FUNCTION Relevant(Relevance.Factor) =
    Relevance.Factor >= Relevance.Threshold OR
    Utterance needs further enrichment.

PROCEDURE Expand.Context(Current.Relevance.Factor)
BEGIN
    Change contextual status of all nodes to Context
    Expand context using all concepts
    Form closure of forward inferences
    Calculate Next.Relevance.Factor
    IF
        (Next.Relevance.Factor >= Current.Relevance.Factor) AND
        NOT Relevant(Next.Relevance.Factor)
    THEN
        Expand.Context(Next.Relevance.Factor)
    END IF
END Expand.Context
```

Figure 4.1: Sketch of the Utterance Processing Algorithm

```
PROCEDURE Handle.Ambiguous(U1, U2, ..., Un)
BEGIN
    Push deductor state onto interpretation queue
    Process.Utterance(U1)
    Push deductor state onto interpretation queue
    Handle.Rest(U2, ..., Un)
    Restore deductive device state to that on top of interpretation queue
END Handle.Ambiguous


PROCEDURE Handle.Rest(Utt1, Utt2, ..., Uttn)
BEGIN
    IF there are any utterances left to process
    THEN
        Restore deductor state by copying from 2nd element of the interpretation queue
        Process.Utterance(Utt1)
        IF Relevance(Utt1) > Relevance(top state of intepretation queue)
        THEN
            Replace the top state on the interpretation queue with the deductor state
        END IF
        Handle.Rest(Utt2, ..., Uttn)
    END IF
END Handle.Rest
```

Figure 4.2: Sketch of the Disambiguation Algorithm

This initial state represents the background context in which each sense of the utterance will be processed. Handle.Ambiguous therefore pushes it onto the interpretation queue initially, and restores it before each sense is processed. After the first sense is processed, the state of the deductive device is again pushed onto the interpretation queue. This represents the most Relevant interpretation so far. Handle.Ambiguous then calls the auxiliary procedure Handle.Rest on the remaining senses.

Handle.Rest restores the initial context from the queue and processes each sense in turn. If the Relevance of the current sense is greater than the Relevance of the utterance in the deductor state at the top of the interpretation queue, it replaces it. In this way, the most Relevant interpretation will be the one on the top of the interpretation queue after Handle.Rest has finished.

As a by-product of this algorithm, the interpretation queue holds a chronological sequence of deductor states representing the contexts invoked by successive utterance representations. The states can be examined and possibly used for backtracking through a discourse to restore to some previous state. In chapter 7, it will be explained how the meanings of certain words might best be described in terms of operators on these contexts.

Handle.Ambiguous is the simple algorithm used by the main control process. An optional log is kept of its major actions, so that the user can monitor how processing progresses.

## 4.8 Summary

SW's description of logical and encyclopædic information is problematic. I therefore employ an established logic in the deductive device. In this logic, it is possible that the inference process will not terminate; this is avoided in Crystal by preventing assumptions that contain very deeply-nested expressions from being added to the deductor memory.

The interface is responsible for adding information to the deductor memory. It treats certain assumptions in a special manner, forwarding them to special handling routines. For example, if an assertion of equality is passed to the interface, it adds it to the deductor memory but also passes it on to the equality subsystem described in section 8.4.5. The interface is also responsible for rejecting assumptions with very deeply-nested expressions before they can be added to the deductor memory.

The inference algorithm employed by the controller of the deductive device forms the closure of deductions in the deductor memory using a left-right, depth-first inference strategy. It forms the entire closure before it examines the Relevance measure in order to decide what to do next.

The non-trivial inferences generated during the deduction process are tagged with a contextual status, which may be of type Utterance, Memory, Context, or Contextual.Implication. It is possible to determine the contextual status of an inference from the contextual statuses of the premises used to generate it. The number of assumptions added with contextual status Contextual.Implication contributes to the overall measure of significant new information and therefore to the overall Relevance measure.

Before processing a new utterance or expanding the context for a second time, all assumptions already present in the deductor memory are assumed to be part of the existing context, and are given a contextual status of type Context. This existing context is augmented with new assumptions when a new utterance is processed. These new assumptions may be contextual implications, utterance representations, or further encyclopædic information fetched from the conceptual memory. The concepts in the new utterance, and if necessary in the entire existing context, are used as indices to fetch this information. Because this growing context maintains the chronological order in which assumptions are added, there is no need for several buffers to hold successive utterance interpretations.

An unambiguous utterance is processed by loading its representation into the deductor memory, loading the encyclopædic entries indexed by its concepts, and taking the closure of their non-trivial forward deductions. During this process, the number of assumptions added with contextual status Contextual.Implication are counted. After the closure operation has finished, the Relevance of the utterance is calculated as a function of this information and the effort expended in obtaining it (see chapter 6). If the utterance is not deemed Relevant enough, the context contained in the deductor memory is augmented by re-indexing into the encyclopædic memory using the new concepts in the deductor memory obtained from the last context expansion. The inference process is then repeated, and the Relevance recalculated. The expansion/inference cycle repeats while Relevance increases.

A simple algorithm is used to process an ambiguous utterance. Every sense of the utterance is processed individually, and the most Relevant utterance interpretation selected as the most apt final interpretation. The operation of this algorithm is shown in detail in appendix B.

I have now given a basic account of Crystal's operation and of its monotonic inference capabilities. In chapter 6, I consider how a Relevance measure can be computed and used. However, because Relevance is highly dependent on the formal properties of the knowledge representation used during the inference process, I first give an account of Crystal's knowledge representation in chapter 5.

# Chapter 5

# Knowledge Representation in Crystal

Crystal's encyclopædic knowledge representation should express the common-sense knowledge that we use to reason. The inferences that can be derived from this representation will determine what can be considered to be a contextual implication, because this is essentially a syntactic notion. Thus, the Relevance of an utterance is contingent on the choice of knowledge representation for logical forms. Different knowledge representations will involve different syntactic organisations, which, in turn, will yield different contextual effects.

In this chapter, the major problems for the knowledge representation will be introduced and a knowledge acquisition strategy for Crystal discussed and explained. A basic syntax and semantics will be provided for the knowledge representation which will be argued to be expressive enough to represent assumptions and assumption schemas. The advantages of augmenting the basic knowledge representation to handle a *hybrid knowledge representation* are explained, and the implementation of a hybrid system for Crystal using *sorted unification* is provided. Next, it is explained how the representation system can be augmented to handle temporal, frame- and script-like reasoning. Finally, various implementation issues are discussed in order to explain how certain representation problems can be overcome.

## 5.1 Problems with Logical Forms

Many of the problems for Crystal's representation are concerned with the implementation of assumption schemas. SW's ideas about the utility of such schemas are exemplified by the following:

> Quite a lot of work has been done in the last ten or fifteen years on the organisation of conceptual information in memory, and various models have been proposed to describe what we are calling encyclopædic entries. These models are intended to answer questions about the structure of the entries, the relations between the various types of assumptions contained in them, and the relations among the entries themselves. Many of the models that have been proposed incorporate such notions as *schema, frame, prototype* or *script*...We do not want to argue for any one of them: in our terms the encyclopædic information contains not only factual assumptions but also assumption schemas which an

appropriate context may convert into fully fledged assumptions. (Sperber and Wilson, 1986, pp. 87–88).

SW aver that assumption schemas can perform the function of all of the various knowledge representation schemes. Unfortunately, they have provided no syntax or semantics for them. Crucial to the function of assumption schemas is the fact that they can be instantiated to form propositional assumptions which might later be erased. This instantiation mechanism is not explained, and the erasure process is underspecified. I will return to these problems in later sections. For the purposes of this section, however, I wish to illustrate two major difficulties associated with building a suitable knowledge base for a Relevance-based system.

1. SW state that frames, etc. can be implemented as assumption schemas. This suggests there is some simple transliteration from frames, prototypes, scripts, etc. into a logic. However, the notions that these frameworks encapsulate are often somewhat vague, lacking adequate semantics. It is often hard to characterise them in a deductive framework.

2. Given a particular semantics for an expression, it is possible to realise this expression in an infinite number of different ways, each exhibiting different inferential behaviour and therefore producing different numbers of contextual effects in identical contexts.

Each of these points is examined in turn.

### 5.1.1 Problems with Semantics

The problems with the structured knowledge representations like frames and scripts boils down to two fundamental problems: exactly what do they mean, and can this be logically expressed? In this section, I shall endeavour to answer these questions.

As SW have pointed out, considerable research effort has been put into knowledge representation. Early attempts to use logic as a knowledge representation language were not very successful because:

1. Unconstrained theorem proving is intractable.

2. FOPC is not an ideal formalism to represent human reasoning.

3. Human reasoning is not perfect; it is essentially invalid.

Typical examples of the early structured formalisms were *semantic nets*, *frames* and *scripts* (Quillian, 1967; Minsky, 1981; Schank and Rieger III, 1974). More sophisticated, but essentially similar formalisms ensued. These systems were intended to be intuitively better models of human knowledge, utilising some tractable inference algorithm that was not always performing deduction in the FOPC sense. Typically, they used some kind of graph-based structuring techniques, and could allow for the expression of defeasible and stereotypical information. It was soon observed that many of these systems lacked adequate semantics and were easily open to misinterpretation (Woods, 1975; Brachman et al., 1983a; McDermott, 1986a).

As the semantics of these systems were given more consideration, so their relationship to FOPC was made clearer. Hayes (1979) showed that many aspects of structured knowledge representations could be expressed in FOPC, and Moore (1982) showed that it could more flexibly handle universals and non-determinacy. Furthermore, it was shown that apparently efficient dedicated subsumption procedures became intractable when certain common and intuitively reasonable graph structures were allowed in a knowledge representation (Brachman and Levesque, 1984; Nebel, 1988). It appears that most adequately expressive structured knowledge representations suffer from the same intractability problems as FOPC. The only aspect of structured knowledge representations that can be proven to be unimplementable in FOPC is defeasibility.

In response to this deficiency, logicians allowed for the expression of defeasible information using *non-monotonic logics* (e.g. McDermott and Doyle (1980)). Within the numerous formalisms that have proliferated from the first introduction of such logics, attempts have been made to formalise frame-like systems (e.g. Brewka (1987)). Much as these endeavours have captured some of the defeasible nature of reasoning, they have not fully accounted for all of it. Even such basic notions as scripts have not been convincingly captured in non-monotonic logic. The translation of these extant knowledge formalisms into some sort of logic is a far from trivial matter. However, what logical formalisms have contributed to the area of defeasible reasoning is the ability to make explicit (through having well-defined semantics) some hidden problematic issues, for example problems due to multiple inheritance (Touretzky et al., 1987).

Whatever the knowledge representation utilised in a Relevance-based system, it is vital that it be provided with a formal semantics, since this defines which inferences can legitimately be licensed by its formulæ. This is of particular importance when considering *hybrid knowledge representations.* These are formalisms which combine the ease of structural description common to many structured formalisms with the more general deductive capabilities of logic (see section 5.4). A particular advantage of these systems is that they can compute certain operations very efficiently, for example whether one structure subsumes another[1]. A useful function of the semantics is to show that inferences are being made implicitly when subsumption checks are performed. Exactly what these inferences are can be expressed in the semantics. I will take up this issue again in section 5.4.

Let me summarise what the knowledge representation for Crystal must cover. It must:

- Provide a syntax and semantics for assumptions.
- Provide the deductive device with the advantages of a hybrid representation scheme (see section 5.4 for justifications for this).
- Explain how important notions such as frames, scripts and various kinds of non-monotonic reasoning can be expressed using assumption schemas.
- Provide a syntax and semantics for defeasible assumption schemas.

In this chapter, I will concentrate on the first three items. The last item will be explained in detail in chapter 7.

### 5.1.2 Problems with Syntax: Counting Information

Having claimed that providing adequate semantics is both difficult and necessary, I shall consider an equally important problem. Given a single semantic expression, there are an infinite number of ways of realising it syntactically. These expressions provide different

[1]The significance of this will become apparent in section 5.4 and chapter 8.

numbers of contextual effects during inferencing. Again, this has serious repercussions for RT, which is based, as described by SW, on the notion of information measured on a syntactic basis.

Let us look at a specific example: given a simple logical semantics for a trivial semantic network, it is possible to syntactically realise an expression (that employs the network relationships) in various plausible ways. These realisations can yield different contextual effects under identical conditions.

Consider a simple 'isa' hierarchy fragment, such as that portrayed in figure 5.1.

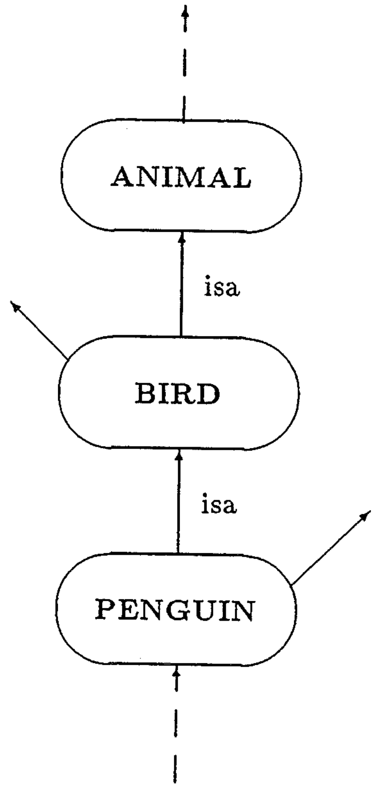



Figure 5.1: A Fragment of an 'isa' Hierarchy

This hierarchy is intended to express the fact penguins are birds, which in turn are animals.

In this case, a translation into logic is straightforward. The information about penguins in this network can be translated into a universally-quantified expression of FOPC, such as:

$$\forall !p.\, \text{Penguin}(!p) \supset \text{Bird}(!p) \wedge \text{Animal}(!p) \tag{5.1}$$

The same hierarchy can also be translated into the two semantically equivalent expressions:

$$\begin{aligned} \forall !p.\, \text{Penguin}(!p) &\supset \text{Bird}(!p) \\ \forall !p.\, \text{Penguin}(!p) &\supset \text{Animal}(!p) \end{aligned} \tag{5.2}$$

Now, consider an utterance which may invoke one of the above realisations:

Tweety is a Penguin. (116)

This utterance might be translated as:

$$\text{Penguin}(\textit{tweety}) \tag{5.3}$$

If this is a fresh utterance representation, it will be introduced into the deductor memory with contextual status Utterance. Assume one of (5.1) or (5.2) is subsequently loaded as part of the information concerning penguins, with contextual status Memory.

If (5.1) is loaded, the following will be inferred:

$$\text{Bird}(\textit{tweety}) \ \wedge \ \text{Animal}(\textit{tweety}) \tag{5.4}$$

$$\text{Bird}(\textit{tweety}) \tag{5.5}$$

$$\text{Animal}(\textit{tweety}) \tag{5.6}$$

(5.4) is inferred from (5.3) and (5.1) using modus ponens. Assumptions (5.5) and (5.6) follow from (5.4) by and-elimination. From the definition of contextual implication provided in section 4.4.3, all of these inferences count as contextual implications; there are three contextual effects in total.

However, if (5.2) is loaded instead of (5.1), only (5.5) and (5.6) are inferred. I assume (5.4) cannot be inferred because it would require an introduction rule. Thus, only two contextual effects result.

This problem is not just that of phrasing translations of frame-like formalisms, it is the general problem of phrasing encyclopædic knowledge. Because of the syntactic nature of contextual implication, contextual effects are highly dependent on the logical form of the encyclopædic knowledge representation. Some policy must be developed concerning how information must be expressed in order to provide sensible results. Furthermore, if a hybrid representation is used, an exact translation into logical form must be provided so that contextual effects can be properly measured (i.e. it is not sufficient to provide logical semantics — the formal realisation must be provided as well).

## 5.2 Knowledge Elicitation

The encyclopædic knowledge used by Crystal must be input by the user. Clearly, its character and quality will determine system performance. It is important that the knowledge for any one encyclopædic entry is of approximately the same precision and detail as for the other entries[2]. If this were not the case, there would be the danger that a particular entry would bias the main control process into selecting readings that involved its associated concept because it would be easier to draw contextual implications from it. This issue will be discussed further in chapter 6. Additionally, since encyclopædic entries are intended to represent folk knowledge about their associated concepts, it is important that the information contained within them is a reasonable description of such knowledge. Users should not simply input encyclopædic knowledge as they see fit, but be constrained by some guiding principles that should ensure approximately even distribution of suitable detail across the entries. I will first consider the constraints that can be placed on the logical rules and then on the encyclopædic information.

[2]In section 6.4.2, I will suggest ways of alleviating this problem using the notion of loading effort.

### 5.2.1 Logical Rules

The system's logical rules are elimination rules that provide meaning for the logical particles.

In principle, any FOPC formula could be expressed solely in terms of material implication. This is undesirable because:

- It can result in cumbersome, counter-intuitive expressions of knowledge, which in turn result in counter-intuitive numbers of contextual effects.
- Although the inference rules of FOPC can be defined solely in terms of modus ponens, most of them cannot be expressed as elimination rules. A reduced set of elimination rules using only modus ponens would be inadequate.

As a simple illustration, consider two of the five axioms that Turner (1984, pp. 15) uses to axiomatise FOPC in terms of material implication:

$$A \supset (B \supset A) \tag{5.7}$$

$$A \supset (B \supset C) \supset ((A \supset B) \supset (A \supset C)) \tag{5.8}$$

These axioms are inference schemas. Their instantiation can be considered to be an analytic implication. So, for example, given $p$, (5.7) can be instantiated in order to infer that $p \supset (z \supset p)$.

These schemata, along with the rule of modus ponens (MP), are used to describe the valid inferences of FOPC[3]. However, (5.7) is an introduction rule, and therefore should not be used in Crystal. Additionally, the way the schemas must be used in order to obtain an "obvious" inference can be rather convoluted. Consider an expression such as:

$$(U \wedge M) \supset B \tag{5.9}$$

This is intended to represent the fact that, if Zog is unmarried and Zog is male, then he is a bachelor. Assume we are given (5.9) and the fact that Zog is male, $M$. It seems reasonable to infer the fact that, if Zog is unmarried, then he is a bachelor, i.e. $U \supset B$. This can be realised in a single inferential step using the rule of conjunctive modus ponens suggested by SW:

**Identifier:** Conjunctive Modus Ponens (b)
**Trigger 1:** $(!A \wedge !B) \supset !C$
**Trigger 2:** $!B$
**Inference:** $!A \supset !C$

Now consider the same expression expressed solely in terms of material implication. In this case, an equivalent form is:

$$U \supset (M \supset B) \tag{5.10}$$

A typical sequence of inferences that would be required to obtain $U \supset B$ given $M$ and the inference rules above is:

[3] In this case, it will be only a subset of the valid inferences that excludes inferences about negation, because I have omitted the other three axioms.

From (5.8), (5.10) and MP: $(U \supset M) \supset (U \supset B)$
From $M$, (5.7) and MP: $U \supset M$
From the Previous 2 steps and MP: $U \supset B$

Since the synthetic rule MP is used at each stage, we potentially have three contextual effects. Furthermore, the intermediate forms derived can hardly be described as those that intuitively follow from the information provided. Indeed, Johnson-Laird (1983) provides experimental evidence that human beings do not make these kinds of inferences in normal circumstances. Furthermore, if introduction rules such as (5.7) are discounted, it becomes impossible to make the above inference.

We can see from the above example that it is important that the logical rules should allow for ease of expression of encyclopædic knowledge. It is tempting to use this criterion to create many convenient logical connectives and redundant inference rules. This would result in two problems:

1. If there is redundancy in the inference rules, inflated amounts of contextual effects will result. This may mean that certain syntactic constructs are more Relevant than others which express the same semantic information, which is clearly undesirable[4].
2. The deductive device must scan the deductor memory for each input of every rule in the logical memory. Increasing the number of logical rules has a seriously debilitating effect on the efficiency of the deductive device.

In conclusion, the logical rules should be an adequately expressive, minimal set of elimination rules that express (some of) the behaviour of the usual logical particles. The set used in Crystal is provided in appendix D.

### 5.2.2 Encyclopædic Entries

Encyclopædic entries should express the meaning of their associated concepts. Each concept should correspond to some (possibly compound) word sense. Conceptual information is defined solely in terms of other concepts and the logical connectives. Therefore, predicates that do not correspond to word senses should not be used at all. This is reasonable because, much as it can be argued that all humans share knowledge of a core set of word meanings, it is far less obvious that they share other information. If speakers can assume that the hearers' encyclopædic entries are structured in similar ways to their own, it is feasible for them to predict what will be mutually manifest.

#### Naïve Physics and Metaphysics

I have taken the approach that encyclopædic information should express folk intuitions about their associated concepts. This approach, when applied to the material world, is known as *naïve physics* (Hayes, 1984). Rather than attempting to express scientific theories about the world, this approach provides the background intuitions that most scientists take for granted. For example, Hayes (1985) attempts to formalise an ontology for liquids, attempting to explain notions of containment, wetness, amounts, directions and other concepts fundamental to liquid shape and flow. A similar approach, called naïve or ethno-metaphysics, can be applied to metaphysical notions such as time, action and belief.

[4]This does not apply to the grammatical structure of the utterance; it is only undesirable for internal logical representations.

For example, Allen and Kautz (1985) have taken such an approach to temporal reasoning. Illustrations of physical and metaphysical knowledge will be provided in section 5.5.

#### Obtaining Knowledge From a Dictionary

I have shown that Relevance is highly dependent on the inferential behaviour of the information in the deductor memory. This information is dependent on the organisation of the conceptual memory because its contents are determined by what information is addressed by a particular concept during context expansion. Consequently, it is important to have a principled way of determining what should be put in an encyclopædic entry. My solution to this problem was to form an encyclopædic entry from the entry corresponding to its word-sense in a dictionary.

In order to elicit knowledge for a particular word sense, I attempted to encode its description directly from Longmans (1987) by hand. This particular dictionary was used because:

1. It is intended for use by non-native speakers, so the philosophy behind the entries was to make them as straightforward and intuitively understandable as possible. Generally, an entry is defined in terms of several other simple concepts, rather than just using synonyms.

2. All entries draw from a core vocabulary of 2000 words. This meant that these words were relatively common and that there would be a higher probability of obtaining a defined concept on context expansion.

3. Attempts had been made to avoid circularities and inconsistencies amongst entries.

4. The dictionary was available in machine readable form so it was possible that, at some time in the future, the lexicon could be extended automatically.

Zadrożny (1987) also argues the efficacy of the dictionary-based approach to encyclopædic knowledge elicitation. However, he uses this knowledge in a different manner.

Elaborate logical schemes for representing knowledge, such as that found in Hobbs et al. (1987), have been developed to handle complex knowledge. Unfortunately, given the limited scope of a PhD project, a rather simpler scheme had to be employed, which is described in section 5.3.

### 5.2.3 A Methodology for Obtaining Encyclopædic Data

Encyclopædic information for a word-sense is established as follows:

1. Conceptual information about the word sense is manually extracted from Longmans (1987) as logical forms. Information that cannot be conveniently translated is either simplified or omitted.

2. Any information obviously excluded from the dictionary entry is added. This includes information specific to a particular person, for example the fact that Buttercup is a Rottweiler may be added under the entry for Rottweiler; such information would not appear in a dictionary, but should appear in an encyclopædic entry. The information may also include typical properties, uses and scenarios involving an instantiation of this concept.

3. The encyclopædic entry is tested by processing utterances that contain it. If the "expected" results do not follow, the entry is updated. There are two major ways that the entry may not perform as expected:

   (a) It does not connect into the context in the expected way. In which case, the entry must be modified or strengthened to ensure that it does. So, for example, it might be the case that Nutkin is hungry and finds an object called a nut. Intuitively, we expect the nut to be the food rather than the metallic object used for fastening. One of the reasons that we expect this is because we know that squirrels eat nuts. Consider a situation where the encyclopædic entry for hunger includes:

   $$\begin{aligned}&\mathrm{Hungry}(!x) \wedge \mathrm{Food}(!f) \wedge \\ &\quad \mathrm{Likes}(!x, !f) \supset \mathrm{Eats}(!x, !f)\end{aligned} \tag{5.11}$$

   Given a sentence like:

   $$\mathrm{Hungry}(nutkin) \wedge \mathrm{Squirrel}(nutkin) \tag{5.12}$$

   it is reasonable to assume that since Nutkin has found the object, then he eats it. However, Longman's does not state that nuts are usually edible or that squirrels like nuts so the expected inference does not go through. Thus, the appropriate information must be added to the relevant entries.

   (b) It connects into the context in an inappropriate way. This is usually due to the presence of inadequate non-demonstrative inferences (see chapter 8). In order to circumvent this, the theory must usually be strengthened with negative information. Consider, for example, a situation where it is known that an island is in the middle of a lake. Since islands are surrounded by water and that lakes are surrounded by land, a typical guess that Crystal could make is that the water which surrounds the island is the water in the lake. However, with Crystal's original encyclopædic entries for "island", "middle" and "lake", it also made the incorrect guess that the land surrounding the lake is the island. Given the definition of "middle" in Longman's, no inconsistency was found. The entry for "middle" had to be strengthened with information that explained that if $X$ contains $Y$, then $Y$ cannot contain $X$.

SW believe in some form of mental logic; the forms in the deductive device represent actual thoughts. They are significant purely due to their syntactic form, which should therefore be something that we judge to be manifest in the circumstances dictated by a particular example.[5]. Therefore when examining the assertions in the deductive device, I checked each assumption to see whether it was intuitively plausible given the situation (e.g. Nutkin may eat the nut in the interpretation of (5.12)) or implausible (e.g. something can contain itself). Implausible inferences include not only those that are clearly false but also those that are not likely to be entertained. For example, assuming *placido.domingo* denoted a human:

$$\mathrm{Day}(placido.domingo) \supset \mathrm{Interval}(placido.domingo)$$

[5]Such judgements are, of course, highly dependent on an individual's intuitions about pragmatic competence. However, there is some psychological evidence in Johnson-Laird (1983), for example, as to which forms we tend to entertain mentally.

is vacuously true, because the antecedent is false. However, it is not the kind of assumption that is normally manifest to us.

This testing phase is repeated until all of the assumptions in the deductive device seem plausible. The testing of the information in encyclopædic entries is distinct from judgements about Relevance. All this testing phase ensures is that the inferences in the deductive device seem reasonable. Consider a situation where there are two plausible interpretations of an utterance concerning Nutkin finding a "nut". Each interpretation must be reasonable according to the criteria above. However, Crystal may still select the wrong interpretation because it yields more contextual effects for less effort. Thus, one of the ways that RT is evaluated using Crystal is by considering which of several putative interpretations is chosen given that they are all reasonable.

## 5.3 An Encyclopædic Knowledge Representation

I have described the logical rules used by Crystal in appendix D. I will now describe the encyclopædic knowledge representation and explain how it can be used to represent assumption schemas.

I begin by sketching a syntax and semantics for the forms. Next, I explain how they are immediately normalised when they are input into Crystal so that they are in a form suitable for inferencing in the deductive device. I then show that the representation can be used to implement assumption schemas. Having done this, I demonstrate the fact that SW's notion of instantiation is somewhat nebulous, covering a variety of deductive and non-demonstrative phenomena; as a result, instantiation may or may not be a contextual effect, depending on the type of instantiation occurring. Finally, I explain how the encyclopædic information is practically interfaced to the deductive device in Crystal.

### 5.3.1 Syntax and Semantics

Knowledge is expressed using a slightly modified syntax of FOPC.

The basic types used by the system are constants, variables, functions, predicates, quantifiers, and connectives and the modal operator $\mathcal{D}$.

- Constants are written as lower case, italicised identifiers, such as *dragon004, bedpan026, aspidistra943, cup.cake007.* By convention, constants representing named individuals inherit that name, for example a person named Włodek would be represented by *włodek.*

- Variables are written as italicised identifiers preceded by an exclamation mark, eg: *!salami, !x, !wimple, !batmobile, !G.String.*

- Functions are written as capitalised identifiers, followed by their arguments in parentheses. The arguments can be any term. Typically, function identifiers are capitalised and end in "Of", e.g.: Intestines.Of(*!x*), Yodelling.Event.Of(*horatio*). Constants are special cases of functions; they can be considered to be functions with no arguments.

- Predicates have the same syntactic realisation as functions, but are internally represented using different types. Examples are: Spotted(*dick021*), Precarious(House.Of($!pig$)), Eats(Wolf.Of($peter$), Duck.Of($peter$)). Certain predicates, such as $<$ and $=$, can be written as infix operators.

- Quantifiers are one of $\exists$ and $\forall$.

- Connectives are one of $\wedge, \vee, \supset$ and $\neg$.

All of the above classes are represented as separate types in Crystal.

The syntax describes terms, atomic well-formed-formulas (WFFs), complex WFFs, default assumptions and assumptions.

- A *term* is a constant, variable or function.

- An *atomic WFF* is a predication of terms.

- A *complex WFF* is either an atomic WFF or a series of assumptions linked using the connectives and quantifiers in the usual way.

- A *default assumption* is a complex WFF preceded by the modal operator, $\mathcal{D}$.

- An *assumption* is either a complex WFF or a default assumption.

Ignoring default assumptions, the syntax and semantics of assumptions correspond to that of ordinary FOPC that can be found in Turner (1984) and Genesereth and Nilsson (1987). The semantics of default assumptions will be explained fully in chapter 7. However, informally, $\mathcal{D}P$ can be considered to mean "P is typically true".

A sample assumption is:

$$\forall !l.\, \text{Liverwort}(!l) \supset \exists !b.\, \text{Body}(!b) \wedge \text{Has}(!l, !b) \wedge \mathcal{D}\text{Lobed}(!b) \tag{5.13}$$

This is intended to be a translation of: "Liverworts typically have lobed bodies".

Terms can also be *sorted*, or given a type. However, sorted terms can be considered to be complex WFFs of ordinary FOPC[6] (see section 5.4 for more detail).

The equality predicate, $=$, is treated specially by the inference engine. The interface passes assertions of equality to the *equality sub-system* which handles them separately (see sections 4.5 and 8.4.5).

### 5.3.2 Normalisation

Information is input to the encyclopædic memory through a dedicated browser (see appendix A). It is input in a form convenient for the user but immediately converted into a form suitable for the deductive device using a simple process which I call *normalisation.* The browsers maintain a copy of the original form for further editing by the user, but only the normalised forms are accessed by the context expansion process; encyclopædic entries contain normalised assumptions. Utterance representations are similarly normalised before input to deductive device. Normalisation preserves the inferential behaviour of assumptions and therefore does not affect the measurement of Relevance. There are five stages of normalisation:

1. Macro expansion.

[6] I generally use the term "complex WFF" or "formula" to refer to the subset of assumptions that correspond to FOPC.

2. Sortal information completion.

3. Variable renaming.

4. Skölemisation.

5. Conversion to prenex form.

Most of these steps are standard practice for deductive systems (Charniak et al., 1987). I shall briefly consider each in turn.

### Macro Expansion

For user convenience, certain macros are defined that are effectively abbreviations for a longer form. For example, $\ni !p$ represents "the unique $!p$", so:

$$\ni !p.\, \mathrm{Paraclete}(!p) \tag{5.14}$$

is an abbreviation for:

$$\exists !p.\, \mathrm{Paraclete}(!p) \wedge \forall !p1.\, \mathrm{Paraclete}(!p1) \supset (!p1 = !p) \tag{5.15}$$

During the expansion phase, macros are recursively rewritten into their expanded forms.

### Completing Sortal Information

All terms are sorted. They have the form *Term:Sort*. However, on input, the sorts of every term need not be completely specified. Sorts are completed as follows:

1. If no sort is provided for an item, in the absence of more specific information it is assumed to be of sort "T", the most general sort.

2. If the term is a variable, a check is made to see if it is in the scope of a quantifier. If so, the sort of the term is compared with the sort of the quantified variable. The most specific sort replaces the sort of all instances of this variable.

For example:

$$\forall !b{:}\mathit{Bee}.\, \mathrm{Furry}(!b{:}\mathit{Bumblebee}) \wedge \mathrm{Insect}(!b) \tag{5.16}$$

will be converted into:

$$\forall !b{:}\mathit{Bumblebee}.\, \mathrm{Furry}(!b{:}\mathit{Bumblebee}) \wedge \mathrm{Insect}(!b{:}\mathit{Bumblebee}) \tag{5.17}$$

because "Bumblebee" is the most specific sort of "T", "Bee" and "Bumblebee".

### Renaming Variables

To prevent any name clashes between assumptions, all variables that occur within different assumptions are given unique identifiers. Variable identifiers consist of a public identifier and a private number. The combination of identifier and number is unique for each universally quantified variable within every assumption. It is only the identifier that is normally displayed, however. The system is responsible for assigning unique private numbers to the identifiers.

### Skölemisation

Skölemisation is the process by which existential variables can be removed from a given assumption or assumption schema. Such variables are replaced by *skölem functions*[7] which, intuitively, represent the anonymous entities referred to by the existential quantifiers. The arguments of a skölem function are dependent on the universally quantified variables within whose scope the existentially quantified variable lies. So, for example:

$$\forall c.\ \text{Cloud}(!c) \supset \exists !l.\ \text{Lining}(!l) \wedge \text{Has}(!c, !l) \wedge \text{Silver}(!l) \tag{5.18}$$

is converted into:

$$\begin{aligned}&\forall !c.\text{Cloud}(!c) \supset \\ &\quad \text{Lining}(\text{Sk.Lining.Of}(!c)) \wedge \text{Has}(!c, \text{Sk.Lining.Of}(!c)) \wedge \\ &\quad \text{Silver}(\text{Sk.Lining.Of}(!c))\end{aligned} \tag{5.19}$$

The existentially quantified variable "$!l$" is replaced by the skölem function Sk.Lining.Of($!c$). It is a function of $!c$ because $!l$ occurs within $!c$'s scope. Further information on this process can be found in Charniak et al. (1987).

Note that, by convention, skölem function identifiers are prefixed with "Sk.". Heuristics are used for naming skölem functions based on their surrounding context. So, for example, Sk.Lining.Of($!c$) was so named because "Lining" was the first property associated of $!l$ (and is probably also $!l$'s sort). Similar heuristics are used for naming skölemised event and interval variables. All of these heuristics can be overridden by the user.

### Conversion into Prenex Form

Name clashes between universally quantified variables within a particular assumption are removed by simply renaming one of the clashing variables. This permits the quantifiers to be moved as far to the left of the assumption as possible. The assumption is now in *prenex* form. The quantifiers need not be explicitly used in an assumption any more since their scope is over the whole sentence.

The assumptions are now in a form that can be used by the deductive device.

## 5.3.3 Implementing Assumption Schemas

In section 2.3.3, I discussed how SW's uninstantiated assumption schemas could be interpreted in a variety of disparate ways. Having made the decision that the deductive device should support a traditional logic, and having defined this logic, I am in a position to be more precise about the meaning of instantiation. It will be seen that SW's notion of instantiation, covers the disparate phenomena of deductive and non-demonstrative reasoning; it is not merely a matter of plugging a value into a box.

The apparently monolithic process of enriching assumptions and assumption schemas can be formalised logically in terms of:

1. Universal Quantification.
2. Default Reasoning.
3. Existential Quantification.

[7]This includes constants.

4. Abductive Reasoning.

In order to demonstrate this, I will translate fragments of definitions from Longman's and show how they can be expressed using both assumption schemas and my knowledge representation. Pieces of text in parentheses are ignored for the sake of perspicuity.

**Universal Quantification**

Consider the definition of "Jive":

A style of (very fast) dancing... (117)

This might be represented by:

The jive $\boxed{\phantom{xx}}$ is a kind of dance. (118)

A reasonable expression of this expressed in Crystal's knowledge representation is:

$$\forall !j.\, \mathrm{Jive}(!j) \supset \mathrm{Dance}(!j) \tag{5.20}$$

which is normalised to:

$$\mathrm{Jive}(!j) \supset \mathrm{Dance}(!j) \tag{5.21}$$

This kind of representation handles all valid generalisations.

In this context, instantiation can be seen to be equivalent to the FOPC notion of *universal instantiation.* This states that any universally quantified expression can be instantiated by a member of the set of quantified objects. For example, (5.21) can be instantiated to:

$$\mathrm{Jive}(\mathit{water.buffalo743}) \supset \mathrm{Dance}(\mathit{water.buffalo743}) \tag{5.22}$$

This highlights an interesting problem: much as the schema can be instantiated to *any* object, we would really only like to instantiate it with plausible objects, i.e. those that are jives[8]. The contents of the deductive device should express reasonable beliefs, as explained in section 5.2.3. However, the overall value of Relevance will only change if instantiation of schemas is regarded as a contextual effect. In this particular case, it is unlikely that any synthetic inferences, potentially altering Relevance, can be derived from (5.22) because the antecedent is false. Nevertheless, it is possible that (5.22) was the result of a synthetic rule application and could therefore be regarded as a non-trivial inference that contributes to the overall information count; this is intuitively not the case. I will address this problem again in section 5.4.

Since universal instantiation is a synthetic rule of inference, can it produce contextual effects? If instantiation always involved suitable objects, then the answer would be positive. However, (5.22) can hardly be regarded as a contextual implication.

Another way of looking at a quantified expression is as a kind of abbreviation or template for all of its instantiations[9]. As an example, consider a rather scanty domain of three

[8] I am assuming that *water.buffalo743* denotes an actual water buffalo rather than a dance called "the water buffalo".

[9] This is a consequence of Herbrand's theorem.

objects denoted by *water.buffalo743*, *jive451* and *agent007*. Under these circumstances, (5.21) can be seen to be a shorthand for:

$$\text{Jive}(\textit{water.buffalo743}) \supset \text{Dance}(\textit{water.buffalo743}) \tag{5.23}$$

$$\text{Jive}(\textit{jive}451) \supset \text{Dance}(\textit{jive}451) \tag{5.24}$$

and

$$\text{Jive}(\textit{agent}007) \supset \text{Dance}(\textit{agent}007) \tag{5.25}$$

Adopting this view, instantiation is not a rule of inference like, say, modus ponens. Indeed, there is no need for an explicit instantiation rule — it is built into the unification algorithm. When instantiation is allowed as a separate rule of inference, applying the logical rules merely requires tests of equality. Alternatively, unification can be regarded as a kind of modified equality test that checks if a particular instantiation of a schema will result in the successful application of a logical rule. Either way, universal instantiation is not an inference, and should not be considered as something which can effect Relevance.

To summarise, universally quantified assumptions express valid generalities. Their instantiation is universal instantiation, but should not be considered to be a contextual effect if the universal quantifier quantifies over the entire domain of objects.

### Default Reasoning

Consider the following definition of "Joust":

(A fight)...*esp.*[10] a sport. (119)

Part of this could be used in a prototyping schema:

The joust $\boxed{\phantom{xxx}}$ is a sport. (120)

This piece of knowledge differs from (118) in that does not reflect a universal generality. Jousts are not *always* performed in sport. Instantiation by a term that is a joust is not a valid rule of inference.

However, if we are writing this in a deductive framework, we would like to express it as a valid assumption. The means for doing this is the $\mathcal{D}$ operator, which can be used to express information that is typically true[11]. $\mathcal{D}P$ can be viewed as an instruction to complete one's knowledge by assuming $P$, if this is consistent. Thus, (120) might be expressed as the normalised form:

$$\text{Joust}(!j) \supset \mathcal{D}\text{Sport}(!j) \tag{5.26}$$

This can be considered to be a valid schema, representing all of its instantiations, in the same way as (5.21). However, depending on the context, a particular joust may or may not be regarded as a sport. Sentences containing default assumptions are used to represent prototypical information of the kind expressed by frames and scripts.

[10]This means "especially".

[11]The semantics of $\mathcal{D}$ will be further discussed in chapter 7.

### Existential Quantification

Consider the definition of "Yuppie":

A (young person) in a (professional) job... (121)

which SW might represent with:

The yuppie $\boxed{\phantom{xx}}$ works as a $\boxed{\phantom{xx}}$. (122)

This is not saying that the yuppie has every job, it is saying that he has a job of which we do not know the identity. SW describe how such objects can be introduced by assumption schemas and later strengthened to produce contextual effects. Thus the last $\boxed{\phantom{xx}}$ is not instantiated in the same sense as for the above schemas — it represents a single anonymous entity. It is possible to refer to this anonymous entity in normal speech. For example, there is no problem finding a co-referent with "her job" in the following utterance:

Charlotte has become a bit of a yuppie.
She's very pleased with her job. (123)

A logical translation of (122) might be:

$$\forall !y.\, \mathrm{Yuppie}(!y) \supset \exists !j.\, \mathrm{Profession}(!y, !j) \tag{5.27}$$

which is normalised to:

$$\forall !y.\, \mathrm{Yuppie}(!y) \supset \mathrm{Profession}(!y, \mathrm{Sk.Profession.Of}(!y)) \tag{5.28}$$

The skölem function, "Sk.Profession.Of" is the anonymous constant described above.

### Abductive Reasoning

Assumption schema (122) can be instantiated, say, by a person named Trotsky to yield:

The yuppie Trotsky works as a $\boxed{\phantom{xx}}$. (124)

Later, it might be independently discovered that Trotsky is a pig farmer. We would then be in a situation where we knew that he had some anonymous profession, and that he was also a pig farmer. We may feel that these two professions are one and the same. Thus, the $\boxed{\phantom{xx}}$ can be instantiated to pig farmer:

The yuppie Trotsky works as a pig farmer. (125)

This version of instantiation amounts to an abductive (non-demonstrative) assertion of equality. This can be demonstrated using Crystal's logical knowledge representation.

Schema (5.28), has been used to infer:

$$\mathrm{Profession}(\mathit{trotsky}, \mathrm{Sk.Profession.Of}(\mathit{trotsky})) \tag{5.29}$$

Now it is also believed that:

$$\mathrm{Profession}(\mathit{trotsky}, \mathit{pig.farmer}) \tag{5.30}$$

These two expressions only become identical if we make the non-demonstrative assertion:

$$\mathrm{Sk.Profession.Of}(\mathit{trotsky}) = \mathit{pig.farmer} \tag{5.31}$$

This cannot be viewed as logical instantiation, but is a significant non-trivial inference and therefore a contextual effect. Note that in this case, the "$\boxed{\phantom{xxxx}}$" translates into an anonymous constant, not an uninstantiated variable. Since it is possible to non-demonstratively "instantiate" logical forms with no free variables, I have not bothered to distinguish assumptions and assumption schemas in the syntax of Crystal's knowledge representation provided in section 5.3.1. For the remainder of this thesis, I use the terms "assumption" and "assumption schema" synonymously.

In conclusion, it is possible to implement SW's nebulous notion of assumptions/assumption schemas in the logic that I have described. Instantiation of assumption schemas can be seen to be either universal instantiation, introduction of anonymous terms or abductive assertions of equality. Only the latter constitutes a contextual effect.

### 5.3.4 Interfacing the Conceptual Memory to the Context Expansion Process

I have broadly introduced the kind of information that may be found in an encyclopædic entry. In this brief section, I will explain how it is interfaced to the context expansion process. This is an implementation detail, included for completeness, and is of no theoretical import.

The conceptual memory will deliver a list of encyclopædic entries when passed a list of concepts by the context expansion process (see chapter 4). However, the conceptual memory also handles lexical ambiguity in a special way.

Concepts representing different senses of a lexical item of a particular category are given identifiers which differ only in their final few characters, which will be numbers in this case.

When an ambiguous lexical item is input to the system, it is represented as the identifier common to both senses. For example, in Crystal the predicate representing "nut" has two senses: "Nut1" and "Nut2". "nut" is represented by the predicate "Nut" in the utterance representation. When this predicate is passed to the conceptual memory, the memory passes back the entries for both "Nut1" and "Nut2". These entries are tagged as ambiguous, so that the context expansion process can pass back two alternative augmentations to the context: one with Nut1, the other with Nut2. The context expansion process also replaces the utterance representation with a list of its possible senses that could arise through lexical ambiguity (by replacing "Nut" with either "Nut1" or "Nut2" in this case).

### 5.3.5 Summary

A suitable encyclopædic knowledge representation is FOPC enhanced with the modal "default" operator, $\mathcal{D}$. Crystal itself uses a normalised version of these forms.

It is possible to implement the various facets of assumption schemas in this representation scheme. Having done this, it is apparent that SW's "$\boxed{\phantom{xxxx}}$"s are a very vague notion. If they represent a free variable, it may be universally quantified, in which case its instantiation is not a contextual effect. They may represent some kind of prototypical schema, in which case its instantiation may be regarded as a specialised sort of universal instantiation. It may be an anonymous constant that represents some implicit object, in which case its instantiation can be an abductive guess of equality and therefore a contextual effect. The way these notions can be used to implement script and frame-like notions will be seen in section 5.5. In chapter 6, I will explain how my descriptions of instantiation can be used in the calculation of a modified information measure that overcomes some deficiencies of RT.

The encyclopædic memory is interfaced to the deductive device in a straightforward way using the context expansion process.

## 5.4 Hybrid Representations: Sorts

Hybrid knowledge representations combine some means of placing structure on information intended to describe world knowledge with a separate deductive capability. Some of the major uses of such representations are to provide efficient inferencing and to prevent certain inferences from taking place. We have already seen in section 5.3.3 that it is undesirable for certain forms to be instantiated by certain objects; this can be realised using hybrid knowledge representations.

In this section, I briefly survey such representations and explain their advantages. I then show by means of a specific implementation of a *taxonomic logic* that it is possible to incorporate such representations into the deductive device as long as certain criteria are fulfilled: the implicit inferences being performed by the structured knowledge representation must be counted or made explicit.

### 5.4.1 A Brief Overview of Hybrid Systems

Frisch (1989) identifies five categories of hybrid representations:

**Sorted Logics** These integrate logical and sortal languages (e.g. Cohn (1987)).

**Taxonomic Logics** These logics restrict the sortal system to a simple taxonomic hierarchy, e.g. Frisch (1986).

**Frame-based Systems** These mix assertional and terminological information (i.e. propositional world information and structural information), for example Brachman et al. (1983b) and Vilain (1985).

**Horn Systems** These mix Prolog-style reasoning using Horn clauses and a taxonomic hierarchy (Frisch et al., 1983).

**Specialised Inference Engines** These utilise special theorem provers for dedicated representations (e.g. Stickel (1985)).

Sorted and taxonomic languages augment a logic, providing it with sorted terms. A separate language is used to describe the relationship between the sorts. This is also true of horn systems, which also allow a certain amount of procedural reasoning.

Frame-based systems attempt to elucidate and formalise the relationship between frame-like notions and deduction. A structural or terminological component provides information about the way terms denoting real-world information can be structured. The terminological component is accompanied by a specialised subsumption-checking mechanism, designed to work efficiently with these representations. Additionally, a deductive component allows assertions and inferences to be drawn from the terminological information.

Specialised inference engines are theorem provers designed to work with specialised knowledge representations. These theorem provers may incorporate some non-deductive inference capabilities based on certain properties of the knowledge representation (Stickel, 1988).

### 5.4.2 Advantages of Hybrid Knowledge Representations

As well as the advantages that structured information representations and logical representations individually possess, hybrid representations have the following advantages:

#### Superior Information Structure

Hybrid systems encourage the user to distinguish between assertional and structural relations in world information, a problem that often leads to obfuscations in knowledge representation (Brachman et al., 1983a).

Subsumption checking for most interesting structures is NP-hard (Nebel, 1988). Hybrid systems alleviate (but do not necessarily solve) this problem by:

1. Making the subsumption mechanism a separate, specialised process which can therefore operate faster than a general theorem prover.

2. Placing restrictions on the structuring mechanism either to make the checking tractable or to make it more tractable for the majority of cases.

#### Undesirable Instantiations can be Prevented

It was shown in section 5.3.3 arbitrary universal instantiation should not be allowed in a Relevance-based system because the resulting forms may not be reasonable. Hybrid knowledge representations may prevent this by disallowing certain instantiations on the grounds that the resulting expressions would be semantically ill-formed. Reconsider:

$$\text{The jive } \boxed{\quad} \text{ is a kind of dance.} \tag{118}$$

This might be represented using a hybrid system that allows sorted terms as follows:

$$\text{Dance}(!j{:}\mathit{Jive}) \tag{5.32}$$

This could never be instantiated as:

$$\text{Dance}(\mathit{water.buffalo{:}Buffalo}) \tag{5.33}$$

assuming Jive and Buffalo are incompatible sorts.

#### Preventing Unwanted Inferences

By extension of the instantiation argument, hybrid knowledge representations can prevent certain inferences being made because the antecedents to the inference rule are incompatible in some way.

Consider:

$$\text{Male}(!x) \land \neg\text{Married}(!x) \land \text{Human}(!x) \supset \text{Bachelor}(!x) \tag{5.34}$$

and

$$\text{Male}(\mathit{terrapin333}) \tag{5.35}$$

Applying conjunctive modus ponens to (5.34) and (5.35), (5.36) can be derived:

$$\neg\text{Married}(\textit{terrapin333}) \wedge \text{Human}(\textit{terrapin333}) \supset \text{Bachelor}(\textit{terrapin333}) \tag{5.36}$$

This might be a contextual implication, however it is clearly ludicrous: only humans can be bachelors.

A solution is to sort the variables in (5.34):

$$\text{Male}(!x{:}\textit{Human}) \wedge \neg\text{Married}(!x{:}\textit{Human}) \wedge \text{Person}(!x{:}\textit{Human}) \supset \text{Bachelor}(!x{:}\textit{Human}) \tag{5.37}$$

This not only prevents unwanted contextual effects, but will speed up the inference engine because less objects should unify.

#### Compatibility Checks can Aid Non-Demonstrative Inferencing

In Crystal, a non-demonstrative inference component makes guesses concerning the equality of certain objects. Of course, guesses are not a valid form of reasoning, but they often amount to non-trivial inferences. Thus, it is important that the guesses are not entirely ludicrous, for example the referent of "he" could never be a woman. Some kind of compatibility-checking mechanism could greatly improve the quality of these guesses.

This also seems psychologically plausible. We can tell that certain objects are incompatible instantaneously, apparently without the need for reasoning about their properties. For example, we know that dogs are not trees, despite the fact that they may share certain properties (like being brown). By contrast, other statements seem to require more extensive reasoning, like:

Margaret Thatcher will never play cricket for Albania. (24)

The instantaneous nature of certain kinds of compatibility checking suggests that we may ourselves have some specialised system dedicated to this task.

Another way of looking at a specialised compatibility checker is as a system for cutting down on unnecessary context expansions. Without compatibility checks, it may require several expansions to obtain a contradiction from two incompatible terms. This can circumvented by a dedicated system which is effectively performing selective "context look-ahead" for these cases.

### 5.4.3 Sorting Terms in Crystal

In Crystal, I have adopted a taxonomic logic to solve the above problems because:

1. It only requires a modification of the unification algorithm.
2. It has a direct translation into FOPC.
3. It is simple and tractable.

I shall now describe the syntax and semantics of the sortal system, and explain its interface with the deductive device.

### The Syntax of Terms

The definition of terms is modified to incorporate a sort. Thus, constants, functions, and variables all have an associated sort. The sort is denoted by an identifier, associated with the term by means of a colon. Thus, the following are terms:

*beast666:Animal, Number.Of(beast666:Animal):Integer, !f:Ferret*

### Semantics

I shall treat unquantified and quantified terms separately, providing their semantics in unsorted FOPC.

- For unquantified terms:

$$\textit{Term:Sort} \equiv \mathrm{Sort}(\textit{Term}) \tag{5.38}$$

  So, for example:

$$\textit{beast666:Yak} \equiv \mathrm{Yak}(\textit{beast666}) \tag{5.39}$$

- For universally quantified terms:

$$\forall !x{:}\mathrm{P}.\,\Phi \equiv \forall !x.\mathrm{P}(!x) \supset \Phi' \tag{5.40}$$

  where $\Phi' = \Phi[!x{:}\mathrm{P}|!x]$, or all free typed occurrences of $x$ in $\Phi$ are replaced by $x$ itself. Thus,

$$\forall !j{:}\textit{Jive}.\,\mathrm{Dance}(!j{:}\textit{Jive}) \equiv \forall !j.\,\mathrm{Jive}(!j) \supset \mathrm{Dance}(!j) \tag{5.41}$$

- For existentially quantified terms:

$$\exists !x{:}\mathrm{P}.\,\Phi \equiv \exists !x.\mathrm{P}(!x) \wedge \Phi' \tag{5.42}$$

  For example:

$$\begin{aligned}&\exists !j{:}\textit{Job}.\,\mathrm{Profession}(\textit{trotsky}, !j{:}\textit{Job}) \equiv\\ &\qquad \exists j.\,\mathrm{Profession}(\textit{trotsky}, !j) \wedge \mathrm{Job}(!j)\end{aligned} \tag{5.43}$$

This translation has certain implications. Sorts are no more than abbreviations for predicates. Thus, the constraints that can be placed on sorts must be at least those placed on predicates. In particular, sorts must represent concepts; "Six.Armchairs.And.An.Aardvark" is not a reasonable concept and therefore not a reasonable sort.

Which concepts can be usefully sorted? This is a difficult question, which poses problems for most sortal systems. The most obvious candidates are natural kinds and categories of object. In particular, terms that are not properties. Hence, "armchair" and "wimple" make reasonable sorts, but "married" or "bachelor" do not. However, certain events might be considered to be sorts. It is immediately obvious to us that "sleeping" is not "killing". This suggests that they should be considered to be distinct sorts. However, it is less clear how "hitting" and "killing" should be related to each other. This issue of what can be sorted is important to a hybrid system, but outside the scope of this thesis.

For the purposes of this thesis, I will suggest, with little justification except for intuitive appeal, which terms might be sorted:

- Terms denoting nouns, especially "natural categories": objects and natural kinds.
- Events and states. These notions are discussed in section 5.5.1.
- I have avoided sorting WFFs because it is less easy to categorise them. Since an appropriate sort structure would be very sophisticated, the subsumption algorithm would become unnecessarily complex. This would detract from the principle that most of the significant inferencing should be performed by the controller rather than the compatibility mechanism (see section 5.4.6). Therefore, constituents such as adjectives and adverbs that translate into WFFs are not sorted.

### 5.4.4 The Sort Hierarchy

#### The "isa" and Subsumption Relations

The sort hierarchy reflects the structural relations between the various sorts. The sort hierarchy is defined in terms of the "isa" relation. Formally:

$$sort_1 \text{ isa } sort_2 \equiv \forall !t.\, !t{:}sort_1 \supset\, !t{:}sort_2 \tag{5.44}$$

Let the "is subsumed by" relation, "$\sqsubset$", be the transitive closure of the "isa" relation, i.e:

$$\begin{aligned}&(sort_1 \sqsubset sort_2) \equiv \\ &\quad (sort_1 \text{ isa } sort_2) \vee \\ &\quad \exists sort_3\, (sort_1 \text{ isa } sort_3) \wedge (sort_3 \sqsubset sort_2)\end{aligned} \tag{5.45}$$

The sort hierarchy can be drawn as a directed graph whose vertices are the sorts, and whose edges are defined by the isa relation such that there is a directed edge from $sort_2$ to $sort_1$ if $sort_1$ isa $sort_2$. In this scenario, the $\sqsubset$ relation states whether there is a path from its second argument to its first.

#### The Structure of the Hierarchy

In order to make the sorts more powerful and ensure the computation of compatibility is tractable, the following extra constraints are placed on $\sqsubset$:

$$\forall s \in \Psi.\, s \sqsubset T \tag{5.46}$$

$$\forall s \in \Psi.\, (s = T) \vee (s \not\sqsubset s) \tag{5.47}$$

$$\forall s_1, s_2 \in \Psi.\, (s_1 \neq s_2) \wedge (s_1 \sqsubset s_2) \supset (s_2 \not\sqsubset s_1) \tag{5.48}$$

$$\begin{aligned}&\forall s_1, s_2, s_3 \in \Psi.\, (s_1 \sqsubset s_2) \wedge (s_1 \sqsubset s_3) \supset \\ &\quad ((s_2 \sqsubset s_3) \vee (s_3 \sqsubset s_2) \vee (s_2 = s_3))\end{aligned} \tag{5.49}$$

The first constraint states that there must be a most general sort, $T$. On a sort graph, every sort is connected to the $T$ vertex. (5.47) states that no sort (except for "T") subsumes itself. (5.48) prevents cyclic definitions of sorts. (5.49) enforces a tree structuring on the graph. Transitivity of $\sqsubset$ follows from its definition.

A fragment of Crystal's sort hierarchy is illustrated in figure 5.2.

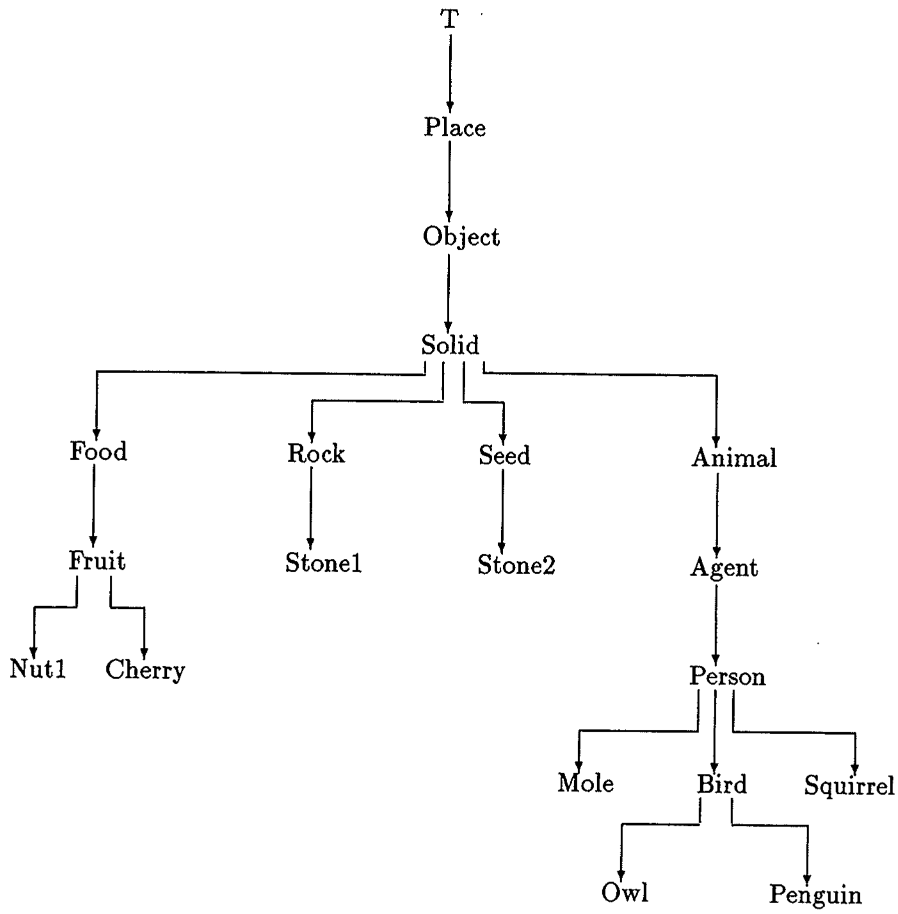



Figure 5.2: A Fragment of a Simple Sort Hierarchy

### Compatibility of Sorts

Formally, compatibility of sorts can be defined using the relation Compatible:

$$\text{Compatible}(s_1, s_2) \equiv (s_1 \sqsubset s_2) \vee (s_2 \sqsubset s_1) \vee (s_1 = s_2) \tag{5.50}$$

Informally, sorts are compatible if they are equal to each other or connected in some way.

## 5.4.5 Constructing the Sort Hierarchy

A general algorithm that checks a graph for cycles is NP-hard. This means that an algorithm that checks a sort hierarchy for conformance to the constraints detailed in section 5.4.4 is intractable.

The solution adopted in Crystal is to allow only a single sort to be added at a time using a *sort browser* (see appendix A). The algorithm for input is as follows:

1. A sort is input by specifying its name and its parent (the vertex with which it is associated via the isa relation).

2. It is checked that the new sort is subsumed by "T".

3. It is checked that the parent is not subsumed by the new node.

If any of these checks fail, the new sort is not added.

## 5.4.6 Deduction, Hybrid Representations and RT

Having rigorously defined sorts, we are left with the problems of determining how sorts should be used and the relationship of their usage with contextual effects.

### How Should Sorts be Used?

When a term is declared as having a particular sort, many other synthetic implications may follow from it. Consider figure 5.2. If the term, *orville:Owl* is present in the deductor memory, we are implicitly inferring that Orville is a bird, a person, an agent, an animal, a solid, an object and a place. If these were explicitly asserted, then they would be regarded as contextual effects.

However, all of the uses of sorts described in section 5.4.2 were to *prevent* unwanted inferences rather than to provide a separate, more efficient inference system. If sorts are only used to block inferences, they are being used in a passive way; the sorts subsumed by a particular sort will not be contextual effects. It is only when a sorted term is physically present in the deductive device that it counts as information.

Thus, following Frisch (1989)[12], the interface to the sortal system is in the unification algorithm[13]. Unwanted inferences are prevented by ensuring that only Compatible terms will unify. During unification, substitutions have the most specific sort of the two terms.

[12] The original inspiration for my approach began with the somewhat different sortal system described in Milward (1986).

[13] The semantics are based on Frisch's, the unification algorithm is loosely based on Milward's.

The consequence of this is that synthetic implications inherit the most specific sort of their antecedents. Thus, for example, given:

$$\text{Male}(!p{:}\mathit{Person}) \wedge \neg\text{Married}(!p{:}\mathit{Person}) \supset \\ \quad \text{Bachelor}(!p{:}\mathit{Person}) \tag{5.51}$$

and

$$\text{Male}(\mathit{piotr}{:}\mathit{Russian}) \tag{5.52}$$

will result in the inference:

$$\neg\text{Married}(\mathit{piotr}{:}\mathit{Russian}) \supset \text{Bachelor}(\mathit{piotr}{:}\mathit{Russian}) \tag{5.53}$$

if the sort Russian is subsumed by the sort Person.

### Sorts and Information

Consider the deductive instantiation of some sorted schema:

$$\text{Watch.Dog}(!d{:}\mathit{Dog}) \tag{5.54}$$

might be instantiated to:

$$\text{Watch.Dog}(\mathit{cerberus}{:}\mathit{Dog}) \tag{5.55}$$

Should this be regarded as a contextual effect? Unlike the case of unsorted schemas, the answer is "yes".

Considering the semantics, (5.54) is equivalent to:

$$\text{Watch.Dog}(!d) \supset \text{Dog}(!d) \tag{5.56}$$

It follows that instantiating (5.54) as (5.55) is implicitly making the inference:

$$\text{Dog}(\mathit{cerberus}) \tag{5.57}$$

which is a contextual effect. Indeed, when an inference inherits the most specific sort, this inference is effectively made explicit.

However, instantiation of quantified variables of the generic sort "T" is not a contextual effect. Consider:

$$\text{Belief}(x{:}T) \tag{5.58}$$

This is equivalent to:

$$\text{Belief}(!x) \supset \text{T}(!x) \tag{5.59}$$

Instantiation of this schema by any belief, $b$, will result in the inference $\text{T}(b)$. However, since $T$ is true of *any* term, it is effectively just "true". The inference "true" is not a contextual effect: it would be rejected by the interface[14].

[14]Terms of sort $T$ are of particular use when describing reified logics (see section 5.5.1).

#### Implementing Sorts

There are two ways of implementing sorts in Crystal.

1. Incorporate an information counter in the unifier/inference controller that measures the information inferred by instantiating sorted variables.

2. Do not treat instantiation as a potential information source, but duplicate the inferences that would be made by sorts. For example, (5.54) would actually be phrased as:

$$\text{Watch.Dog}(!d{:}\mathit{Dog}) \supset \text{Dog}(!d{:}\mathit{Dog}) \tag{5.60}$$

   This would ensure that when this schema is instantiated, its contextual effects are made explicit.

In practice, the second option was taken because it meant that the interface could count all of the contextual implications and the sole function of the sortal system was to prevent certain inferences being drawn.

### 5.4.7 Hybrid Representations: Conclusion

It is possible to use hybrid representations within the deductive device if:

1. The hybrid representation can be expressed in the logic of the deductive device.

2. EITHER the terminological component is only used to block inferences,

3. OR the implicit inferences made by operations such as instantiation are either counted or made explicit through redundancy.

## 5.5 Representing Time, Events, Frames and Scripts

The logical rules of the deductive device describe an incomplete implementation of FOPC. This is coupled with the reason maintenance process to support the operator $\mathcal{D}$, which provides some non-monotonic reasoning capability. In this section, I will discuss how more complex reasoning about beliefs, actions, time and prototypes have been incorporated into the deductive device, and the problems introduced by these enhancements.

First of all, I will consider the important issues for a temporal logic that is compatible with my configuration of the deductive device. I will then select Allen's temporal logic as a suitable logic, and suggest how the deductive device might be further extended to handle other modalities. At this point, Crystal's entire knowledge representation will have been explicated, and I go on to show how it can be used to provide a logical interpretation of frame- and script-like notions.

### 5.5.1 Temporal Logic

The external world cannot conveniently be regarded as a static entity: it continuously changes state. Furthermore, our actions and other events can effect such state changes. FOPC has no built-in capability for reasoning about chronological change and events, so this must be incorporated into the deductive device.

There are two ways that a temporal logic can be incorporated into the deductive device.

1. By altering the logical rules to express a temporal logic.
2. By expressing the semantics of the temporal logic in FOPC at the encyclopædic level.

Each possibility will be dealt with in turn.

### Modification of the Logical Rules

The logical rules could be modified to handle modal tense operators, such as "past" and "future" (Van Benthem, 1982). In this way, utterances such as:

Zadok was yet to be a priest (126)

would be translated into:

$$(\mathbf{PAST}\,(\mathbf{FUTURE}\;\mathrm{Priest}(\mathit{zadok}))) \tag{5.61}$$

The main advantage of this approach is that the logical translations of utterances seem intuitively appropriate, in the sense that they have a close relationship to the linguistic entities expressed by the utterance; this often results in inferences yielding intuitively plausible contextual effects[15].

The main disadvantage of such approaches is that the convenience of using FOPC is lost and there is much controversy as to exactly which temporal concepts should have dedicated operators. Many of the major temporal logics such as those described in Kowalski and Sergot (1986), Moens and Steedman (1987) and the philosophical logics described in Van Benthem (1982) have very different approaches.

### Adding Extra Encyclopædic Information

Various researchers have resorted to expressing what RT considers to be logical information at the encyclopædic level in order to take advantage of the convenient, well-understood properties of FOPC (Moore, 1985a; Shoham, 1986). Such logics are known as *reified* logics.

Four of the most popular reified temporal logics are those of McDermott (1982b), Allen (1984), Shoham (1986) and Reichgelt (1986). These all have in common the fact that they have complex axiomatisations and that expressions in these languages are less intuitive, for example (126) might be phrased as:

$$\exists t_1, t_2.\,(t_1 < \mathit{NOW}) \land (t_1 < t_2) \land \mathrm{HOLDS}(\mathrm{Priest}(\mathit{zadok}), t_2) \tag{5.62}$$

Note that notions of states, events and times are made explicit in this logic, even when they are not explicitly mentioned in an utterance.

### Choosing a Logic

Ultimately, I implemented an incomplete version of Allen's logic (using only elimination rules) because:

1. It has well-defined semantics in terms of FOPC, thus no changes needed to be made to the deductive device, and its behaviour was still predictable.

[15] Much as appeals to intuition are not very satisfying arguments, it is often the case that such appeals must be made when discussing the plausibility of contextual effects.

2. Allen's logic is an example of a theory of naïve metaphysics. It has a relatively elegant axiomatisation as compared with the other reified logics, and was intended for naïve physics reasoning.

3. It has a reasonable ontology of events, processes, intervals etc. Turner (1984) states:

   > I prefer the account of Allen's to that of McDermott if only because events are taken as primitive (which seems intuitively sound) and 'chunks of time' are utilised instead of instants.

   The adoption of intervals as a temporal representation is also endorsed in Dowty (1979) and Van Benthem (1982).

4. Allen's logic axiomatises only the basic concepts of the language, so there is considerable freedom of expression. There are several ways of implementing the notions which are often frozen into modal logics of time, like "past", "before" etc.

5. The logic makes it possible to distinguish between the occurrence of an event and its mention. This proves very useful when describing prototypical events such as scripts (see section 5.5.3).

Turner and other researchers criticised Allen's original formulation for having inadequate semantics and not sufficiently distinguishing between intervals, moments (intervals of short duration) and points. These criticisms have largely been overcome in Allen and Hayes (1987), Hayes (1987) and Tsang (1987).

However, there are disadvantages in using Allen's logic:

1. Since the logic is reified, assumptions are expressing what would normally be regarded as logical rules. As a result, inferences that would normally be regarded as analytic implications become synthetic implications.

2. It is necessary to load the encyclopædic information about time when it is mentioned in an utterance, even if this reference is only implicit (unless it has already been loaded).

3. In chapter 6, it will be explained how Crystal favours interpretations which continue to refer to encyclopædic information already present in the deductive device, i.e. it favours sticking to a particular subject. Making implicit references to temporal notions in an utterance is hardly the same as continuing to talk, say, about fish, but both are regarded as maintaining a conversational strand (to some extent).

4. Much as events are treated as primitive objects, states are not. However, this can be overcome by creating a "State" sort and allowing "HOLDS" to apply to terms of type state rather than to assumptions.

In appendix B, it is shown how Allen's logic can give rise to unrealistic Relevance measurements. Fortunately, this biasing effect tends to be common to all interpretations: most discourses make some explicit or implicit reference to time in their first utterance, so the appropriate encyclopædic entry remains permanently loaded. Furthermore, where utterances are syntactically ambiguous, all the alternative readings tend to refer to time or events (if they refer to such notions at all). Thus, since Relevance is usually used only comparatively (see chapter 6), the adoption of Allen's logic did not have serious consequences. However, in retrospect, it would have been better to employ a temporal logic at the meta-level, had a suitable logic been available.

For the remainder of this thesis, I will assume that the reader is familiar with Allen's temporal logic as described in Allen (1984). A minor difference between the logic I employ and the logic described in Allen's paper is that I have not used the predicates CAUSE (and ACAUSE). This is because there seems to be no useful difference between:

$$\mathrm{CAUSE}(event_1, event_2) \tag{5.63}$$

and

$$event_1 \vdash event_2 \tag{5.64}$$

### Encyclopædic Information about Verbs

The meaning of most verbs involves temporal notions. In Crystal, I have adopted Davidson's approach of representing events as distinct terms which occupy an argument place of the predicate representing the verb.

Consider two alternative fragments of encyclopædic information for the verb "go", represented in Allen's logic:

$$\begin{aligned}
&\forall !agent, !from, !to, !event, !t_1 .\, \mathrm{Go}(!agent, !from, !to, !event) \wedge \\
&\mathrm{OCCUR}(!event, !t_1) \supset \\
&\quad \exists !t_2 .\, \mathrm{HOLDS}(\mathrm{At}(!agent, !from), !t_1) \wedge \\
&\quad \mathrm{HOLDS}(\mathrm{At}(!agent, !to), !t_2) \wedge \\
&\quad (!from \neq !to) \wedge (!t_2 > !t_1)
\end{aligned} \tag{5.65}$$

and

$$\begin{aligned}
&\forall !agent, !from, !to, !e, !t_1 .\, \mathrm{Go}(!e) \wedge \mathrm{OCCUR}(!e, !t_1) \wedge \\
&\mathrm{Agent}(!e, !agent) \wedge \mathrm{From}(!e, !from) \wedge \mathrm{To}(!e, !to) \supset \\
&\quad \exists !t_2 .\, \mathrm{HOLDS}(\mathrm{At}(!agent, !from), !t_1) \wedge \\
&\quad \mathrm{HOLDS}(\mathrm{At}(!agent, !to), !t_2) \wedge \\
&\quad (!from \neq !to) \wedge (!t_2 > !t_1)
\end{aligned} \tag{5.66}$$

Both of these definitions state that "going" from "!from" to "!to" is an event such that the agent is at "!from" at a particular time and at "!to" sometime later, and that "!to" is not the same place as "!from".

(5.65) differs from (5.66) in that all of the arguments have been included in the predicate denoted by the verb. In (5.66) the thematic roles[16] are separate predications of the event variable.

The two representations will behave differently during inferencing. This can be observed by considering an utterance in which "go" only takes a single (mandatory) argument — the agent.

[16] I am not making any commitment to any general theory of thematic roles here. For example, the "agent" role for the "go" event is not intended to capture some general notion of an "agent" common to many events.

Thelonius went out. (127)

Considering representation (5.65), (127) may be represented as:

$$\begin{aligned}&\text{Go}(\mathit{thelonius}, \mathit{sk.from001}, \mathit{sk.to3333}, \mathit{go.event341}) \wedge \\ &\quad \text{OCCUR}(\mathit{go.event}341, \mathit{go.interval}452) \wedge \\ &\quad (\mathit{go.interval}452 < \mathit{NOW})\end{aligned} \tag{5.67}$$

In order to successfully translate (127), all of the arguments to Go *must* be supplied. Thus, for the statement (127), the "from" and "to" thematic roles would have to be supplied as skölem constants. Once this is done, modus ponens could be applied to yield the information on the right hand side of the implication, which is part of the meaning of the verb. Obtaining this information would constitute two contextual effects (applying conjunctive modus ponens to (5.65) twice: once for each conjunct of (5.67)).

Considering (5.66), (127) may be represented by:

$$\begin{aligned}&\text{Go}(\mathit{go.event}341) \wedge \text{OCCUR}(\mathit{go.event}341, \mathit{go.interval}452) \wedge \\ &\quad (\mathit{go.interval}341 < \mathit{NOW})\end{aligned} \tag{5.68}$$

This time, there is no obligation to supply the unknown thematic roles as anonymous constants. However, without them, the schema (5.66) cannot be fully instantiated. If they were all to be separately assumed, the number of contextual effects obtained in the derivation of the right-hand side of the material implication would be four: one for each application of conjunctive modus ponens to (5.66).

Examination of these representations raises two important questions:

1. Which representation yields the intuitively correct number of contextual implications?
2. Which process supplies the skölem constants or appropriate thematic roles? The supplying of this information is a non-deductive process.

My solution has been to adopt Davidson's suggestion that the mandatory arguments to a verb should fill argument places in the predicate representing that verb. Adjuncts appear as predications of the event. Since certain thematic roles are implicitly filled when an event occurs, these are represented as existentially quantified variables on the RHS of the implication. In this case, "from" and "to" are such roles, because "going" always involves going *from* somewhere, *to* somewhere. Thus, the normalised encyclopædic information about Go would be:

$$\begin{aligned}&\text{Go}(!\mathit{agent}, !\mathit{event}) \wedge \text{OCCUR}(!\mathit{event}, !\mathit{time}) \supset \\ &\quad \text{Agent}(!\mathit{agent}) \wedge \\ &\quad \text{From}(!e, \text{Sk.Place1.Of}(!\mathit{agent}, !\mathit{event}, !\mathit{time})) \wedge \\ &\quad \text{Place}(\text{Sk.Place1.Of}(!\mathit{agent}, !\mathit{event}, !\mathit{time})) \wedge \\ &\quad \text{To}(!e, \text{Sk.Place2.Of}(!\mathit{agent}, !\mathit{event}, !\mathit{time})) \wedge \\ &\quad \text{Place}(\text{Sk.Place2.Of}(!\mathit{agent}, !\mathit{event}, !\mathit{time})) \wedge \\ &\quad \text{HOLDS}(\text{At}(!\mathit{agent}, \text{Sk.Place1.Of}(!\mathit{agent}, !\mathit{event}, !\mathit{time})), !\mathit{time}) \wedge \\ &\quad \text{HOLDS}(\text{At}(!\mathit{agent}, \text{Sk.Place2.Of}(!\mathit{agent}, !\mathit{event}, !\mathit{time})),\end{aligned}$$

$$
\begin{aligned}
&\quad \text{Sk.Arrival.Time.Of}(!agent, !event, !time)) \wedge \\
&(\text{Sk.Place1.Of}(!agent, !event, !time) \neq \\
&\quad \text{Sk.Place2.Of}(!agent, !event, !time)) \wedge \\
&(\text{Sk.Arrival.Time.Of}(!agent, !event, !time) > !time)
\end{aligned} \tag{5.69}
$$

This answers the above questions in the following ways:

1. Skölem constants never have to be magically supplied because:
   (a) The predicate representing the verb only has argument places that will be filled by every usage of the verb.
   (b) Necessary roles that are implicitly present are made explicit on the right-hand side of the implication.

2. The left-hand side of the implication need only contain the predicate representing the verb and the "OCCUR" relation. This means that obtaining the right-hand side information by modus ponens will always yield the same contextual effects, independent of the verb described[17]. This is only true for the description of the meaning of a verb with no adjuncts. Additional assumption schemas may be provided to represent the meaning of the verb when modified by an adjunct. In this case, the LHS of the implication may include extra conjuncts in order to restrict the application of modus ponens so that it is only applied when the particular adjunct is present.

   So, for example, an utterance such as "Florence went to Jamshedpur by car" would access the encyclopædic entry for "go". Not only might this contain an assumption schema such as (5.69) but it may include a "by" schema, which provides further information, which would begin:

$$\text{Go}(!agent, !event) \wedge \text{By}(!event, !i) \wedge \text{OCCUR}(!event, !time) \supset \ldots$$

When an adjunct supplies a role already represented by a skölem constant, it is up to the non-demonstrative inference generator to notice the equivalence of the anonymous constant and the supplied role (see chapter 8). In practice, such inferences might not always be necessary because it may be possible to derive the assumption through syntactic and semantic constraints in the utterance. However, these constraints are not exploited in Crystal.

### 5.5.2 Other Modal Concepts

Other modal concepts, such as belief and desires could be similarly implemented as reified logics. For example, Moore's logic of belief (Moore, 1985a) could be straightforwardly incorporated into the deductive device. Some of the problems that are incurred by the use of reified logics are illustrated in appendix B.

### 5.5.3 Frames and Scripts

Having defined a logic which is capable of default and temporal reasoning, it is now possible to discuss how frame-like and script-like notions can be implemented in Crystal's logic.

[17] Of course, the amount of information on the RHS may vary between different verbs.

### Frames

A logical characterisation of frames can be found in Hayes (1979). This does not account for the prototypical nature of frames.

I will show how frame-like notions can be built up from a dictionary definition[18]. Consider the Longman's definition of the noun "lake":

> A large area of water, *esp.* non-salty water, surrounded by land.

Note that:

- Certain properties are always true, like the fact that the lake is an area of water. This can be expressed through predication of universally quantified variables.
- Other properties are prototypical, such as the fact that it is non-salty. This can be modelled using the $\mathcal{D}$ operator.
- Certain anonymous objects are introduced by the mention of the concept. For example, every lake is surrounded by a specific piece of land. This can be modelled by normalised existentially quantified variables.

These are the typical kinds of things that can be expressed by most frame-based languages.

Thus, the encyclopædic information for "lake" might be[19]:

$$\mathrm{Lake}(!i) \supset \mathrm{Water}(!i) \tag{5.70}$$

$$\mathrm{Lake}(!i) \supset \mathrm{Land}(\mathrm{Sk.Land.Of}(!i)) \wedge \mathrm{Surrounds}(\mathrm{Sk.Land.Of}(!i), !i) \tag{5.71}$$

$$\mathrm{Lake}(!i) \supset \mathrm{Area}(!i) \tag{5.72}$$

$$\mathrm{Lake}(!i) \supset \mathcal{D}\neg\mathrm{Salty}(i) \tag{5.73}$$

As Moore (1982) points out, logical representations of such information also allow for non-determinism using disjunction. Crystal's logic is a superset of many frame-like languages.

### Scripts

The notion of scripts is more intangible than that of frames. However, it is generally the case that scripts describe stereotypical uses of objects or typical actions. They can therefore be considered as a kind of frame that involves temporal aspects of the objects it describes.

A standard example of a script involves some kind of restaurant scenario. The script states that, typically, going to eat at a restaurant involves waiters approaching you to seat you, providing you with a menu, taking your order, bringing the food, giving you the bill, and finally taking your payment for the bill. Clearly, at any stage, it is possible to divert from the script. This has caused many problems for researchers who believe that understanding is merely a matter of detecting conformance to a script (Schank (1982), for example, discusses these problems and suggests a solution).

In my opinion, the role of scripts may be viewed differently under RT: scripts provide coherence to a discourse whether the actions within it conform to them or not. They are

[18] I assume the reader is familiar with the basic characteristics of frames.

[19] Of course, the following definition is not a completely adequate description of a lake.

used to predict events that may form part of a stereotypical sequence. However, we often find interest in stories because the stories do *not* obey these predictions. In RT terms, a prediction made from a script may be retroactively strengthened if it later proves to be right, or it may be later erased if it proves to be wrong. Both of these are contextual effects in their own right. The purpose of the script is to provide a framework in which these effects can occur.

Consider a script-like schema involved with the definition of "hungry". The dictionary definition of "hunger" merely states that it is the feeling of hunger, hunger being the desire for food. However, this definition does not include the kinds of general eating script that may go with hunger. Sometimes, when we are hungry, we eat food to satisfy it.

$$\begin{aligned}
&\mathrm{HOLDS}(\mathrm{Hungry}(!p), !t) \supset \\
&\quad \mathrm{Food}(\mathrm{Sk.Food.Of}(!p, !t)) \wedge \\
&\quad \mathrm{Eat}(!p, \mathrm{Sk.Food.Of}(!p, !t), \mathrm{Sk.Eat.Event.Of}(!p, !t)) \wedge \\
&\quad [\mathrm{OCCUR}(\mathrm{Sk.Eat.Event.Of}(!p, !t), \mathrm{Sk.Eat.Interval.Of}(!p, !t)) \supset \\
&\quad (\mathrm{Sk.Eat.Interval.Of}(!p, !t) < !t_1) \supset \mathcal{D}\mathrm{HOLDS}(\neg\mathrm{Hungry}(!p), !t_1)]
\end{aligned} \tag{5.74}$$

This states that if person, $p$, is hungry at time $!t$, we can imagine some event where they eat some food. This event does not necessarily "OCCUR". If it does, then if consistent, it is assumed that $p$ is not hungry afterwards.

Now, if it is known that Miranda is hungry at time $i$, the schema will be instantiated and modus ponens applied to infer, among other things:

$$\mathrm{Food}(\mathrm{Sk.Food.Of}(miranda, i)) \tag{5.75}$$

$$\mathrm{Eat}(miranda, \mathrm{Sk.Food.Of}(miranda, i), \mathrm{Sk.Eat.Event.Of}(miranda, i)) \tag{5.76}$$

$$\begin{aligned}
&\mathrm{OCCUR}(\mathrm{Sk.Eat.Event.Of}(miranda, i), \mathrm{Sk.Eat.Interval.Of}(miranda, i)) \supset \\
&\quad (\mathrm{Sk.Eat.Interval.Of}(miranda) < !t_1) \supset \\
&\quad \mathcal{D}\mathrm{HOLDS}(\neg\mathrm{Hungry}(miranda), !t_1)
\end{aligned} \tag{5.77}$$

What exactly is Sk.Eat.Event.Of($miranda, i$)? I consider it to be a possible future event. Because it does not necessarily "OCCUR", it is not a necessity, merely a possibility.

The script cannot be deductively instantiated; it must be instantiated using abductive equality as described in section 5.3.3. If some object appears in the discourse that could match part of the script, we can assume (non-monotonically) that it does. This allows us to predict what *might* happen. So, for example, say we know that Miranda has the rusk, *rusk042*. The non-demonstrative inference generator might guess that:

$$rusk042 = \mathrm{Food.Of}(miranda, i) \tag{5.78}$$

From this, we can infer (by substitution of equals) that Miranda *may* eat the rusk. This is weaker than the default inference that she *will* eat the rusk. (5.78) is more non-committal because the actual eating event does not necessarily happen ("OCCUR"). Thus, it is possible to distinguish possibilities from default assumptions.

Later in the discourse, it may be discovered that Miranda cannot eat the rusk at all during the interval $i$. In this case, the negation of (5.78) will be assumed by the reason maintenance process (see chapters 7 and 8).

Scripts do not just talk about possibilities, they can be used to explain certain events. We may be told that Miranda has a rusk, and that she eats. It seems reasonable to assume that, since she had the rusk, it was the food that she ate. The abductive inference generator could manage this by assuming that Miranda's actual eating corresponds to Sk.Eat.Event.Of($miranda, i$) and that the anonymous food mentioned in her actual eating act is equal to Sk.Food.Of($miranda, i$).

#### Summary

In my view, frames describe necessary and default properties of some object. Scripts describe possibilities. The two can be represented separately in Crystal. Defaults are assumed to be true when it is consistent to do so. Events that have not "OCCUR"ed are regarded as possibilities. I should, perhaps, point out here that this use of Allen's logic has no foundation in the semantics that he has provided and is rather ad-hoc. A more satisfactory approach would be to use a modal possibility operator, which would require modification of the meta-rules.

Crystal uses scripts in two ways:

1. It can guess that a particular object or action will form part of a future possibility.
2. It can guess that certain utterances cohere because one can be used in the explanation of the other.

The key to both of these uses is abductive guesses of equality.

## 5.6 Implementation Issues

I will now discuss how conceptual information described using Crystal's knowledge representation must be phrased. First, I will show that it is sometimes necessary to strengthen encyclopædic information to prevent unwanted abductive inferences. Secondly, I discuss how RT places constraints on how it should be phrased.

### 5.6.1 Strengthening Encyclopædic Information

We have already seen that script-like information is often not included in an encyclopædic entry. For example, the definition of "restaurant" does not include a stereotypical dining situation. Thus, it is often necessary to enhance encyclopædic entries with extra default and script-like information. Most of this information will be in the form of positive assertions.

However, it is sometimes necessary to strengthen encyclopædic information with negative assertions to prevent unwanted non-demonstrative inferences.

Consider the following script about bolts:

Bolt($!b$) $\supset$
Nut1(Nut1.Of($!b$)) $\wedge$
Solid(Solid.21.Of($!b$)) $\wedge$
Solid(Solid.22.Of($!b$)) $\wedge$
Adjacent(Solid.21.Of($!b$), Solid.22.Of($!b$)) $\wedge$

$$\begin{aligned}
&\mathrm{Attach}(\mathrm{Attach.Event.Of}(!b)) \wedge \\
&\mathrm{Using}(\mathrm{Attach.Event.Of}(!b), !b) \wedge \\
&\mathrm{Using}(\mathrm{Attach.Event.Of}(!b), \mathrm{Nut1.Of}(!b))
\end{aligned} \tag{5.79}$$

i.e. The bolt might be used with a nut to attach two solids together. Now, this may seem to be a reasonable expression, however it gives rise to some abductive inferences:

Consider a situation where the following are asserted from the script:

$$\mathrm{Solid}(\mathrm{Solid.21.Of}(!b)) \tag{5.80}$$

$$\mathrm{Solid}(\mathrm{Solid.22.Of}(!b)) \tag{5.81}$$

There is nothing to stop the non-demonstrative inference generator producing the following inference:

$$\mathrm{Solid.21.Of}(!b) = \mathrm{Solid.22.Of}(!b) \tag{5.82}$$

This is an undesirable inference: it must be ruled out by adding the negation of (5.82) to the script.

### 5.6.2 Phrasing Encyclopædic Knowledge

It was shown in section 5.1 that the syntactic realisation of encyclopædic information is significant. Different ways of phrasing information may result in different numbers of contextual effects.

Recall the assumptions (5.1) and (5.2). These yielded different numbers of contextual effects with assumption (5.3). The reason for this is that (5.1) yields an extra effect because it first yields a conjunction when modus ponens is applied.

Although (5.1) yields more contextual effects, the conjunction that it yields is of no use; it should not have been inferred as a "plausible inference". Conjunctions are not required because the rule of conjunctive modus ponens can be used to progressively derive the RHS of a material implication if its LHS contains a conjunction of WFFs. Since the deductive device has limited memory size, it is better not to make inferences that are of no use, since they only occupy space. Thus, (5.2) is a better representation of the utterance.

By a similar argument, it is better to phrase a conjunction as separate assumptions, one for each conjunct.

To conclude, when phrasing information, its possible inferential behaviour must be considered. Forms should be avoided that result in undesirable inferences, such as conjunctions of assumptions[20]. This problem is addressed in more detail in section 6.4.2.

## 5.7 Summary

Crystal uses a logic to describe commonsense knowledge. The logic can describe stereotypical inferences and chronologically ordered sequences. In order to prevent certain inferences, a sort hierarchy has been employed within the unifier. The inferences made

[20]In Crystal, it was *only* conjunctions of assumptions that were definitely of no use in themselves. Thus, it was only these types of forms that needed to be written concisely.

implicitly using this hierarchy are duplicated in the deductive device so that information change is accurately measured.

I have endeavoured to form a logical characterisation of frame- and script-like inferences. Unfortunately, trying to bolt together all of the off-the-shelf logical apparatus has resulted in something of a Frankenstein's monster: the deductive device and knowledge representation can perform the job required of them, but not in the most elegant or satisfactory way. Ideally, the logic intrinsic to the deductive device should have had notions such as possibility, belief, and time built into its meta-rules. There is no reason why this should not be possible in a future implementation.

At various points in this chapter, I have illustrated how an acceptable syntactic realisation of a particular form suitable for inferencing is problematic. It appears that its inferential behaviour must be known in advance in order to determine the best way of phrasing it. In chapter 6, I suggest a heuristic means of overcoming this problem. However, an alternative, more radical approach would be to measure change in *semantic* rather than syntactic information; I will further justify this in section 7.6.1 and suggest a way of measuring semantic information change in chapter 9.

# Chapter 6

# Measuring Relevance

Now that I have expounded Crystal's knowledge representation, I can go on to discuss the measurement of Relevance itself, and how it controls context expansion.

As mentioned in chapters 1 and 2, much of the criticism of RT is centred on the fact that notions such as information and effort have not been defined adequately enough by SW. Their reply to this appears to be that Relevance should not be described numerically. I begin this chapter by arguing the converse: a numeric measure of Relevance seems as good as any. Following this, I define a Relevance function which I utilise in the remainder of the chapter.

The next two sections consider different approaches to information and effort measurement. In order to evaluate these measurements, I consider whether they can be used to successfully interpret some of the significant utterances that critics have put forward as counter-examples to RT. Each section concludes with an appropriate measure.

Another criticism levelled at RT is that it never appears to be worth expanding the context. I therefore suggest a modification to the theory that overcomes this criticism. Following this, I explain the relationship between Relevance and plausibility.

I conclude with a brief summary and argue that the changes I have made to RT do not change it in an inappropriate manner.

## 6.1 Can Relevance Be Measured?

SW avoid providing a definition of Relevance that might be computationally evaluated. The reason for this is that they speculate that:

> ...contextual effects and mental effort, just like bodily movements and muscular effort, must cause some symptomatic physico-chemical changes. We might assume the mind assesses its own efforts and their effects by monitoring these changes. (Sperber and Wilson, 1986, pp. 130).

In other words, the magnitude of contextual effects, effort and therefore of Relevance associated with an utterance in a given situation can vary from person to person. This variation is accounted for by the fact that we are talking about physical changes in a particular person's brain.

Unfortunately, this argument can be seen only as a criticism of RT. If it is not possible for RT to make any generalisations about the Relevance of an utterance, or classes of utterances, it would become vacuously correct. Any utterance could be argued to produce certain numbers of contextual effects or require a certain amount of effort and thus become trivially Relevant.

SW are similarly vague about how and when the context should be expanded. All they are prepared to say is that information that should make an utterance more Relevant should be brought in during context expansion. They do not explain when context expansion should take place or when it should stop.

For RT to have any general explanatory or predictive power across individuals, some sort of generic base *must* be defined. If this measure is numeric, it does not mean that we actually have numbers in our heads, merely that it is a good way of modelling what is actually going on. Consequently, in the remainder of the chapter, I will define a Relevance measure and eventually describe how information and effort change are calculated. I will then relate Relevance and context expansion.

## 6.2 A Relevance Measure

In this section, a measure of Relevance at context expansion $n$, Relevance($n$), is defined. I will use the most obvious definition, suggested by Gazdar and Good (1982). It is the simplest equation that conforms to SW's descriptions. This definition will be assumed for the remainder of this chapter.

Let $\delta inf_i$ be the change in contextual effects obtained at context expansion number $i$ and $\delta eff_i$ be the change in effort required to derive those effects.

We can then define Inf($n$) and Eff($n$) to be the total information derived in $n$ expansions, and the effort required to derive that information, respectively.

$$\mathrm{Inf}(n) = \sum_{i=1}^{n} \delta inf_i$$

and

$$\mathrm{Eff}(n) = \sum_{i=1}^{n} \delta eff_i$$

Given these definitions, the simplest definition of the Relevance at the $n^{th}$ context expansion is:

$$\mathrm{Relevance}(n) = \frac{\mathrm{Inf}(n)}{\mathrm{Eff}(n)} \tag{6.1}$$

## 6.3 Measuring Contextual Effects

Crystal takes a simple approach to information counting: every contextual effect is regarded as a single unit of information. A criticism of measuring information in this way is found in Gazdar and Good's (GG's) "Experts and Obsessives" problem. One method of overcoming this problem is to redefine contextual implication (see section 6.3.3). A different, more adequate method is discussed in section 6.4.2.

### 6.3.1 Experts and Obsessives

Gazdar and Good (1982) point out an interesting problem:

> ... consider the following example:
>
> (23) Sue has married Bill, who is a Mason.
>
> Now, intuitively, this seems to be a relevant thing to say to person A who is known to be obsessed with Masonry but actually knows almost nothing about it. This person will not be able to derive many non-trivial contextual implications from the utterance of (23). However, intuitively, it will not be as relevant a thing to say to a person B who knows all about masonry but is completely disinterested in the topic. Such a person, will, nevertheless, be in a position to derive numerous non-trivial contextual implications from the utterance of (23).

Sperber and Wilson (1982b) reply that:

- Much as the obsessive, $A$, makes many inferences, he may not properly understand the utterance; he may not be able to resolve the necessary references or disambiguate concepts correctly. $A$ may not even be able to make any connections with previous utterances.
- GG's description is assuming the natural sense of relevance, rather than the non-natural meaning, Relevance. Even if we are disinterested in the subject area of an utterance, this may not prevent it being Relevant to us, even though it is not relevant.
- SW doubt that an obsessive really would have little encyclopædic information about a subject with which he was obsessed.

We shall see in the next section that SW have not given an entirely satisfactory response.

### 6.3.2 Problems with Information Yield

GG have obfuscated the issue by introducing the obsessive and confusing the senses of "relevance". Despite these facts, they have correctly identified a serious problem with RT which SW have not convincingly overcome. Consider the following in an empty context:

Damon found a nut. (128)

Assume we know nothing about Damon. We are left with the problem of deciding whether "nut" refers to the fruit, $Nut_1$, or the object used for attaching objects to each other, $Nut_2$.

RT, as described by SW, appears to predict that the chosen sense will be that which yields the most contextual effects, which in turn will be dependent on the number of assumption schemas and associated logical forms that can be instantiated by the nut. For example, if the encyclopædic entry for $Nut_1$ contains five more mutually consistent assumption schemas than that of $Nut_2$, it may turn out that we prefer a reading containing $Nut_1$. This seems too arbitrary; other factors seem to be at least as significant, for example which word sense is the most common, and the "priming" of senses other words.

Let us consider a practical example. Imagine a situation where an expert on $Nut_2$s reads a childrens' story, which contains the following:

Nutkin, the squirrel, found a nut. (129)

Even this expert, who knew far more about $Nut_2$s than $Nut_1$s, should be able to work out that "nut" is more than likely a $Nut_1$ because squirrels like to eat nuts. It is harder to imagine what a squirrel has to do with a $Nut_2$.

The sense $Nut_1$ should be obtained because the inferences that connect the "finding" event with Nutkin are more important than the inferences that are simply obtained by instantiation of encyclopædic information.

Part of this problem is due to SW's vague notion of instantiation, the development of assumption schemas. I have already illustrated in chapter 5 that instantiation is generally equivalent to non-trivial inferencing. However, if we did not consider the information gained from instantiation to be significant, the instantiation of an experts knowledge would not bias Relevance any more than the instantiation of a layman's knowledge because neither would be counted as a contextual effect. This suggests a possible solution to the problem I have just outlined. We might invent a new category of implication known as a *simple implication* which is an implication that is the instantiation of encyclopædic knowledge. A new definition of contextual effect would then exclude simple implications.

### 6.3.3 A Solution: Simple Implications

Let us call the more trivial inferences derived by simple instantiation of assumption schemas by the deductive device *simple implications.* These might be formally defined as follows:

*Simple Implication:* A set of assumptions, $P$, all of identical status, simply implies an assumption $q$ in the context of a set of assumptions $S$, all of identical status, iff:

1. $P \cup S \vdash q$
2. $P \not\vdash q$
3. $S \not\vdash q$
4. Given that Status is a function from sets of assumptions of identical status which returns that status (see section 4.4.3):

$$\begin{aligned} \text{Status}(P) &\neq \text{Status}(S) \wedge \\ \{\text{Status}(P)\} &\cup \{\text{Status}(S)\} = \{\textit{memory, utterance}\} \end{aligned}$$

Intuitively, simple implications are inferences derived directly from the memory and the utterance.

The notion of simple implication can solve the above problems if there are not considered to be contextual effects. Reconsider utterance (128). Discounting the simple implications, each reading yields the same number of contextual effects. No more contextual implications follow from any particular sense of "nut"; the utterance has been badly phrased and is truly ambiguous.

Now consider (129). Since simple implications are not counted, the reading with $Nut_1$ will be more significant because it can be connected with a script about squirrels finding and storing nuts, for example.

Thus, the notion of contextual implication can be revised to yield the necessary results:

*Contextual implication (3):* A set of assumptions $P$, all of identical status, contextually implies an assumption $q$ in the context of a set of assumptions $S$, all of identical status, iff:

1. $P \cup S \vdash q$
2. $P \not\vdash q$
3. $S \not\vdash q$
4. **Either**

$$\begin{aligned} \text{Status}(P) = \text{Status}(S) &= \text{Contextual.Implication} \vee \\ \text{Status}(P) = \text{Status}(S) &= \text{Simple.Implication} \end{aligned}$$

**or**

$$\begin{aligned} \text{Status}(P) &\neq \text{Status}(S) \wedge \\ \{\text{Status}(P)\} &\cup \{\text{Status}(S)\} \neq \{\mathit{memory},\ \mathit{utterance}\} \end{aligned}$$

Unfortunately, this definition is not very elegant and occasionally can deliver incorrect results, because not all forms of inference from items of status Memory and Utterance are necessarily instantiations, although this is what is implied by the formal definition provided above. As an example, consider the following possible piece of encyclopædic information for the entry Rottweiler:

$$\text{Rottweiler}(\mathit{buttercup}) \tag{6.2}$$

This information would be invoked by an utterance representation such as:

$$\text{Rottweiler}(!r) \supset \text{Vicious}(!r) \tag{6.3}$$

which might be paraphrased as "all Rottweilers are vicious". By applying Modus Ponens to these two forms, the following can inferred:

$$\text{Vicious}(\mathit{buttercup}) \tag{6.4}$$

In no way can this be regarded as instantiation of (6.2). Thus, (6.4) should not be regarded as simple implication, but rather as a contextual implication. An obvious modification to the definition of simple implication that avoids this problem would be to insist that instantiation of encyclopædic information either involves universal instantiation of an assumption with contextual status Memory or a non-demonstrative assertion of equality concerning one of its constituent terms (see chapter 8). I will not provide a precise reformulation because there is a more elegant solution that I have adopted in Crystal which is explained in section 6.4.2. However, assuming that the definition of simple implication above is adequate, let us see how this circumvents the experts and obsessives problem.

### 6.3.4 Experts and Obsessives Reconsidered

The "experts and obsessives" problem can be cast in a new light if we do not consider simple implications to be contextual implications.

According to Longman's, an obsession is:

> A fixed and often unreasonable idea with which the mind is continually concerned.

Obsessives' unreasonable ideas usually involve forming unlikely connections between what has been said by a speaker and the object of their obsession, which I assume to be part of their current context. If such connections were purely deductive, they could only be due to incorrect encyclopædic knowledge (misunderstanding) rather than obsession. Such deductive links would not pervade all conversations unless all the appropriate encyclopædic entries are incorrect. A more plausible explanation is that the obsessive's ability to form non-demonstrative or inductive inferences is impaired; obsessives constantly form unwarranted guesses concerning the connection of an utterance with their current context. Thus, an obsessive is likely to make *irrational* connections with the context frequently. The result of this will be that, since these connections are certainly not simple implications, every utterance will yield a multitude of contextual effects. However, these contextual effects may not have any rational justification. The obsessive person will find many utterances Relevant because of the many irrational contextual effects he finds. Subjectively, the obsessive has "understood" the utterance. However, to the speaker, the obsessive is not understanding the correct sense of what is being said. Communication is failing because the speaker cannot predict what is in the mutual cognitive environment because the hearer is behaving irrationally.

As for the expert, she will make rational non-demonstrative connections when necessary. Furthermore, using the revised definition of contextual effect, she will not be biased towards any particular concept purely on the grounds of how much she knows about it. This is because the mere instantiation of her knowledge is not important. Rather, it is the contextual effects that are then derived from this instantiated knowledge that are important.

In my opinion, RT predicts what should be communicated to a rational agent during utterance interpretation. An obsessive is not a rational person, thus he may misunderstand an utterance. This, of course, is exactly what obsessives tend to do.

I reconsider this topic in section 6.5 after I have considered the measuring of effort in the next section.

## 6.4 Measuring Effort

In this section, SW's notion of cognitive effort is examined. The possible effort-consuming processes are enumerated, and various methods of quantifying their effort consumption examined.

### 6.4.1 Effort-Consuming Processes

SW's notion of effort is somewhat intangible. According to them, certain processes do not contribute to effort measurements. In particular:

1. The assertion of new assumptions.

2. The establishment of how much effort is consumed.

If these processes consumed effort, it is possible that the amount of information yielded by an utterance could never overcome the effort required to establish this information.

In the ideal schema described in section 3.3, possible sources of effort consumption are:

- Work done by the input processes.

- Work done in expanding the context.
- Work done in applying various inference rules.
- Work done by the reason maintenance process.

Each of these will be discussed in turn.

### The Input Processes

Part of Grice's maxim of manner, "be brief", can be explained by considering the fact that the input processes require an amount of effort dependent to the syntactic complexity of an utterance. Consider two utterances, $U_1$ and $U_2$ that yield the same contextual effects but $U_1$ is syntactically simpler than $U_2$. $U_1$ is the more Relevant because it yields the same number of contextual effects for less processing effort.

I shall illustrate how Crystal might be used to explore the maxim of manner using RT concepts. However, I have not tested these ideas in any detail. Consider the following two utterance representations:

$$\text{Human}(rajeev) \wedge \text{Man}(rajeev) \tag{6.5}$$

$$\text{Man}(rajeev) \tag{6.6}$$

(6.5) might represent "Rajeev is a human man", and (6.6) might represent "Rajeev is a man". Considering Grice's maxim, (6.6) is better phrased because it expresses the same information as (6.5) but is more syntactically succinct. This can be explained using RT as follows. (6.6) will result in the loading of the encyclopædic entry for "man" into the deductive device. Subsequent inferencing will produce assumptions that include the fact that Rajeev is human and not a woman etc. Next consider (6.5). This will load the entry for "man" and the entry for "human". The entry for "human" may include information about "women" and "children" as well as "men". However, this information will not be instantiated by Rajeev, who is a man. All that will be instantiated is the fact that Rajeev is a man and that he is not a woman etc. Thus, the contents of the deductive device will contain similar non-trivial inferences for both (6.5) and (6.6), but (6.5) contains five words and is syntactically more complex than (6.6), which contains four. Consequently, (6.5) is less Relevant than (6.6) in a null context.

However, there may be a reason why (6.5) has been uttered: the speaker may deliberately want to make the apparently useless generic information about humans available to the hearer. For example, the discourse might continue:

However, sometimes he acts like an alien man. (130)

In this case, "however" is contrasting certain properties of "alien" with properties of "human" which have been made contextually salient by the last utterance[1].

These kind of examples have not been tested in Crystal, which only compares the Relevance of lexically ambiguous utterances. In such cases, the syntactic complexity of the utterances is identical for all senses. The effort expended in parsing can therefore be ignored.

---

[1] I am assuming here that the entry for "human" contains a superset of the information about "man". However, it might be argued that it contains a subset of this information — the less specific information about "man" that is common to all humans. If this were the case, (6.5) could never be used to bring in extra information to be used later, and would always be considered to be phrased badly unless it was used poetically.

### Expanding the Context

If expanding the context consumed no effort, there would be no reason to stop expanding it (except, perhaps, some arbitrary time limit). Therefore, accessing and loading encyclopædic data must consume effort.

In my opinion, the fact that context expansion consumes effort can be used to explain why people tend to stick to particular subjects in a discourse. If the loading of conceptual information requires effort, then it is cheaper to express new information in terms of conceptual information already present in the deductive device. Consider:

Zebedee drank water from the mountain spring. (131)

After that, he came across a dirty spring. (132)

When (132) immediately follows (131), we tend to assume the same sense of "spring" in the appropriate utterances. At least part of the explanation for this is that this sense of "spring" is still in the deductive device and therefore requires no loading effort.

This will be further discussed in section 6.4.2.

### Applying Various Inference Rules

There are three possibilities concerning the application effort of logical rules:

1. Obtaining the contextual implications of any particular $\delta inf_i$ requires an approximately constant amount of effort.

2. Applying particular inference rules requires a fixed amount of effort per rule application (independent of the particular rule).

3. The amount of effort consumed is dependent on the particular rule.

The first option is the simplest and has a particular advantage: the syntactic organisation of the encyclopædic knowledge is less critical. It is therefore the approach adopted in Crystal[2].

### The Cost of Reason Maintenance

SW do not discuss belief revision in any detail, but presumably it must consume effort. If this were not the case, there would be no harm in selecting the wrong sense of an utterance.

The cost of reason maintenance can also be used to explain why the positioning of certain constituents within an utterance determines which particular reading of the utterance is obtained (see section 7.6.4).

## 6.4.2 The Processing Cost of Context Expansion

In this section, I will show how consideration of loading effort can provide an alterative to the "simple implications" approach to determining contextual effects and solve various other problems introduced in earlier sections.

There are various factors that can contribute to the cost of context expansion:

---

[2]SW also suggest that inferencing costs at any particular expansion level are approximately constant, although they do not justify this.

1. The level of context expansion (the number of times the context is expanded).
2. The number of encyclopædic entries that are loaded.
3. The number of assumptions that are loaded.
4. The complexity of the loaded clauses.
5. The utility or salience of the loaded assumptions.

Each of these will be examined in turn.

### Level of Context Expansion

If this were the only significant factor, and it was constant, this would mean that it would generally be worth expanding the context because it would almost always result in a significant increase in Relevance. Generally, for a constant amount of effort, a possibly exponentially larger amount of assumptions would be loaded[3]. Thus, if it is the only factor, effort proportional to expansion level yields counter-intuitive results. However, if it is not the only factor or it is not constant, it is made redundant by the other, more specific, factors listed above.

### The Number of Encyclopaedic Entries

If the number of complete entries[4] loaded are taken into account, the following desirable consequences will generally be obtained:

1. Effort will not increase in a linear fashion with context expansion. Consider the somewhat unlikely situation where encyclopædic entries consisted, on average, of one assumption and that, on average, each of these assumptions contained only two distinct concepts. Even under these circumstances, worst case context expansion effort would increase exponentially. The worst case would be when every encyclopædic entry accessed was not yet loaded into the deductor memory. On the first context expansion, two encyclopædic entries would be loaded. On the next, since each assumption contains two concepts and each encyclopædic entry contains one assumption, four encyclopædic entries would be loaded. Similarly, the next expansion would load eight entries, and so on.

   This non-linearity is exactly what is required by RT. Utterances are expected to produce as much information as possible in the initial context. The hearer should be discouraged from arbitrarily expanding the context as this makes the utterance increasingly more vague and is a time-consuming process.

2. It is generally more Relevant to "stick to the same subject" (see section 6.4.1). For the case of interleaved conversations, SW suggest that one or more context buffers are used to switch contexts. Although I have not investigated this phenomenon with Crystal (because of its lack of context buffers), this seems practical. The biggest difficulty in performing this action would be in distinguishing between situations where a context switch is required from situations where some other form of context expansion is required.

[3]The exponential cost of context expansion is explained in the next section.

[4]By complete, I mean an entire encyclopædic entry, rather than parts of it.

### The Number of Loaded Assumptions

This has all of the advantages of measuring the number of encyclopædic entries, but can also be used to overcome the problem concerning "simple implications" and certain aspects of information structuring.

In section 6.3.3, I suggested a somewhat cumbersome redefinition of contextual implication that overcame the problems associated with differing syntactic complexity between encyclopædic entries. A simpler solution involves associating a fixed amount of effort with every loaded clause. Considering the definition of Relevance($n$), it becomes clear that if every piece of loaded encyclopædic information were to be instantiated, the contextual implications that constitute these instantiations would be offset by the effort in loading these clauses. Consequently, significant Relevance could only be achieved by extra contextual effects resulting from the further combination of this instantiated information. Thus, if we disregard the effort of the input processes, if Relevance(1) $\leq 1.0$ in an empty context, the loading effort has not been offset by the resulting contextual effects[5].

Another advantage of this approach is that it encourages encyclopædic entries to be phrased so that every assumption is utilised when loaded. If an assumption is loaded, but not utilised, it consumes effort but does not produce any contextual effects.

### The Complexity of Loaded Clauses

This has all the advantages of measuring the number of assumptions loaded, but places additional constraints on how information should be phrased in encyclopædic memory.

Recall the problems relating to the syntax of expressions pointed out in section 5.1 and partially overcome in section 5.6.2. The solution was that forms such as:

$$A \supset P \wedge Q \tag{6.7}$$

should be phrased as:

$$A \supset P \tag{6.8}$$
$$A \supset Q \tag{6.9}$$

An alternative solution is to associate an effort penalty with assumptions phrased like (6.7). Rather than associating one effort unit with (6.7), three effort units are associated, one for each contextual implication that will be derived after modus ponens. This means that the contextual implications will balance the effort after simple instantiation, i.e. in a null context, disregarding other types of effort, Relevance($n$) will be the same for both (6.7) and (6.9).

This approach to effort is the one adopted by Crystal.

### Utility and Salience

The system could be made even more efficient by taking notions of utility and salience into account. If the assumptions with the highest utility are loaded, (by the very meaning of utility) this should result in the most contextual effects. Thus, loading effort could also be related to utility. Salience is a contributing factor to utility because prominent information should be that which is currently being used. This can also make context expansion more selective: only clauses requiring the least effort should be loaded.

[5] I am assuming that loading an assumption requires exactly one effort unit.

The "complexity" based approach currently adopted by Crystal can be regarded as a kind of expected utility value. The loading effort can be viewed as an expected inference yield in a null context. If the actual information yield exceeds the expected yield, then the utterance representation should be connecting into the context.

Although formal notions of salience and utility have not been built into Crystal, I will discuss a way of extending the knowledge representation to do so in chapter 9.

## 6.5 Relevance and Context Expansion

In this section, the relationship of Relevance to context expansion is examined. A problem for SW's exposition of context expansion is examined. Then, in the light of SW's own statements and associated work by Blakemore, a revised approach to Relevance and context is suggested which solves the problem.

### 6.5.1 Problem: Expansion is Too Costly

Among GG's criticisms of RT is a very significant observation: in general, Relevance($n$) will decrease as $n$ increases. The reason for this is that context expansion cannot be easily directed. Vasts amounts of information will be loaded into the context, but little of it will be of much use. In other words: why, if Relevance is unlikely to increase, should the context *ever* be expanded?

SW themselves seem to support the idea that context expansion is generally to be avoided. Sperber and Wilson (1982a) states:

> Given the cost of expanding the context and the lack of flexibility in amount of processing, the search for an adequate context tends to remain within a predictable and generally narrow domain. In trying to maximise relevance, the speaker must adapt to this fact, and the hearer can assume that he has. (pp. 77)

This is tantamount to saying that most of the information that the speaker intends to communicate should be very accessible.

### 6.5.2 Another Approach To Context Expansion

It was established in chapter 4 that, in order to understand an utterance, the context must be expanded (from the encyclopædic memory) at least once. Thus, there is always an initial estimate of Relevance, Relevance(1).

Using this initial estimate, it must be decided whether the context should be expanded, and if so, how it should be expanded. However, I believe that the proper function of Relevance is not generally to determine when the context should be expanded, but rather to determine when expansion should stop. SW generally neglect the fact that we often have particular goals in mind during communication. For example, if I ask someone a question, I do not necessarily stop processing their reply when it becomes Relevant, I stop when my question has been satisfied in some way. Under these circumstances, Relevance can play a rather different role: it can determine when to stop attempting to satisfy a goal because it has become too costly.

There are three basic situations that require context expansion:

1. Relevance(1) is grossly low (ie. well below the point where amount of information change balances exactly with amount of effort expended).

2. There is some directive which instructs the context to be expanded.

3. Some goal has not been satisfied.

I argue that for each case, Relevance is necessary for determining when context expansion should stop.

### Low Relevance

If the Relevance of an utterance is extremely low, this indicates that the situation is abnormal in some way. Presumably, no links into the context have been made. There are several reasons why this may be the case, among them:

- This is the first utterance in a new discourse segment, for example the first sentence of a story.

- The previous context is no longer valid, for example a new conversation topic has arisen.

These situations can be made Relevant by meta-reasoning about the Relevance of the utterance itself. For example, in the first case, it can be reasoned that, since this is the first sentence of a story, the Relevance is not expected to be high. This in itself is a contextual implication that will make the utterance more Relevant.

Context expansion in these cases is a specialised affair. It is not necessary to expand the context with arbitrary encyclopædic information, it is only necessary to expand it with information about Relevance. Thus, expansion (if any) is directed and its cost is minimal.

Let us consider a trivial example.

Once upon a time, there was a vicious poodle named Percival. (133)

If this is uttered in an empty context, it will not immediately result in any non-simple contextual implications and Relevance will be relatively low. Let us assume the hearer is aware of this fact and it is therefore an assumption in the deductor memory. In this particular case, the utterance is delivered in a style consistent with it being the first sentence in a story. If some specialised rules are available to reason about this kind of phenomenon, they may derive this fact. Such a derivation is a contextual implication derived from (133) and the belief that the Relevance is low. Thus, the fact that (133) is initially not Relevant results in restricted reasoning in order to infer the reason why. If a reason can be found, it may make the utterance Relevant. If not, the utterance is indeed incomprehensible.

### Directives for Context Expansion

Blakemore (1987) suggests that certain utterance representations include directives concerning context expansion (see chapter 7). Indeed, much of RT as practised by linguists concerns the constraints utterances place on contexts.

A typical directive is 'Expand the context around $A$ until it contradicts the context associated with $B$'. In this case, Relevance is not used to determine if the context needs to be expanded. It tells the hearer when to *stop* expanding the context. If the Relevance gets too low, the hearer should, perhaps, revert to the meta-reasoning approach.

### Satisfying a Goal

SW discuss a situation where Peter utters:

Do you want some coffee? (134)

They state the following about this utterance:

> ...Peter makes it manifest that he wants an answer to his question, and that nothing less than such an answer would be Relevant enough for him at that point. (Sperber and Wilson, 1986, pp. 167)

If this is to be taken at face value, it is claiming certain things about the way Peter understands the answer. Even if Relevance(1) is sufficiently high, Peter may not accept this interpretation. He will only accept an interpretation that answers his question in some way. In other words, Peter will expand the context until he either finds some suitable answer to his question or he decides that it is not worth the effort (i.e. not Relevant) to go on.

Therefore, I suggest an alternative interpretation of the Principle of Relevance: it is not the first Relevant interpretation that is chosen, it is the first *suitable* Relevant interpretation. By suitable, I mean one in which all of the required goals have been satisfied. The context is only expanded when various goals must be satisfied in some way. The kinds of goals I envisage include finding answers to questions, obeying discourse processing directives associated with words such as "but" and "before" (see section 7.6.4), and the resolution of definite references such as "he" and "it".

### Implementation Details

In Crystal, some of the above features have been implemented. Specifically, Crystal utilises certain directives to control context expansion (see Chapter 7) and only expands the context if it needs to satisfy certain goals such as the resolution of definite references. If Relevance($n$) becomes too low, context expansion is stopped. For the case of definite references, this means that the referent is just assumed to exist (see chapter 8). An obvious extension to Crystal would be to add goals that enable it to attempt question answering during a dialogue.

I have not implemented any meta-reasoning about Relevance.

## 6.5.3 Plausibility

SW claim that Relevance and plausibility are related concepts. However, their definitions of information and effort are so vague that it is not possible to evaluate this claim. The methods I have described for measuring information and effort in Crystal allow this claim to be better evaluated. However, my predictions are not in total accordance with those of SW. Consider SW's example:

My son has grown another foot. (135)

There are two readings of "foot" here: the appendage and the measure. SW claim that in a normal context, the appendage reading will be the most Relevant one, for it would have the most contextual repercussions. The sentence is therefore, according to them, badly phrased.

However, I (and many other people) do not find this utterance problematic. They do not accept the "appendage" interpretation because:

- Under normal circumstances, it is not consistent. Most people do not believe that it is possible to grow extra appendages.

- The effort required to revise the appropriate encyclopædic entries to incorporate the idea that people can grow extra appendages is vast. We are generally reticent to give up long-standing (encyclopædic) information about the world because it would make one particular interpretation of an utterance yield more contextual effects.

In Crystal, the "appendage" reading would fail to be Relevant for the following reason. The "appendage" reading would bring in extra information about "sons", "growing", "another" and "feet". Among the information about "growing" is the fact that human beings do not grow new appendages. Thus, the deductive device infers a contradiction from the utterance and the encyclopædic information. The reason maintenance process is subsequently invoked to erase a suitable cause of the contradiction (see section 2.5.6), which will be either the utterance or the encyclopædic information about human appendage growth. The latter is selected because, in Crystal, utterances are taken to be undisputably true. However, the erasure of non-default encyclopædic information incurs a severe effort penalty (see section 7.5.2). Thus, although a consistent reading *can* be found, it is not a very Relevant one because of the effort involved in obtaining it.

By contrast, the alternative "height" reading is not exceptional; information about "growing" includes the fact that people can grow taller. The utterance representation is easily accommodated into a "growing taller" schema, thus becoming more Relevant than its alternative.

Thus, in practice, only one "plausible" interpretation is obtained. The "appendage" reading is implausible because it requires too much effort in order to make sense of it.

Consider another "counterexample" from GG:

George Best walked to the ball. (136)

GG say that, even if there are more contextual implications for the "football" than the "dance" interpretations of (136), this has no bearing on determining the best interpretation.

Using the "simple implication" or "effort" based means of calculating Relevance, this is not true. In a null context, the ball interpretation is preferred because it is easy to connect a football player with a football (using an appropriate script-like schema). However, there is no such connection between a football player and a dance. In the "dance" reading, there is no immediate referent for "the ball". The context will therefore be expanded in order to find one (see section 8.3.2). A referent is still not found, but a huge effort penalty is incurred in context expansion. The Relevance is now too low to continue any further, and the referent is just assumed to exist. The "football" reading is preferred because it provides more information for less effort.

In a fuller context, such as George Best having an evening out with his wife, the "dance" reading matches a script concerning "evenings out". This time, there are two consistent readings. The one preferred will be the one that yields the most inferences: the one that connects into the context best. If the relative Relevance of the two interpretations is similar, the utterance is genuinely ambiguous. However, note that playing football is normally inconsistent with "evenings out": the hearer will expend a lot of effort in trying to find a referent for "the (foot)ball" before deciding that there is no explanation for its presence and just assuming it exists[6]. Thus, the "dance" reading may well be preferred.

From these examples, the following can be concluded. If simple implications are not considered to be contextual effects (or the effort-based approach to Relevance is used), the amount of information inferred *is* significant. Typically, implausible readings will either not provide *any* non-simple implications or provide them for a huge effort penalty, because the necessary context is hard to access. If two interpretations have similar Relevance factors, then they are genuinely ambiguous. Consequently, Relevance and plausibility are related concepts.

## 6.6 Conclusion

In this chapter, I have described some apparently drastic modifications to RT in the area of context expansion. The reader may even be inclined to say that Crystal can no longer be considered to be an instantiation of the theory. My answer to this is that my modifications are not as drastic as they appear.

In their book, SW do not provide many concrete details about the context expansion process. The most significant aspect of their descriptions is that the processing of an utterance should finish when *the hearers believe* that they have processed the utterance so as to achieve optimal Relevance. My particular interpretation of this does not contradict this view, but takes into account an individual's goals. Without any high-level goals, there is never any point in expanding the context with encyclopædic information; a similar point has also been noted by Gazdar and Good. When a particular goal must be satisfied, it is still important that the utterance is optimally Relevant in the context. However, in this modified situation, "optimally Relevant" means that the goal can be satisfied using the existing processing mechanism with minimal expenditure of effort. If too much effort is expended, then the hearer does not find it worth satisfying.

I am certainly not claiming that Crystal's Relevance/context expansion mechanism manages to process an utterance optimally, but it may pave the way for future research in this area. Such research should particularly concentrate on the problem of selectively expanding the context, probably using notions of manifestness or salience. I shall discuss this further in chapter 9, where I explain how the knowledge representation can be enhanced to handle such notions.

I will end this chapter by summarising the significant points discussed in it.

Information and effort can be represented as integer quantities. It is convenient to calculate Relevance as the information change divided by the effort required to obtain this change.

Most of the information change discussed in this chapter was concerned with contextual implications. We have seen that instantiation of encyclopædic information can be considered to be providing contextual information, resulting in counter-intuitive results with the "experts and obsessives" problem. One solution to this problem is to revise the definition of contextual effects so that simple implications, which are instantiations of encyclopædic assumption schemas, are not considered to be contextual effects.

An even better solution is to associate a loading effort with every assumption based on its expected inference yield in a null context. Utterance representations are then considered

[6] Crystal, however, can find a referent for "the ball" – it assumes it is the ball that Best uses in an imaginary football match. In order to prevent this, Crystal requires the extra information that people do not carry redundant objects around with them.

to be Relevant if they can offset the loading effort with extra information derived from the utterance and the existing context. This approach also explains why it is often intuitively more Relevant to stick to a particular subject: loading effort only has to be expended once for the concepts being discussed.

Relevance is used in two ways:

1. To distinguish between different candidate interpretations.

2. To control context expansion.

The particular method for calculating Relevance is not critical for the first case, because it is only necessary to compare the relative Relevance of various interpretations.

For the second case, Relevance is used to detect exceptional circumstances (like the use of tropes, the beginning of stories, changes of context, etc) and also to determine when to stop expanding the context.

Finally, Relevance is indicative of the plausibility of an utterance. Implausible utterances require a large amount of effort incurred by context expansion or belief revision in order to modify the context in order to explain them. Thus, if there is a simpler explanation, it is usually preferred. However, an utterance that is implausible in one context might be considered plausible in another.

I have now discussed the design of Crystal, its knowledge representation, and the purely deductive components of its inference engine. The next two chapters are directed towards non-demonstrative inference in Crystal.

# Chapter 7

# Non-Monotonic Reasoning

## 7.1 Introduction

This chapter considers the non-monotonic reasoning that is required of the deductive device and any inference engine that performs common-sense reasoning. In chapter 5, I informally described the modal default operator, $\mathcal{D}$. In the course of this chapter, I will provide a semantics for this operator, and show how it can be employed in Crystal's knowledge representation to solve various problems with reasoning about the world. I first discuss the areas of language and reasoning that involve non-monotonic reasoning, and then examine the currently available formalisms for handling it. One of these, autoepistemic logic, is selected as best suited to the needs of an RT approach to language processing. The chapter concludes with a detailed account of the ways in which the chosen formalism handles the reasoning issues discussed earlier.

### 7.1.1 The Need for Non-Monotonic Logic

Non-monotonic reasoning is required of human beings because they are not omniscient. Normally, with mathematical theories, it is assumed that the domain being reasoned about is completely described by a theory. However, humans can never know everything and consequently have to guess how to complete their knowledge. Their guesses are not always correct, hence their inferences must be defeasible.

The following all come under the domain of non-monotonic reasoning:

1. Non-monotonicity in discourse.
2. Defaults and prototypes.
3. Indexical reasoning about belief.
4. Knowledge completion.
5. The solution to the frame and qualification problems.
6. Resolution of inconsistencies.

Each is discussed in turn.

### Non-monotonicity in Discourse

In discourse processing, people invoke notions of prototypical events and properties in order to cope better with certain situations. The application of these prototypes results in defeasible inferences; people's expectations can prove incorrect. For example, O'Rorke (1983) demonstrates the need to maintain non-monotonic dependencies between utterances in a story. He cites the following example from Collins et al. (1980):

He plunked down \$5 at the window. (137)

She tried to give him \$2.50, but he refused to take it. (138)

So when they got inside, she brought him a large bag of popcorn. (139)

People typically make the mistake of assuming that the "she" in (138) is referring to a cashier behind the window. This, of course, is refuted by (139) (unless one assumes that the cashier also follows "him" in). In RT terms, the reason for this is that there are two possible referents invoked by the assumption schemas associated with paying and windows. One is a cashier, the other a person possibly accompanying the payer. A typical non-demonstrative inference that an understanding system must make is that "she" is the cashier. However, it is defeasible; the inference can be revoked in the light of further evidence, namely (139)[1].

Another facet of non-monotonicity in discourse is presuppositions, which have been regarded as inferences of a non-monotonic logic by Mercer and Reiter (1982) and Joshi et al. (1986). The latter also endeavour to explain how people attempt to prevent false inferences involved with interpretations of clauses like "most $P$'s are $Q$"; without reason to believe otherwise, an individual who knows of a $P$ may assume it is a $Q$, making a defeasible inference.

Non-monotonic formalisms have also been used to describe speakers' and hearers' attitudes as they are revised by speech acts (Appelt and Konolige, 1988). Much as RT does not explicitly utilise the notion of speech acts, it is still necessary for it to model the revision of peoples' attitudes.

### Defaults

Default rules are those that generate defeasible inferences, typically used to provide some stereotypical or prototypical information about an entity. A standard example is:

Typically, Birds fly. (140)

Given (140) and the bird Tweety, one can assume that Tweety can fly. If the premise is added that Tweety is a flightless penguin, this will not result in inconsistency; the fact that Tweety can fly will simply not be valid any more.

[1] O'Rorke explains this by assuming that some sort of script has been inappropriately applied, rather than using the notions of assumption schemas and non-monotonic logic. However, I have already explained the correspondence between assumption schemas and script-like mechanisms in section 5.5.

### Indexical Reasoning about Belief

We are capable of reflecting on our own beliefs and drawing conclusions from this. Such inference normally involves reasoning about the presence or absence of beliefs. A typical example is:

$$\text{If John is my brother, I would know it.} \tag{141}$$

As opposed to default reasoning, which is a form of invalid reasoning, indexical reasoning can be regarded as purely deductive (Moore, 1988).

### Knowledge Completion

A more general capability that humans have is being able to decide that they know everything that is required to solve a particular problem. For example, given the information available, they might assume that everything they don't know is not true. Recall the well-known "missionaries and cannibals problem"[2]:

> Three missionaries and three cannibals are trying to cross a river. They have a boat which can hold no more than two people. If the cannibals outnumber the missionaries on either side of the river, the missionaries will be eaten; this is to be avoided. Find a way of getting them across.

Valid, but rarely used, solutions to the problem are "get them to cross the bridge over the river together", or "get them to build a bigger boat". The reason why these solutions are not usually adopted by us is because we assume that the information provided above is all the information that is valid. We also assume that the objects mentioned in the problem are the only objects that need be considered. Furthermore, we have assumed that the standard scenarios for crossing a river in a boat are the only scenarios.

This is a rather extreme case; sometimes it is assumed that everything is known about only one aspect of the problem. In some senses, default reasoning can be thought of in this way. People decide that, for example, all of the birds that are abnormal with respect to flying are known, therefore it can be assumed that all of the birds that are not abnormal can fly. The mode of reasoning that involves assuming that the information known is all that is required is is often called *circumscriptive reasoning* (McCarthy, 1980).

### The Frame and Qualification Problems

So far, I have described modes of reasoning with incomplete knowledge. The prediction of properties and changes in state are also problems which can be avoided by employing a non-monotonic logic.

The real world changes state all of the time. Even a reasoning system that had complete information would find that its data would become outdated very rapidly. It would be ludicrous to attempt to recompute each new state from scratch since this would require vast amounts of processing effort and time; if processing were to be performed in this way, the information regarding a particular state would be outdated by the time it was completed. Most propositions true of a particular state will be true in the following state, which gives an indicator as to how computation of successive states might be effected. The *frame problem* is that of how to reason efficiently about changes in state. One of

[2]This example is due to John McCarthy.

the earliest computer systems to do this was STRIPS (Fikes and Nilsson, 1971) which computed updates of state by rewriting only the germane propositions and leaving the remainder untouched. Unfortunately, STRIPS was not given adequate semantics and its behaviour was only described in terms of the primitive operations it performed. Another system developed to handle the frame problem was the reason maintenance system (RMS) developed in Doyle (1979)[3]. This system attempted to propagate the consequences of change around its data structures efficiently, using associated *dependency information.* More recently, logicians have introduced declarative approaches with the invention of non-monotonic logics (NMLs) that can formalise notions such as "things tend to remain the same over successive state changes".

A similar problem is the *qualification problem.* This is the difficulty in producing rules which may have an indefinite number of exceptions. In classical logic, it is easy to express a rule such as:

$$\forall b.\mathrm{Bird}(b) \supset \mathrm{Flies}(b) \tag{7.1}$$

Unfortunately, there are exceptions to this rule. For example, the bird may be a penguin or an emu. As a consequence, the rule must be revised:

$$\forall b.\mathrm{Bird}(b) \wedge \neg\mathrm{Penguin}(b) \wedge \neg\mathrm{Emu}(b) \supset \mathrm{Flies}(b) \tag{7.2}$$

This is still not sufficient because it is possible that the bird is dead, asleep, or confined in a very small area. It may have broken or lost its wings. It may be tired. In fact, there are an infinite number of possible exceptions to the rule. What is required is a means of expressing the fact that a bird can be assumed to fly unless there is some means of showing that it cannot. RMS systems can tackle the qualification problem by employing the same mechanisms that they use to approach the frame problem.

### Resolving Inconsistencies

In some reasoning systems, information may be received from several different sources of varying reliability. Such information could be integrated into the reasoner's knowledge base which may consequently become inconsistent. Since anything can formally be inferred from an inconsistent theory, it is necessary that the inconsistencies be resolved.

Usually, it is desired that minimal changes should be made to the theory to resume consistency. This is often managed by finding the putative *culprits* which caused the contradiction (Doyle, 1979). The culprits are the set of premises which have ultimately resulted in the problem. Having located the culprits, the inconsistency can be removed by rejecting one or more of them. However, given a set of culprits, and faced with no more than their logical form, it is impossible to make anything more than an arbitrary choice of the culprit.

Consider a simple inconsistent theory:

$$\begin{aligned}
\mathrm{been_outside} \wedge \mathrm{wet} &\supset \mathrm{raining} && (7.3)\\
\mathrm{forecast_sunny} &\supset \mathrm{sunny} && (7.4)\\
\neg(\mathrm{sunny} \wedge \mathrm{raining}) & && (7.5)\\
\mathrm{been_outside} \wedge \mathrm{wet} & && (7.6)\\
\mathrm{forecast_sunny} & && (7.7)
\end{aligned}$$

[3]Doyle called the system a *truth maintenance system* (TMS) in this original paper, but later argued that the title RMS was more appropriate (Doyle, 1983).

A theorem of the above theory is sunny $\wedge$ raining, which contradicts (7.5). The culprits in this case are (7.5), (7.6) and (7.7), but which one should be rejected? Normally, one would assume that personal experience should be put above second-hand knowledge from a weather forecast, but this information is not encoded in the theory. One solution to the problem is to associate measures of reliability of the information in some way.

#### Belief Revision

A central idea of RT and similar theories is that communication essentially involves the communicators updating their belief sets in the light of fresh information. New information is presented to the agents, and they must decide whether to accept it, and if so, how to efficiently propagate the consequences of this belief around their current belief set. This, in some senses, summarises all of the other aspects of non-monotonicity mentioned so far.

Some systems designed to model human reasoning associate degrees of confidence with their individual beliefs. Such degrees can be used to determine which culprit to reject or which of several alternative belief sets is the most appropriate in a given context. Such belief measures may be explicit local number representations as used in the Mycin project (Buchanan and Shortcliffe, 1984), ordering relations as in Gärdenfors and Makinson (1988) or more global criteria based on various factors (Galliers, 1990).

## 7.2 Issues for Non-Monotonic Logic

The previous section illustrated the various ways in which non-monotonicity manifests itself in common-sense reasoning and discourse processing. However, there is currently a proliferation of such formalisms. In this section, I will enumerate some general criteria that can be used to compare these logics, and also the specific criteria that they must meet in order that they be accommodated into Crystal's schema comfortably.

### 7.2.1 General Considerations

The following issues pertain to any non-monotonic formalism:

- How are *multiple extensions* handled?
- Should they be *cautious* or *brave*?
- Should they be *subjective* or *objective*?
- Should non-monotonic reasoning be a meta-level notion?

These will be examined in turn.

#### Multiple Extensions

McDermott (1986b) was one of first advocates of non-monotonic logic to highlight a problem intrinsic to all of the relevant formalisms at that time, the *multiple extension problem*. Unlike the case for monotonic logics such as FOPC, there may be several maximal (with respect to set inclusion) sets of theorems, known as *extensions*, which are mutually incompatible but internally consistent. This was exemplified by the so-called "Yale Shooting Problem". McDermott introduced a scenario where a man loads a gun, waits, and then fires it at someone. Using the standard default persistence axioms, two extensions of the theory can be found: one has the gun successfully shooting the other person, the other has the gun unloading itself during the waiting period. This meant, according to McDermott, that the very formalisms introduced to overcome the frame problem did not work. McDermott even went as far as suggesting that a logical approach to knowledge representation was doomed to failure[4].

The consequence of McDermott's controversial comments was a resurgence of interest in new formalisms which placed priorities on the applications of defaults in order to determine a unique extension (e.g. Konolige (1988a)), or placed more importance on notions of causality and forward-chaining (Shoham, 1988). When considering a non-monotonic formalism, it must be considered if and how it tackles the multiple extension problem.

#### Cautious vs Brave Logics

The possibility of obtaining multiple extensions from a single set of premises gives rise to another problem: what exactly is a non-monotonic theory? Is it the intersection of all extensions, or any single one of them? If the former is the case, it is said that the logic takes a *cautious* approach. If the latter is the case, it is said that the logic is *brave.* Cautious NMLs are not necessarily safe, in the sense that they cannot go wrong. It is still possible for a default to be successfully applied in all extensions, yet complete the information in a way that ultimately proves to be inappropriate. Furthermore, restricting a logic to be cautious may result in a situation where it is only the truly deductive (i.e. FOPC) theorems of the theory that hold in all extensions. On the other hand, there is the problem of how to select between multiple extensions in brave theories. I have already described how a single extension can be selected if a preference order is placed on them, but a problem with such orderings is that they tend to violate modularity constraints. In order to operate successfully, the relationship between all defaults has to be known and expressed in absolute terms (Doyle and Wellman, 1989).

#### Subjective vs Objective Logics

Another issue is whether a logic is *subjective* or *objective* (Levesque, 1990). Premises of a subjective logic implicitly represent the beliefs of an agent at a particular time. Premises of an objective logic do not implicitly represent any agents' beliefs; objective logics explicitly use modal operators to describe the beliefs of a particular agent.

#### Logical vs Encyclopaedic Non-Monotonic Rules

It is not clear at which level non-monotonic reasoning should occur. Some researchers view it as a rational individual reflecting on their own beliefs (e.g. Moore (1988)), others prefer the idea of a theory of theory change (e.g. Reiter (1980)). The former approach tends to result in the actual object language being enhanced with special indexical operators, the latter in enhancements at the meta-level describing how to complete the theory in the object language.

### 7.2.2 Relevance Considerations

A non-monotonic logic that is to be integrated into Crystal must not only provide some solution to the problems expressed in section 7.1.1, it must also fit comfortably into the

[4]He later retracted this view.

system as already described without the need for major modifications to either the theory of Relevance or the existing deductive device.

The logic must therefore have the following properties:

- It must subsume FOPC.
- Its (useful) theories must be tractably computable.
- It must be possible to recognise when default have been overridden in order to calculate information change (see section 7.6.1).
- Its behaviour under a depth-first inferencing strategy must be both predictable and suitable, given that it may be used to determine the particular extension selected.
- It must support the function of the reason maintenance process.

## 7.3 Review of Non-Monotonic Logics

Non-monotonic formalisms have been developed to handle various aspects of the problems already outlined. Many of these systems, while outwardly different, are essentially similar. In this section, I will provide a critical survey of the popular formalisms, placing special emphasis on how they meet the criteria described in section 7.2.

### 7.3.1 Default Logic

Default Logic, in its original formulation (Reiter, 1980), is a typical example of NML as a logic of theory completion. Reiter presupposes the existence of a database of premises in FOPC. Separate meta-level defaults are then applied to it to yield a particular extension. Defaults are of the form:

$$\frac{\vec{F} : \mathcal{M}\vec{P}}{\vec{Q}} \tag{7.8}$$

where $\vec{F}, \vec{P}$ and $\vec{Q}$ are vectors of propositions or formulæ.

(7.8) is interpreted to mean "if $\vec{F}$ is present in the database, and $\vec{P}$ is consistent, then add $\vec{Q}$ to the database". In this case, "$\vec{P}$ is consistent" means "$\neg\vec{P}$ is not part of the final database[5]".

Given this definition, the default rule "typically birds can fly" would be phrased thus:

$$\frac{\mathrm{Bird}(x) : \mathcal{M}\mathrm{Flies}(x)}{\mathrm{Flies}(x)} \tag{7.9}$$

and the persistence of facts can be expressed as:

$$\frac{\mathrm{Holds}(\mathrm{P}(x), t_1) \wedge (t_2 > t_1) : \mathcal{M}\mathrm{Holds}(\mathrm{P}(x), t_2)}{\mathrm{Holds}(\mathrm{P}(x), t_2)} \tag{7.10}$$

Reiter provides an algorithm for applying these defaults to the database. If Bird(*tweety*) is a member of the initial database, then given no evidence to the contrary, the fact Flies(*tweety*) will be added. If the database is enhanced with information that

[5] $\neg\vec{P}$ means the negation of each of the individual elements of $\vec{P}$.

Tweety cannot fly, then the fact that it can fly can no longer be consistently added; hence, the non-monotonicity. In the original formalism, Reiter provides no means of reasoning about provable theorems at the encyclopædic level or for reasoning about the defaults themselves. Clearly, one disadvantage of this from the RT point of view is that default rules would either have to be a separate part of every encyclopædic entry, or defaults would have to be universal rules that form part of the logical rules. This would mean that it would either be the case that only generalities could be expressed or the logical memory would have to be massive enough to contain all of the necessary default rules, which would incidentally make the inference process unacceptably slow. However, it is easy to detect when a default is applied and, with a little more difficulty, when it has been overridden.

Reiter identified an interesting subset of the set of defaults known as *normal* defaults. These are defaults of the form:

$$\frac{\vec{P} : \mathcal{M}\vec{W}}{\vec{W}} \tag{7.11}$$

These have the property of *semi-monotonicity*. That is, the successful application of a default rule can only result in more theorems, it never disables the application of other default rules. Much as this leads to convenient, predicatable behaviour, it does not allow for the definition of priorities between defaults. Nevertheless, many interesting defaults, including the frame persistence axioms and closed-world assumptions can be expressed using normal defaults.

A property of default logic extensions is that they are *orthogonal*; the union of any two extensions of the same theory is inconsistent.

Default logic was finally provided with an adequate semantics in Etherington (1987b). Because he accounts for theories that are not just semi-monotonic, Etherington's semantics are *global*; the meaning of each default cannot be determined independently. However, Lukaszewicz (1985) claims that, in any case, truly non-normal defaults do not make any sense, and attempts a rational translation of all rules into normal defaults.

Enhancements of default logic have included the ability to explicitly name defaults and prevent their application by assertion at the encyclopædic level. If this is allowed, priorities can be expressed using normal defaults (see Poole (1988)).

I have discussed default logic in order to introduce some of its key properties. However, since it is subsumed by autoepistemic logic, the reader is referred to the following section for a discussion of how it meets the criteria enumerated in section 7.2.

### 7.3.2 Nonmonotonic and Autoepistemic Logic

Because autoepistemic logic is used by Crystal, I shall describe Doyle's RMS, nonmonotonic[6] logic and its successor, autoepistemic logic, in some detail.

#### Doyle's Reason Maintenance System

In order to explain the ideas behind nonmonotonic logic, it is useful to illustrate its connection with the RMS. McDermott and Doyle (1980) invented nonmonotonic logic to provide semantics for the original TMS.

[6] I have adopted McDermott's spelling of "nonmonotonic" here so as to contrast his particular logic with general, non-monotonic logics.

The TMS consisted of a set of nodes, each of which had some proposition associated with it. A node was either regarded as a premise, or the result of some logical inference. Each node consequently had an associated *justification set*, a set of possible reasons for believing the contents of the node to be true[7]. The node also had a field known as the *support status*, which could hold the values "in" or "out". A node was "in" if there was some valid reason for believing it to be so, and "out" otherwise. The reasons, known as *justifications* were 2-tuples of the form:

SL(*inlist*, *outlist*)

"SL" stands for "support list". This was used to distinguish it from non-standard justifications which I do not mention here. The inlist was a list of nodes, all of which had to be "in" for the justification to be valid. The outlist was a list of nodes, all of which had to be "out" for the justification to be valid. A *well-founded valid justification* was a justification whose inlist and outlist were both valid. Thus, a node was labelled "in" if there was a single well-founded-valid justification in its justification set.

Given a set of nodes, Doyle provided an algorithm which updated them as new information came along. Using the justification-based structure, the consequences of new beliefs could often be efficiently propagated around the system. Furthermore, justifications could easily be used to express non-monotonic dependencies.

For example, given a rule that typically birds can fly, and given an extant node Bird(*tweety*), an inference engine might add the node Flies(*tweety*) to the TMS as in figure 7.1. Flies(*tweety*) is justified by Bird(*tweety*) being "in" and ¬Flies(*tweety*) be-

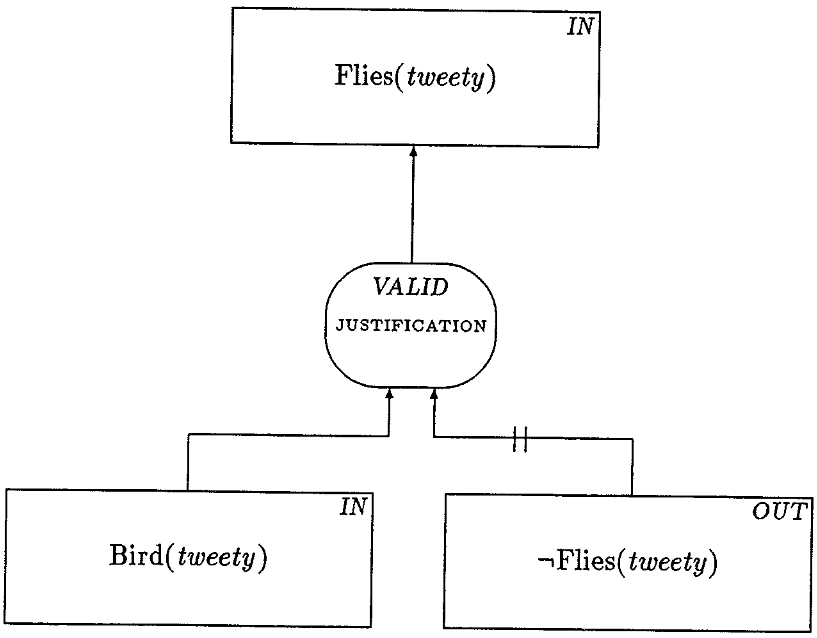



Figure 7.1: An RMS Representation of a Default Inference

ing "out". Nodes in the justification's inlist are shown connected to it by an uncrossed

[7]By contents, I mean the logical proposition associated with the node.

vector; those in its outlist are connected by a doubly-crossed vector. At this point, the justification is well-founded-valid.

The addition of the information that Tweety is a non-flying penguin might result in the state illustrated in figure 7.2 where the Flies(*tweety*) node has been made "out" The justification is no longer well-founded-valid because a node in its outlist is "in".

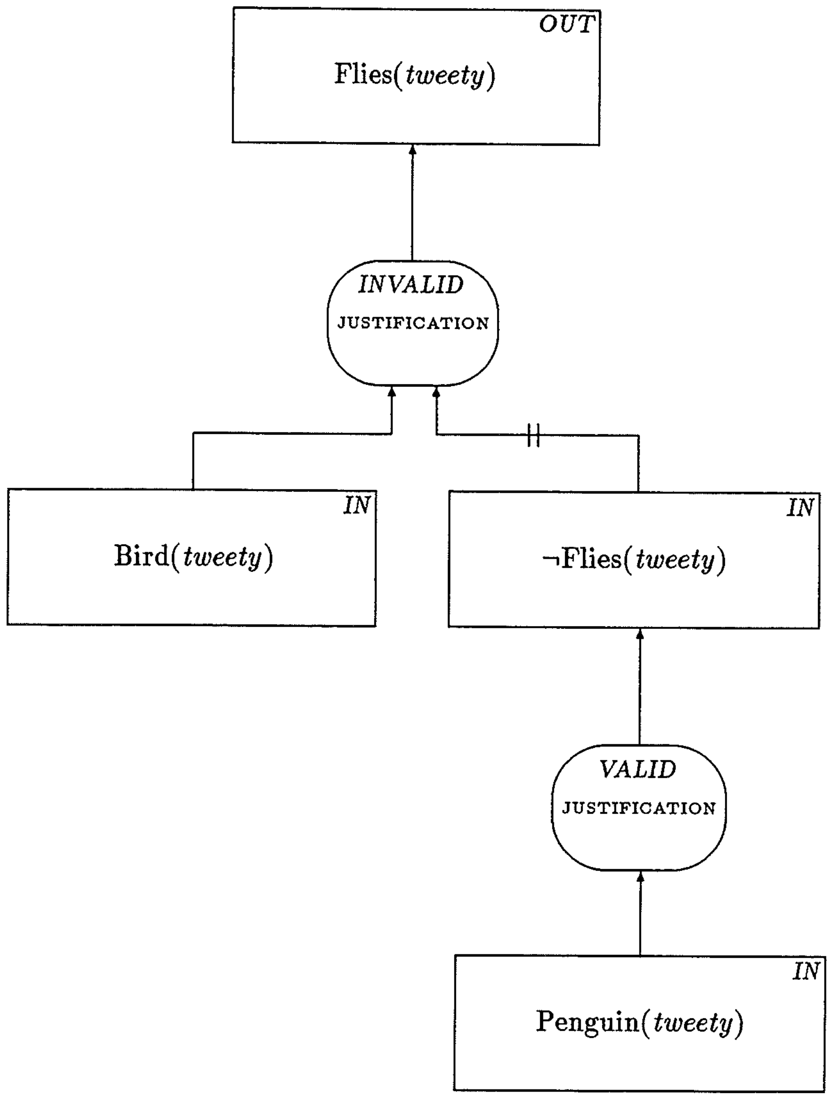



Figure 7.2: The Default Inference is Overridden

It is straightforward to detect the overriding of defaults when using an RMS.

Generally, as new information is added to the system, only the nodes that need be altered are examined. This means that theories can be changed incrementally, solving certain aspects of the frame problem.

Note that the function of an RMS is to maintain the consistency of nodes. The RMS itself has no knowledge of the actual content of the nodes and no logical deductive capability (although a slightly different approach is taken in McAllester (1980)). It is up to a separate inference engine to make any necessary inferences and provide sensible justifications. Given these justifications, an RMS will maintain "consistency".

### Nonmonotonic and Autoepistemic Logic

Doyle's original intention was that the contents of the nodes should be logical propositions. It was with this in mind that he and McDermott invented nonmonotonic logic. This rationalisation of the function of the RMS augmented propositional calculus with the modal consistency operator "$\mathcal{M}$".

$\mathcal{M}P$ can be interpreted to mean "$\neg P$ cannot currently be proved". In their initial paper, McDermott and Doyle do not provide strong enough semantics; their semantics allow $\mathcal{M}P$ and $\neg P$ to be true simultaneously. McDermott (1982a) attempts to improve on his previous attempts to provide a semantic foundation for the logic by considering the standard modal logics T, S4 and S5 (see Hughes and Cresswell (1968) for more detail on these logics).

Moore (1985b) identifies certain problems with McDermott's approach: McDermott includes $P \vdash \neg\mathcal{M}\neg P$ as a rule of inference, and only considers T, S4 and S5. Even though he felt S5 might be the most suitable logic, McDermott discovered its use resulted in a monotonic logic.

Moore goes on to invent a new logic of reasoned belief known as *autoepistemic logic.* He circumvents the need for McDermott's incorrect axiom by providing a new means of defining extensions, which he calls *stable expansions.* Moore defines the operator "$\mathcal{B}$" as an operator of belief (defined such that $\mathcal{B}P \equiv \neg\mathcal{M}\neg P$). The semantics of a logic with this operator are those of *weak S5*, S5 without the axiom $\mathcal{B}P \vdash P$ which would result in monotonicity.

### Autoepistemic Logic and Defaults

The flying bird rule can be phrased in autoepistemic logic as:

$$\forall x.\text{Bird}(x) \wedge \neg\mathcal{B}\neg\text{Flies}(x) \supset \text{Flies}(x) \tag{7.12}$$

which can be read as "if $x$ is a bird, and I have no reason to believe that it cannot fly, then it can fly".

This looks similar to a normal default in default logic. In fact, Konolige (1988b) shows that under certain circumstances they are formally equivalent. The default:

$$\frac{\alpha : \mathcal{M}\beta}{\omega} \tag{7.13}$$

translates directly into:

$$\mathcal{B}\alpha \wedge \neg\mathcal{B}\neg\beta \supset \omega \tag{7.14}$$

This is a little different from the autoepistemic version of defaults, a typical example of which was (7.12), because of the $\mathcal{B}\alpha$ rather than $\alpha$. Konolige states that autoepistemic defaults are more in accordance with his intuitions.

Moore claims that default reasoning and autoepistemic reasoning differ in that defaults are defeasible and therefore cannot be regarded as valid inferences. Much as this seems intuitively true, there is nothing in Moore's semantics that supports this distinction. All types of reasoning within autoepistemic logic are valid in the sense that all premises and theorems proved from those premises will be true under all interpretations.

In Moore (1988), he also claims that quantifying into the modal operator should be disallowed because of confusions due to possible equality of objects with different names. A way around this for simple cases is to allow schema that represent all of their ground instances, so (7.12) would be rephrased as:

$$\text{Bird}(x) \wedge \neg\mathcal{B}\neg\text{Flies}(x) \supset \text{Flies}(x) \tag{7.15}$$

### Other Autoepistemic Logics

By contrast to McDermott's logic, autoepistemic logic is brave. Any stable expansion is regarded as a theory. One problem with this is that it does not offer any insight into the problem of dealing with conflicting defaults.

Konolige (1988a) proposes a solution that involves the use of ordered autoepistemic subtheories. Sentences are effectively labelled with a priority level. Defaults at higher levels take precedence over those at a lower level. Non-determinism of extensions is no longer possible because conflicting defaults are not allowed to occupy the same level. The result is that a single extension is always obtained, whether this is a desirable property of the logic or not.

Levesque (1990) provides another kind of autoepistemic logic which is objective. It offers the advantages that it is monotonic, allows expressions such as "$P$ is all that is believed" (i.e. $P$ is all of the relevant information), and allows quantification into modal contexts (by assuming that all names are distinct).

### Dependency-Directed Backtracking

Much as autoepistemic logic describes the way justifications can be used to maintain consistency of a theory, there is another aspect of reason maintenance that it ignores: this is the restoration of consistency in the face of a contradiction. Doyle's original RMS could attempt this if informed of a contradiction by the separate inference engine.

The RMS restored consistency by tracing back through the justifications to find nodes that were not circularly dependent on any other nodes. These were the candidate culprits. A particular culprit would then be chosen and subsequently rejected by creating a node that made its justification invalid. The consequences of its change in support status would be propagated around the system. Should another inconsistency occur, the RMS would be recursively reinvoked. Ultimately, either a suitable culprit would be rejected or the theory found to be irreparably inconsistent. Doyle's original formulation only allowed nodes that had justifications with a non-empty outlist to be rejected, since this was possible by bringing one of the outlist's nodes "in".

The RMS does not backtrack through inferences in chronological order, but maintains declarative information about how the information was derived and which items are incompatible. This enables the revision mechanism to avoid reconsidering already-computed combinations. This process is known as *dependency-directed backtracking* (Stallman and Sussman, 1977).

The backtracking mechanism ideally should try out alternative extensions of a theory. However, some such mechanisms actually modify the premises to achieve a new consistent theory.

Much as there is very little research on consistency restoration, it is a significant aspect of a Relevance-based reasoning system; new information and non-demonstrative inferences can lead to inconsistencies which need to be resolved in some way.

### 7.3.3 Circumscription

Circumscription is a second-order theory about FOPC theories that completes them by minimising some aspect of the logic. The earliest notion of circumscription was *predicate circumscription* (McCarthy, 1980), which minimises the extension of a particular predicate in all models. For example, given the following theory, $\Delta$:

$$\forall x.\mathrm{Lion}(x) \supset \mathrm{Mammal}(x)$$

$$\neg\mathrm{Lion}(\mathit{tweety})$$

$$\mathrm{Mammal}(\mathit{leo})$$

The circumscription of "Mammal" in $\Delta$ yields:

$$\Delta \wedge (\forall x.\mathrm{Mammal}(x) \Leftrightarrow (x = \mathit{leo}) \vee \mathrm{Lion}(x))$$

In other words, "Mammals are only either lions or Leo". The circumscription of the predicate has resulted in an augmented theory which carries the extra assertion that all that is true of "Mammal" is contained in $\Delta$.

A commonly used subset of circumscriptive techniques is the *closed-world assumption* (CWA). Theories that invoke the CWA assume that if the truth of a formula cannot be proved, it can be assumed false. A typical example of the CWA is the use of negation-by-failure in databases queries. Unfortunately, such reasoning can lead to inconsistencies. Given the theory $P \vee Q$, both $\neg P$ and $\neg Q$ can be inferred using the CWA. However, $\neg P \wedge \neg Q$ is inconsistent with $P \vee Q$. Nevertheless, there is an interesting subset of FOPC that will yield consistent solutions when circumscribed (see Genesereth and Nilsson (1987)).

#### The Circumscription Schema

McCarthy's original paper defined circumscription such that the circumscription of $P$ in $A(P)$ is the sentence schema:

$$A(\phi) \wedge \forall \vec{x}.(\phi(\vec{x}) \supset P(\vec{x})) \supset \forall \vec{x}.(P(\vec{x}) \supset \phi(\vec{x})) \tag{7.16}$$

where $P(x_1, x_2, \ldots, x_n)$ is written $P(\vec{x})$, and $A(\phi)$ is the result of replacing all occurrences of $P$ in $A$ by $\phi$. The above can be interpreted as meaning that, assuming $A$ is true, the only $\vec{x}$ that satisfy $P$ are those that have to.

Given the above formula, a suitable $\phi$ must be found to satisfy (7.16). So, for example, given the theory:

$$\mathrm{Ewe}(\mathit{agnes}) \vee \mathrm{Ewe}(\mathit{eunace}) \tag{7.17}$$

circumscribing "Ewe" in (7.17) we get:

$$\begin{aligned}&(\phi(\mathit{agnes}) \vee \phi(\mathit{eunace})) \quad \wedge \\ &\quad \forall x.(\phi(x) \supset \mathrm{Ewe}(x)) \quad \supset \quad \forall x.(\mathrm{Ewe}(x) \supset \phi(x))\end{aligned} \tag{7.18}$$

Letting $\phi(x)$ be $x = \mathit{agnes}$ gives:

$$\begin{aligned}&(\mathit{agnes} = \mathit{agnes}) \vee (\mathit{agnes} = \mathit{eunace}) \quad \wedge \\ &\qquad\qquad \forall x.(x = \mathit{agnes} \supset \mathrm{Ewe}(x)) \quad \supset \quad \forall x.(\mathrm{Ewe}(x) \supset x = \mathit{agnes})\end{aligned} \tag{7.19}$$

This reduces to:

$$\text{Ewe}(agnes) \supset \forall x.(\text{Ewe}(x) \supset x = agnes) \tag{7.20}$$

Similarly, substituting $x = eunace$ for $\phi(x)$, the result is:

$$\text{Ewe}(eunace) \supset \forall x.(\text{Ewe}(x) \supset x = eunace) \tag{7.21}$$

From, (7.17), (7.20) and (7.21) we get:

$$\forall x.(\text{Ewe}(x) \supset (x = agnes)) \vee \forall x.(\text{Ewe}(x) \supset (x = eunace)) \tag{7.22}$$

i.e. Either Agnes or Eunace is the only sheep. Predicate circumscription is more powerful than the CWA since it can handle the cases that lead to inconsistencies when using negation by failure. It is immediately apparent that finding $\phi$ is a non-trivial task. In many ways, one has to know the correct substitution before one tries it out.

### Circumscription and Defaults

It turns out that McCarthy's original formulation of circumscription is not powerful enough to deal with even the simple flying bird example. This is because the circumscription of a particular predicate in a theory cannot alter any other predicates (Etherington et al., 1984). This was overcome by the introduction of *parallel circumscription* which allows some predicates to be minimised while others may vary, and *formula circumscription*, which allows a formula to be circumscribed in a similar fashion (McCarthy, 1984).

Defaults can be phrased in terms of an abnormality predicate. A theory of birds may look something like:

$$\forall x.\text{Bird}(x) \wedge \neg\text{Abnormal}(x) \supset \text{Flies}(x) \tag{7.23}$$

$$\forall x.\text{Penguin}(x) \supset \text{Bird}(x) \tag{7.24}$$

$$\forall x.\text{Penguin}(x) \supset \text{Abnormal}(x) \tag{7.25}$$

Given Bird(*tweety*), nothing can be inferred as to whether it can fly as it is not known whether it is abnormal[8] or not. One solution is to circumscribe the abnormality predicate so that only penguins satisfy it (in this case). For more general taxonomic hierarchies, parallel circumscription in conjunction with several different abnormality predicates can be used to obtain the necessary results.

### Computing Circumscription

The major difficulties with circumscription involve deciding what to circumscribe and then finding a suitable substitution. Lifschitz (1985) shows that there are several stereotypical situations where particular substitutions are required. He goes on to define an interesting subset of theories that always result in consistent circumscriptions that are trivially calculated, known as *separable formulas*.

More recently, methods have been discovered for converting certain circumscribed theories into logic programs (Gelfond and Lifschitz, 1989) and building more general circumscriptive theorem provers (Przymusinski, 1989).

[8] To be precise, the use of "abnormal" here means "abnormal with respect to flying".

### Other Types of Circumscription

There are several other types of circumscription. These include:

- Protected Circumscription. This allows the user to state which objects are and are not included in the process (Minker and Perlis, 1984).
- Domain Circumscription. This assumes that the only individuals that exist are those mentioned by the formulæ. (McCarthy, 1984).
- Pointwise Circumscription. Viewing a predicate as a set of propositions, this makes false as many of these propositions as possible, hence making the predicate "stronger" (Lifschitz, 1986).

### Circumscription and Relevance

Circumscription is cautious; given conflicting defaults, all it will say is that only one of them can apply, it does not say which one. Hence, the Yale shooting problem cannot be solved by the above methods of circumscription. As with the other logics mentioned above, preferences can be put on which predicates to prefer using *prioritised circumscription* (McCarthy, 1984). This suffers from the same problems as hierarchical autoepistemic logic.

From a Relevance viewpoint, there are three significant problems with circumscription.

1. Since the result of circumscribing something is a normal FOPC theory, it is impossible to distinguish a defeasible inference from any other.
2. If new information is brought into the system, the circumscription must be recalculated which would be an expensive process.
3. The "abnormality" predicates used to define hierarchies do not correspond to any natural language concepts.

Much as research is not complete in this area, there is some evidence that circumscription is subsumed by autoepistemic logic (see Etherington (1987a) and Konolige (1989)), within which it possible to overcome the above problems.

## 7.3.4 Other Formalisms

There are several other interesting formalisms that have been developed to cope with non-monotonic reasoning.

### Taxonomic Hierarchies

It is possible to augment the taxonomic hierarchies discussed in chapter 5 so that they can handle exceptional cases. These systems have the advantages of structured knowledge representations, but some also have the benefit of firm mathematical underpinnings (Touretzky, 1986).

Taxonomic hierarchies can suffer from a variant of the multiple extension problem. Touretzky et al. (1987) provides a semantics for taxonomic hierarchies in terms of default and nonmonotonic logic. He shows that where our intuitions clash about which properties should be inherited, the equivalent logical theories have multiple extensions.

The problems of using structured representations were dealt with in chapter 5; the only use of the representations would be to augment Crystal's sort hierarchy. Even under these circumstances, either the inferences made would have to be duplicated at the encyclopædic level using some other formalism or some method for calculating information change would have to be added to the appropriate functions. This would remove most of the efficiency of these hierarchies.

### Poole's Framework

This approach adopts the view that defaults are ways of augmenting FOPC theories to yield new FOPC theories.

Poole (1988) describes a system, implemented in a program called THEORIST, that takes a FOPC theory and adds further sentences to it as long as they maintain overall consistency. Sets of sentences may be expressed as schemas that represent all of their ground instances. A subset of the ground instances may be added to the original FOPC theory. Defaults are expressed as a subset of normal defaults, which I shall call *simple defaults.* These can be formalised in autoepistemic logic as:

$$\neg\mathcal{B}\neg P \supset P$$

Such defaults are semi-monotonic, thus they cannot deal with the multiple extension problem. However, Poole allows these defaults to be named and explicitly referred to, and introduces *constraints*, which limit the applicability of some defaults in particular contexts. In this way, it is possible to express the equivalent of Reiter's defaults and therefore an arbitrary theory of autoepistemic logic with ordered defaults.

The technique of progressively adding sentences from a subordinate theory can be extended to include several different levels of sub-theories, each having some priority in relation to the other. This would provide another means of solving the multiple-extension problem.

From the RT point of view, this is a nice way of looking at defaults. The FOPC contents of the deductive device could be augmented by adding less specific information from encyclopædic entries when required.

Having added these, though, there does not appear to be a way of making efficient updates in the light of new evidence (and therefore no efficient way of checking if defaults have been overridden).

### Epistemic Entrenchment

This is an approach which addresses the problems of general belief revision (Gärdenfors and Makinson, 1988). It has most in common with Poole's system of adding consistent information. However, its chief theoretical contribution concerns its approach to belief revision. Given a new sentence that is inconsistent with the current database, the database is updated according to a principle of minimal change.

According to Gärdenfors, a database can be *expanded*, by adding consistent information, *contracted* by removing information, or *revised* by first contracting the relevant aspects of the database and then adding information that would have otherwise resulted in inconsistency. Clearly, revision is contraction followed by expansion.

*Rationality postulates* are provided that describe the properties of the functions that modify a database, known as *updating functions.* The rationality postulates do not identify a single updating function, but several classes of them. A constructive way of determining a single updating function involves the use of *epistemic entrenchments*. Informally, these describe the relative importance of a sentence to a theory; a more important sentence is one that would cause more changes if it were to be removed. Such a sentence is said to be more epistemically entrenched. Five axioms define the ordering relation for entrenchment.

The interesting thing about epistemic entrenchment is that it attempts to provide a uniform way of looking at belief revision. This is not the case with RMSs, which separate the propagation of new information from the removal of conflicting assumptions via dependency-directed backtracking.

Gardenförs has a different view of belief revision that does not look at data in terms of defaults and non-defaults. This proves problematic for RT; since there is no notion of defaults, it is impossible to detect when defaults are being overridden. If an addition causes a contradiction, the "culprit" will be the least epistemically entrenched cause of the inconsistency. This, however, is not guaranteed to be the item regarded in non-monotonic logics as "default information"[9]. Thus, one way of viewing default rules is that they give explicit indicators of which information should be given up when an inconsistency occurs.

As new information is input to the system, information change could be easily measured as items are explicitly added or rejected. However, the ordering relation will have to be continually updated, which may not be an efficient process. Despite all of these problems, epistemic entrenchments are one of the most intuitively appealing mechanisms available to date.

### First-Order Conditional Logic

This is an approach due to Delgrande (1988). A theory of this logic comes in two parts: a FOPC theory and a default theory. Defaults use the *conditional operator*, "$\Rightarrow$". For example:

$$(x)\mathrm{Bird}(x) \Rightarrow \mathrm{Flies}(x)$$

is read as "if x is a bird then it normally flies".

Delgrande's defaults are expressed in the conditional logic, $N$. Properties of default inference rules are then separately axiomatised. This allows independent reasoning about the default rules, which can then be applied to the FOPC theory or vice-versa. Both directions of application yield identical results. Delgrande argues that many of the problems that are assumed to require ordering of defaults in other logics are automatically handled by $N$.

In order make $N$ act as a logic of defaults, Delgrande makes two assumptions.

1. That the model of the theory is one of the least exceptional ones consistent with the default theory.

2. Only those sentences known to bear on the truth value of a conditional relation are assumed to have a bearing on its truth value.

This logic can be used to provide a solution to the Yale shooting problem and provides a notion of adequate defaults. However, a tractable theorem prover has yet to be published for it.

[9]A reason for this may be the fact that epistemic entrenchment is a syntactic property of a formula, whereas strength of belief appears to be a semantic property. I suggest a semantic approach to belief revision in chapter 9.

## 7.4 Choosing a Logic for Crystal

I am now in a position to evaluate the logics described in section 7.3 with respect to the criteria listed in section 7.2. I conclude that non-hierarchical autoepistemic logic is the most suitable for Crystal.

### 7.4.1 Applying The Criteria to the Proposed Logics

In this section, I only consider the formalisms that have not already been dismissed as unfeasible, e.g. reasoning in conditional logic cannot presently be automated and taxonomic hierarchies would need to be replicated at the encyclopædic level, making them redundant; they are consequently beyond consideration.

#### Hierarchic Theories

Logics which place explicit orderings on default applications are not suitable for a system such as Crystal. Given different contexts, the priorities on the defaults may be different (see section 7.6.4).

In my opinion, it is not the responsibility of a general non-monotonic logic to choose between the possible extensions. A separate system should perform this role; it could use meta-logical criteria to select a particular extension if necessary. Thus, ordered theories, prioritised circumscription, and hierarchic autoepistemic logic are ruled out as candidate logics as they place to much responsibility on the logic.

#### Measuring Contextual Effects

One argument for cautious logics is that contextual effects are only counted if they occur under any circumstances. An argument against them is that they may not necessarily represent the state of knowledge of a single agent, but rather what must be true of any one of them.

The only cautious logics discussed were the nonmonotonic logic of McDermott and circumscription. McDermott's logic has been superceded by autoepistemic logic, which is brave (since it represents the beliefs of a single agent). Within circumscription, it is difficult to measure information change.

Of the brave logics, it is easy to measure information change in autoepistemic logic and default logic using the available mechanisms for computing their theorems, but autoepistemic logic subsumes default logic.

#### Subjective Logics

Despite the advantages of objective logic, it is not suitable for Crystal because it does not represent the state of mind of a particular agent. Thus, Levesque's logic is ruled out.

#### Level of Non-Monotonic Reasoning

There are arguments for and against viewing non-monotonic reasoning as a meta-level notion. In its favour, reasoning about defaults is not the same as reasoning about general encyclopædic information. Against it is the the fact that autoepistemic reasoning is sometimes a more appropriate notion. In fact, both modes of reasoning are necessary in

Crystal. Prototypical reasoning is a typical example of completing encyclopædic information (i.e. instantiated prototypes can be regarded as theorems provided by a meta-level reasoner). In section 7.6.4, it will be seen that the semantics of certain natural language constructs may be thought of in autoepistemic terms.

#### Relevance Considerations

Of the logics described, all of them can support FOPC. However, non-monotonic inference engines are widely available for only default logic, autoepistemic logic, circumscription, taxonomies and Poole's defaults. Of these, it is simple to measure information change in only default and autoepistemic logic. Autoepistemic logic has the further advantage that it has a simple RMS implementation; the functionality of the RMS is similar to that of the reason maintenance process.

### 7.4.2 Autoepistemic Logic in Crystal

From the material presented in sections 7.3 and 7.4.1, it is apparent that autoepistemic logic is the logic that seems most suited to Crystal. I have therefore adapted the deductive device to handle a subset of autoepistemic logic. The advantages and disadvantages of the logic and my implementation of it are discussed below.

### 7.4.3 Advantages of Employing Autoepistemic Logic

Autoepistemic logic is suitable for Crystal because:

1. Aside from the most current unexplored theories, there is considerable evidence that most non-monotonic logics are equivalent to, or subsets of, autoepistemic logic.
2. It is easy to measure contextual effects.
3. It has a well-defined semantics.
4. It has close associations with the RMS.

#### Advantages of Using an RMS

A subset of autoepistemic logic can easily be implemented using an RMS. This provides the following useful features:

1. The RMS can propagate the consequences of incremental change through the system in an efficient manner. Thus, successive contexts can be augmented with relative ease.
2. The RMS can interact with a non-demonstrative inference component by employing dependency-directed backtracking when inconsistencies occur. Indeed, the RMS is a more adequate version of SW's erasure algorithm discussed in section 2.5.6.
3. It is a trivial matter to check when a node changes its belief status in order to check when defaults have been overridden.
4. The controller needs no modification to handle non-monotonicity.

5. The interface is the sole point of communication between the RMS and the rest of the deductive device; it only needs to hand control over to the RMS when information in the deductor memory is rejustified, or when a default assumption is asserted.

6. The RMS mechanism requires that every node has a justification set. These sets are already available since they are required by the interface (see section 4.5). In fact, much of the RMS structure is already present.

7. The RMS structures can be used to aid the querying system (see appendix C).

8. The RMS can be adapted to support belief measures, which can be used to select between alternative expansions and guide consistency restoration (see chapter 9).

### 7.4.4 Disadvantages of Employing Autoepistemic Logic

- Proofs and reasoning about provability are allowed at the same level. This conflation of meta- and object level reasoning is arguably undesirable because it treats defaults in the same way as one treats normal encyclopædic knowledge.

- The very notion of a valid default is a little bizarre. The validity of a default certainly does not mean that such rules hold in the real world. If a theory is intended to represent the state of an ideally rational individual's beliefs about the world, it may not be appropriate to give default rules the same status as real-world information. Paraphrasing Moore, if a default rule *always* holds, why is it a *default* rule?

- The semantics of autoepistemic logic (and most of the other default theories) cannot distinguish between appropriate and inappropriate defaults. Consider:

$$\text{Happy}(!x) \supset \mathcal{D}\text{Sad}(!x) \tag{7.26}$$

This can be read as expressing the (ridiculous) fact that happy people are typically sad.

This will be valid even when combined with:

$$\neg(\text{Happy}(!x) \wedge \text{Sad}(!x))$$

$$\text{Happy}(!x) \supset (!x = vlad)$$

The above states one cannot be both happy and sad, and that Vlad is the only happy person. Given the latter two assumptions, it is clear that Vlad is the only happy person, and will never be sad. The consequent of (7.26) will never be true, yet the whole assumption is still valid. Clearly, (7.26) is not an expression of a good default; typicality cannot be phrased solely in terms of consistency and material implication.

## 7.5 Implementing a Subset of Autoepistemic Logic

My implementation of autoepistemic logic is described in this section. A glossary of useful terminology and further implementation information is provided in appendix C.

### 7.5.1 Simple Defaults

If an extra-logical mechanism decides which extension is appropriate, then Poole-style simple defaults are sufficient for Crystal's knowledge representation. Simple defaults have the advantage that they are straightforward to implement and their theories are semi-monotonic, which makes the reason maintenance algorithm simpler.

Simple defaults can be represented using the $\mathcal{D}$ operator, defined in autoepistemic logic by:

$$\mathcal{D}P \equiv (\neg\mathcal{B}\neg P \supset P)$$

So, for example, a persistence axiom can be phrased as:

$$\text{Holds}(!P, !t_1) \wedge (!t_2 > !t_1) \supset \mathcal{D}\text{Holds}(!P, !t_2)$$

#### Extension Selection

In Crystal, the ordering of encyclopædic information combined with the left-right depth-first forward-chaining strategy is used to determine the extension.

In particular, within an encyclopædic entry, later defaults take priority over earlier ones. This is because the entry is loaded backwards into the encyclopædic memory. The depth-first strategy will then attempt to apply the last default first and so on.

Consider the following:

$$\text{Engine}(!e) \supset \exists !f.\, \text{Fuel}(!f) \wedge \text{Contains}(!e, !f) \wedge \mathcal{D}\text{Diesel}(!f) \tag{7.27}$$

$$\text{Engine}(!e) \supset \exists !f.\, \text{Fuel}(!f) \wedge \text{Contains}(!e, !f) \wedge \mathcal{D}\text{Petrol}(!f) \tag{7.28}$$

$$\text{Petrol}(!p) \supset \neg\text{Diesel}(!p) \tag{7.29}$$

$$\text{Diesel}(!p) \supset \neg\text{Petrol}(!p) \tag{7.30}$$

These might form part of an encyclopædic entry about engines. Now, if this were the actual ordering of the entry, it would first be assumed that an engine is a petrol engine. If this is not consistent, it would next be assumed that the engine was a diesel engine.

#### Implementing Simple Defaults

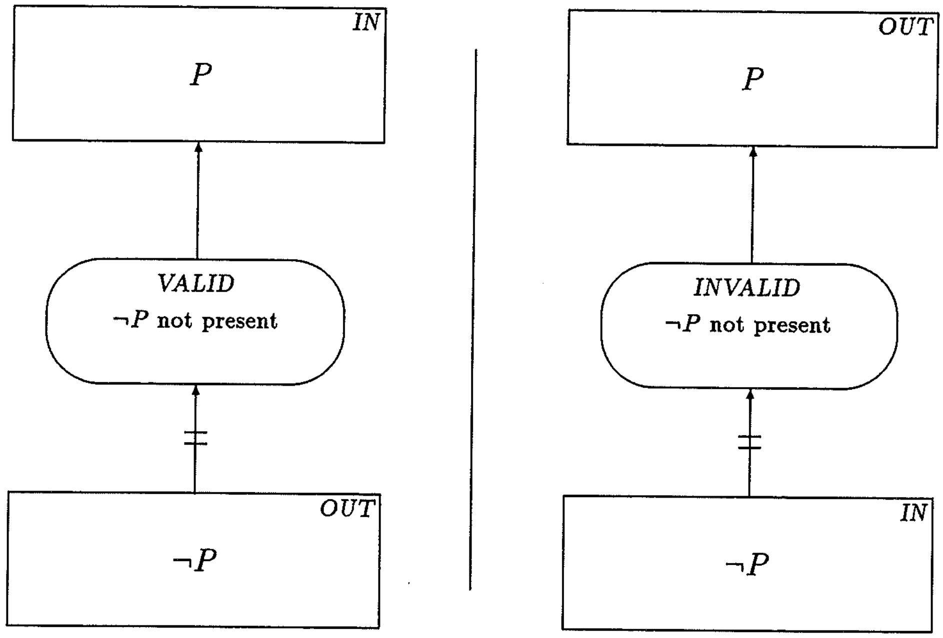
The following modifications are necessary in order for the deductive device to handle simple defaults.

- A meta-rule is present that can detect inconsistencies, explicitly asserting the contradiction (described in chapter 4). This is used by the dependency-directed backtracking process as the root node from which the search for candidate culprits begins.

- The interface is modified to detect assumptions of the form $\mathcal{D}P$ and pass them on to the RMS process rather than merely adding them as single nodes. If $\mathcal{D}P$ occurs as part of a larger assumption, then it is treated as a normal assumption; eventually, meta-rules like and-elimination will isolate the bare default, if possible.

- The RMS handles $\mathcal{D}P$, which is identical to $\neg\mathcal{B}\neg P \supset P$, as follows:

1. It is established whether $\neg\mathcal{B}\neg P$ is the case. This will be true if $\neg P$ is not present (i.e. "in") in the deductor memory, which can be established by the interface.
2. If it is not present, the consequent, $P$ is added as a new node with belief status "in". It is justified as in figure 7.3 (a) to be dependent on $\neg P$ being "out". If any of these nodes are already present, then their justifications sets are merely augmented and their belief status changed.
3. If it is present, the previous step is repeated, but node $P$ has belief status "out"[10] figure 7.3 (b).




(a) "In" Default Added

(b) "Out" Default Added

Figure 7.3: Adding Default Inferences

Hence, the bird example of figure 7.1 would be implemented as in figure 7.4 in practice.

- If a node's support status changes when a new justification is added, the RMS is invoked to propagate the consequences of this change around the deductor memory. The algorithm used for this is similar to that found in Doyle (1979) except that it is simplified by only having "SL" justifications. Furthermore, the "constraint relaxation" part of the algorithm is not necessary with Crystal's use of simple defaults[11]. The reader is referred to Doyle's paper for more detail.

[10] It is assumed that no effort is consumed in adding this default since it has never been "in".

[11] However, this section of the algorithm was implemented in case the defaults were later complicated.

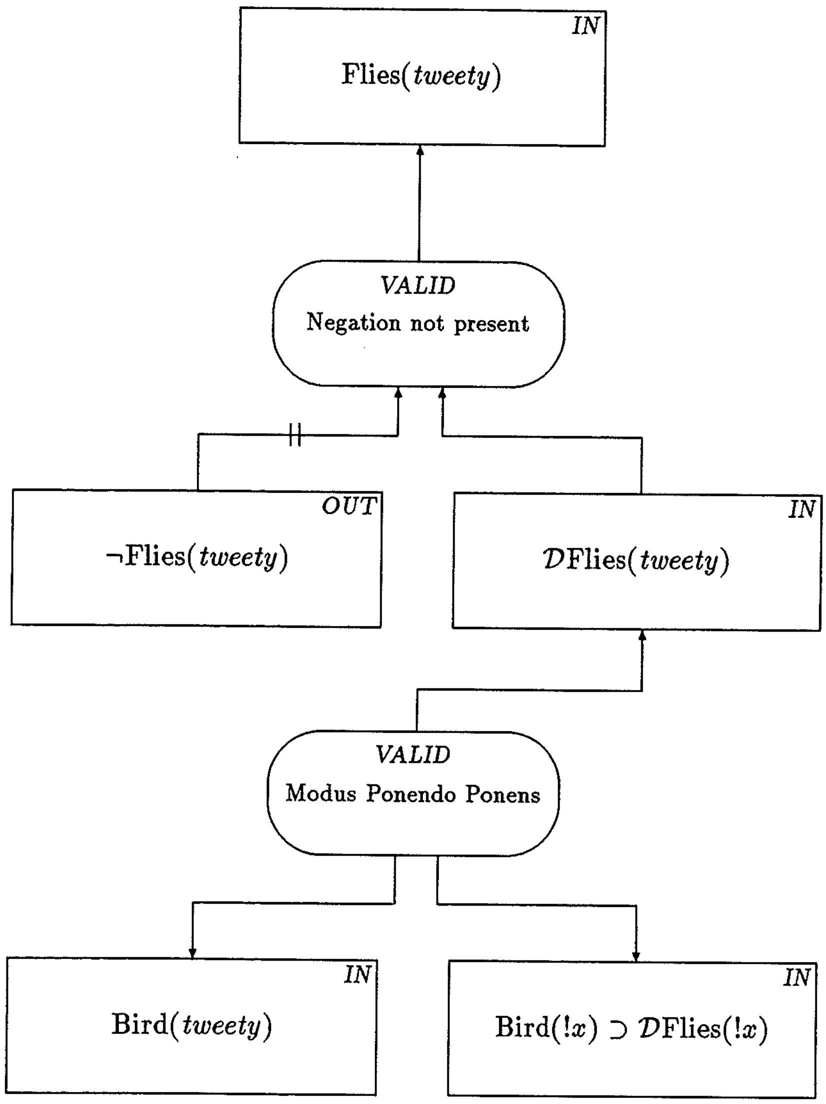



Figure 7.4: A Typical Default Inference Structure in Crystal

### 7.5.2 Restoring Consistency

#### Overview

There are three possible origins of inconsistency in the deductive device:

1. An incoming utterance is inconsistent with the theory in the deductor device. Practically, this means that the input contradicts what the hearer currently believes.
2. Information from different encyclopædic entries results in inconsistency. Effectively, this is similar to the notion that people may hold inconsistent beliefs, but do not realise this until some appropriate stimulus brings them together.
3. Some non-demonstrative inference results in inconsistency; some guess about the world was wrong.

In all cases, the same mechanism must handle the restoration of consistency. In Crystal, simple heuristics are used to select an appropriate culprit. A better motivated scheme should take into account the reliability and importance of the sources of the problem. In chapter 9, I will suggest a revision of this mechanism which uses these concepts.

#### Doyle's Dependency-Directed Backtracking

Doyle's dependency-directed backtracking algorithm used special justifications known as *conditional proof* or CP justifications. These are of the form $CP(C, S)$, where $C$ is a single node and $S$ is a list of nodes. A CP justification is valid if $C$ is "in" whenever the nodes in $S$ are "in".

Given a contradiction node $X$, the algorithm is as follows:

1. A list of putative culprits called the *nogood set* is found. Given the nogood set, $C_1, C_2, C_3, \ldots, C_n$, it must be the case that:

   $$C_1 \wedge C_2 \wedge C_3 \wedge \ldots \wedge C_n \supset \mathit{false}$$

   and since, $A \supset B \equiv \neg A \vee B$:

   $$\neg(C_1 \wedge C_2 \wedge C_3 \wedge \ldots \wedge C_n) \tag{7.31}$$

2. Assumption (7.31) is asserted as a special *nogood* or NG node. It is justified with:

   $$CP(X, [C_1, C_2, \ldots, C_n])$$

   Even if one of the nogood set subsequently goes "out", the CP justification will still be valid.

3. The next step is to choose a particular culprit. In Doyle's system, all of the putative culprits must have justifications[12] with non-empty outlists. This allows the node to be rejected in a straightforward manner.

   Doyle does not say how this selection must be made; in his original TMS, the user was prompted for an answer if there were several candidates.

[12]Much as each node has a set of justifications, only one is used to make the node "in", known as the *supporting justification*. It is this justification that I am referring to.

4. The culprit is rejected by making a node in its justification's outlist "in". Let the culprit $C_i$ be the $i^{th}$ element of the nogood set. $C_i$ will have a justification with outlist $[D_1, \ldots D_m]$. One of these nodes, say $D_j$, will be brought "in" to make the culprit "out". $D_j$ will be justified with the SL justification:

$$\begin{aligned}&\mathrm{SL}([C_1, C_2, \ldots, C_{i-1}, C_{i+1}, \ldots, C_{n-1}, C_n],\\&\quad[D_1, D_2, \ldots, D_{j-1}, D_{j+1}, \ldots, D_{m-1}, D_m])\end{aligned} \tag{7.32}$$

5. The inconsistency node, $X$ should now go "out". However, it is possible that bringing $D_j$ "in" may result in the inconsistency being maintained for a different reason or another inconsistency occurring. If this is the case, the dependency-directed backtracking algorithm is recursively reinvoked. If this is necessary, the algorithm will not merely repeat itself, because the NG nodes keep track of previous attempts to restore consistency.

CP justifications are inefficient and difficult to evaluate, because of their conditional nature.

Given that the *consequences* of a node are the nodes which mention that node in their justification, CP($C$, $S$) is converted into an SL justification as follows:

1. The transitive closure of the justification for $C$, $C_t$ is found.

2. The transitive-closure of the consequences of $S$, $S_t$ is found.

3. $I = C_t - S_t$ is calculated. $I$ consists of the nodes which support $C$ but do not depend on $S$.

4. $I$ is used to form the SL justification. The "in" nodes form the inlist, the "out" nodes the outlist.

CP justifications were converted into SL justifications as they were entered into the system.

### Doyle's Establishment of Candidate Culprits

Doyle defined the putative culprits to be the set of *maximal assumptions* related to the contradiction node. A node is an *assumption*[13] if its supporting-justification has a non-empty outlist, i.e. it has non-monotonic dependencies.

Let the *antecedents* of a node be the nodes mentioned in its supporting justification, and its *foundations* be the transitive-closure of its antecedents.

$$\left\{ m \in \text{foundations}(C) \;\middle|\; \begin{array}{l}\text{assumption}(C) \wedge \\ \neg\exists B \in \text{foundations}(C).\, m \in \text{foundations}(B)\end{array} \right\}$$

The maximal assumptions are those that are inferentially "closest" to the contradiction node that can also be made "out". Since they are inferentially close, making one of them "out" should remove the contradiction with minimal change to the system. This is a heuristic rather than a mathematically proven fact[14].

[13] Doyle's use of "assumption" is different to the SW notion.

[14] The degree of epistemic entrenchment of the foundations of the contradiction node would be a better measure; the foundation with the lowest entrenchment being rejected will guarantee minimal change.

### Modifications to Doyle's Algorithm

Doyle's NG nodes are justified with a CP justification, which is valid if the contradiction is "in" whenever the maximal assumptions are "in". In Crystal, this will always be the case, so NG nodes are asserted as premises with justification:

SL([ ], [ ])

and reason "incompatible assumptions".

Sometimes, there may be no maximal assumptions because all of the foundations of the contradiction have empty outlists. The root causes of such a contradiction will be either the utterance or one of the loaded conceptual entries. All of these will be justified as premises and hence cannot be rejected using Doyle's algorithm. A culprit, $C$, is selected and removed as follows:

1. The supporting justification is removed from the justification set of $C$.
2. $C$ is now converted into the default inference $\mathcal{D}C$ as explained above.
3. $\neg C$ is justified according to the usual dependency-directed backtracking process, making $C$ go "out".

Since it is essential to restore consistency, Crystal is prepared to view any premise as a default. The changing of an encyclopædic premise into a default does not change that item in the conceptual memory; it is merely a temporary measure. The conversion of encyclopædic information into defaults was a relatively rare occurrence, which incurred a severe effort penalty[15].

### Heuristics for Culprit Selection

The putative culprits are collected as a list in the reverse order from which they were inferred. This is possible because:

1. The inferencing strategy is depth-first.
2. Progressive utterances are processed by appending them to the deductor memory.

The head of the list will be the most recently asserted clause.

I will distinguish two types of defeasible inferences:

1. Hypotheses about the referents of referring expressions.
2. All other types of defaults.

Given these, the simple heuristics for rejection are as follows:

- If possible, the most recent non-referring default is rejected.
- Otherwise, the most recent default inference is rejected.
- Otherwise, the most recently asserted putative culprit is rejected.

[15]Five effort units was found to be an appropriate amount.

The reasons for this are as follows. As inferences are made, they will effectively become more entrenched in the context as inferences are progressively made from them. Rejecting the most recent default should result in the least upheavals to the deductor memory, since less information should depend on it as compared with the older defaults. Practical experience supports this: it is generally true that people are more ready to give up contextual information that has just been inferred than they are to give up information from some time back.

The reason why non-referring defaults are preferred for rejection is that these will usually be the least connected. The purpose of many referents is to actually make the discourse cohere. A more principled rejection technique is given in chapter 9.

## 7.6 Applications of the NML in Crystal

Having selected a suitable NML and described its implementation within the reason maintenance process and the deductive device, I now describe the application and utility of this logic in Crystal.

I begin by considering how reason maintenance effects the information and effort measurements in the deductive device. It will be seen that autoepistemic logic, supported by an RMS, can provide a concrete interpretation of SW's ideas concerning defaults and also provide a Relevance theoretic explanation of some other pragmatic phenomena which, to the best of my knowledge, have not previously been given such a treatment.

Following this, I briefly recap how the $\mathcal{D}$ operator is used to help implement assumption schemas and then describe how they can be used to provide some solutions to the problems enumerated in section 7.1.1. Finally, a novel use for autoepistemic logic will be shown that views certain natural language constructs as operators on the context and hence on beliefs.

### 7.6.1 Measuring Changes of Information and Effort in the RMS

In this section, I discuss the way Crystal handles some pragmatic phenomena with the assistance of its RMS, and hence provide a more detailed treatment of them within the framework of RT than has been hitherto proposed. I also provide a unified account of all contextual effects in terms of overall change in belief.

#### Overriding Defaults

SW note that sentences such as (142) are intuitively more informative than (143) and hence more Relevant.

There is a crypt underneath the bungalow. (142)

There is a crypt underneath the church. (143)

The reason for this is that it no surprise that a crypt is underneath a church, but it is not usual for a crypt to be underneath a bungalow. SW claim default assumptions are being overridden in (142) while they are not in (143). Thus, the cancellation of defaults should be regarded as a contextual effect.

### The Cost of Dependency-Directed Backtracking

It is possible to extend SW's ideas to explain why certain utterances containing definite references can be understood but are better phrased in an alternative manner. Consider the following:

Olga took the bucket to her car and washed it. (1)

Olga took the bucket to her car and washed the car (144)

Intuitively, (144) is a better phrased utterance than (1). The reason for this is that the hearer naturally guesses that "it" refers to "the bucket" rather than "the car". It is only common sense reasoning about the fact that people do not usually wash buckets but often wash cars that makes us realise that "it" must refer to "the car".

In my opinion, this can be explained in RT terms by considering the wastage of effort in dependency-directed backtracking. (1) is less Relevant than (144) because it yields the same information for less processing effort.

Note that retraction, when not caused by the cancellation of defaults, is not regarded as a contextual effect (unless information is retracted from a *previous* context, i.e. information of contextual status "Context" is retracted).

### Unifying the Different Contextual Effects

SW treat assertion, retraction and strengthening as distinct contextual effects. However, the notion of strength can be used to unify them. All three effects result in a change in the magnitude of overall strength, $S_t$, of the assumptions in the deductive device: assertion increases $S_t$ by the strength of the new assumption, retraction[16] similarly decreases it, and strengthening changes $S_t$ by the increment in strength of the reinforced assumption. Consequently, the unified information measure is $\Sigma\delta|S_t|$.

### Measuring Information Change

Since Crystal counts all new justifications as contextual effects, it effectively regards strengthening and assertion as the same effect.

For erasures, strength changes are calculated as follows. If a default is overridden, then all the repercussions that change support status are counted as contextual effects.

During dependency-directed backtracking, if any nodes marked "nil"[17] change their support status and were of context type "Context" then this counts as a contextual effect. Thus, when a culprit is rejected, that node and all its repercussions may count as contextual effects.

### Information and Non-monotonic Logic

Autoepistemic logic regards every stable expansion as a valid theory; it takes the brave approach. This is reflected in the RMS algorithm by the fact that it uses a single justification, the supporting justification, to determine a node's support status. The other

[16] Retraction includes both the overriding of defaults and rejection of culprits.

[17] "nil" means marked for consideration by the reason maintenance algorithm; see appendix C for more detail.

justifications are only considered during dependency-directed backtracking, which is a search for an alternative consistent extension.

However, measurement of information uses every valid justification in the justification set. $S_t$ is calculated from one of the internally consistent intersections of the possible stable expansions. This seems to indicate that the non-monotonic logic used by Crystal should be cautious. In such a logic, there would be no notion of supporting justification, since all nodes would contribute to the support status[18]. Crystal's logic does not support this fully. In chapter 9, I will sketch a new, cautious non-monotonic logic with belief measures that overcomes the deficiencies of SW's strength model.

#### Measuring Effort

If an assumption is of contextual status "Context" and changes its belief status, this counts as information. If it is not of this status, then it is an assumption that has been asserted and retracted within the initial context of the current utterance. This assertion and retraction is undesirable; the reasoner is effectively making a bad guess for no benefits.

It is therefore reasonable to assume that retraction also consumes effort. In the initial empty context, this will serve to lower the overall Relevance. In an expanded context, it may also yield contextual effects which overcome this effort.

### 7.6.2 Assumption Schemas

In chapter 5, it was explained how the $\mathcal{D}$ operator could be used to implement assumption schemas, frames, and script-like mechanisms. The precise implementation and meaning of this operator has now been described. The interface and reason maintenance process handle assumptions containing this operator; the inference controller is not affected. The interface effectively regards assertions of the form "$\mathcal{D}P$" as an instruction to assert $P$ if this is consistent.

### 7.6.3 Simple Defaults and Nonmonotonicity Problems

I now show how the default assumptions can be used to provide solutions to some of the problems outlined in section 7.1.1.

#### Nonmonotonicity in Discourse

Assumption schemas can be used to predict properties of both objects and events, using frame- and script-like logic representations. In a discourse, such defeasible inferences are what Griceans call conversational implicatures. The RMS, performing the function of the reason maintenance process, maintains the consistency of the assumptions in the deductor memory.

As an example, consider the following discourse fragment:

John felt suicidal. (145)

There was a gun beside him. (146)

It had run out of caps. (147)

[18] This casts a different light on the role of a cautious logic; it is being used here as a logic describing the information that *must* be inferred by any agent.

Let us consider the inferences which Crystal generates that we might make when reading the above discourse.

To begin with, in utterance (145) we tend to imagine that John is an adult. This is partially because there is no reason to believe he is not an adult, and partially because children do not tend to be suicidal. These are generalised conversational implicatures. We also believe that he might attempt to kill himself, which is a particularised conversational implicature, inferred by applying a script-like assumption schema.

Sentence (146) will be Relevant because we can fit it into the possible killing event (as the instrument), which is a particularised conversational implicature. We also tend to imagine the gun to be some sort of working pistol, rather than, say, a cannon. This is a generalised conversational implicature (and a default). The fact that the gun is next to him means that it can easily fit into some shooting scenario. If the sentence had read:

> There was a gun five thousand miles away,
> buried under six tonnes of concrete. (148)

it would be difficult to fit it into a suicide scenario; in practice, this would result in a contradiction when fitted into a suicide script.

Once again, the conversational implicature that John is an adult is reinforced by the fact that he has access to a weapon.

Some defeasible inferences in these sentences do not fit into the Gricean framework: "him" is assumed to be John and "it" is assumed to be the gun.

So far, each utterance has been Relevant because it connects into the context to produce contextual implications. However, this is not the case for the next utterance, (147). Assuming, "it" is the gun, we are confronted with two possibilities: it has either run out of hats, or it has run out of artificial ammunition. The second reading is preferred, because it connects into some scenario involving toy guns. However, if it is assumed that the gun is a toy gun, it follows that either "it" is not the gun mentioned in (146) or that gun is not a real pistol but a toy gun. Most people decide that the gun is a toy gun and that it therefore cannot form part of the suicide scenario.

Overriding the contextual information derives new information. The net result is that the utterance is Relevant because it contradicts our expectations (in a plausible way). Scripts only suggest the way things may go in unexceptional circumstances. Given no viable alternative, an utterance can make sense because it deliberately flouts our expectations.

### Defaults

The relationship between frames, scripts and defaults has already been considered.

Another use of defaults in Crystal concerns the temporal relationship between successive events. Consider, the following:

> Rambo went indoors. He washed his poodle. (149)

In the absence of any better information, it is assumed that Rambo washed his poodle after he went home. This sort of inference is defeasible. For example, the dialogue might continue:

> He cleaned it before he went inside because he didn't want
> it to cause a mess on his Afghan rug. (150)

(149) might be represented as follows:

$$\begin{aligned}&\mathrm{Go}(\mathit{rambo}, \mathit{Sk.Indoors012}, \mathit{Sk.Go.Event003}) \wedge \\ &\quad \mathrm{Indoors}(Sk.Indoors012) \wedge \\ &\quad \mathrm{OCCUR}(Sk.Go.Event003, Sk.Go.Interval004) \wedge \\ &\quad (Sk.Go.Interval004 < NOW)\end{aligned} \tag{7.33}$$

$$\begin{aligned}&\mathrm{Male}(Sk.Male046) \wedge \mathrm{REF}(Sk.Male046) \wedge \\ &\quad \mathrm{Poodle}(Sk.Dog012) \wedge \\ &\quad \mathrm{Wash}(Sk.Male046, Sk.Dog012, Sk.Wash.Event023) \wedge \\ &\quad \mathrm{OCCUR}(Sk.Wash.Event023, Sk.Wash.Interval023) \wedge \\ &\quad (Sk.Wash.Interval023 < NOW) \wedge \\ &\quad \mathcal{D}(Sk.Go.Interval004 < Sk.Wash.Interval023)\end{aligned} \tag{7.34}$$

The segment of this translation that provides the default temporal ordering is $\mathcal{D}(Sk.Go.Interval004 < Sk.Wash.Interval023)$. Given that there is only one main event described by an utterance, the addition of this default to each successive utterance representation in a particular conversational strand is straightforward.

### Autoepistemic Reasoning and Knowledge Completion

Crystal is capable of handling restricted autoepistemic reasoning. For example, (141) could be rephrased (using contraposition) as:

If I do not believe John is my brother, then he is not my brother. (151)

This can be expressed as:

$$\mathcal{D}\neg\mathrm{Brother}(\mathit{john}) \tag{7.35}$$

Similarly, certain aspects of circumscriptive reasoning can be implemented. (7.36) expresses the fact that everything true of a predicate is known.

$$\mathrm{P}(!x) \supset \mathcal{B}\mathrm{P}(!x) \tag{7.36}$$

Now, by contraposition:

$$\mathcal{D}\neg\mathrm{P}(!x) \tag{7.37}$$

This is similar to the predicate completion aspects of circumscription.

However, there are some things that cannot easily be expressed using the simple default mechanism with skölemisation. It is not possible to express what is not believed. Consider the following from Moore (1988):

$$\exists !x.\mathrm{P}(!x) \wedge \neg\mathcal{B}\mathrm{P}(!x) \tag{7.38}$$

This states that there is at least one object which satisfies $P$ which is not currently known.

After normalisation, this is transformed into:

$$\mathrm{P}(\mathit{skolem002}) \wedge \neg\mathcal{B}\mathrm{P}(\mathit{skolem002}) \tag{7.39}$$

In this case, skölemisation leads to strange results. Applying the and-elimination meta-rules to (7.39) yields two assumptions: the first states that $P$ is true of something, the second that the former fact is not known. This contradiction is due to naming problems when quantifying into the $\mathcal{B}$ operator. It would be better to phrase the statement in terms of the equivalent schema:

$$\neg\mathcal{D}\neg\mathrm{P}(!x) \tag{7.40}$$

This sort of statement is not handled in a totally satisfactory manner by Crystal, which only interprets the $\mathcal{D}$ operator properly if it appears at the outermost textual level of a logical form. All Crystal could realise would be that it is not consistent to assert:

$$\mathcal{D}\neg\mathrm{P}(!x) \tag{7.41}$$

or any of the assumptions it subsumes.

It is possible to enhance Crystal to cope with more complex autoepistemic statements by augmenting the interface to handle embedded operators. There would be no need to change the RMS or inference engine. This modification was not found necessary for the examples processed by Crystal.

### The Frame and Qualification Problems

A default frame axiom can be expressed as:

$$\mathrm{Holds}(!P, !t_1) \wedge (!t_2 > !t_1) \wedge \mathcal{D}\mathrm{Holds}(!P, !t_2)$$

However, in practice this is not very helpful because it is too general. As a result, the system becomes rapidly clogged with information that is still assumed to hold. Furthermore, it is not the case that everything persists indefinitely (McDermott, 1982b; Shoham, 1988). As Shoham points out, whereas it is reasonable to assume that boulders tend to remain where they are indefinitely, it is less clear that it reasonable to assume that a hand-dryer in lavatory will blow hot air indefinitely.

Thus, Crystal only applies persistence to more specific cases, for example:

$$\begin{aligned}&\mathrm{Boulder}(!b) \supset \mathrm{HOLDS}(\mathrm{At}(!b, !p), !t_1) \wedge \\ &\quad (!t_1 < !t_2) \supset \mathrm{HOLDS}(\mathrm{At}(!b, !p), !t_2)\end{aligned} \tag{7.42}$$

Many aspects of the qualification problem can be solved by statements of the form:

$$C \supset \mathcal{D}P$$

where C is a conjunction of assumptions, and $P$ is the formula to be assumed.

The particular advantage of using a Doyle-style RMS to express these formulæ is that as new evidence comes in, it is not necessary to recalculate the entire theory from scratch. Many of the other non-monotonic formalisms expressed in the literature do not tackle the problem of handling incremental updates efficiently.

### Resolving Inconsistencies

The most common cause of inconsistencies in Crystal is the adoption of incorrect hypotheses. This is frequent enough for efficiency considerations to be of some importance. Provan (1988) and Elkan (1990) have shown the dependency-directed backtracking process (and even the general reason-maintenance algorithm) to be NP-hard. The only documented situations where the computations become practically intractable have been for certain computer vision applications. Crystal's RMS system performed quickly during incremental updates, partially due to the restriction to simple defaults.

The backtracking aspects of the RMS are extra-logical. Elkan shows that this procedure is effectively a search for what he calls *grounded models*, sets of un-negated non-modal atoms unique to a single expansion of an autoepistemic theory. He goes on to show that some algorithms can generate spurious grounded models or even exclude legitimate grounded models. Informally, this is because some backtracking algorithms will modify the premises of an autoepistemic theory when a nogood is asserted even though they could have switched to an alternative extension which did not include it. This, he claims, is against the spirit of human reasoning. It is not clear to me that humans would necessarily switch to an alternative extension; there is very little psychological evidence to support either view.

## 7.6.4 Autoepistemic Logic, Context and Relevance

I have now shown how simple defaults can be used in Crystal to model assumption schemas and erasures according to RT, and also how they handle some general problems of commonsense reasoning. I will now go on to describe a novel use for indexical belief operators as a semantic directive on pragmatic processing.

The $\mathcal{D}$ operator can be thought of in a procedural manner: $\mathcal{D}P$ means "add $P$ if its negation is not present". As well as default-style reasoning, such an operator can be used as part of the semantics of an utterance by regarding it as an autoepistemic operator on the cognitive environment.

### Relevance and Context

Most of the linguistic literature about Relevance assumes that the ideas in SW's book are totally consistent and implementable. Papers about RT are usually concerned with the way utterances manipulate the context.

Researchers such as Blakemore (1987) claim that the forms delivered by the input processes may contain semantic directives concerning how to go about pragmatic processing. For example, she interprets "but" as "$\wedge$" plus directives on how to expand the context as follows:

> ... the hearer is instructed to process the proposition *but* introduces in a context in which she can derive a proposition logically inconsistent with one assumed to have been derived from the proposition expressed by the utterance of the first clause (p. 130).

Consider, for example:

Angus is from Edinburgh but he doesn't have a strong accent. (152)

Presumably, the speaker believes that people from Edinburgh usually have a pronounced accent, but Angus is the exception. If she had rather said:

Angus is from Edinburgh. He doesn't have a strong accent. (153)

this would be more ambiguous. It is not clear whether the second sentence of (153) is an explanation of the first, a consequence of the first, or Relevant in a more indirect manner.

The effect of the "but" in (152) is, at least, to restrict how the hearer must expand the context. It must be expanded until the hearer believes the speaker believes that the hearer will believe that Angus should have a strong accent[19].

A broad class of "but" cases can be handled in Crystal by processing the first conjunct and then processing the second conjunct. The context is expanded until some default inference of type "context" (i.e. from the first clause) is overridden.

### "$\mathcal{D}$" as an Operator on the Context

Crouch (1990) suggests that part of the semantics of certain temporal connectives might best be thought of in a procedural manner. Some of these ideas have been formalised in a RT framework in Crouch and Poznański (1990) using the above ideas of "semantic directives".

Consider the use of "before" in the following two utterances:

Shylock dropped his wallet before he left the ghetto. (154)

The car stopped before it hit the tree. (155)

Under normal circumstances, a purely factual reading is obtained with (154) and a counterfactual reading obtained with (155); Shylock seems to leave the ghetto, but the car does not appear to hit the tree.

It is tempting to provide a semantic analysis of "before" as follows:

$$\begin{aligned}&(A \text{ before } B)(t_1) \equiv \\ &\quad A(t_1) \wedge \forall t_2.\, B(t_2) \supset (t_1 < t_2)\end{aligned} \tag{7.43}$$

In other words, for "$A$ before $B$" to be true at $t_1$, it must be the case that $A$ is true at $t_1$ and, if $B$ is true at time $t_2$, then $t_2$ be after $t_1$. For example, (155) might be translated as:

$$\begin{aligned}&\mathrm{Car}(Sk.Car023) \wedge \mathrm{Stop}(Sk.Car023, Sk.Stop.Event021) \wedge \\ &\quad \mathrm{OCCUR}(Sk.Stop.Event021, Sk.Stop.Interval037) \wedge \\ &\quad (Sk.Stop.Interval < NOW) \wedge \\ &\quad [\forall !e, !t.\, \mathrm{Hit}(Sk.Car023, Sk.Tree077, !e) \wedge \mathrm{Tree}(Sk.Tree077) \wedge \\ &\qquad \mathrm{OCCUR}(!e, !t) \supset (Sk.Stop.Interval < !t)]\end{aligned} \tag{7.44}$$

Which means that the car stopped, and if it hit the tree, this was after it stopped.

However, much as the above semantics covers both the factual and counterfactual readings, it is insufficient because:

[19] Blakemore does not actually use the notions of speakers and hearers beliefs, but this must be the case for her interpretation to be meaningful. All a "but" can tell a hearer is that the speaker assumes that the hearer will make a particular inference. In (152), it is not necessary for the hearer to believe that people from Edinburgh have strong accents, only that the speaker believes that the hearer believes this. Presumably, one function of "but" is to make some of the speaker's implicit beliefs mutually manifest.

1. People tend to assume either a factual or a counterfactual reading of such sentences; the above semantics seems to suggest that it is possible to remain neutral (e.g. it is not known whether the car hit the tree or not).

2. Sentences like (155) tend to support conditionals like "if the car had not stopped, it would have hit the tree".

The solution to the above problems is the addition of an autoepistemic directive concerning processing, which directs the deductive device to assume, given "$A$ before $B$", that $B$ actually occurs (or holds). This is exactly the function of the operator $\mathcal{D}$. The revised semantics is therefore:

$$\begin{aligned}&(A \text{ before } B)(t_1) \equiv \\ &\quad A(t_1) \wedge \forall t_2 .\, B(t_2) \supset (t_1 < t_2) \wedge \\ &\quad \mathcal{D}\exists t_3 .\, B(t_3)\end{aligned} \tag{7.45}$$

Thus, (155) is translated as:

$$\begin{aligned}&\text{Car}(Sk.Car023) \wedge \text{Stop}(Sk.Car023, Sk.Stop.Event021) \wedge \\ &\quad \text{OCCUR}(Sk.Stop.Event021, Sk.Stop.Interval037) \wedge \\ &\quad (Sk.Stop.Interval < NOW) \wedge \\ &\quad [\forall !e, !t.\, \text{Hit}(Sk.Car023, Sk.Tree077, !e) \wedge \text{Tree}(Sk.Tree077) \wedge \\ &\qquad \text{OCCUR}(!e, !t) \supset (Sk.Stop.Interval < !t)] \wedge \\ &\quad \mathcal{D}(\text{Hit}(Sk.Car023, Sk.Tree077, Sk.Hit.Event044) \wedge \\ &\quad \text{OCCUR}(Sk.Hit.Event044, Sk.Hit.Interval051))\end{aligned} \tag{7.46}$$

Let us consider how this will be processed. Amongst the encyclopædic information loaded will be information about stopping and hitting, e.g.

$$\begin{aligned}&\text{Stop}(!s, !e) \wedge \text{OCCUR}(!e, !t) \supset \\ &\quad \text{Moving}(!s, \text{Sk.Moving.Interval.Of}(!s, !e, !t)) \wedge \\ &\quad \neg\text{Moving}(!s, \text{Sk.Not.Moving.Interval.Of}(!s, !e, !t)) \wedge \\ &\quad \text{MEETS}(\text{Sk.Moving.Interval.Of}(!s, !e, !t), !t) \wedge \\ &\quad \text{MEETS}(!t, \text{Sk.Not.Moving.Interval.Of}(!s, !e, !t))\end{aligned} \tag{7.47}$$

$$\begin{aligned}&\text{HOLDS}(\neg\text{Moving}(!o), !t) \wedge \text{MEETS}(!t, !t_1) \supset \\ &\quad \mathcal{D}\text{HOLDS}(\neg\text{Moving}(!o), !t_1)\end{aligned} \tag{7.48}$$

$$\begin{aligned}&\text{Hit}(!h, !o, !e) \wedge \text{OCCUR}(!e, !t) \supset \\ &\quad \text{HOLDS}(\text{Moving}(!h), \text{Sk.Moving.Interval.Of}(!h, !o, !e))\end{aligned} \tag{7.49}$$

Since the inference controller processes the utterance from left to right, the default inference about the car still being stationary will precede the assumption that the car hit the tree. Therefore, it will be inconsistent to believe the car hit the tree because the car is stationary. This immediately goes "out" and its negation is brought "in". Thus, the counterfactual reading is obtained.

Partee (1984), amongst others, has observed that the syntactic positioning of constituents seems to effect the truth-conditions of an utterance. Some aspects of this can be explained elegantly using RT.

Consider the following:

Before the car hit the tree, it stopped. (156)

Much as semantics above seem to suggest that this should have an identical interpretation, the factual reading is far more prominent. This can be explained by considering effort consumed by the reason maintenance process.

Utterance (156) is processed from left to right. As a consequence, the defeasible inference about the car hitting the tree is asserted first, and the persistence default asserted later. Crystal's heuristics for culprit selection choose the candidate which should be least "epistemically entrenched". As explained in section 7.5.2, these heuristics are biased towards the rejection of the most recently asserted candidate because this should require the least revision effort. The most recent candidate for utterance (155) was the fact that it hit the tree. However, for (156) it is the persistence axiom. Thus, rather than assume that the car does not hit the tree, it assumes that the car must have started again[20].

Note that sentences such as (157) generally have factual readings no matter what the constituent ordering.

The car stopped before the cyclone began. (157)

This is because there is no reason to believe that the car stopping can affect a cyclone.

#### Extending the Contextual Operations

There is some evidence that other temporal connectives, such as "when", can also be treated in this way (Crouch and Poznański, 1990).

The following are possible extensions to the current system that would enable it to handle more sophisticated directives:

- A facility could be added that allows the context to be expanded until a certain event occurs (e.g. a default is overridden) or Relevance is too low. This would be useful for interpreting words like "but" (and, in chapter 8, it will be seen that it could be used to establish definite references).
- Improved access could be provided to the interpretation queue, so that more control could be exercised over previous contexts. This would be useful for interpreting words like "moreover" and "also" (Blakemore, 1987).
- A more flexible system would allow context control directives to be directly asserted. The interface would detect these and pass them on to a special context control mechanism. Thus, the first stage of processing "$A$ before $B$" might be a representation "BEFORE($A$, $B$)" which has special interpretation rules associated with it.

## 7.7 Summary

There are many problems that plague common-sense reasoning systems. Some of them are general problems concerning non-monotonicity. Non-monotonic logics have been developed to tackle these problems. Having reviewed these logics, I argued that autoepistemic logic is the most viable for Crystal.

[20] I have omitted one of Crystal's defaults that assumes $A$ meets $B$ if $A$ is before $B$ for perspicuity; it does not affect the above arguments.

A subset of autoepistemic logic using simple defaults, expressed using the modal operator $\mathcal{D}$, has been implemented in Crystal. Extra-logical properties of the system determine which extension is ultimately selected, when there is a choice. These properties are the order of inferencing, the order of assumptions within the encyclopædic entries, and the way new information is appended to the deductor memory.

The logic is supported by a Doyle-style RMS which efficiently propagates the consequences of incremental change around the system. Additionally, when an inconsistency is found, it can perform dependency-directed backtracking to locate the possible causes and select a culprit. The extra-logical properties of assumptions, such as the order in which they were inferred, play an important role in culprit selection; it is reasonable to reject the least "epistemically entrenched" putative culprit, and this is likely to be the temporally most recently applied default.

I have shown how simple defaults can handle many of the general common-sense reasoning problems and additionally provide an implementation of SW's defaults. The RMS can be used to detect the overriding of defaults, which is deemed to be a contextual effect. Furthermore, it can provide an estimate of the effort required during belief revision, which can explain why certain utterances, while identical in meaning to others, seem to be less appropriate.

The three contextual effects, implication, strengthening and erasure, can also be viewed in a single framework in terms of overall change in belief.

Finally, I have shown how simple defaults, when considered in a procedural manner, can be used to implement some of Blakemore's semantic directives. Such ideas can also be applied to provide a novel interpretation of the word "before".

In the next chapter, I will show how simple defaults can also be used as an interface between a separate non-demonstrative inference engine and the deductive device.

# Chapter 8

# Hypothesis Formation

## 8.1 Introduction

As indicated in chapters 2 and 3, the non-demonstrative inference process is a critical component of Crystal. The inferences generated by this process often provide vital links with the context that cannot be proved deductively. However, deduction still plays a role in non-demonstrative inference. The deductive device draws the non-trivial consequences from the hypothesis and the context in order to check that their combination is logically consistent, and to measure the resulting information change.

In this chapter, I provide a detailed description of non-demonstrative inferencing in Crystal. First, I identify the need for hypotheses in Crystal, and enumerate the constraints that the non-demonstrative inference process must satisfy in order to be integrated into the system. Next, I review some systems that incorporate an abductive inferencing capability, contrasting them with an ideal Relevance-based system. Using a modified version of a system introduced in this section, I illustrate how abductive equality can be used to generate many of the hypotheses necessary for Crystal to operate effectively. I then explain how the abductive equality process can be implemented and interfaced to the deductive device.

### 8.1.1 Types of Non-Demonstrative Inference

For convenience, I will divide SW's implicated premises as described in chapter 3 into two categories:

**Contextual premises:** These are assumptions that are not derived using a deductive rule; they have been added by some process external to the deductive device. In Crystal, there are two external sources: the context expansion process and the abductive component of the unifier.

**Defeasible Instantiations:** These are the result of instantiating free variables in assumption schemas; if these instantiated assumptions were derived with the use of the $\mathcal{D}$ operator, they may be retracted at a later stage. It can be argued that even these retractable inferences are deductive, since the logics used to formulate them are based on this notion[1]. They are the only means that SW have explicitly mentioned for making hypotheses.

[1] A weaker claim is that they are not non-demonstrative since a proof for any such implication can be supplied.

Chapter 7 dealt with defeasible instantiations. This chapter will deal with abductive contextual premises. In particular, it is explained how the question/answer phase is incorporated into the deductive device using the notion of *abductive unification.*

### 8.1.2 The Need for Contextual Premises

Contextual premises in Crystal can serve two functions:

1. They can establish the connection of an utterance with the context when they do not follow deductively.

2. They can reduce the number of distinct objects being considered by the system.

These will be discussed in turn.

#### Establishment of Links with the Context

SW claim that non-demonstrative inference forms the basis for utterance understanding. In chapter 3, it was explained how part of the process of utterance understanding is a question/answer phase that attempts to explain the utterance. In order to answer these questions, it may be necessary to form certain hypotheses. The deductive device may later establish that these hypotheses are inconsistent with the rest of the contextual information.

SW regard the enrichment of logical forms and the resolution of definite references as separate processes. However, it will be seen that the mechanism employed by Crystal for performing enrichment is identical to that used to establish candidate referents. This is also true of many of the related systems described in section 8.2.

Non-demonstrative inference is often necessary to establish *how* the utterance is Relevant during the question/answer phase. Much of the connection of an utterance with the preceding context may be through contextual premises. Consider:

Edwina was a shark. (158)

She went hunting for small fish. (159)

Without any contextual premises, (159) will not connect with (158) at all. It is only when it is postulated that "she" is Edwina that the connection can be deduced.

Thus, when (159) is input, questions such as "who is 'she'?", "what did she hunt for?" and "why did she hunt?" should be asked. It should consequently be inferred that Edwina went hunting. Information about how and why sharks hunt will be instantiated, and from this it may be hypothesised that Edwina may be hungry and hunting fish for food. The deductive component of the inference engine will check whether the hypotheses are consistent with the context and further establish their Relevance.

As I observed in chapter 3, SW provide little insight into the mechanism of the question/answer process. All they mention is that the questions should be "relevant"[2]. It will be seen that Crystal gives this important process substance.

[2]It is not clear from SW's definitions what it means for a question to be Relevant.

#### Plurality of Objects

A problem with Crystal's representation language, common to most frame-like formalisms, is that many anonymous terms (manifested in Crystal as skölem constants) are introduced through the use of existential quantifiers in assumption schemas[3]. Many of these names may be referring to identical objects in actuality. The problem of identifying such equivalent objects is the *internal co-reference* problem.

This can be demonstrated using a simple example:

Pepys went into the inn. (160)

He asked for the roast ox. (161)

Conceptual information about "inn" in (160) might include information that inns are run by innkeepers. Information about "ask" in (161) might include the information that it includes two people in its arguments: one is the asker, the other the person being asked. These two pieces of information might appear as follows after normalisation:

$$\text{Inn}(!i) \supset \text{Person}(\text{Sk.Person.Of}(!i)) \tag{8.1}$$

$$\text{Inn}(!i) \supset \text{Owns}(\text{Sk.Person.Of}(!i), !i) \tag{8.2}$$

$$\begin{aligned}&\text{Ask}(!p1, !s, !e) \supset\\ &\quad \text{Person}(\text{Sk.Addressee.Of}(!p1, !s, !e)) \wedge\\ &\quad \text{Addressee}(\text{Sk.Addressee.Of}(!p1, !s, !e))\end{aligned} \tag{8.3}$$

Part of the translation of (161) will include:

$$\text{Ask}(\mathit{Sk.Person002}, \mathit{Ox005}, \mathit{Ask.Event004}) \tag{8.4}$$

(8.1) will be instantiated by a suitable "inn" object, say *inn023*. From the discourse, it is apparent that:

$$\text{Person.Of}(\mathit{inn023}) = \text{Sk.Addressee.Of}(!p1, !s, !e) \tag{8.5}$$

This relation cannot be deduced, it can only be guessed. Such inferences are somewhat similar in nature to reference resolution, except that the objects are all implicit in the discourse rather than being explicitly mentioned. The observation that some of the participants are identical across successive utterances also results in added coherence, thus leading to more contextual implications.

### 8.1.3 Hypotheses for Crystal

Given that a non-demonstrative inference engine is needed to supply hypotheses to the deductive device for the two purposes just described. This engine must satisfy the following criteria:

- The inference process must be distinct from the deductive device, so it does not interfere with its modularity.
- Its defeasible nature must be expressible within the extant mechanisms in the deductive device so that it need not be altered or specialised.
- The interaction of the extra-logical and deductive components must be compatible with RT.

[3]The existential quantification is usually implicit in the structured formalisms.

## 8.2 Related Work

Before discussing hypothesis generation within Crystal, I shall consider some current natural language processing that bear some relation to Crystal. At the end of this section, I contrast and compare these other computational approaches with that taken by Crystal. It will be seen that Crystal bears some resemblance to these other systems, but adopts a different perspective which places more emphasis on how successive contexts change.

Most of these systems that I survey use some notion of *abduction* in order to form non-demonstrative inferences. Abduction is exemplified by the following inference schema:

Given:

$$\begin{array}{ll} \forall x.\mathrm{P}(x) & \supset \ \mathrm{Q}(x) \\ \quad \mathrm{Q}(a) & \end{array}$$

Infer: $\mathrm{P}(a)$

This inference, being non-demonstrative, is not valid. However, we have already seen that such inferences are often necessary, so it is desirable to find some constraints on the application of such schema that maximise their reliability. For the abduction schema to be effective, the set described by the predicate $Q$ should be as large as possible and the set described by $P$ should be as small as possible (Thagard, 1978). These constraints are known as *consilience* and *simplicity* respectively. $Q$ is as general as possible while $P$ is as specific as possible.

In the following subsections, I shall review the following systems that utilise abduction or some similar scheme:

1. The TACITUS project.
2. The Frail and Wimp systems.
3. Some more tangentially related systems.

### 8.2.1 The TACITUS Project

The sentence interpretation component of the TACITUS system uses an abductive scheme to establish its logical form in the face of incomplete information (Hobbs et al., 1988) regarding an utterance. In particular, abduction is used to solve problems of what Hobbs calls *local pragmatics* (Hobbs and Martin, 1987): this includes ambiguity, reference, metonymy and compound nominals.

TACITUS uses a hybrid logic representation which requires a dedicated inference engine (Stickel, 1988). In this, WFFs can have associated weights and costs. If a WFF cannot be proved, it may be assumed for its associated cost. Additionally, if the predicate to be proven is not a premise, the cost of assuming it can be passed back to its foundations using their associated weights. For example, consider the WFF:

$$P_1{}^{w_1} \wedge P_2{}^{w_2} \supset Q$$

Given that $Q$ has cost $C$ associated with it, and that real number weights are superscripted above the appropriate WFF, the cost of assuming $P_1$ and $P_2$ is the product of the cost of $Q$ with their weights. Therefore, the cost of assuming both of them is:

$$w_1.C + w_2.C$$

Much as the cost *can* be passed to the antecedents, it is not compulsory. Generally, the antecedents will be worth inferring if $w_1 + w_2 < 1$. As a general rule for language interpretation, the least specific explanation is preferred (i.e. the sum of the weights are not less than one). Thus, the cost/weight mechanism effectively places a preference ordering on the possible extensions of a theory.

### Interpretation Mechanism

The basic mechanism is as follows.

Sentence interpretation involves obtaining a logical form, adding further information, then finally trying to prove the enhanced form.

Hobbs describes the process as follows (Hobbs et al., 1988):

Derive the logical form of the sentence,
    together with the constraints predicates impose on their arguments,
    allowing for coercions,
Merging redundancies where possible,
Making assumptions where necessary.

This process, which solves problems of local pragmatics, performs a function similar to SW's notion of enrichment.

The above algorithm operates as follows:

- As with Crystal, the process begins with a literal representation of the utterance, the "logical form". However, there are various areas of the representation that could be made more explicit, for example the relations between elements of compound nominals and the coercions that occur during metonymy. These are made explicit by postulating a relation between the objects involved, as follows.

- Compound nominals are handled using the "nn" predicate. For example, "car wheel" would be expressed as:

$$\exists c, w.\, \mathrm{car}(c) \wedge \mathrm{wheel}(w) \wedge \mathrm{nn}(c, w) \tag{8.6}$$

  "nn" expresses the implicit relation between the arguments of the nominal. It may be entailed by any one of several relations, for example "part of", "sample of" and "used for". TACITUS' knowledge base consists of assertions such as:

$$\forall x, y.\, \mathrm{part}(x, y) \supset \mathrm{nn}(x, y)$$

$$\forall x, y.\, \mathrm{used}(x, y) \supset \mathrm{nn}(x, y)$$

  which indicate some of the possible relationships between elements of compound nominals.

  During the interpretation process, attempts will be made to prove some "nn" relation when necessary. If it cannot be proved deductively, the abduction mechanism will prefer the cheapest assumed explanation (given appropriate weights). In this case "part of" is the correct relationship for the "nn" in (8.6).

- If the argument of a predicate is of the wrong type, it is assumed that it must be related to the desired type in some way. Hobbs calls this relation "rel". As with "nn", there is a set of axioms like:

  $$\forall x, e.\, \text{function}(e, x) \supset \text{rel}(e, x)$$

  which can be read as "e is related to x if it is a function of x.

  Using abduction, the particular relation can be inferred. Hence, coercion of arguments is effectively achieved.

- References are established by attempting to prove the existence of the relevant object. This may involve the use of the abduction mechanism in order to assume its existence if a suitable referent cannot be found.

- The weights are organised so that it is generally not worth assuming more specific information. However, when there is redundancy between clauses, it is sometimes cheaper to make more specific inferences. Consider the following example from Hobbs et al. (1988):

  $$P_1^{0.6} \wedge P_2^{0.6} \quad \supset Q_1 \tag{8.7}$$
  $$P_2^{0.6} \wedge P_3^{0.6} \quad \supset Q_2 \tag{8.8}$$

  Assume that we are trying to prove $Q_1 \wedge Q_2$, and that the cost of assuming each $Q$ is 10 units. It follows that assuming $Q_1 \wedge Q_2$ will cost 20 units. However, assuming $P_1 \wedge P_2 \wedge P_3$ will cost 18 units since $P_2$ in (8.7) is unifiable with $P_2$ in (8.8). Much as the weights in this case sum to more than one, the more specific reading is preferred because it can be explained by conflating certain entities.

#### Overall Operation

Enrichment of a form is effected by fleshing out compound nominals and other non-specific items and then trying to prove each conjunct of the resulting logical form. As a rule, TACITUS prefers the most general explanation. However, it is possible to infer more specific information when it is common to several rules. Hobbs justifies this by claiming that redundancy is intrinsically part of language. Indeed, the very reason for such redundancy may be to assist the abductive interpretation process, providing it with more detailed information.

The TACITUS interpretation system is currently limited to the processing of isolated sentences. Thus, whereas TACITUS can be argued to have a theory of utterance interpretation, it cannot presently be regarded as having a theory of discourse interpretation. It is clear that a major difference between the principles underlying RT and TACITUS is that RT always considers an utterance in the context created by, amongst other things, the previous utterance.

#### Abduction in TACITUS

It is illustrative to express the abduction mechanism in terms of an NML such as that of Crystal, thus making them easier to compare. The guessing of a suitable predicate for relating nominals and coercion can be explained in terms of predicate circumscription. For example, the "nn" predicate might be circumscribed to yield something like:

$$\text{nn}(x, y) \equiv \text{part}(y, x) \vee \text{sample}(y, x) \vee \ldots \text{for}(y, x)$$

The fact that only one of these need be chosen can be expressed, say, in terms of Crystal's logic as[4]:

$$\mathcal{D}\text{part}(y, x) \wedge \mathcal{D}\text{sample}(y, x) \wedge \ldots \mathcal{D}\text{for}(y, x)$$

and

$$\neg\text{part}(y, x) \wedge \neg\text{sample}(y, x) \wedge \ldots \neg\text{for}(y, x)$$

Abduction in TACITUS may be considered to be an aspect of non-monotonic reasoning. The interpretation problem then boils down to selecting between one of several conflicting extensions, in this case by using the cost mechanism. However, the weights and costs in the knowledge representation seem rather ad-hoc. Hobbs states that costs are intended to be the cost of assuming the entity, whereas the weights appear to be more like probabilities, expressing "semantic contribution". As yet, no semantics have been provided for this aspect of the knowledge representation.

The ordering of steps in a proof is significant in the TACITUS system. Different results can be obtained by starting with the representation of the verb or noun phrases. No explanation for this is given.

The interpretation mechanism itself is a little like "Relevance in reverse". Rather than forward-chaining on the utterance, an explanation is obtained through backward-chaining (Hobbs, 1988). The entire mechanism is very elegant, since it relies on a few orthogonal principles. However, because there is no theory of how successive utterances connect, TACITUS must rely on ad-hoc weights to direct the interpretation process.

### 8.2.2 Frail and Wimp

In contrast with TACITUS, the Frail and Wimp systems have less adequately motivated foundations, but are more ambitious in their coverage. These systems attempt to analyse and comprehend children's stories using scripts, frames and plans to predict future events and individuals' goals.

#### History

The Wimp systems follow a line of systems that began with Schank and Wilensky's early programs (Wilensky, 1978; Schank and Abelson, 1977). They all have in common the fact that they perform *motivation analysis.* This is an attempt to infer an individual's motivations and goals from a discourse. They are *explanation-based* systems in that they attempt to fit explanatory inferences to sequences of utterances.

Frail is a knowledge representation acted on by a separate inference engine, Wimp. The earliest versions of Wimp were based on dæmons and simple frame representations. Later on, efficient inferencing algorithms were implemented using marker propagation techniques (Charniak, 1986). However, it was discovered that for large systems, the techniques were

[4]I have not continued to use circumscription here as its cautious approach cannot be used to select between these alternatives. It would just yield their disjunction.

inefficient and lacked adequate semantics (Goldman, 1987). This led to the development of the logic-based Wimp2 and Frail3 (Charniak and Goldman, 1988). Rather than describing a particular version of these systems, I will pick out the significant features of all of them[5].

### Knowledge Representation

I shall illustrate typical aspects of Frail by reference to Frail2.1 and Frail3.

Frail2.1 (Charniak et al., 1982) uses a simple frame-based language, which can be translated into typed FOPC terms using the predicates "inst" and "sub-step". "inst" is identical to "isa" (see chapter 5). "sub-step" is a binary predicate asserting that its first argument is a step in achieving a plan, identified by its second argument.

So, for example, a sub-step of an "eating" plan might be "masticating":

$$\text{sub-step}(e{:}\,\textit{eating}, \text{masticating.step}(e))$$

This might be one of a class of "mouth movement" actions:

$$\text{inst}(\text{masticating.step}(e), \textit{mouth.movement})$$

The agent of such an action is the agent of the eating action:

$$\text{agent.of}(\text{masticating.step}(e)) == \text{agent.of}(e)$$

Frail2.1 allows certain terms to be typed. These types can be converted into non-typed FOPC expressed in terms of the "inst" predicate and material implication. Thus, a frame typically translates into a conjunction of simple statements in predicate calculus.

The semantics of "==" are not those of "=". "$A == B$" means "$B$ is a better name for $A$". It has the procedural effect that $B$ is substituted for $A$ in all logical forms that contain $A$, which is a valid action in non-modal contexts. However, no semantics of actions, events or time are provided in any versions of Frail.

Frail3 (Charniak and Goldman, 1988) is logic-based and adds the following additional features[6]:

- Both forward and backward chaining are made explicit in the knowledge representation.
- Forward-chaining parsing rules are introduced that provide semantic analyses using cases.
- Exclusive disjunction is allowed using the operator "$\vee$". Probabilities may be used to guide the search for an appropriate disjunct.
- For.Some($x$, $y$) finds all derivable variable bindings for the formula $x$ and asserts $(y_1 \vee y_2 \vee \ldots y_n)$ where each $y_i$ is the result of substituting a distinct variable binding of $x$ in $y$.
- **exists** and **pexists** assert the existence of an object and the existence of an object that has already been mentioned, respectively. **pexists** backward-chains to prove the necessary fact.

[5]Where it makes sense, I will also use "Frail" and "Wimp" to refer to the generic systems.

[6]I have altered the notation for clarity.

- Role.Inst($x$, *slot*, *frame*) is a predicate that is true if "$x$" can fill "*slot*" in "*frame*".

Charniak expresses a rule of explanation/reference as follows:

$$\begin{array}{l} \text{inst}(x, \mathit{frame}) \wedge (\mathit{frame} \neq \textbf{garbage}) \supset \\ \quad [\textbf{pexists}\ y{:}\mathit{frame}.\ (x == y)]_{\mathbf{0.9}} \\ \qquad\qquad \vee \\ \quad [\text{For.Some}(\text{Role.Inst}(x, \mathit{slot}, \mathit{superframe}), \\ \qquad\qquad \textbf{exists}\ s{:}\mathit{superframe}.\ (\mathit{slot}(s) == x))]_{\mathbf{0.1}} \end{array}$$

Informally, this expresses the fact that, if $x$ is a meaningful frame, then it is either identical to a previous frame (which has already been explained) or can be explained by filling the slots in other suitable frames. The probabilities subscripted to the disjuncts are intended to indicate the fact that reference to a previous frame is preferred.

### The Inference Engine

As far as Wimp is concerned, an utterance has been understood if it can be explained. An explanation is effectively matching the semantic representation to some actions or events within a form of script.

Wimp has been drastically altered several times. Earlier versions used simple marker passing schemes, making decisions based on the intersection of these markers. These intersections could amount to deductive or non-deductive inferences.

In later systems, abduction is affected through non-monotonic equality. This allows the unifier to make guesses of equality. Charniak (1988) adapts the standard unifier in the following way:

- Unification is performed in the usual manner, but when two constants[7], $X$ and $Y$, do not unify, the following step is taken.
- It is checked to see if it is consistent to believe $X == Y$. If this is the case, then $X == Y$ is asserted, and they are allowed to unify.
- Consistency checking is performed by backward-chaining. In order to prove that $P$ is consistent:

  1. If $P$ is not of the form $A == B$ and has no free variables, then attempt to prove $\neg P$. If this fails, $P$ is consistent.
  2. If $P$ has free variables, then for every suitable instantiation $P_n$, prove that $P_n$ is consistent.
  3. If $P$ is of the form $A == B$ then for very predicate that has $A$ as an argument, either prove it with $B$ substituted for $A$, or at least prove that it is consistent.

With Frail2.1, if a bad decision is made about equality, it almost impossible to retract this decision.

Presumably, the unification algorithm is abductive because we desire:

Can.Unify($p(A), p(B)$)

---

[7]Constant functions are regarded as constants.

and it is true that:

$$(A == B) \supset \text{Can.Unify}(p(A), p(B))$$

so we assume:

$$(A == B)$$

if this is consistent.

Earlier versions of Wimp over-committed themselves to a particular course of actions. The first consistent one was selected, without even considering the others. Wimp2 (Goldman, 1987; Charniak and Goldman, 1989b) avoids this by allowing several alternatives to be entertained simultaneously. Probabilities are used to select the most appropriate viewpoint. In order to hold parallel conflicting viewpoints, Wimp2 uses the ATMS (deKleer, 1986a). Every disjunction is implemented as a separate ATMS extension. The probability mechanism, similar to Pearl's Bayesian inference networks (Pearl, 1988), is used to select the most appropriate extension. The disambiguation/reference rule is intended to prefer existing extensions rather than new ones.

### Probabilities for Extension Selection

When using the ATMS, the extension with the most likely event will generally be the one selected. However, extensions that offer explanations in terms of a single event are preferred over explanations using several events, no matter what the probabilities involved.

Charniak and Goldman (1989a) distinguish "life" from "stories", claiming that many inferences are possible in stories that are not possible in life. In life, the likelihood of two objects being equal or a particular object taking part in a particular event is almost zero. In stories, this likelihood is much higher because fewer objects are being considered by the audience. In order to bias the probabilities, the "fundamental constant", $E$, is introduced, biasing the probability that an object satisfies a particular predicate. The higher the value of $E$, the more "story-like" the behaviour of the objects. Charniak calls this the *knob theory* because changing the value of $E$ is like turning a knob to alter the probabilities.

Charniak notes that even the knob theory does not account for why people make certain inferences in language. The very *mention* of an object that is compatible with another object within a plan makes it more likely that the two objects are identical. Thus, the knob theory can be modified so that only certain objects will be biased through mention. Charniak calls this the *mention theory.*

### Discussion

Frail and Wimp have much in common with RT and Crystal. Wimp is essentially a forward-chaining inference engine which hypothesises abductive explanations of actions in utterances. Where it differs from RT is that Relevance attempts to connect an utterance with the context rather than explain the actions. Furthermore, given alternative possibilities, the one that will be selected will be that which is the most Relevant rather than the most "probable". Frail and Wimp are better suited to motivation analysis than as a model of human utterance understanding. The knob and mention theories are attempts to explain why the discourse itself leads to certain inferences. However, Goldman and Charniak are not totally satisfied with their rather ad-hoc applications of probability theory to a logical system. This has recently led them to develop a purely probabilistic version of Wimp, Wimp3, solely based on Pearl's conditional probabilistic networks (Goldman and Charniak, 1990) which can be incrementally updated in the light of new evidence.

### 8.2.3 Other Related Systems

The system described in Pereira and Pollack (1988) uses the notion of *conditional interpretations*: these consist of the "sense" or basic logical form of the utterance, and "assumptions" which place constraints on what must be true of the sense in the context. These include binding and sortal restrictions. The assumptions can be removed using *pragmatic discharge rules* which take the interpretation and context and yield a new interpretation. So, for example, a definite reference may be discharged to yield a new logical form with the appropriate referent substituted. The conditional interpretation can be seen as imposing constraints on the possible models of the sense of the utterance. Thus, a class of suitable models are described by the interpretation, in much the same way as a default theory may lead to multiple extensions. The system differs from Crystal, Frail and TACITUS in that it does not attempt to use common sense knowledge to enrich, or commit to any particular extension (as this is not its task).

The FAUSTUS system (Norvig, 1987) is another marker propagation system. It uses a knowledge representation based on KL-ONE. The inference system places potential inferences, detected by the collision of markers, onto an agenda (see Charniak et al. (1987) for an explanation of agendas). The pattern of inferences leading to the collision determine the inference type. Some of these types are purely deductive, others are not (for example "concretion" is a type of non-deductive enrichment). Potential inferences on the agenda can either be endorsed, ignored or deferred. Deferral occurs because there are several conflicting possibilities that cannot be distinguished on the basis of current information. Inferences are ignored if their negation is already present. Otherwise, inferences are accepted. This is a kind of non-monotonic inference. No semantics are provided for the system, but it is possible that deferral can be regarded as the consideration of multiple extensions (or at least allowing non-determinacy within an extension).

Kautz and Allen (1986) presents a principled approach to recognising agents' plans. Allen's temporal logic is used to define a database of actions and plans. A hierarchy of plans is constructed using a version of Charniak's "inst" predicate. Plans can be broken into component actions using a "sub-step" relation. A version of circumscription is used to close the hierarchy[8]. The only plans and actions that can be carried out are the ones that are mentioned. Thus, it is possible to explain any action by assuming it is part of one of the larger plans described in the database. Since circumscription is cautious, it is often the case that an explanation of an action is a disjunction of alternative possibilities; if an action forms part of several plans, its explanation is that it could be part of any one of them. So for example, if it is known that a tomato sauce is being made, it is assumed that it is part of any one of the actions that involve preparing a meal with tomato sauce.

This cautious approach has the advantage that it does not suffer from *over-commitment*. The first consistent view to be found is not necessarily adopted. However, this can be a disadvantage; as information is added to the system, it is likely that more and more disjunctions will result, without any particular view being adopted. The system avoids the multiple extension by allowing non-determinacy, but this is costly.

[8] Circumscription of the database is carried out by hand before its contents are used.

### 8.2.4 Comparing the Systems

Most of the systems considered here can be viewed in terms of the multiple extension problem. For systems such as Wimp2, this is explicit. For systems such as TACITUS, it is implicit.

As a way of comparing the system, it is useful to consider the problem of interpreting an utterance in two separate parts:

1. Determining the possible extensions through some suitable non-monotonic logic.

2. Selecting a particular extension.

This process can be viewed as abduction for the following reasons. A certain set of assumptions, $C$, will make the utterance $U$, true. ie:

$$C \models U$$

and by the deduction theorem:

$$\vdash C \supset U$$

Often, $C$ cannot be proved deductively. There are many possible $C$'s which could satisfy the above equation. Utterance interpretation involves a search for a suitable extension $C$. Assuming that $C$ is true is abduction. As shown for TACITUS, it can also be described in terms of NML.

TACITUS attempts to use heuristics to guide a search for a minimal $C$ that implies the utterance. The Wimp systems attempt to find some $C$ (not necessarily minimal), that explains $U$. In my analysis, RT attempts to find a $C$ which implies $U$, but also results in the most information yield in terms of connectivity and non-trivial consequences for the effort in obtaining $C$.

RT is the only one of the approaches described that attempts to explain *why* people choose a particular extension. The others rely on some statistical notion of what people are likely to do, without justifying this.

RT also allows for the possibility that there may not be a specific $C$ that supports the utterance (see chapter 2); the choice is non-deterministic. There are two ways in which the context can be made more determinate:

1. $C$ is a partial description of the world; more details can be added about entities in the domain. In RT, this occurs when further assumptions are brought into the deductor memory.

2. Given two incompatible ways of viewing the world from the available facts, one of them can be preferred. In RT, this occurs when erasures are carried out; in Crystal this manifests itself either as the overriding of defaults or as dependency-directed backtracking.

Situation semantics also adopts this two-dimensional viewpoint (Barwise and Perry, 1983; Fenstad et al., 1987). It is possible to have partial extensions and competing extensions.

It is impractical and psychologically implausible to be totally non-committal; Wimp2's storage of all possible extensions can be very time consuming and inefficient as the number of utterances build up. Kautz and Allen's maintenance of a single non-determinate extension becomes increasingly more difficult to reason with.

Having reviewed the most prominent systems, it is now possible to choose the most suitable means of generating abductive inferences in Crystal. Since Crystal does not need to completely explain an utterance representation, it seems inappropriate to apply the abductive explanation mechanism of TACITUS. The various non-committal approaches are also not suitable, since according to RT, one commits to the extension in which the utterance is deemed most Relevant. This leaves us with Charniak's abductive equality mechanism. As illustrated above, this mechanism is used to match a logical form to its possible explanations. However, this mechanism could just as easily be used to provide non-demonstrative inferences that connect an utterance representation with the context, and guesses that certain objects are identical. In the forthcoming section, I show how abductive equality, when suitably co-ordinated with the deductive device, can be used to develop a non-demonstrative interpretation system that is as elegant as that of TACITUS, but works with discourse segments rather than unitary utterances.

## 8.3 Providing Hypotheses for Crystal using Abductive Equality

SW postulate what I have called the question/answer phase in utterance interpretation, where the hearer attempts to guess how the utterance representation fits into the context by means of suitable questions. Questions are backward-chaining queries. However, the deductive device is essentially a forward-chaining inference engine. What questions should be asked, and how are the forwards- and backwards- chaining processes to be integrated?

I will answer these questions by considering the points at which the deductive device currently queries its memory. I will then show that, if these queries are made abductive, it is equivalent to implementing a version of the question/answer phase.

There are three points in the deductive process during which the deductor memory is queried in a very simple backwards-chaining manner:

1. When scanning for assumptions that match the triggers of the inference rules.
2. When passing new information to the interface. The deductor memory is queried to check if it is subsumed by, equivalent to, or subsumes an already-present assumption.
3. When performing consistency checks for the $\mathcal{D}$ operator.

These are the ideal points at which to integrate some kind of non-demonstrative inference engine. It can also be noted that the three actions listed above all use the unification process. If the call to the non-demonstrative inference process was placed in the unifier it could be called at all these points (if so desired).

Let us consider which non-demonstrative inferences might usefully be made at these points.

Checking for antecedents: When considering two antecedents to a synthetic rule, it might be worth assuming that they can be made to unify in order for certain inferences to go through.

During subsumption checks: Assuming that one form is equal to, or subsumes another reduces the number of entities in the deductive device while providing more contextual connectivity, thus allowing more information to be held in it. This will certainly be the case if the forms under consideration can be unified.

During Consistency Checks: When checking for the absence of information during consistency checks associated with the $\mathcal{D}$ operator, there is no advantage in assuming that the object being searched for is present since this generally results in the blocking of some default inference, but an object of RT is to maximise information yield.

Thus, when not considering consistency, it can be useful to make non-demonstrative inferences concerning the fact that two assumptions can unify. This can be regarded as either assuming that different names refer to the same object or as making the simplest assumptions that would allow one assumption to explain another. Every time a new assumption is added to the deductor memory, an attempt is made to explain it or match it with existing assumptions. This means that an attempt will be made to question every non-trivial implication, which is a possible interpretation fo the question/answer phase.

The above views can be seen as an application of the the principle of Occam's Razor, which states that the simplest solution to a problem should be taken. The understanding systems described earlier in this chapter have all applied this principle in some way. Hobbs argues that the redundancy in natural language may be deliberately present to aid the human abduction process, which generally prefers the least specific (i.e. simplest) interpretation. Charniak attempts to match logical forms to corresponding frames, assuming that a simple fit is desirable between the objects explicitly and implicitly mentioned. Crystal adopts both approaches by assuming that the number of objects in its deductor memory should be as small as possible[9]. This application of Occam's Razor to the number of objects can be used in the follow ways:

1. It can be used to obtain the effects described by SW as enrichment.
2. It can perform reference resolution.
3. It solves the internal co-reference problem.
4. It can be used to explain certain events and predict future ones.

Each of these will be considered in turn.

### 8.3.1 Enrichment and Explicatures

#### Enrichment

Enrichment is the process by which an ambiguous, vague logical form is reified. The form is vague because the predicates are underspecified and references have not been resolved.

Reconsider the following from section 8.1.2:

Pepys went into the inn. (160)

He asked for the roast ox. (161)

This might result in an inference like the following:

$$\begin{aligned}&\text{Ask}(\textit{pepys}, \textit{ox}, \textit{Ask.Event}004) \supset \\ &\quad \text{Addressee}(\text{Sk.Addressee.Of}(\textit{pepys}, \textit{ox}, \textit{Ask.Event}004)) \wedge \\ &\quad \text{Person}(\text{Sk.Addressee.Of}(\textit{pepys}, \textit{ox}, \textit{Ask.Event}004))\end{aligned} \tag{8.9}$$

which can be regarded as one of SW's assumption schemas:

[9]The only exception to this rule are certain indefinite references.

$$\text{Pepys asked the person } \boxed{\phantom{xxx}} \text{ for the roast ox.} \tag{162}$$

This is considered to be an explicature that is enriching the utterance. Applying modus ponens to (161) and (8.9) followed by and elimination will yield:

$$\text{Addressee(Sk.Addressee.Of}(pepys, ox, Ask.Event\text{004})) \tag{8.10}$$

$$\text{Person(Sk.Addressee.Of}(pepys, ox, Ask.Event\text{004})) \tag{8.11}$$

Now, inferencing from the frame for "inns" will include the information that:

$$\text{Person(Sk.Person.Of}(inn\text{023})) \tag{8.12}$$

i.e. the innkeeper is a person.

When (8.11) is added to the interface, (8.12) is already present in the deductor memory. The interface then scans the deductor memory to check for subsumption or equality. If unification is used, it can be noted that (8.11) will unify with (8.12) only if:

$$\text{Person.Of}(inn023) = \text{Sk.Addressee.Of}(pepys, ox, Ask.Event\text{004})$$

If this is asserted via the interface, it is as if the above schema (162) has been instantiated to:

$$\text{Pepys asked the person, Person.Of}(inn023)\text{, for the roast ox.} \tag{163}$$

By linking the addressee to the innkeeper, the utterance is not only enriched in the sense that the two entities are interchangeable. It is also enriched because it automatically inherits all of the properties of the terms with which it co-refers. So, for example, if it was typical for innkeepers to be men, then it would also be typical for Pepys' addressee.

### Explicatures

As seen from the last example, with the introduction of an equality operator, there is no need to physically modify an utterance representation in order to enrich it.

This makes the notion of *explicature* rather difficult to define. According to SW, explicatures are contextual implications that contain some portion of the original logical form. The more of the original form that is present in the inference, the more "explicit" that inference is. Using the notion of equality, it is possible for two syntactically different forms that differ only in their constituent terms to be (semantically) equivalent. It is not clear whether such forms should be regarded as identical, and therefore equally as explicit as each other.

The notion of explicature is intended to contrast developments with inferences drawn using commonsense encyclopædic knowledge. If enrichment involves asserting equality, it is difficult to bifurcate the implications using notions of explicitness. This way of categorising implications is also criticised in Levinson (1987); he also notes implicatures and explicatures are very tightly interwoven concepts that are mutually dependent and therefore cannot be separated easily.

SW appear to believe that the enrichment process has some explicit termination point; when an utterance is Relevant it must have been "fully developed". However, there is no limit to the number of assertions of equality that can be associated with objects in the utterance: it can be enriched indefinitely. It seems unlikely that there is a single point at which the utterance is "completely" developed.

One solution to this problem is to treat the disambiguated logical form plus the assertions of equality as explicatures and everything else as implicature. The reasoning behind this is that the logical form is clearly an explicature and the assertions of equality relate it to the encyclopædic and contextual information. Using this approach, an utterance representation can be enriched as far as the capacity of the deductive device will allow.

The equality-based approach to enrichment seems more adequate than SW's view. The context consists of various assumptions, which include an individual's plans and various other stereotypical information. It is invariably the connection of the utterance with these assumptions that makes it comprehensible. Some of the connections will be implicit inferential connections (implicatures), and others will be putative matches with extant contextual information (explicatures)[10], expressed using the equality relation[11].

Both the notion of explicature and finite development are not critical to Relevance theory. Thus, as they are difficult to deal with, they can be abandoned. It will be seen that the built in equality mechanism described in section 8.4.5 automatically identifies my revised notion of developments from an assumption using substitution of equals. There is therefore no need to treat explicatures as separate entities — during enrichment, the utterance simply becomes equivalent to increasingly more forms as progressively more assertions of equality are made.

### 8.3.2 Reference Resolution

Enrichment in Crystal involves making assertions of equality of terms. The same assertions of equality can be used to resolve definite references. Definite references differ from general developments in that they are more constrained. Whereas enrichment is not a directed process, there is usually a goal associated with definite references: as long as Relevance is maintained, a context should be found that contains an entity with which the reference co-refers. As before, it is required that the co-referring terms unify; in order for this to be the case, it is often necessary for some terms to be assumed equal.

Thus, resolving a reference $\dot{R}$ can be seen as a search for a context which contains an entity $P$, such that $R = P$. A definite reference is explicitly asserted in Crystal using the "REF" and "IT" predicates. The IT predicate differs from the REF predicate in that its argument can only co-refer with terms that are explicitly mentioned in an utterance, and is therefore more suitable for describing pronominal references.

Let us consider the "REF" predicate first. This asserts that its argument is expected to co-refer with *any* compatible entity in the system. Thus "the dog is barking" would be expressed as:

$$\begin{aligned}&\mathrm{Barks}(\mathit{dog046}, \mathit{barks.event021}) \wedge \\ &\qquad \mathrm{OCCUR}(\mathit{barks.event021}, \mathit{NOW}) \wedge \mathrm{REF}(\mathit{dog046})\end{aligned}$$

REF(*dog046*) means that "dog046" is a reference that must be resolved. IT can be used similarly. For example, "it was strange" would be expressed as:

$$\mathrm{IT}(\mathit{Sk.Event023{:}Event}) \wedge \mathrm{Strange}(\mathit{Sk.Event023{:}Event}) \tag{8.13}$$

Note that when adverbial modifiers are used with "it", I assume the input systems can supply the information that the referent must be an event.

---

[10]This approach also counters some of Levinson's criticisms of RT in the above paper.

[11]There are other possibilities for connecting an utterance to the context non-demonstratively, for example using generalisations and analogical techniques. These are not considered in this thesis.

Definite references are resolved without the requirement of modifying the inference engine significantly. All that is needed is is some process or co-routine that is capable of making hypotheses concerning equality of certain objects. Processing proceeds as follows:

The context is expanded as usual, and the closure of inferences taken. If the resulting context does not contain a suitable assertion of equality, the context is expanded again, and the inference closure retaken. This is repeated until the reference is resolved or the Relevance becomes too low. In the latter case, it is assumed that there is no suitable referent.

The interface detects terms that are definite references when they are asserted using the "IT" and "REF" predicates. These terms are added to a list of definite references called a REF list[12]. This is used by the dependency-directed backtracking process (see section 8.4.6) and the main controller to check whether the context needs further expansion because some definite references have not been resolved. In addition, a list of entities that could be potentially referred to by a pronoun such as "it", the IT list, is also kept. The IT list consists all of the terms contained within every assumption of type Utterance. When an assertion of equality is made that involves a definite reference which is predicated "IT", the assertion is only allowed if the other co-referring term is on the IT list. Thus, terms predicated IT can only refer to items on the IT list, which are themselves explicitly mentioned within an utterance.

A critical assumption that is made for the reference resolution process to work effectively is that the most salient compatible entity is the correct referent. An alternative would have been to select several salient candidate referents and process each, selecting the most Relevant extension. This would be treating references in the same way as syntactic ambiguities[13]. However, it is less in accordance with the Principal of Optimal Relevance, so I have not adopted it in Crystal.

The mechanism just described can also handle some cataphoric references, such as "he" in the following:

Before he conjured up the demon, Tim lit the candles. (164)

When the the first clause is added, "he" does not co-refer with anything. However, processing continues until the entire form has been processed. When, "Tim" is added, it is inferred that he is male (since Tim is the name of a male). It is also known that the object denoting "he" is male. Thus, the unifier can guess that "he" is "Tim".

### 8.3.3 Internal Co-reference

Co-referring internal skölem constants are similar to definite references. However, it is not *necessary* that such objects should refer to others as with definite references, but it may well be desirable. As for the other cases, guesses of equality might be made during unification. If a referent for an entity cannot be found, it is assumed to be a new object.

Crystal is different from fully explanation-based systems in that it is merely desirable that it find matching entities, it is not the sole aim of the inference process. It is not necessary to explain the presence of every object or justify every action that is present in the deductor memory.

[12] It is also noted on the REF list whether the reference was a REF or an IT.

[13] Processing several alternatives in this way might be better implemented using a backtracking ATMS.

Let us consider an example that illustrates how internal co-reference might be established. The following are fragments of the encyclopædic entries for "stone" and "cherry":

$$\text{Cherry}(!c) \supset \text{Stone}(\text{Stone.Of}(!c)) \wedge \text{Has}(!c, \text{Stone.Of}(!c))$$

i.e. "Every cherry has a stone" and:

$$\text{Stone}(!s) \supset \text{Cherry}(\text{Cherry.Of}(!c)) \wedge \text{Has}(!c, !s)$$

i.e. "Every stone has a cherry".

Now, consider the phrase, "the stone of a cherry":

$$\text{Stone}(s) \wedge \text{Cherry}(c) \wedge \text{Has}(c, s) \wedge \text{REF}(s)$$

From these encyclopædic fragments and the above phrase, the following can be inferred:

$$\text{Cherry}(c) \tag{8.14}$$

$$\text{Stone}(s)$$

$$\text{Cherry}(\text{Cherry.Of}(s)) \tag{8.15}$$

$$\text{Stone}(\text{Stone.Of}(c))$$

$$\text{Has}(c, s)$$

$$\text{Has}(\text{Cherry.Of}(s), \text{Stone.Of}(c))$$

It is apparent that:

$$\text{Stone.Of}(c) = s$$

and

$$\text{Cherry.Of}(s) = c \tag{8.16}$$

The guesses of equality could be made as for the other cases of the application of Occam's razor. So, for example, assume that (8.14) is added first, and then (8.15) is passed to the interface. (8.15) is passed to the abductive equality process which notices that it would unify with (8.14) if (8.16) were the case. If it is consistent to assert (8.16), this is added to deductor memory via the interface. After this, (8.15) is merged or added as appropriate.

The cherry and stone problem also has a purely deductive solution. This is because any particular cherry has exactly one stone, i.e.

$$\begin{aligned}&\text{Cherry}(!c) \supset \text{Stone}(\text{Stone.Of}(!c)) \wedge \text{Has}(!c, \text{Stone.Of}(!c)) \wedge \\ &\quad [\text{Stone}(!s) \wedge \text{Has}(!c, !s) \supset (!s = \text{Stone.Of}(!c))]\end{aligned}$$

However, it is not generally possible to find a purely deductive solution.

### 8.3.4 Event Prediction

In order to comprehend a dialogue, it is often necessary to understand how current events are related to each other and to predict what may well happen. This is often regarded as part of *plan recognition*, the problem of establishing an individual's intentions. As explained in section 5.5.3, this is usually realised by fitting people's actions to stereotypical sequences of actions and events. The stereotypical sequences do not necessarily occur, but are merely possibilities.

Thus, using the eating script described in section 5.5.3, it is possible to process the following discourse fragment, similar to that described in chapter 1.

Tweety felt hungry. (165)

He found a nut. (166)

Part of the translation of (165) and (166) is:

$$\text{Feels}(tweety, hunger, Sk.Feel.Event077)\cdots \tag{8.17}$$

$$\begin{aligned}&\text{Male}(Sk.Male021) \wedge \text{IT}(Sk.Male021) \wedge \\ &\quad \text{Find}(Sk.Male021, Sk.Nut033, Sk.Find.Event021) \wedge \\ &\quad \text{Nut}(Sk.Nut033)\cdots\end{aligned} \tag{8.18}$$

During inferencing, schema (5.74) will be applied, yielding:

$$\text{Food}(\text{Sk.Food.Of}(tweety, Sk.Find.Interval042)) \tag{8.19}$$

$$\begin{aligned}&\text{Eat}(tweety, \text{Sk.Food.Of}(tweety, Sk.Find.Interval042), \\ &\qquad \text{Sk.Eat.Event.Of}(tweety, Sk.Find.Interval042))\end{aligned} \tag{8.20}$$

$$\begin{aligned}&\text{OCCUR}(\text{Sk.Eat.Event.Of}(tweety, Sk.Find.Interval042), \\ &\qquad \text{Sk.Eat.Interval.Of}(tweety, Sk.Find.Interval042)) \supset \\ &\quad [(\text{Sk.Eat.Interval.Of}(tweety, Sk.Find.Interval042) < !t_1) \supset \\ &\qquad \mathcal{D}\text{HOLDS}(\neg\text{Hungry}(tweety), !t_1)]\end{aligned} \tag{8.21}$$

It will also be known that:

$$\text{Food}(Sk.Nut033)$$

Now, when this is added, it will unify with (8.19) if:

$$Sk.Nut033 = \text{Sk.Food.Of}(tweety, Sk.Find.Interval042) \tag{8.22}$$

This results in the prediction that Tweety may eat the nut. Now assume "he" is guessed as being equal to Tweety. If (8.22) is assumed to hold, then it will be understood that if Tweety has the nut, he will eat it; the nut has become part of a potential eating event. Thus, predicting actions becomes part of the internal co-reference problem, and is solved using the same mechanism. In this example, it is the fact that the "food" reading of nut may be associated with a script in this way that allows it to be distinguished from the "fastener" reading which will yield less contextual implications.

## 8.4 Implementing Abductive Equality

In the previous section, I showed how abductive equality can be used as the basis for a non-demonstrative inference process for Crystal that solves the problems enumerated at the beginning of the chapter. In this section, I show how an abductive equality process can be co-ordinated with the deductive device. It is then shown how the process can be integrated with the unifier and how its guesses can be improved with the assistance of the sort hierarchy. Next, it is explained how partially-ordered preference relations can be placed on hypotheses in much the same way as they were placed on expressions of defaults (by treating their order within an encyclopædic entry as significant). The detailed algorithm for the unifier is then provided. In order that assertions of equality are interpreted both efficiently and correctly within the deductive device, it is provided with a separate *equality sub-system* that maintains a list of all equal entities within the deductor memory. The operation of this system is explained. Finally, it is explained how incorrect hypotheses are handled using the RMS and how abductive equality is utilised by the querying system.

### 8.4.1 Interfacing the Inference Processes

The interface and inference algorithms must instruct the unifier to pass control to an *abductive equality process* (AEP) when unification fails. The abduction process is handed the first two terms that do not unify.

The AEP hazards a guess as to whether any incompatible constants can be made equal. However, since it has extremely limited deductive capability and no access to the deductor memory, this guess may well be incorrect. Thus, given two constants, $a$ and $b$, the AEP asserts:

$$\mathcal{D}(a = b) \tag{8.23}$$

Consequently, if the negation is inferred, the guess does not make the theory inconsistent. In general, any hypothesis, $H$, can be asserted as $\mathcal{D}H$. (8.23) is asserted in the usual way via the interface, justified by "abductive guess" and the nodes that were the source of the constants.

If (8.23) is asserted during unification, it does not mean that the unification will necessarily succeed. For example, unifying:

$$p(a) \vee q(a)$$

and

$$p(b) \vee r(b)$$

will result in the assertion of:

$$\mathcal{D}(a = b)$$

during the unification of $p(a)$ and $p(b)$. However, it is impossible to unify $q(a)$ and $r(b)$, even if $a$ and $b$ are identical. This is not regarded as a problem; it is quite possible that $a$ and $b$ are equal even though the entire unification fails.

The operation of the deductive device is therefore independent of the AEP. Once a hypothesis is added to the deductive device it is treated as any other non-monotonic assumption.

### Crystal's Abductive Unification

The AEP can be regarded as a development of Charniak's abductive unification algorithm. It differs from it in that it uses real equality and its operation can be described partially in terms of AEL. It therefore has a well-defined interface with an RMS[14]. Furthermore, unlike Charniak's system which has no notion of current context, the abductive equality process is constrained to work only within the confines of the deductor memory rather than the whole knowledge base.

### 8.4.2 Improving the Abductive Unifier's Guesses

The simplest abductive unifier will assert (8.23) whenever $a$ and $b$ do not unify. Therefore, given:

$$\mathrm{In}(\mathit{chair005}, \mathit{house014}) \wedge \mathrm{Chair}(\mathit{chair005})$$

and

$$\mathrm{In}(\mathit{table024}, \mathit{house014}) \wedge \mathrm{Table}(\mathit{table024})$$

the unifier will infer:

$$\mathcal{D}(\mathit{chair}005 = \mathit{table}024)$$

even if elsewhere it has been asserted that:

$$\neg(\mathrm{Table}(!f) \wedge \mathrm{Chair}(!f))$$

Intuitively, there is no point in assuming that a table is a chair because they are of different sorts. The performance of the RMS and unifier can be greatly improved by taking the sorts of the constants into account. Thus, the modified unifier only makes guesses of equality if the sorts of the constants involved are "Compatible", as defined in chapter 5. The main use of sorts in Crystal is therefore to *block* unwanted hypotheses. This approach has also been taken in Frail and TACITUS[15]. However, neither system allows guesses to be retracted if they have not been successfully filtered by the sort system.

### 8.4.3 Preferences on Hypotheses

In section (7.5) I explained how the local ordering of defaults along with the structure of the utterance and the context determined which extension was selected. A similar mechanism is used for the AEP.

The subsumption and equality checking part of the interface scans backwards for a compatible assumption in reverse chronological order to that in which the assumptions were initially added. Consequently, information inferred from later parts of an utterance is considered before information from earlier parts of it. Furthermore, within information

[14]Charniak's abductive unifier in Charniak (1988) merely backward chains to check for consistency. Should a later inconsistency occur, there is no RMS or AEL to handle the problem.

[15]Hobbs goes as far as saying that the sorts are only used as heuristics to speed up inferencing. Furthermore, they are the *only* means of checking for consistency (Hobbs, 1990).

loaded from an encyclopædic entry, information is loaded in reverse order, so earlier information is scanned before later information. Consider the following part of an entry about voles (the ordering is significant):

$$\text{Vole}(\textit{victor}) \tag{8.24}$$

$$\text{Vole}(\textit{vladimir}) \tag{8.25}$$

This not only states the fact that Victor and Vladimir are both voles, it states that Victor is more mentally prominent than Vladimir. If a definite reference is made to "the vole", the first Vole picked up will always be Victor.

If subsequently, the assertion that Victor is the vole proves incorrect, the fact that an abductive assertion of equality has been undone is noted (because the invalid equality node is tagged) and the subsumption/equality check resumes from where it left off, picking up the fact that Vladimir may be the Vole.

The particular use of the above backtracking mechanism during subsumption checks is not just to find candidate referents but to consider the application of competing scripts. For example, initially it may be predicted that a brick will be used as part of some building action, but after backtracking it may be predicted that it is used as a kind of blunt weapon.

The inferencing order combined with the notion of processing around the focal stress point of the utterance and the abductive unification process might be used to explain some aspects of Sidner's reference resolution algorithm (Sidner, 1979), since it naturally produces ordering in the deductor memory like those of her focus stack.

### 8.4.4 The Unification Algorithm

A sketch of the unification algorithm is provided in figures 8.1, 8.2 and 8.3. The algorithm works with typed entities. The types are constant, function, variable and, by default, list. Additionally, entities of the first three types have a sort associated with them. This is the sort of entity within Crystal's knowledge representation using its "isa" hierarchy.

#### Explanation of the Functions

- Unify takes two patterns and a set of existing bindings and returns a new set of bindings or fails[16].
- Unify calls Var.Unify, Const.Unify or Fn.Unify according to the type of the entities being unified.
- When two constants or functions cannot be unified, it is then checked if they are "=" in Const.Or.Fn.Unify. If they are not, but they are of Compatible sorts, an abductive guess concerning their equality is made by Non.Monotonic.Equal.

#### Explanation of the Auxiliary Functions

- Type.Of returns the type of an entity.
- Compatible is as defined in chapter 5.

[16]The code to handle failure is not included here.

```
FUNCTION Unify(Pat1, Pat2, Environment)

CASE
    Type.Of(Pat1) = variable: Var.Unify(Pat1, Pat2, Environment);
    Type.Of(Pat2) = variable: Var.Unify(Pat2, Pat1, Environment);
    Type.Of(Pat1) = Type.Of(Pat2) = constant:
                              Const.Unify(Pat1, Pat2, Environment);
    Type.Of(Pat1) = Type.Of(Pat2) = function:
                              Fn.Unify(Pat1, Pat2, Environment);
    {Type.Of(Pat1)} ∪ {Type.Of(Pat2)} ⊂ {constant, function}:
                              Const.Or.Fun.Unify(Pat1, Pat2, Environment)
OTHERWISE
    Unify(Tail(Pat1), Tail(Pat2), Unify(Head(Pat1), Head(Pat2), Environment))
END CASE
```

```
FUNCTION Var.Unify(Vpat, Pat, Environment)

IF Vpat = Pat
THEN
    Environment
ELSE
    IF Compatible(Sort.Of(Vpat), Sort.Of(Pat))
    THEN
        CASE
            Bound(Vpat, Environment):
                        Unify(Binding.Of(Vpat, Environment), Pat, Environment);
        OTHERWISE
            Change.Sort(Pat, Most.Specific.Sort(Sort.Of(Vpat), Sort.Of(Pat)))
            Add.To.Bindings(Vpat, Pat, Environment)
        END CASE
    END IF
END IF
```

Figure 8.1: Sorted Abductive Unification (1)

```
FUNCTION Const.Unify(Pat1, Pat2, Environment)

CASE
    Pat1 = Pat2: Environment;
    Name.Of(Pat1) = Name.Of(Pat2):
                IF  Subsumes(Sort.Of(Pat1), Sort.Of(Pat2))
                THEN
                    Add.To.Bindings(Pat1, Pat2, Environment)
                ELIF
                    Subsumes(Sort.Of(Pat2), Sort.Of(Pat1))
                THEN
                    Add.To.Bindings(Pat2, Pat1, Environment)
                END IF
OTHERWISE
    Const.Or.Fn.Unify(Pat1, Pat2, Environment)
END CASE
```

```
FUNCTION Fn.Unify(Pat1, Pat2, Environment)

IF Compatible(Sort.Of(Pat1), Sort.Of(Pat2)
THEN
    LET
        Env = Unify(Function.Of(Pat1), Function.Of(Pat2), Environment)
    IN
        IF Valid(Env)
        THEN
            IF Subsumes(Sort.Of(Pat1), Sort.Of(Pat2))
            THEN
                Add.To.Bindings(Pat1, Pat2, Env)
            ELSE
                Add.To.Bindings(Pat2, Pat1, Env)
            END IF
        ELSE
            Const.Or.Fn.Unify(Pat1, Pat2, Environment)
        END IF
    END
END IF
```

Figure 8.2: Sorted Abductive Unification (2)

```
FUNCTION Const.Or.Fn.Unify(Pat1, Pat2, Environment)

IF Unifies(Pat1, Pat2)
THEN
    IF Depth(Pat1) > Depth(Pat2)
    THEN
        Add.To.Bindings(Pat1, Pat2, Environment)
    ELSE
        Add.To.Bindings(Pat2, Pat1, Environment)
    END IF
ELSE
    Non.Monotonic.Equal(Pat1, Pat2, Environment)
END IF
```

```
FUNCTION Non.Monotonic.Equal(Pat1, Pat2, Environment)

IF Compatible(Sort.Of(Pat1), Sort.Of(Pat2))
THEN
    IF Abduction.Enabled
    THEN
        Add.Abductive.Inference(Pat1, Pat2)
    END IF
END IF
```

Figure 8.3: Sorted Abductive Unification (3)

- Bound checks if a constant, function, or variable is bound in the environment passed as an argument.
- Binding.Of gives the value a constant, function, or variable is bound to. Bindings are arranged so that the more syntactically complex entity is bound to the less complex entity.
- Change.Sort changes the sort of the entity referred to by its first argument to the sort expressed by its second argument.
- Add.To.Environment returns an environment which is the environment passed as its third argument augmented by binding its first argument to its second.
- Unifies returns true if its first argument is "=" to its second argument and false otherwise.
- Abduction.Enabled returns true if abduction is currently switched on. Without this, the algorithm merely functions as a sorted unification system.
- Add.Abductive.Inference creates an appropriate assumption that its arguments are equal and passes it to the interface. The interface will pass this information to the equality sub-system (see chapter 4)[17].

### 8.4.5 The Equality Sub-System

The equality relation, "=", is problematic because it cannot easily be described using elimination rules but must be provided with adequate meaning because it is utilised by the non-demonstrative inference process. My solution has been to dedicate a separate section of the deductive device, the *equality sub-system*, to providing meaning for this relation. In this section, I justify the need for the separate system and briefly describe its implementation.

#### Problems With the Equality Relation

For full effect, the equality relation, "=", cannot simply be asserted; its encyclopædic entry must include axioms stating the fact that it is an equivalence relation that licenses substitution of its arguments in non-modal contexts.

An equivalence relation is reflexive, symmetric, and transitive. i.e.

$$\forall x. (x = x)$$

$$\forall x, y. (x = y) \supset (y = x)$$

$$\forall x, y, z. (x = y) \land (y = z) \supset (x = z)$$

The rule of substitution is:

$$\forall x, y, p. (x = y) \supset (p(x) = p(y)) \tag{8.26}$$

The latter rule is problematic: it is not an elimination rule and it is expressed in higher-order logic. Unfortunately, it is vital for describing the meaning of "=" and there is no way of converting it into a first-order elimination rule. However, without the rule, the enrichment mechanisms of Crystal cannot operate. For example, consider:

[17] I have not shown here how justifications for equality are passed around.

Tweety is a bird. He is hungry. (167)

This would translate as:

$$\text{Bird}(\mathit{tweety}) \tag{8.27}$$

$$\text{HOLDS}(\text{Hungry}(\mathit{sk.male087}), \mathit{NOW}) \wedge \\ \text{Male}(\mathit{sk.male087}) \wedge \text{IT}(\mathit{sk.male087})$$

Now, assuming that it is asserted that:

$$\mathit{tweety} = \mathit{sk.male087}$$

without the rule (8.26), it is not possible to infer as obvious an assumption as:

$$\text{Male}(\mathit{tweety})$$

Another problem is that the definition of "=" can result in an inference explosion. As equivalences are progressively asserted, the information in the deductor memory can expand at an alarming rate. This not only slows down the operation of the inference engine as a whole, but also tends to bias the information count in an undesirable manner. Since an abductive assertion of equality is a contextual effect, it follows all the inferences drawn from it are also contextual effects. Thus, consider a situation where the database consists of information concerning the entity "$a$":

$$p(a), q(a), r(a), s(a), \ldots$$

Given that $p(b)$ is also true it may be inferred that $(a = b)$. In this case, the following will also be asserted:

$$p(b), q(b), r(b), s(b), \ldots$$

Intuitively, each of these inferences should not be counted as new information. Indeed, they are unnecessary altogether if the unifier itself is aware of the notion of equality[18].

A final problem is that allowing forward-chaining about the equivalence of functions and constants can result in an infinite number of substitutions. To give a practical example, consider the equivalences that are derived by the "stone and cherry" example:

$$\text{Stone.Of}(c1) = s1, \text{Cherry.Of}(s1) = c7$$

Given the rule of substitution, it follows that:

$$\text{Stone.Of}(\text{Cherry.Of}(s1)) = s1, \text{Stone.Of}(\text{Cherry.Of}(\text{Stone.Of}(c7))) = s1, \ldots$$

[18]This notion is often integrated in resolution theorem provers as *paramodulation*.

### Integrating Equality into the Deductive Device

Since equality is such an important notion for the deductive device, Crystal has a separate sub-system to handle it. The interface between the equality sub-system (ES) and the deductive device can be implemented as part of the unification algorithm. During the unification of two constants, the ES is first queried to check if the two constants are equal. If so, it returns a suitable justification of equality. This justification will be incorporated into the final justification for the inference if the top level meta-rule application succeeds.

The ES is a specialised inference engine that understands only equality, supported by a simple RMS. Every time an equality is asserted into the deductive device, the interface passes the details to the ES which makes any necessary inferences and maintains its own justifications for them. These justifications will be passed by the unifier to the controller so that when a meta-rule is successfully applied, the justification for the new assumption will include the use of the germane equality assertions. Consequently, if an assertion of equality proves wrong, all of the conclusions dependent upon it can be retracted.

### Implementation Details

The interface calls the ES via the function "Add.Equivalence($f_1, f_2, j$)". $f_1$ and $f_2$ are the two equivalent formulæ and $j$ is the justification.

Equivalences are contained in database entries of the form (item, equivalence list). The equivalence list gives details of constants that are equal to the "item". An equivalence is of the form: (name, justification list). "name" is the name of the constant that is equivalent to "item". The justification list contains reasons for believing the equivalence.

1. The first step taken by "Add.Equivalence" is to update the database. If an entry with item $f_1$ is not already present in the database, a new entry for it is created and added to the database. If the item is already present, the new equivalence is merged with the equivalence list, after "Add.Equivalence" is called on the old equivalence list as follows.

   Given an equivalence list of the form $[(e_1, jl_1), \ldots (e_i, jl_i), \ldots (e_n, jl_n)]$ and a new equivalence $(f_2, j)$, "Add.Equivalence($f_2, e_i, j \cup jl_i$)" is executed for each equivalence. This will have the effect that there is a database entry for every constant that is involved in an equivalence relation.

2. The next step in Add.Equivalence($f_1, f_2, j$) is to complete the relation, by making it symmetric and transitive.

3. In order to make the relation symmetric, Add.Equivalence($f_2, f_1, j$) is called.

4. In order to make the relation transitive, all of the equivalences relating to $f_2$ are extracted from the database (this merely involves retrieving the equivalences for entries with item $f_2$).

   For every equivalence, $(e_i, j_i)$, Add.Equivalence($f_1, e_i, j \cup j_i$) is called.

5. For every database item, ($k_i$, Equivalence.List$_i$), calculate $S_i$, the result of substituting $f_1$ for $f_2$ in $k_i$. If $S_i \neq k_i$ then call Add.Equivalence($k_i, S_i, jl$)

   This means that if, for example, $p(a)$ is in the database, and $a = b$, then so should $p(a) = p(b)$. This can cause a potentially infinite chain of substitutions if, say, $a = p(a)$. The solution adopted in Crystal is to limit the depth of nesting of the database to an arbitrary number. A depth of four was chosen after experimentation since higher levels seemed of little use and slowed down the system too much (growth is exponential).

"Add.Equivalence" is a closure operation. All possible equivalences (to a certain depth of nesting) are present as database entries. Therefore, given a single item, it is a trivial matter to find out all of the other items to which it is equivalent. Consequently, it is easy to check if two objects are equal. When a query is successful, the appropriate justification set is returned.

The final operation that the ES must support is retraction of an equivalence. In order to simplify the system, a node is not made "out", it is simply erased. This is reasonable because invariably equivalences are permanently retracted. Many guesses of equality are wrong in *any* context because of the inherent properties of the objects involved. For example, Crystal has guessed that an island in the middle of a lake is the land that surrounds that lake. The guess was made because both objects are pieces of land. The inference is finally retracted when the deductive device infers that something cannot surround itself.

When an equivalence is retracted, all of its consequences are retracted. This merely involves erasing every justification that includes the retracted node. If a justification is removed and there are no other justifications, then the associated equivalence is also removed by calling the retraction operation recursively.

With the introduction of equality, it is possible for justifications to have more than two nodes in their inlist; they may also include any number of nodes which are assertions of equality that allow the antecedent clauses to unify.

### 8.4.6 Dependency-Directed Backtracking

When assertions of the form (8.23) are made which are inconsistent with the current theory, it should be possible to derive their negation and hence make them "out". However, since the inference system is incomplete, this may not be possible. As a consequence, an inconsistency can occur because the appropriate hypothesis has not been retracted. During dependency-directed backtracking, the rejection of abductive guesses is preferred, especially if they are not binding an object with property "REF". Furthermore, more recent hypotheses are always rejected in preference to older ones because it is less likely that they have many consequents, thus change should be minimal (see chapter 7 for more precise details).

In order to correctly handle rejection of hypotheses, whenever an item of the form "REF($p$)" is passed to the interface, $p$ is added to a "REF" list, which contains all definite references. After inferencing, it is a simple task to see if an object has been bound to anything by querying the ES to check whether it has been asserted equal to another term.

### 8.4.7 Queries

The AEP can also assist the querying system (see appendix A).

Various sorts of queries to the system are allowed, among which are queries of a formula. Such queries return the set of all assumptions in the deductor memory which unify with the formula. The use of abduction is optional. A query such as:

What happened? (168)

might be phrased as:

$$\text{OCCUR}(\mathit{skolem.event012}, \mathit{skolem.interval01})$$

Given this, the AEP can be used to find matching events, as described in Charniak (1988). However, the querying mechanism is simple and does not properly handle conjunctions etc.

## 8.5 Summary and Conclusion

I have now completed my description of Crystal's implementation. In this section, I will summarise the key points of this chapter, consider an alternative implementation, and compare Crystal with some related system by reference to the multiple extension problem. In chapter 9, I will suggest some modifications to Crystal that will provide it with a knowledge representation that incorporates a formal model of quantitative belief.

### 8.5.1 Deduction and Hypotheses

A modification of Charniak's abductive unification algorithm has been adopted as the most suitable non-demonstrative inference scheme for Crystal. Crystal's unifier has been modified in such a way that it will occasionally guess that certain entities, denoted by terms, are equal. The guess is passed to the deductive device as a default assumption.

This suggests a general way of interfacing non-demonstrative inferences to the deductive device; they can be asserted in the deductor memory prefaced by the autoepistemic default operator, $\mathcal{D}$. In this way, if they prove incorrect, they will be automatically overridden. The $\mathcal{D}$ operator is effectively stating the fact that its argument should be given up in the light of contradictory evidence.

The modified unification process handles both deduction and abduction, making the operation of the inference engine both simple and elegant. In the course of inferencing, hypotheses are added to the deductive device. They are then verified or rejected in a purely deductive manner.

Abductive unification is an expensive process. However, it is made tractable by employing the sort hierarchy to prevent incompatible terms from unifying and limiting the scope of the unifier to the deductor memory.

### 8.5.2 Parallelism

The current implementation of Crystal only considers a single putative reference at a time, using crude heuristics to guide its search. The algorithm could be modified to consider multiple referents simultaneously. The Relevance of several candidate references could be examined simultaneously using a backtracking ATMS (deKleer, 1986b). Salience and strength of belief could guide this process so that it only considers the most accessible interpretations.

### 8.5.3 The Multiple Extension Problem

It has been seen that the general problem of utterance interpretation can be viewed in terms of the multiple extension problem; a suitable non-monotonic theory must be used to establish contexts that logically imply the utterance. Different systems have different criteria for selecting the contexts.

Viewing interpretation this way, it can be seen that the main difference between RT and its related computational approaches is that the computational approaches use notions of probability and utility to select a particular extension that makes the utterance true, whereas Relevance balances the contextual effects against the effort expended in establishing them; the search for contextual effects favours contexts which are heavily connected to previous contexts and that yield many non-trivial inferences. However, these connections may not be explanations in the sense of TACITUS or Wimp.

RT places particular emphasis on the notion of context, previous and current; it is not just important to explain an utterance, but to explain what it has to do with previous contexts. This is not handled by systems such as TACITUS, which only considers single sentences, and is not made explicit in systems such as Wimp.

SW use the Principle of Optimal Relevance in order to make the choices available to an individual at any one time determinate. It was argued in chapter 6 that this was not necessarily useful when incrementally handling linguistic ambiguities. In this chapter, it has been shown that many pragmatic ambiguities can be handled in this more deterministic manner by assuming that there is an ordering on information as it is placed in the deductor memory, and that there are priority relations between indexical rules. Should a choice prove wrong, dependency-directed backtracking and the subsumption/equality system recover the next interpretation in the ordering.

The priority relations and orderings are closely related to the cognitive notion of salience, which will be further discussed in chapter 9.

# Chapter 9

# Strength and Belief Measures

There are several aspects of the ideal system outlined in chapter 3 that I have not handled. In this chapter, I propose an extension to Crystal to deal with the aspects of RT that relate to the notion of strength of belief. I describe several related problems, and show how a partial solution can be obtained by augmenting the existing AEL with belief measures. I then sketch this new logic and some details of its future implementation. Finally, I compare this augmented version of Crystal to some related systems.

## 9.1 Issues Relating to Strength

### 9.1.1 Strength Has No Semantics

Much as strength is vital to RT, SW have given it no theoretical justification. Their model of strength is underspecified and has no formal substance.

The important aspects of their description of strength are:

- It should only be used comparatively.
- Strengths should not be considered as precise values, but as gross estimates.
- It is dependent on processing history.
- There must be some means of modelling dependent and independent strengthening.
- New strengths can be inferred by considering best and worst case scenarios involving the entities concerned.

SW shy away from the use of numbers on the grounds that it is unlikely that humans entertain strengths as, say reals. Instead, they propose a model using something a little like "enumerated types" in programming languages. They provide some simple inference rules, but only justify them by appeals to intuition.

It is clearly desirable to have a formal theory of strength that provides a precise account of its behaviour under various conditions.

I see no reason why numbers cannot be used to model strengths. If a numerical model proves successful, it does not mean that people actually have numbers in their heads. It merely means that this is an adequate way of modelling human mental processes, as has been done in many current expert systems.

### 9.1.2 Contextual Effects

The various contextual effects of an utterance can be considered as a single contextual effect, since they all effect a change in overall belief (see section 7.6.1). A formal model is then required so that change in belief can be quantified.

An important issue with RT is whether information change should be measured syntactically or semantically. It was shown in section 5.1 that measuring syntactic information change can lead to problems. Semantic information change is more difficult to quantify, but it is possible to relate this to change in strength. By adopting the unified approach to contextual effects using change in strength, there is an implicit commitment to measuring change in semantic information.

### 9.1.3 Utility, Manifestness and Salience

SW do not relate manifestness to any cognitive notion. The closest such notion is *salience*, which is mental prominence. Clearly, salience and strength are related notions. However, they are not identical; tautologies must be entertained at full strength, yet they may not be at all salient. Indeed in Crystal, tautologies are not even input into the deductive device, thus they have minimal salience.

The degree of belief in an assumption may actually affect its mental prominence. Conclusions derived from an uncertain assumption will be equally uncertain. They will be even more uncertain if further uncertainty is introduced via the meta-rules or other premises. They may even become totally uncertain. Generally, less useful information will follow from uncertain assumptions. Thus, there is little point in such assumptions being mentally prominent, because they will merely be occupying unnecessary space.

These points suggest that the notion of salience is connected with that of *utility*, which is intended to be a measure of the usefulness of an assumption. Clearly, it is desirable for the assumptions with the highest utility to be the most salient. Ideally, perhaps, only the most useful assumptions should be salient, although it is difficult to envisage a cognitive mechanism that could realise this.

### 9.1.4 Extension Selection

Faced with a choice of several candidate extensions, a theory of understanding must select the extension which it deems most appropriate. Strength can be used to assist this choice in the following ways:

- In RT, total information change will be total change in the magnitude of the strength of all present assumptions.

- Given an inconsistency, a culprit must be selected and rejected. The putative culprit held with the least strength should be rejected since less inferences are generally dependent on it; this should result in minimal semantic information change to the system.

Thus, the choice of extension should become deterministic. Clearly, for these ideas to be implemented requires a more precise notion of strength.

### 9.1.5 The Adequacy of Elimination Rules

Elimination rules are enforced because it is necessary that the inference process eventually terminates. However, there are alternative, arguably more satisfactory ways of limiting inference.

Rather than restricting the rules in the logic itself, it is possible to associate values with every assumption. These values, which can still be considered as measures of strength, decrease with depth of inference and will only increase if independent support is provided for the assumption. This mimics SW's descriptions of dependent and independent strengthening. The inference process can now be forced to terminate by incorporating *acceptance functions* into the interface (see Loui (1987) and Kyburg (1987b)). Such functions would examine the strength of an assumption and choose not to add it to the deductive device if it were below some threshold value. Thus, assuming strength of assumptions decreases with depth of inference, the inference process must eventually terminate.

It is not always the case that strength of a consequent will be less than that of the antecedents. The strength will only be less if the antecedents are not totally certain. Thus, it is possible to relax SW's restrictions on inference rules such that it is only necessary for them to be elimination rules when dealing with information held at full strength. In all other cases, the inference process must terminate anyway.

The advantage of this approach is that, whereas the inference system is complete, the restrictions on knowledge are weaker. Thus, for example, introduction rules can be applied to yield one or two levels of introduction before their inferences become too uncertain. It is these low levels that are of use in making "plausible" inferences.

## 9.2 Belief Measures as a Unifying Approach

In order to measure strength, I propose an approach which utilises probabilities to model beliefs. The belief measures have the advantage that they can easily be incorporated into a Relevance-based system as described in section 3.3 and can be used to measure changes in semantic information, the unifying concept underlying all conceptual effects.

### 9.2.1 Degrees of Belief and Confidence

SW's model of strength can be interpreted using probabilities. In figure 9.1, a typical SW-style scale is depicted. One way of modeling this is to assume that the real numbers between 0.0 and 1.0 are divided into equivalence classes, in this case four of them. Each equivalence class contains probabilities that are all regarded as parts of the same category, e.g. probabilities between 0.0 and 0.25 are "weak".

Unfortunately, SW's approach is not specific enough. Are these equivalence classes all of the same size? A perfectly reasonable implementation might have the equivalence classes of a width shown in figure 9.2.

Moreover, this model can be generalised to incorporate the width information as an aspect of the strength itself. The width of a region would give some indication of how precise it was. This is known as *confidence* in a belief. It would clearly be convenient if a theory of belief measures could take confidence into account along with strength. I will formally develop these notions in section 9.4.

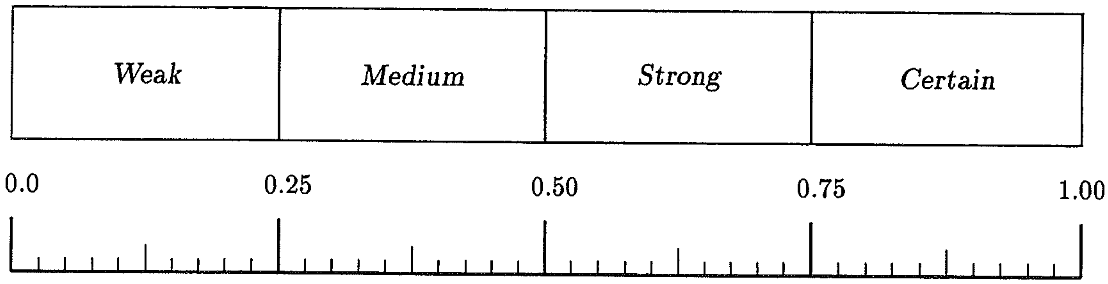



Figure 9.1: Strength Scale Divided Equally

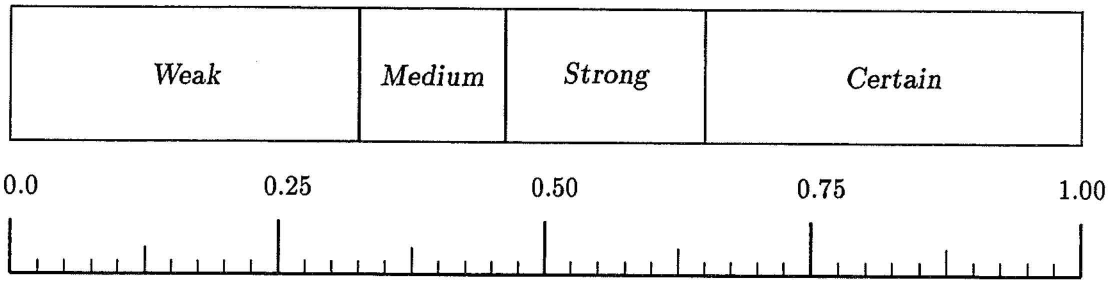



Figure 9.2: Strength Scale With Unequal Divisions

### 9.2.2 Integrating Belief Measures into the Deductive Device

The best method for integrating belief measures into the deductive device is to augment the existing logic to handle them. Some kind of prefix could be added to assumptions to express their associated belief measure. The meta-rules, which do not themselves have any associated belief measure, operate on assumptions in the deductor memory to produce new assumptions with appropriate belief measures. Thus, though belief measures are represented at the encyclopædic (object) level, they cannot be directly manipulated at this level.

### 9.2.3 Logic, Probability and Semantic Information

#### The Carnapian View of Probability

Carnap (1956, 1962) proposed a logical view of probability that considers probabilities to be valuations over sets of *possible worlds.* These worlds are valid states of a some real-world system.

A possible world consists of a complete specification of a state of the system using either instantiated FOPC predicates or propositions. Each state must be internally consistent and can be regarded as a canonical model. An elementary event of probability theory can be seen as the conjunction of all of the negated and unnegated atomic literals true in a particular world. Thus, each world describes a distinct event.

A probability distribution can be induced over the universe of possible worlds that yields the probability that any particular world corresponds to the actual state of the real-world system it denotes. The probability that an arbitrary formula is true can be seen as the sum of the probabilities of all of the worlds that satisfy that formula. The simple sum can be taken because the worlds represent distinct events.

Notions of *certainty* and *precision* can be applied to information evaluated with respect to these worlds. Precise and certain information identifies a particular world. Certain, imprecise information identifies a set of possible worlds. Uncertain information induces a probability distribution over the set of worlds. Unfortunately, there is no measure of precision in Carnap's system. It is the notion of imprecision that has presumably led SW to talk about gross estimates. The logic that I develop in this chapter will explicitly represent precision.

### Semantic Information and Inference

Semantic information content is usually defined as some function of the possible worlds that can satisfy a logical form (see Bar-Hillel (1964) for more detail).

In order to illustrate the Carnapian view of probability and its relation to semantic information, I will examine how Johnson-Laird postulates semantic constraints on inference rules. This, of course, is also of significance to RT, which has different views concerning inference.

Rather than as a basis for selecting particular interpretations of a language, semantic information has been considered as the basis for formulating "psychologically plausible" inference rules by Johnson-Laird (1983). He suggests that meaningful inference rules should increase semantic information, or *informativeness*, while keeping the resulting forms as succinct as possible. Much as he suggests that the suitability of rules can be defined in terms of the Gricean notion of informativeness, it is more in keeping with the spirit of Relevance Theory:

> No conclusion contains less semantic information than the premises on which it is based or fails to express that information more parsimoniously.

Johnson-Laird goes on to give a metric for comparing the informativeness of formulæ in propositional logic. Propositions represent states of affairs. Expressions involving propositions place constraints on the possible states of affairs that the propositions can represent. For example, the propositions $p$ and $q$ can represent four possible states, but the formula $p \vee q$ is satisfied by only three: the states where $p$ or $q$ are true. These states of affairs are the equivalent of Carnapian possible worlds.

A function $P(f)$ is defined to yield the *truth-table probability* of $f$. It is calculated as the number of worlds[1] that satisfy $f$ divided by the total number of possible worlds (which will be $2^n$, where n is the number of unique propositions in $f$). This assumes the a priori probability of each proposition in $f$ is equal. The semantic informativeness of a formula $f$, $\mathcal{S}(f)$ is then defined as:

$$\mathcal{S}(f) = 1 - P(f) \tag{9.1}$$

(9.1) can be used to calculate the informativeness of $p \vee q$ as 0.25. An or-introduction rule which introduces an arbitrary concept $q$ given a proposition $p$ would be ruled out by this restriction because $\mathcal{S}(p) = 0.5$ and $\mathcal{S}(p \vee q) = 0.25$ so the disjunction is less informative. This is illustrated in figure 9.3.

[1] These can be regarded as complete truth assignments or states.

| State | $p$ | $q$ | $p \vee q$ |
|---|---|---|---|
| 1 | false | false | false |
| 2 | false | true | true |
| 3 | true | false | true |
| 4 | true | true | true |

| | |
|---|---|
| Total number of states: | 4 |
| States that satisfy $p$: | 3, 4 |
| Number of states that satisfy $p$: | 2 |
| States that satisfy $p \vee q$: | 2, 3, 4 |
| Number of states that satisfy $p \vee q$: | 3 |

$$\begin{aligned} P(p) &= \frac{\text{Number of states that satisfy } p}{\text{Total number of states}} = \tfrac{2}{4} = 0.5 \\ P(p \vee q) &= \frac{\text{Number of states that satisfy } p \vee q}{\text{Total number of states}} = \tfrac{3}{4} = 0.75 \end{aligned}$$

$$\begin{aligned} \mathcal{S}(p) &= 1 - P(p) &= 1 - 0.5 &= 0.5 \\ \mathcal{S}(p \vee q) &= 1 - P(p \vee q) &= 1 - 0.75 &= 0.25 \end{aligned}$$

Figure 9.3: $p \vee q$ is semantically less informative than $p$

Johnson-Laird uses the "parsimoniousness" in the definition in a similar way to that used by SW to relate effort to information. Given a choice between two semantically equivalent assumptions, the assumption that requires the least effort to express should be chosen in favour of the alternative. For example, $p$ is preferred to $p \wedge p$ which is as informative but less succinct.

General introduction of conjunctions is ruled out on the grounds that it does not yield new information[2], but the expression $p \wedge q$ is less economic in form than its concepts expressed individually.

The informativeness constraint also dismisses rules which SW consider plausible, like *and-elimination* rules, which for example, would yield $p$ from $p \wedge q$. However, Johnson-Laird's theory also has its faults, which may explain partially explain why it diverges from SW's ideas[3]. There is also little explanation of how to compute parsimoniousness, since it is given no formal definition. Furthermore, if the logic is extended to include quantifiers and predicates, it may be a lot more difficult to calculate the number of possible worlds which satisfy a logical form.

Perhaps the most fundamental flaw with this approach to semantic information is that it assumes a uniform distribution. Without this assumption, it is impossible to make his generalisations. This is compounded by the fact that distributions will vary between various independent systems or *frames of discernment.* In section 9.7.1, I will describe an approach to measuring semantic information content which does not make the above simplifications and is more apt for RT while still providing a means for defining acceptable inferences.

[2]Johnson-Laird gives a procedure for calculating $\mathcal{S}$ for single antecedent rules, but does not describe how the informativeness of multiple antecedents is calculated procedurally. However, it is clear that the same worlds that satisfy $p$ and $q$ will satisfy $p \wedge q$.

[3]However, Johnson-Laird provides some psychological evidence to support his theory, whereas SW do not.

## 9.3 Foundations for a Theory of Strength

This section provides a brief background which justifies my rationalisation of SW's strengths in Crystal; a thorough survey of this field is beyond the scope of this thesis. Throughout this section I assume that it is best to draw on available theories of strength, which invariably use real numbers to express belief measures.

### 9.3.1 Early Systems

Early systems tended to confuse notions of salience, utility and probability. Typical examples of such systems are Schank's ad-hoc *interest* factors (Schank, 1979) and the *certainty factors* of early versions of Mycin (Buchanan and Shortcliffe, 1984). All of these systems use numbers to represent their factors, but no theoretical justification is provided. This is particularly obvious with Schanks's interestingness, which was intended to model salience and utility, yet Schank states in conclusion to his paper on this topic:

> Perhaps ultimately our systems using interestingness will be similar to the way certainty factors are used in MYCIN.

Yet the use of certainty factors in Mycin was much closer to applications of conventional probability theory. Indeed, recent versions of Mycin have utilised such theories.

### 9.3.2 Bayesian Inferencing: Conditional Probabilities

One interpretation of degree of belief in a sentence $S$ is that it is the probability that that sentence $S$ is true. Using simple statistics, it is a straightforward task to calculate the probability of a propositional sentence being true if the probabilities of all of its constituent literals are known. However, this is information that is normally not available.

To complicate matters further, there is some doubt as to the suitability of FOPC for statistical inference. There is considerable controversy amongst researchers as to whether conditional sentences should be interpreted in terms of the material implication of FOPC or in terms of conditional probabilities. When using conditional probabilities, rules of the following form are employed:

$$P \Longrightarrow Q$$

Given $P$ is true with probability $p(P)$, then $Q$ can be inferred with probability $p(Q|P)$, which is the probability that $Q$ is true given that $P$ is true. Using classical probability theory, such a probability can be calculated using *Bayes' rule*:

$$p(Q|P) = \frac{p(P|Q).p(Q)}{p(P)}$$

Unfortunately, the dependencies between events such as $P$ and $Q$ are not always known. As opposed to the case of material implication, the probability of every literal need not be known. However, some conditional probabilities must be obtained, but this is not always possible. It can also be shown that probabilistic inference employing networks of conditional rules is NP-hard (Cooper, 1990). However, for the difficult cases, approximation techniques can be used. This is often the case for typical expert system applications (Pearl, 1988). A further disadvantage of the "conditional" approach is that there is no way of expressing degree of belief in the rules themselves.

Much of the problem here comes down to the inherent *uncertainty* regarding the beliefs present in a knowledge base. Because the dependencies between the data are often not known, it is difficult to fix a point probability that a belief is true.

### 9.3.3 Probabilistic Logic

Nilsson has suggested an alternative approach, which implicitly incorporates uncertainty, but unfortunately removes the uncertainty in its very last stages. However, in the next section, I shall build on the concepts introduced here in order to explicitly maintain an uncertainty measure.

Nilsson's approach is based on Carnap's model of probabilities. A probabilistic theory can be regarded as a probability distribution over a set of worlds, which correspond to propositional calculus models. The distribution indicates the probability that a particular world corresponds to the actual state of affairs. The belief in a sentence is then the sum of the probabilities of all the worlds that satisfy the theory. This can be seen as an augmentation of classical logic. Nilsson (1986) introduces *probabilistic logic* based on this approach.

Assume there are $k$ consistent worlds that interpret a theory. A vector, $\vec{P}$, can be provided such that its $n^{th}$ element, $p_n$, gives the probability that a particular world corresponds to the "real world". Since worlds are similar to minimal models, they must be disjoint. Therefore:

$$\sum_{n=1}^{k} p_n = 1 \tag{9.2}$$

Nilsson now introduces *truth valuation vectors.* Each truth valuation vector is associated with a unique possible world and the sentences of the theory. Each element of the vector corresponds to a unique sentence of the theory. An element of a valuation vector is 1 if its corresponding world makes the sentence true and 0 otherwise. From the $k$ valuation vectors, $\vec{I_1}, \ldots, \vec{I_k}$, a matrix, $\vec{I}$ of all the interpretations can be formed by making each of the individual vectors one of its columns. If there are $l$ sentences in the theory, the vector will have $l$ rows and $k$ columns.

Now, the probabilities of each sentence in the theory, $\pi_1, \ldots, \pi_l$ can be considered as a single vector, $\vec{\Pi}$, of size $l$. It can be calculated by using the formula:

$$\vec{\Pi} = \vec{I}.\vec{P} \tag{9.3}$$

Given that probabilities must be between 0 and 1, and that the probabilities in column $\vec{P}$ must sum to 1, constraints can be placed on the values of $\vec{P}$ for a given $\vec{I}$. For example, given the theory $P$, $P \supset Q$, $Q$, there are four consistent possible worlds:

$$\vec{I} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Substituting this into the equation (9.3), using constraint (9.2), and the fact that probabilities must lie in the range [0..1], it can be inferred that:

$$p(P \supset Q) + p(P) - 1 \leq p(Q) \leq p(P \supset Q)$$

where

$$\vec{\Pi} = \begin{bmatrix} p(P) \\ p(P \supset Q) \\ p(Q) \end{bmatrix}$$

This inequality was obtained by considering values of $\vec{P}$ where it is certain which possible world must be the "real" world, i.e.

$$\vec{P} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The extreme values of the elements of $\vec{\Pi}$ can be established by substitution of the limits of $\vec{P}$ given above. All other values of these will lie between these extremes. A graph can be constructed by plotting the values of the various elements of $\vec{\Pi}$ along different dimensions. In this case, the extreme values of these elements can be circumscribed by a convex hull, within which the actual probability values must lie[4]. More generally, the solution of equation (9.3) becomes intractable for large matrices and numerical approximations must be used.

Any consistent assignment of probability values that satisfies the resulting inequality is selected to provide a final value for $\vec{\Pi}$. This approach has the advantage that it expresses the inherent *uncertainty* or lack of precision present in the theory. The exact probability values for the sentences are not known. All that is known is the range of values that are acceptable. Unfortunately, this information is lost when final values are selected.

Consider a sentence that is assigned a degree of belief of 1. If it was constrained to be 1 no matter what the degree of belief of the other sentences in the theory, it is clearly different to a belief whose strength could vary between 0 and 1 depending on the strength of other beliefs in the theory. In other words, a belief that is certainly true is different to a belief that might, or might not, be true depending on the circumstances. The more certain we are about a particular belief, the more useful that belief is likely to be, since the stronger the inferences we can reasonably derive from it.

The amount of uncertainty, the degree to which the probabilities can vary, are useful as guides of utility and contributors to salience. Furthermore, it provides a better model of SW's inexact strengths. It is not surprising that people are bad at comparing strengths when they turn out to be non-scalar quantities. Strengths can be seen as comprising a degree of belief together with some *uncertainty* or *confidence* measure. However, there is no reason to believe that these confidence factors are fixed.

### 9.3.4 Dempster-Shafer Ranges

Recent work in statistical inference techniques has examined non-Bayesian statistical inferencing (see, for example, Tong et al. (1983), Dubois and Prade (1985), Dubois and Prade (1987), Kyburg (1987a), Smets and Kennes (1990) and Smets and Hsia (1990)). One of the most popular of these methods involves the use of *Dempster-Shafer* ranges (Shafer, 1976); I have adopted a variant of this approach because the ranges combine notions of degree of belief and uncertainty.

[4]The reader is referred to Nilsson's paper for more detail.

A Dempster-Shafer range (DSR) is expressed as [*support, plausibility*], associated with a statistical event. The *support* is the minimal probability value for that event, the *plausibility* is the maximum. The *uncertainty* is defined to be *plausibility* − *support*[5]. A succinct introduction and justification for this approach can be found in Zadeh (1986). It will become apparent that a DSR is a theoretically-motivated representation of the strength regions depicted in figure 9.2.

The remainder of this chapter describes how DSR's can be handled by Crystal's deductive device and integrated into a knowledge representation such as that described in chapter 5 so that it can handle degrees of belief and uncertainty. I will then show how DSRs can contribute to the solution of the problems enumerated in section 9.1.

## 9.4 A Logic Employing Dempster-Shafer Ranges

Crystal's deductive device is a forward-chaining inference engine that manipulates formulæ of autoepistemic logic present in its deductor memory. New information is added to the deductor memory via the interface, which can invoke a separate RMS in order to preserve or attain logical consistency.

I am now going to consider how to modify autoepistemic logic to handle degrees of belief and uncertainty. I shall do this progressively. First, I consider how propositional calculus can be subscripted with DSRs, and how such a logic could be integrated into the deductive device by showing how to derive any necessary inference rules. Next, I consider how the theory might be extended to handle quantifiers or schemas. I then show that, even though the logic described so far is monotonic, an inference engine using DSRs benefits from the support of a novel RMS; the RMS allows for efficient incremental updating and revision in the face of inconsistencies. Finally, I will extend the logic to handle a modal operator that assumes evidence is present when such evidence cannot be logically established. The RMS is easily modified to handle this new logic.

### 9.4.1 Enhancing Propositional Logic

It is possible to enhance normal propositional calculus by associating DS ranges with every syntactically valid sentence in the language. A sentence can now be written:

$$A_{[s(A),p(A)]}$$

The ranges can be interpreted as providing evidence for or against the proposition, $A$, which they are associated with (Garvey et al., 1981). It is even possible for the ranges to indicate evidence both for and against a belief. For example:

| | |
|---|---|
| $S_{[0.7,0.7]}$ | $S$ is believed with point probability 0.7. |
| $S_{[0,0]}$ | $\neg S$ is true. |
| $S_{[1,1]}$ | $S$ is true. |
| $S_{[0,1]}$ | There is no information about $S$. |
| $S_{[0.7,1]}$ | $S$ is believed with uncertainty 0.3. |
| $S_{[0,0.6]}$ | $\neg S$ is believed with uncertainty 0.6. |
| $S_{[0.4,0.6]}$ | There is some evidence for $S$, and some for $\neg S$, with uncertainty 0.2 |

[5]It will be seen that I do not actually use Dempster's rule of combination (Dempster, 1968). Thus, it may be better to view this representation as upper/lower probability bounds.

DSRs can be given a model-theoretic semantics by enhancing Carnap's view to incorporate the state of an agent's knowledge of a particular state of affairs. Possible worlds now assign truth values not only to normal propositions, but to *epistemic propositions* (Ruspini, 1986). An epistemic proposition describes a belief of an ideally rational agent[6]. Typically, an agent will not have complete knowledge of the state of affairs in every possible world. As with Carnap's model, a probability distribution can provide the likelihood that any particular possible world corresponds to the real world. Since possible worlds must be distinct, the sum of their probabilities must be 1.

The DS support for a proposition $P$ can be interpreted as the sum of the probabilities of each possible world where $P$ is known to be true. The DS plausibility is the sum of the probabilities of each world where $P$ could consistently be known to be true. These notions are quite closely related to the modal notions of necessity and possibility (Dubois and Prade, 1987).

Given two sentences, $S_1$ and $S_2$, which express the same proposition, $S$, with different DS ranges it is possible to conclude that $S$ can be believed with a DS range that is the intersection of the DS ranges of $S_1$ and $S_2$, since it is merely giving more accurate information as to what must be the lower and upper bounds on belief. If there is no overlap between the ranges of two otherwise identical propositions, this can be considered the same as a contradiction in propositional calculus. An inference rule can be devised to formalise this argument:

$$\frac{\begin{array}{c} S_{1[s(S_1),p(S_1)]} \\ S_{2[s(S_2),p(S_2)]} \end{array}}{S_{[\max(s(S_1),s(S_2)),\min(p(S_1),p(S_2))]}} \tag{9.4}$$

The smallest strength of belief that can be associated with a proposition, $S$, is *support*($S$). Given that the worlds are disjoint and their probabilities sum to 1, it follows that the maximum belief in $\neg S$ will be $1 -$ *support*($S$). In other words:

$$\mathit{plausibility}(\neg S) = 1 - \mathit{support}(S)$$

Similarly,

$$\mathit{support}(\neg S) = 1 - \mathit{plausibility}(S)$$

By considering the ways that a given proposition can effect the support and plausibility of other propositions, it is possible to devise inference rules that allow these other propositions to be inferred with consistent DS ranges. For example, given the inequalities concerning the constraints on $S$ and $\neg S$, the following inference rule can be constructed:

$$\frac{S_{[s,p]}}{\neg S_{[1-p,1-s]}} \tag{9.5}$$

Now consider the sentences, $A_{[s(A),p(A)]}$ and $A \vee B_{[s(A\vee B),p(A\vee B)]}$. From these, it is possible to place constraints on the DS range of $B$ alone. Since *plausibility*($A \vee B$) is calculated by considering all of the worlds in which $A \vee B$ could possibly be assumed,

[6]It is assumed here that an agent is never wrong about the state of the actual world, only possibly ignorant about it. It is also assumed that the agent is logically omniscient, understanding all of the logical consequences of his knowledge.

$B$ certainly cannot be known in more worlds than $A \vee B$. The lowest belief in $B$ can be calculated by considering the minimum number of worlds in which $A \vee B$ is known. The minimum value of $B$ will be found in the situation where $A \vee B$, the union of all the worlds that satisfy $A$ or satisfy $B$, is at a minimum and $A$ is believed in as many worlds as it consistently can be. It follows that $\mathit{support}(B) \geq \mathit{support}(A \vee B) - \mathit{plausibility}(A)$. Clearly, the belief in $B$ can never be less than 0. This argument can be summarised by the following inference rule:

$$\frac{\begin{array}{l} A \vee B_{[s(A\vee B),p(A\vee B)]} \\ A_{[s(A),p(A)]} \end{array}}{B_{[\max(0,s(A\vee B)-p(A)),p(A\vee B)]}} \tag{9.6}$$

Inference rules such as (9.6) are always obtained by considering the worst and best case situations of the agent's knowledge state. Such rules are not generally exact in the sense that the provide the most accurate possible statement about their conclusions, they merely place certain constraints on what must be true of their conclusions.

Given the standard identities and tautologies of propositional calculus and rules (9.4), (9.5) and (9.6) it is possible to derive any other necessary inference rules. Consider, for example, modus ponens. This allows $B$ to be inferred from $A$ and $A \supset B$ in the following manner. Given $A_{[s(A),p(A)]}$, it is possible to infer $\neg A_{[1-p(A),1-s(A)]}$ using (9.5). By a standard identity of propositional calculus, $A \supset B_{[s(A\supset B),p(A\supset B)]}$ is identical to $\neg A \vee B_{[s(A\supset B),p(A\supset B)]}$. Consequently, using (9.6), $B$ can be inferred from $\neg A$ and $\neg A \vee B$ with range $[max(0, s(A \supset B) - (1 - s(A))), p(A \supset B)]$. This can be simplified and summarised as:

$$\frac{\begin{array}{l} A_{[s(A),p(A)]} \\ A \supset B_{[s(A\supset B),p(A\supset B)]} \end{array}}{B_{[\max(0,s(A\supset B)+s(A)-1),p(A\supset B)]}} \tag{9.7}$$

Much as such rules allow information to be expressed about propositions which would not normally be entailed by similar theories in propositional calculus (for example, rule (9.6)), the logic is still monotonic. Adding premises to the theory only results in more conclusions.

It can also be observed that all inference rules, with the exception of (9.4) will result in increasing uncertainty, since the uncertainty of the antecedents is combined in the consequent. It is only when one or more of the antecedents is totally certain that the conclusions drawn will maintain certainty levels.

### 9.4.2 Expressing Generic Information

Given sentences such as:

$$[\forall b.\mathrm{Bird}(b) \supset \mathrm{Flies}(b)]_{[s_1,p_1]} \tag{9.8}$$

and

$$\mathrm{Bird}(\mathit{tweety})_{[s_2,p_2]}$$

it is not possible to apply modus ponens as in the previous section because the rule is expressing facts about sets of clauses, rather than a single clause. The outcome of this is that, for example, given all birds cannot fly, it still may be possible that a given single bird might be able to fly. The modus ponens rule with quantification becomes:

$$\frac{\begin{array}{l} A(c)_{[s(A(c)),p(A(c))]} \\ \forall x.A(x) \supset B(x)_{[s(\forall x.A(x)\supset B(x)),p(\forall x.A(x)\supset B(x))]} \end{array}}{B(c)_{[\max(0,s(\forall x.A(x)\supset B(x))+s(A(c))-1),1]}} \tag{9.9}$$

The support is identical to the propositional case, but nothing can be said about the plausibility.

In order to avoid the problems with interpreting modalities explained in chapter 7, it is also convenient to use the alternative notion of *schemas*. As before, these are sentences that contain unbound variables. They represent every ground clause that can be formed by instantiating them. Schemas represent facts about every constant rather than about sets of them. They are more meaningful than the quantifier representations in cases where the number of objects in the domain are not known in advance since they express information about any single object rather than the set of objects as a whole.

For example, taking the schema:

$$\mathrm{Bird}(b) \supset \mathrm{Flies}(b)_{[s_1,p_1]} \tag{9.10}$$

to represent, amongst many other sentences, the sentence:

$$\mathrm{Bird}(tweety) \supset \mathrm{Flies}(tweety)_{[s_1,p_1]}$$

and given $\mathrm{Bird}(tweety)_{[s_2,p_2]}$, it is now possible to apply (9.7) to yield:

$$\mathrm{Flies}(tweety)_{[\max(0,s_1+s_2-1),p_1]}$$

Unlike the sentence (9.8), if it is discovered that it is inconsistent to believe that a single bird can fly, a theory using (9.10) will be made inconsistent.

Clause (9.10) will yield clauses with different plausibilities to those inferred by (9.8) after modus ponens is applied. It is up to the system-builder to decide which of these schemes is the most suitable. Both systems have the rather useful property that inferences drawn from uncertain information will be increasingly uncertain.

## 9.5 Maintaining a DS Theory with an RMS

For convenience, let the logic just described be called *DS logic*, and any theory within such a logic be known as a *DS theory*. The deductive device must be modified to support DS theories and DS logic.

This modification introduces a new kind of RMS, dedicated to the handling of DS theories. In the next section, DS logic will be augmented to handle non-monotonicity, and the RMS will be revised to cope with this. The use of these logics and the RMS is not just limited to RT and Crystal, but can be applied to expert system problems.

Even a monotonic DS theory can benefit by being supported by an RMS. There are two major reasons for this:

1. As more evidence for a particular proposition is added, its consequences must be updated. Rather than recalculate the theorems from scratch, the consequences of the change need only be propagated around the germane nodes in the system. This is really an aspect of the frame problem. It is more significant with DSRs because it is more likely that new information with the same propositional content will have a new DSR, the effects of which will change other dependent DSRs.

2. The RMS can be used to revise the theory if inconsistencies are discovered.

It is a straightforward task to modify the deductive device to handle the type of proof rules used in a DS-theory. Specifically, the meta-rules must be modified to calculate new DSRs from old ones using suitable proof rules, but the inferencing algorithm remains unchanged. It remains to explain how DS theories can be combined with these components to gain the maximum advantage.

The system that will be described is as yet unimplemented, so rather than give an algorithm, I will sketch over the intended operation of the system using examples. A glossary of RMS terminology is provided in appendix C. The process I propose is a constraint-satisfaction algorithm, with the inference rules acting as constraints on the theories they describe.

### 9.5.1 Data Structures

The following alterations would have to be made to the original structures:

Nodes would have an associated DSR. A node's DSR's degree of uncertainty corresponds to how "out" the node is. An uncertainty of 1 means totally "out" or unknown, an uncertainty of 0 means totally "in". The support status attribute, originally used to indicate whether a node is "out", "needing evaluation", or "in" is still used, but in the following ways:

1. When marked with "out", it indicates that the node is currently not believed. This means that it can be ignored by the inference engine, thus speeding it up. Its main use is to mark a node as (possibly temporarily) unbelieved by the system. This might be the case because the node has been rejected to restore consistency[7]. Later on, the node might be reinstated, in which case it is simply made "in" making the previous DSR now valid. Without the support status attribute, the rejected node's DSR would have to be remembered somewhere in case it was needed again were the node to be reinstated. "out" nodes are considered to have a DSR of [0, 1], no matter what the value of their DSR attribute.

2. When marked with "nil", it is an indication to the system that it is in the process of re-evaluating the node because a new justification has been added to the system. "nil" nodes are considered as having a DSR of [0, 1] if evaluated as part of a justification. This is not strictly necessary for a monotonic system, but is necessary when defaults are introduced.

[7] In this approach to reason maintenance, non-monotonic inferencing is distinct from consistency maintenance. "out" nodes represent nodes that have been totally rejected from a theory to keep it consistent. When nodes are rejected, the RMS is not switching between extensions of a theory, but between theories themselves. It will be seen that non-monotonic DS theories are the intersections of consistent extensions. Non-monotonic theorems may be revised to be totally uncertain, but they are never rejected "out".

3. When marked as "in", it serves no purpose.

Justifications also need to be altered so that they have an associated DSR derived from the nodes in their "in" lists. They have to be re-evaluated when any of their constituent nodes change their DSR or support status. Justifications only require an "in" list. However, nodes, as opposed to justifications, have an "out" list. If any of the nodes in the "out" list are "in", the node becomes totally uncertain, or "out". "out" lists are only used for restoring consistency in a manner to be described in section 9.5.3.

To calculate the support status of a node, the "out" list is first evaluated. If it is not valid, the node is marked "out". Otherwise, the DSR of the node is calculated from all of its component justifications. Any changes to the DSR or the support status are passed on to the nodes dependent on this DSR. This approach to evaluation is different from Doyle's in two ways.

1. Evaluation is "cautious" rather than "brave". The intersection of all valid justifications is taken rather than a search being made for a single valid justification.

2. Evaluation is always similar to Doyle's "constraint relaxation" case in his algorithm, since "nil" nodes are regarded as "out".

### 9.5.2 Normal Operation

Let us now consider the operation of the new RMS as a DS theory is incrementally modified. There are two basic situations that might occur:

1. The new information is consistent with the existing theory. All that is required is that the new information is efficiently propagated to any relevant nodes. I call this *normal operation.*

2. The new information is inconsistent with the existing theory. This requires the invocation of the dependency-directed backtracking mechanism.

In this section, I address the normal operation of the RMS; in the next section I describe the dependency-directed backtracking mechanism.

The DSR of a node is calculated to be the intersection of the DSR's of its justifications. This can be calculated by progressive applications of rule (9.4). If any justification is invalid (i.e the support is greater than the plausibility), the entire node becomes invalid and dependency-directed backtracking is invoked.

When the DSR or support status of a node changes, the transitive closure of its consequences — its *repercussions* — are marked as "nil". These "nil" nodes are added to an agenda for re-evaluation. The node is then re-evaluated and subsequently the agenda is processed. The marking/re-evaluation process will terminate because, with the exception of applications of rule (9.4), rule applications never result in an increase of certainty, and never cause an extant node to become less certain. This means that any cycles in the updating process will eventually be halted because the affected nodes' DSRs do not change.

As a simple example of the updating procedure, figure 9.4 illustrates an RMS state that is built up by progressively adding nodes and making inferences from them.

To start with, $P$ is added with DSR $[0.7, 1.0]$. $P$ was not present beforehand, so a new node is constructed to represent it with the appropriate DSR. Next, $P \supset Q$ is added with

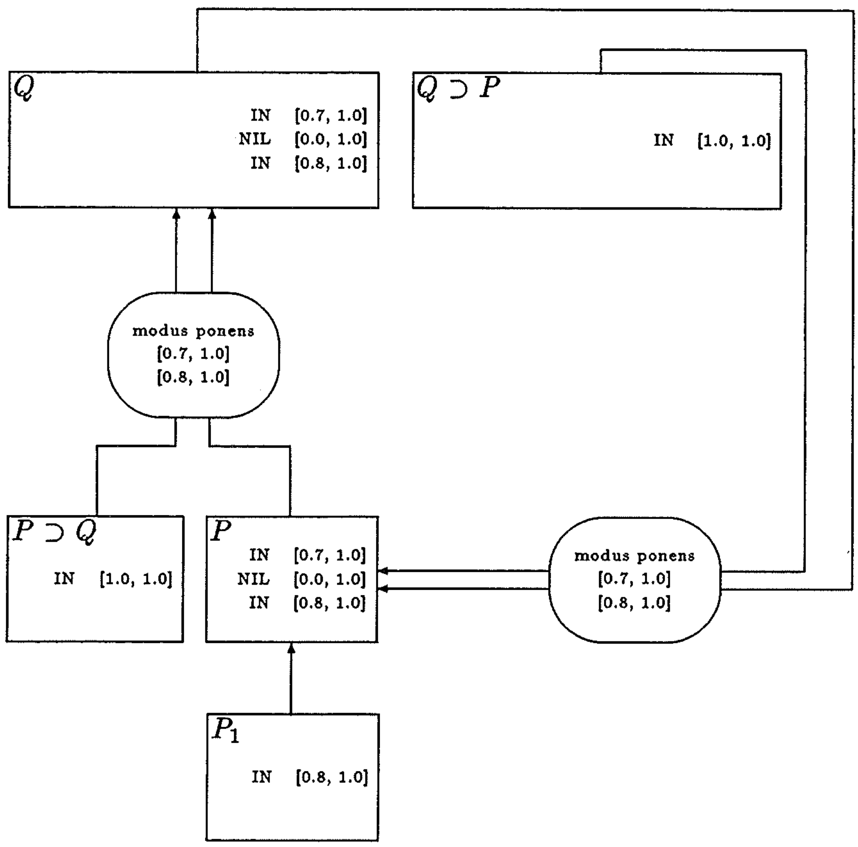



Figure 9.4: An RMS State with Circular Dependencies

total certainty. As before, a new node is constructed to represent this. However, now the rule of *modus ponens* applies to $P$ and $P \supset Q$, resulting in $Q$ being inferred with DSR [0.7, 1.0] (its justification is depicted by an oval). If the node, $Q \supset P$ is added with total certainty, $P$ can be inferred with range [0.7, 1.0] using modus ponens. This time, $P$ is already present. The new justification is added, but no overall change to $P$'s DSR occurs, so the updating process terminates.

Now consider what would happen if the DSR of $P$ is updated, say by the introduction of more evidence in the form of $P_1$[8]. $P$ is re-evaluated after its non-"nil" repercussions are marked with "nil" and added to the agenda. In this case, the only such repercussion is $Q$. When $P$ is re-evaluated, it is calculated from its two justifications. One is from $P_1$, with justification [0.8, 1.0]. The other is the modus ponens justification which was [0.7, 1.0] but has now become [0.0, 1.0] since $Q$ became "nil". The result is that $P$ is [0.8, 1.0]. The agenda is then processed. $Q$ is re-evaluated and becomes "in" with DSR [0.8, 1.0]. Its repercussion, $P$, is marked "nil" and processed again. This time, there is no change and it becomes "in".

It should be noted that, as explained above, the state represented in figure 9.4 contains cyclic structures, but the updating process still terminates.

### 9.5.3 Resolving Inconsistencies using the RMS

We can now consider how the new RMS introduced in the previous section can handle the addition of information that results in inconsistent theories using dependency-directed backtracking.

Given a node with DSR $R_1$, a new justification might be added with DSR $R_2$. Given a consistent theory, $R_1$ is updated to be $R_1 \cap R_2$. However, if the theory expressed by the RMS is inconsistent, it may be the case that $R_1 \cap R_2 = \emptyset$. If the intersection is calculated using rule (9.4), then an empty intersection will manifest itself by the resulting plausibility being less than the resulting support. Given that such an inconsistency is detected, the RMS can be invoked to attempt to restore consistency in the following way.

The justifications are traced back to establish the culprits in exactly the way as Doyle finds them. One of the culprits is selected to be rejected. This is effected by creating a "nogood" (NG) node that summarises the inconsistency. The "out" list of the node to be rejected is modified so that it contains this NG. Consequently, it is impossible for both the inconsistent nodes to be "in" simultaneously.

Given a choice of putative culprits, the most obvious choice would be the culprit which is the most uncertain. The reasoning behind this is that, generally, inference chains will grow less and less certain with increasing depth of inference. Eventually, the deepest inferences will be rejected either by the use of acceptance functions or because they are totally uncertain. Thus, inference chains that begin with the most uncertain premises will generally be the shortest. Rejecting the most uncertain premise should therefore result in minimal change in terms of both syntactic and semantic information[9]. Given two equally uncertain putative culprits, one of which has to be rejected, a random choice must be made or some other criterion used. This should be a relatively infrequent occurrence, however.

To illustrate the consistency maintenance process, a simple example will be given. I will assume that justifications only consist of single nodes so they do not need to be illustrated

[8] Since $P_1$ is merely evidence for $P$, I have not bothered drawing in the justification, since it merely contains the DSR of $P_1$.

[9] The relation of DS ranges to semantic information content will be explained in section 9.7.1.

since they can be inferred from the dependencies that are provided. An example of a situation using such justifications is the progressive addition of evidence from various sources to the RMS.

In the following diagrams, $P$ represents the combination of all the evidence, and $P_n$ a particular piece of evidence. NG $(P_i, P_j)$ represents a node indicating that $P_i$ and $P_j$ are inconsistent. This node's DSR will be a copy of the DSR of $P_j$. The uncrossed directed arcs represent a justification whose DSR is merely a copy of the DSR of the node it originates from; their "in" lists contain only this node and they have no "out" lists. The horizontally-crossed directed arcs represent parts of the "out" list of a node; each of their sources forms a node in their destination's "out" list.

Consider the stable situation depicted in figure 9.5.

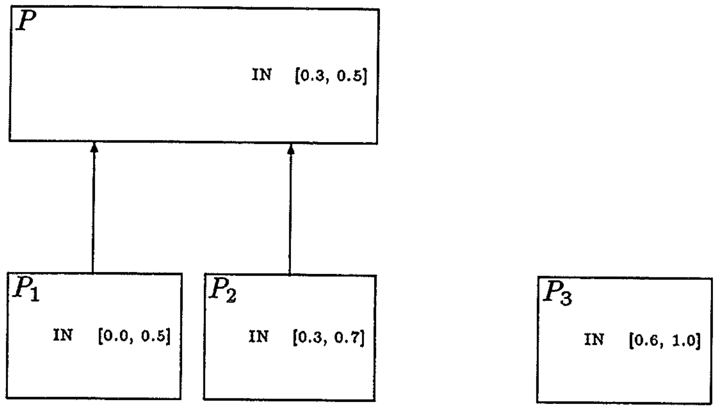



Figure 9.5: The Introduction of Node $P_3$ Results in Inconsistency

In this figure, $P$ has a DSR [0.3, 0.5], which is the intersection of the consistent DSRs of $P_1$ and $P_2$. Now consider the introduction of node $P_3$ with DSR [0.6, 1.0]. This results in an inconsistency between $P$ and $P_3$, so the RMS is invoked to resolve it.

First, the culprits must be found. In this case they are $P_1$, $P_2$ and $P_3$. The least certain of them is $P_1$, so this is chosen for rejection. Next, the appropriate nogood, NG $(P_1, P_3)$ is constructed and justified by $P_3$. This nogood is added to the "out" list of $P_1$, as shown in figure 9.6, resulting in it becoming "out".

$P_1$'s sole repercussion, $P$, is subsequently made "nil" and re-evaluated. Since $P_1$ is now out, $P$ can consistently be evaluated to be [0.6, 0.7]. The process now terminates, since it is not one of its own repercussions and it does not change.

Now consider the addition of the node $P_4$ with DSR [0.9, 1.0]. Once again, the theory becomes inconsistent (see figure 9.7).

This time, the candidate culprits are $P_2$, $P_3$ and $P_4$ ($P_1$ is "out"). $P_2$ and $P_3$ are equally uncertain, but $P_2$ is the only node inconsistent with $P_4$, which will remain "in" as it is more certain. Consequently, $P_2$ is rejected using the appropriate nogood, and $P$ is re-evaluated to be [0.9, 1.0].

Finally, consider the introduction of a further node, $P_5$, with justification [0.0, 0.0]. Again, this results in inconsistency (see figure 9.8).

The candidate culprits are $P_3$ and $P_4$. Much as $P_3$ is the least certain, in order to illustrate the full power of the dependency-dependent backtracking procedure, I will assume that $P_4$, rather than $P_3$, is first rejected. $P_4$ is rejected using the appropriate nogood, NG $(P_4, P_5)$. However, this time, the re-evaluation of the repercussions is more

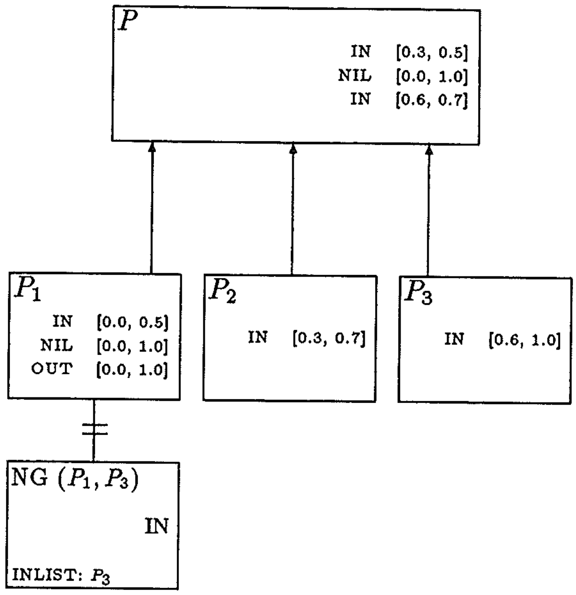



Figure 9.6: Consistency is Restored by Rejecting Node $P_1$

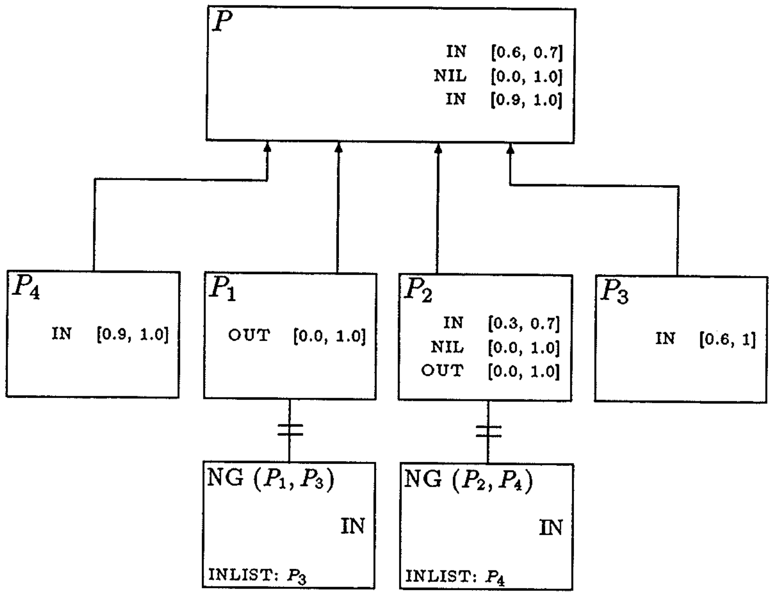



Figure 9.7: The RMS State after Node $P_4$ has been Introduced

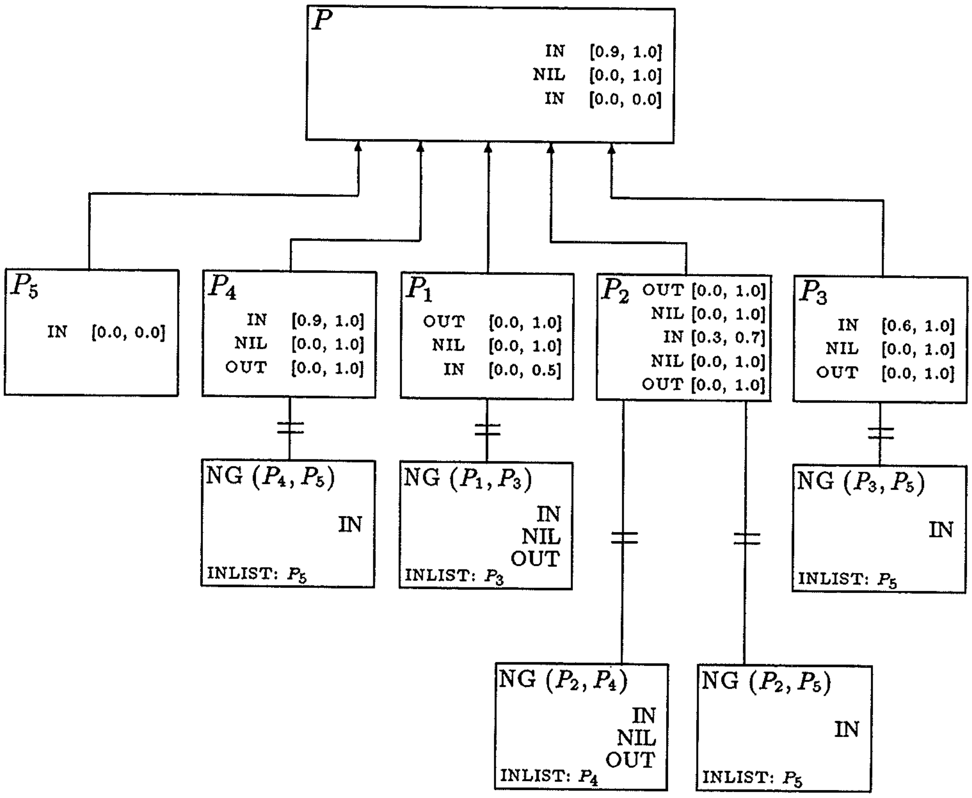



Figure 9.8: The RMS State after Node $P_5$ has been Introduced

complex. The nogood, NG $(P_2, P_4)$ is amongst the repercussions of $P_4$ and it goes "out" on re-evaluation. This results in $P_2$ being added to the agenda and re-evaluated. $P_2$ comes back "in", since the node with which it was inconsistent has gone "out". However, when $P$ is re-evaluated, another inconsistency is discovered. This time, the candidate culprits are $P_2$, $P_3$ and $P_5$. $P_2$ and $P_3$ are the most uncertain, and both are inconsistent with $P_5$. Since they are equally as uncertain, an arbitrary choice is made of the node to be rejected, in this case $P_2$. The nogood, NG $(P_2, P_5)$ is added and $P_2$ goes "out". However, once again an inconsistency is found when $P$ is re-evaluated. This time, the putative culprits are $P_3$ and $P_5$. $P_3$ is rejected, since it is the least certain. Its repercussions include NG $(P_1, P_3)$ which goes "out" when re-evaluated resulting in its repercussion, $P_1$ coming back "in". When $P$ is re-evaluated, the only nodes (excluding nogoods) that are "in", $P_1$ and $P_5$, are consistent. Thus, $P$ is evaluated as having DSR [0.0, 0.0].

It can be observed that the dependency-directed backtracking process can bring nodes back "in" when the nodes with which they were inconsistent are taken "out". The rejection process always favours the most certain single node, rather than the best combination of nodes. Thus, in this scheme evidence obtained from a single source is preferred to evidence combined from several sources.

Much as "out" nodes are regarded as having range [0, 1], the fact that they are "out" signifies that they were inconsistent with the current theory, not because they were defaults, but because the original theory was inconsistent. It will be seen that default inferences will not be handled in this way, thus making a convenient distinction between the maintenance of a consistent default theory and the maintaining of the consistency of an otherwise inconsistent theory.

We have now seen that it is possible to label a FOPC theory with DSRs. A new RMS has been proposed to handle this logic. The advantage of using such a logic is that, not only can degrees of belief be modelled, but uncertainty can be used as a principled method of determining the culprit during dependency-directed backtracking. This kind of modification was suggested by Doyle in his original paper (Doyle, 1979), and has wide application in the general area of uncertain inferencing and belief revision.

## 9.6 Adding Non-Monotonicity to DS Logic

In the previous section, I explained how DS theories can be supported by a new kind of RMS, and how the consistency of such theories might be maintained. I shall now explain how the RMS apparatus set up to handle DS theories can also be used to handle an enhanced logic that facilitates non-monotonic reasoning. I shall first describe how the logic might be augmented with a modal consistency operator, then how this might be implemented using the existing RMS.

It would be convenient if McDermott's modal $\mathcal{M}$ operator could be adapted to suit DS logic. In his original paper on nonmonotonic logic (McDermott and Doyle, 1980), he takes $\mathcal{M}P$ to mean "it is consistent to believe $P$". In propositional calculus, this amounts to the fact that $\neg P$ is not provable or believed, and evaluates to a simple truth value. In DS logic, the operator $\mathcal{M}$ must have an associated DSR. According to my intuition, this range should be the most certain belief that can consistently be believed to the highest degree. This tallies with the usual usage of defaults: default beliefs involve the reasoner jumping to some conclusion, which may be revised in the light of new evidence.

Given an existing belief $P_{[s(P),p(P)]}$, by definition the most it can be believed is $p(P)$. Therefore, the highest certainty it can be believed with is $[p(P), p(P)]$. It follows that a reasonable definition for the consistency operator might be such that:

$$\mathcal{M}(P_{[s(P),p(P)]})_{[p(P),p(P)]}$$

Given the epistemic possible-world semantics of DS theories, models of $\mathcal{M}P$ must be such that $P$ is known in all possible worlds where it would be consistent to know it.

To show how a DS default theory might be supported by an RMS, consider a theory consisting of the single default:

$$\mathcal{M}P \supset P_{[1.0,1.0]}$$

This expresses the fact that $P$ should be believed as much and as certainly as possible given the evidence. A possible implementation of this is shown in figure 9.9.

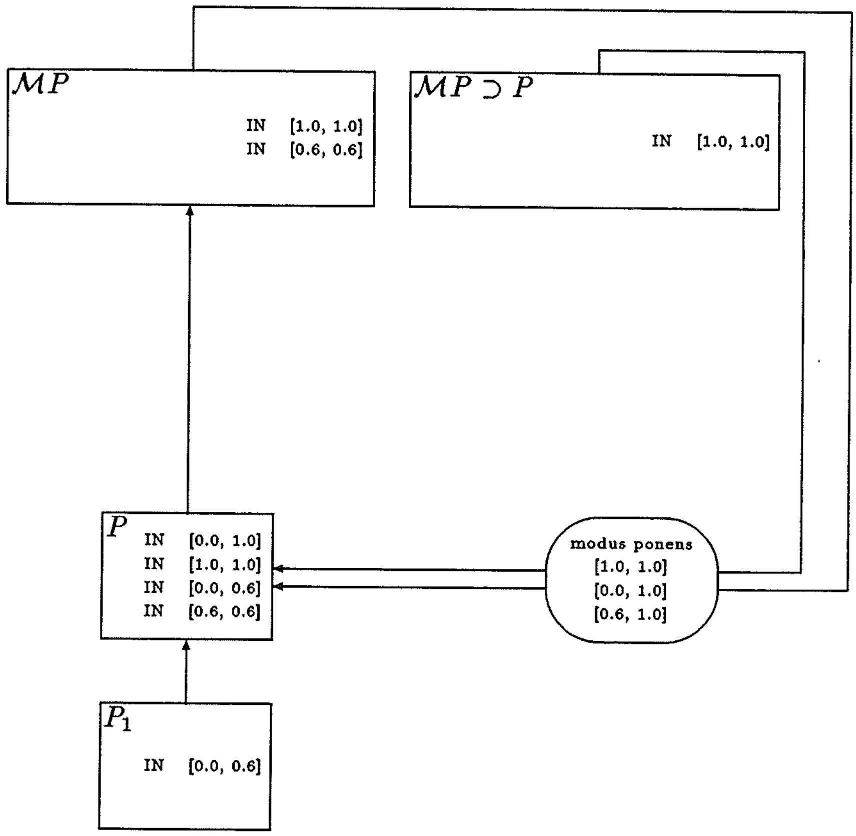



Figure 9.9: A Simple DS Default Theory

Ignoring $P_1$ for the time being, it can be noted that two extra nodes have been introduced, $\mathcal{M}P$ and $P$. I am assuming that the interface automatically adds these after noticing $\mathcal{D}P$ ($\mathcal{M}P \supset P$). Whenever the node $\mathcal{M}P$ is added, it must be justified by the node $P$. The initial state of the system will be constructed as follows. $\mathcal{M}P \supset P$ is added. The inference engine then adds $P$ with DSR [0, 1], since nothing is known about it. Next, $\mathcal{M}P$ is added, justified by $P$. Its initial value, using its definition, will be [1, 1]. Modus ponens can now apply, and $P_{[1.0,1.0]}$ can be inferred. This justification is added to $P$, which is updated, and its repercussions marked "nil". Consequently, $\mathcal{M}P$ will be re-evaluated, but will not change, terminating the updating process. This will be the stable state of the

RMS given the above theory: given no other evidence for $P$, $P$ can be assumed totally true.

Next, consider the addition of further evidence in the form of:

$$P_{[0.0,0.6]}$$

The appropriate node, $P_1$, will be constructed, and $P$ updated after its repercussions have been marked "nil". Since "nil" is interpreted as [0, 1], $P$ is re-evaluated to be [0.0, 0.6]. $\mathcal{M}P$ is re-evaluated to be [0.6, 0.6]. The modus ponens justification then changes value to [0.6, 1.0] then $P$ and its repercussion re-evaluated. $P$ is now $[0.0, 0.6] \cap [0.6, 1.0] = [0.6, 0.6]$. $\mathcal{M}P$ is re-evaluated but not altered and the process terminates. The result is that given further evidence regarding $P$ in the form of $P_1$, $P$ can now only be believed as strongly as [0.6, 0.6].

The algorithm sketched so far is rather inefficient because a node must be evaluated more than once if it changes state. An alternative means for updating the RMS might be as follows: Normally, when a justification is added to a node, if the DS range or status does not change, then the process terminates. Otherwise, after updating the node, the repercussions are marked "nil" and then processed. If a contradiction is found, it is first checked to see if the nodes causing the inconsistency are justifying a "$\mathcal{M}$" node. If so, all the repercussions are marked "nil" and the nodes re-evaluated and then the agenda processed. This may resolve the inconsistency because the default nodes will be re-evaluated purely on the basis of the existing evidence. If the contradiction is still present, the dependency-directed backtracking mechanism must be invoked.

Default rules which are not totally certain will result in the DSR's of their defeasible inferences also being uncertain. If $\mathcal{M}P \supset P$ was uncertain, it would express the fact that it is not always the case that when it is consistent to believe $P$ as much as possible that $P$ is always the case. So, for example, given no information about $P$, it would still not be possible to believe it fully.

Much as I have not proved this, I believe that just as propositional calculus is a subset of DS logic, McDermott's non-monotonic logic is a subset of DS default logic. It differs from Moore's autoepistemic logic in the sense that it is cautious rather than brave, but this is exactly what is required of a logic for the deductive device (see sections 4.5.1 and 7.4.1).

## 9.7 Applications of Non-Monotonic DS Logic

In the course of this chapter, I have developed a new non-monotonic logic with degrees of belief that can assist culprit selection during belief revision. I now show how this logic provides some insight into the solution of the problems listed in section 9.1.

### 9.7.1 Strength, Uncertainty and Information

DSR's provide a formal notion of SW's strengths. In chapter 7, I explained how change in strength could be viewed as a unified way of considering contextual effects. Given that DSR's are used to model strength, it is the change in uncertainty that is of importance. Most DSR's within the system will be of the form $[X, 1.0]$[10]. People generally hold their beliefs as strongly as possible; it is their confidence in these propositions that

[10]Negative assertions of the form [0.0, Y] can be converted into the positive form using rule 9.5.

vary (Gärdenfors and Makinson, 1988; Galliers, 1990). Thus it is generally the confidence, rather than the plausibility, of the DSR's within the deductor memory that will change. The change in confidence is a measure of the change in the number of possible worlds that will satisfy the theory (see section 9.4.1), and therefore a measure of semantic information. Consequently, contextual effects can be quantified in terms of the absolute change in semantic information within the deductor memory.

Using semantic information overcomes problems derived from dependent strengthening. If, say, conjunctive modus ponens is used to derive an assumption via two different routes (one for each conjunct), the final clause will not be strengthened more than is warranted. This solves many of the syntactic inference problems described in chapter 5. As an example, consider the following three assumptions:

$$(A \wedge B) \supset C \tag{9.11}$$

$$A \tag{9.12}$$

$$B \tag{9.13}$$

In the current version of Crystal, $C$ will be justified twice, the rules of conjunctive modus ponens will be applied to (9.11) (see appendix D) to yield $B \supset C$ and $A \supset C$. These two inferences can then be combined with (9.12) and (9.13) using modus ponens to infer $C$. $C$ is apparently inferred from two different sources, and therefore is considered to be independently strengthened twice. Unfortunately, both inference paths provide identical information about $C$, but the inference engine does not realise this.

Now consider a similar problem expressed using DS logic.

$$A \wedge B \supset C_{[1.0,1.0]} \tag{9.14}$$

$$A_{[0.7,1.0]} \tag{9.15}$$

$$B_{[0.6,1.0]} \tag{9.16}$$

The rules of conjunctive modus ponens for DS logic are identical to those for modus ponens because of the identity $(A \wedge B) \supset C \equiv A \supset (B \supset C) \equiv B \supset (A \supset C)$. Therefore, it is possible to infer $B \supset C_{[0.7,1.0]}$ using (9.14), (9.15) and (9.7). Then, again applying (9.7) to this conclusion and (9.16) we get $C_{[0.3,1.0]}$.

Reasoning along the other chain, it is possible to infer $A \supset C_{[0.6,1.0]}$ using (9.14), (9.16) and (9.7). A further application of (9.7) to this conclusion and (9.15) again yields $C_{[0.3,1.0]}$. This conclusion is merged with the extant node representing $C$ using the rule (9.4) with no change. Thus, $C$ is is not unnecessarily strengthened by the second inference strand because it provides no new semantic information.

### 9.7.2 Utility, Salience and Inference Explosions

An investigation of salience and utility were beyond the scope of the project. However, I shall make a few speculative comments.

- Uncertainty is a major contributing factor to utility. A totally uncertain assumption is clearly of no use. A certain assumption can result in potentially large increases in semantic information via its derived inferences.

- Uncertainty should therefore contribute to salience. A certain assumption should generally be more prominent than an uncertain one. DSRs could, perhaps, be integrated into a system like that of Alshawi (1987) in order to achieve this.

- Some of SW's elimination rules need not be so restricted. Most rules now result in an increase in uncertainty. These rules need not be elimination rules so long as their premises are also uncertain, since uncertainty is cumulative. The interface can employ acceptance functions that reject assumptions that exceed a particular uncertainty threshold.

### 9.7.3 Extension and Theory Selection

In chapters 7 and 8 it was seen how somewhat ad-hoc local orderings of encyclopædic assumptions along with the processing strategy of the inference engine determined which extension was selected. It was mentioned that the orderings were effectively determining mental prominence of the defaults and scripts involved. A more serious salience measure based on notions such as uncertainty and temporal proximity could give these orderings a better theoretical justification.

Utilising belief ranges would make Crystal even more similar to the programs described in chapter 8. Taking TACITUS, and Wimp as typical examples of abductive language interpretation programs and comparing them with Crystal, it can be seen that all three employ: deductive inferencing, where possible; non-monotonic inferencing to fill in the gaps; belief/utility measures to restrict the search space given several candidate extensions.

In general, utterance interpretation can be viewed as an attempt to prove (as shown in chapter 8):

$$\vdash C \supset \mathrm{Enrich}(U)$$

"Enrich" is a function which takes an utterance, $U$, and returns an enriched utterance where the references are resolved, implicit relations made explicit and vague relations reified. This assumes that there is a single, complete development of the utterance. Non-determinism could be introduced by making the "Enrich" function a relation.

$$\vdash (C \supset U) \wedge \mathrm{Enriches}(C, U)$$

where "Enriches" is a predicate that states whether a context has enriched an utterance sufficiently. This second formulation has the advantage that the exact degree of enrichment need not be specified. There is not a single, "correct" enrichment: a point which I have argued in chapter 8 and which SW seem to argue when discussing non-determinacy of certain utterances.

As explained in section 8.2.4, calculation of $C$ can be divided into two phases:

1. Determination of possible $C$s using suitable inference techniques, yielding multiple extensions.

2. Selection of a particular extension.

To give some examples:

Wimp has no notion of development. It tries to prove the logical representation of the utterance by matching to scripts and frames represented in Frail. If the match cannot be proved deductively, matching is done abductively. With Wimp2, the possible matches are represented using the ATMS. The most likely extension is selected using probabilities. A more deterministic search for $C$ using probabilities is performed with Wimp3.

TACITUS takes the functional view of enrichment[11] on how the logical form is proved; different extensions can be obtained by starting with different sub-clauses of $U$. The inference engine attempts to prove the enriched form of the utterance. $C$ contains all of the information required for this proof. If the proof cannot be carried through deductively, the necessary assumptions are made. Much as there are several candidates, the search becomes deterministic through the use of costs and weights.

Crystal takes the relational view of enrichment. The utterance is assumed true, and $C$ consists of connections from the utterance and encyclopædic information about the utterance to previous values of $C$. Candidate $C$s are distinguished on the basis of the change in semantic content from previous $C$s against the effort required to derive the new $C$. Thus, a major difference between RT and the other systems is that extension selection is explicitly determined by considering previous extensions.

Less obviously related systems can also be put in this abduction/enrichment framework. For example, systems such as those of Gärdenfors and Makinson (1988) and Galliers (1990) are only interested in the process of selection between candidate extensions using notions of minimal change and coherence.

A more formal way of comparing all of these systems would be to put them into something like the preference framework in Shoham (1987). This places a partial order on models of particular logics, the ordering expressing the fact that one set of models is preferred over another. The actual model will be the one most preferred. This framework could be used to model both non-monotonic inferencing and selection between alternative extensions using notions such as utility or probability simultaneously[12].

## 9.8 Conclusion

In this chapter, I have proposed a new non-monotonic logic of belief which would give Crystal and the ideal Relevance-based system schema a formal basis. The logic has general application in the area of belief revision, but can also be used to precisely express the notion of change in semantic information content.

In chapter 6, I suggested that a Relevance measure depends on the value of:

$$\mathrm{Inf}(n) = \sum_{i=1}^{n} |\delta inf_i|$$

where $\delta inf_i$ was the change in significant information. However, in chapter 5, it was seen that measuring information change on a purely syntactic basis resulted in some problems for RT which are difficult to resolve. However, if $\delta inf_i$ is treated as change in semantic information, then many of these problems disappear; there is also the additional benefit

[11]It might be argued that TACITUS adopts the relational view because different extensions can be reached by changing the order of proof of distinct parts of the utterance representation. However, it is not clear that this is a desirable feature in the system, since it is not fully understood what determines the selection of a particular extension or whether this apparent non-determinacy of interpretation is necessary.

[12]At the time of writing, Appelt (1990) has used these notions to provide a semantics for some aspects of TACITUS' knowledge representation. This semantics provides no justification for the weights themselves, only for the general behaviour of Stickel's inference engine.

that contextual effects can be placed in a unifying framework. A contextual effect is a significant change in semantic information; non-monotonic DS logic can be used to measure such change in semantic information. This is realised by calculating the overall change in confidence of the DSRs in the deductive device (ignoring the direction of change).

This new approach to Relevance makes it clear how RT differs from other computational approaches to utterance comprehension. RT essentially claims that we attempt to maximise the significant changes in semantic information between the contexts invoked by successive utterances for the minimum amount of effort. Most other systems do not attempt to provide a theory which relates successive contexts in this way.

The deductive device can be adapted to handle non-monotonic DS logic by modifying the meta-rules to compute new DS ranges and by substituting the new RMS mechanism described in this chapter for the one described in chapter 7.

DS logic should also facilitate the building of a more sophisticated context model that incorporates notions of salience, because DSRs are contributing factors in the calculation of salience and utility measures.

# Chapter 10

# Concluding Evaluation

The research described in this thesis has been directed towards the development of the ideas constituting RT to the point where they can be computationally applied, and with building an initial, experimental implementation, Crystal. The first section of this conclusion summarises and evaluates the tests carried out with the Crystal system. The second attempts to evaluate the research as a whole, from the specific point of view of its motivation in SW's RT, and more generally, as a theory of computational discourse processing and the knowledge representation and reasoning systems this requires.

## 10.1 Testing Crystal

As mentioned earlier, tests with Crystal are performed via an interactive, menu-driven interface. This is illustrated in appendix A. Crystal has a vocabulary of around 150 words, around 20 of which represent the meaning of verbs. I have not performed extensive tests on the entire vocabulary, but have developed particular examples that are intended to demonstrate some of the key concepts in this thesis.

In appendix B, I provide a detailed example of the processing of a short discourse using both the "simple implication" and "effort-based" approaches described in chapter 6. Appendix E contains a further 28 annotated examples, that illustrate various features of Crystal's processing discussed in the course of the thesis.

### 10.1.1 Evaluating the Tests

Tests were carried out as follows. A trial discourse was set up in an utterance browser (see appendix A). Each utterance representation in this discourse was processed in turn, and the contents of the deductive device examined. In this examination, I checked that the assumptions in the deductor memory were reasonable given the utterance and context (see section 5.2.3). The easiest assumptions to check were the ones representing the sense of the utterance finally selected and the non-demonstrative inferences concerning reference resolution. So, for example, given the following two utterances in sequence:

Squirrel Nutkin felt hungry. (169)

He found a nut. (170)

it is reasonable that the "nut" is the fruit, rather than the metal artefact, and that "he" is Nutkin. These readings can be checked by examining the relevant forms. However, in the case of sense disambiguation, it must be checked *why* this sense has been chosen. Of course, according to RT, the answer is simply that the preferred sense results in the most contextual implications for the least effort. However, in order to ensure that the information counts are plausible, the contextual implications that contribute to this measure must be checked to see if they are reasonable. The significant contextual implications are found by ignoring the "instantiations" or simple implications in the deductive device and examining those that remain. These remaining assumptions will consist of the abductive inferences and the inferences derived from instantiations. For example, the (non-simple) contextual implications in the discourse fragment above that are critical for choosing the appropriate sense are the ones that predict that Nutkin may eat the nut, which seems reasonable. It can be seen in appendix A that the "metal artefact" reading for (170) does not yield as many (non-simple) contextual implications. This can be informally summarised by stating that with the "fruit" reading, the "nut" can be incorporated into a script-like assumption concerning hunger, whereas with the "artefact" reading it cannot. These are the kind of annotations that I have used in appendix E.

I should emphasise that the examples are limited in many ways. Where there are sense ambiguities, there are usually no more than two, and these are clearly distinct. Also, the sequences of utterance representations are no more than discourse *fragments*; if the sample discourses were to be significantly extended, Crystal's abductive unification and reason maintenance processes would become unacceptably slow[1]. Thus, the examples provided in appendix B and E should be regarded as illustrative rather than demonstrative.

## 10.2 Evaluation of Crystal and RT

I stated at the beginning of this thesis that my intention was to investigate whether a computational utterance processing system based on RT was possible, and if so whether it offered any fresh insights into the area of computational language processing. In this section, I consider the extent to which I have achieved these goals, review some of the significant points raised in the thesis and provide some pointers to future work on RT.

### 10.2.1 Is Crystal a Relevance-Based System?

I believe Crystal is essentially based on RT. This is perhaps best left to the reader to decide, but let me argue my reasons for believing so.

I have developed a schema for what I believe to be an ideal Relevance-based language processing system. I have postulated some extra processes, but these are only there to support some of the functionality that SW assume is present, like belief revision and abduction.

Crystal is a very basic instantiation of this schema, supporting a very small subset of its functionality. Whenever I have diverged from the ideal schema and algorithms, I have attempted to justify the alterations and have argued that they maintain the spirit, if not the exact word of RT.

To amplify these summary claims, it is necessary to say what the "spirit of RT" is. I take it to be the following:

1. Systems that attempt pragmatic processing should endeavour to adopt a cognitive approach, because natural language can only be understood in human terms.

[1]A solution would be an implementation of the rejection mechanism I discuss in section 2.6.

2. Language users distinguish consistent candidate utterance interpretations by considering their Relevance. The most Relevant interpretation will offer the maximal significant changes in semantic information between the immediately previous context and the current context for the minimal effort expended in obtaining it.

   Language users tend to have similar mental architectures and apply similar heuristics to direct the search for the best interpretation[2]. An utterance must be phrased so that people can take advantage of these heuristics.

3. Gricean conversational implicatures can be calculated in terms of SW's notions of implicature. The derivation uses the Principle of Relevance along with certain assumptions about mental organisation.

4. Utterance representations are made more concrete using a process of enrichment. Thus utterance interpretation not only modifies the context, but also modifies the utterance representation itself.

5. The semantics of certain utterances can contain procedural directives concerning the manipulation of the context. Many phenomena traditionally regarded as conventional implicature can be handled this way.

The detail in my development of RT is not only compatible with the the above ideas, but often makes them more concrete. I have reorganised SW's descriptions of mental architecture to make them correspond more closely to Fodor's modular mind and built a simple implementation which includes a conceptual memory, context expansion process and deductive device. I have described a logic and inference engine which implements the notions of assumption schemas and instantiation. I have used autoepistemic logic supported by an RMS to make the details of belief revision more specific. The same logic can conveniently describe certain semantic directives on pragmatic processing. I have employed abductive unification, a method for producing the non-demonstrative inferences vital for language interpretation according to RT. I have suggested a Relevance measure and quantified notions of effort in a manner that overcomes some of the apparent problems with the theory. Finally, I have suggested a new knowledge representation and associated RMS that could be integrated into the deductive device in order to quantify degrees of belief as well as mental representations of states of affairs. The overall significant change in belief during utterance processing would be a good measure of change in semantic information.

### 10.2.2 Does RT Have Anything to Offer?

RT can be considered from two different viewpoints:

1. As an alternative to a Gricean-based language interpreting system.

2. As an alternative to other computational language interpreting systems.

In relation to the first point, no computational language interpreting system that I know of works by overtly implementing all of Grice's maxims. Similarly, Crystal is not a complete implementation of RT. However, it seems that Crystal has far more potential for

[2]These shared heuristics may be a consequence of their mental architectures

expansion. Many deficiencies are due to its rather naïve implementation of the modules in the ideal schema. I feel optimistic about the possibility of better implementations. The same cannot be said for a Gricean approach; it appears that the maxims are inherently non-computable (see chapter 2).

As for the second point, I have shown in chapters 8 and 9 that Crystal has much in common with other abductive language processing systems like TACITUS and Wimp. All of these offer a non-Gricean approach to pragmatics. However, the crucial difference between Crystal and the other systems is that it concentrates more on the relationship between the current context and previous contexts in order to determine the utterance interpretation.

Crystal has the additional advantage that it can import Relevance-theoretic linguistic ideas and straightforwardly implement them. For example, the interpretation of the words "before" and "but" (see chapter 7).

### 10.2.3 Is RT of Any Value to the AI Community?

I hope to have convinced the reader that a concrete interpretation of RT does make an interesting contribution to the field of NLP. RT can hardly be considered to be a universal panacea to all problems in this area, however it does offer:

1. An explicit representation of changing contexts during language processing. This approach may prove to be more fruitful than the currently available alternatives, such as DRT (Kamp, 1981).

2. A unified approach to pragmatics that is simpler and more elegant than the Gricean maxims.

3. A computational approach to linguistics. This could hopefully result in more productive exchanges between artificial intelligence researchers and linguists.

Against RT are the points that:

1. Although RT is supposed to be a cognitive theory, it is not backed up with any hard evidence. For example, SW's conjectures concerning the structure of the deductive device and its associated meta-rules are not supported by any experimental psychological facts. Indeed, in Johnson-Laird (1983), evidence is provided which suggests that, in some circumstances, elimination-rules are not utilised (although neither are introduction rules). This does not have incredibly damaging consequences for RT, but the theory would be provided with considerably more support if it could be reconciled with the available psychological and psycholinguistic evidence.

2. It is underspecified and apparently self-contradictory in places. This has been overcome, to some extent, in this thesis.

3. RT (and Gricean pragmatics) completely ignores how an individual's goals effect utterance interpretation. I have speculated on some solutions to this in chapter 6.

4. RT claims to be a completely original theory of pragmatics, yet it is strikingly similar to some extant work, for example Horn (1984) and Levinson (1985). These articles detail a theory like Relevance, balancing information derived using a rule known as the Q principle against a principle of least effort known as the R principle. In this

thesis, I have avoided making any judgements concerning the status and adequacy of RT as a specifically linguistic theory. It would be useful if SW could relate their theory to other radical pragmatic theories.

### 10.2.4 Future Work

Whether or not Crystal is taken to be an implementation of a Relevance-based system, I envisage future work could be usefully invested in developing Crystal's approach to discourse processing in the following ways:

1. Developing a better context model based on notions of salience and strength. I hope to have begun this in chapter 9.

2. Developing a better model of the structuring and accessing of conceptual information.

3. Attempting to discover exactly which processes consume effort and to what extent. In this thesis, I merely made some basic conjectures in order to obtain reasonable results in the simple domain of Crystal's encyclopædic memory.

4. Investigating what constitutes non-trivial implications, as I do not find SW's definitions entirely convincing. This endeavour might benefit from psychological and psycholinguistic evidence.

5. Developing a logic for the deductive device which has a fuller coverage. Such a logic could properly handle beliefs, desires, time etc.

6. Discovering exactly how an individual's goals in a discourse are handled by RT.

7. Attempting to import more linguistic examples into the Relevance framework. This would involve the search for more semantic directives for handling words such as "therefore" and "consequently" and methods for detecting and handling indirect speech acts, tropes, etc.

8. Clearly, to complete the RT view of language processing, RT must also provide a model of language generation. This is a topic that I have not addressed at all in this thesis.

# Appendix A

# A User Level Description of Crystal

## A.1 The User-Level Perspective

The user-level perspective of Crystal is summarised in figure A.1.

The system is built in a windowing environment using menu-driven editors and browsers. Browsers are special kinds of windows that can be considered to be viewers on segments of a database. The browser window contains rows of information. Operations can be performed on the database by selecting the relevant rows in the browser window and using menus to operate on the selected rows.

The system is divided into five main components:

**Controller Browser:** This is a single browser from which the rest of the system can be controlled. It contains various menus and a browser onto the encyclopædic memory.

**Concept Browser:** This is a browser and editor on a single encyclopædic entry.

**Utterance Browser:** This allows for the editing and ordering of a stream of utterance representations.

**Sort Editor:** This allows for the graphical construction of a sort hierarchy.

**Deductor Browser:** This is a browser on the contents of the deductor memory and the reason maintenance system. Further browsers, representing answers to queries, can be summoned from this browser.

Each of these will be discussed in turn.

### A.1.1 The Controller Browser

The controller Browser is depicted in figure A.2. The main viewing area of the browser represents information about entries on the encyclopædic memory. The first three columns contain information about the concept name, arity and category. The final two columns explain the location, and viewing status (ie. being viewed or not) of the encyclopædic information.

The combination of concept name and arity provide a unique key that can be used access encyclopædic entries. This information is determined from the utterance representations or assumptions in the deductor memory during context expansion.

The right-hand menu allows manipulation of the lexicon and also provides access to other system editors and browsers. The menu items are as follows:

- *create, modify, delete, undelete* provide basic control over the lexical entries.
  - *create* adds a new lexical entry to the browser, and creates a concept browser for this entry. Entries of type "utterance" automatically invoke utterance browsers, rather than concept browsers.
  - *modify* allows renaming or modification of the parameters associated with an encyclopædic entry. Renaming an entry also renames all of the appropriate predicates within that entry.

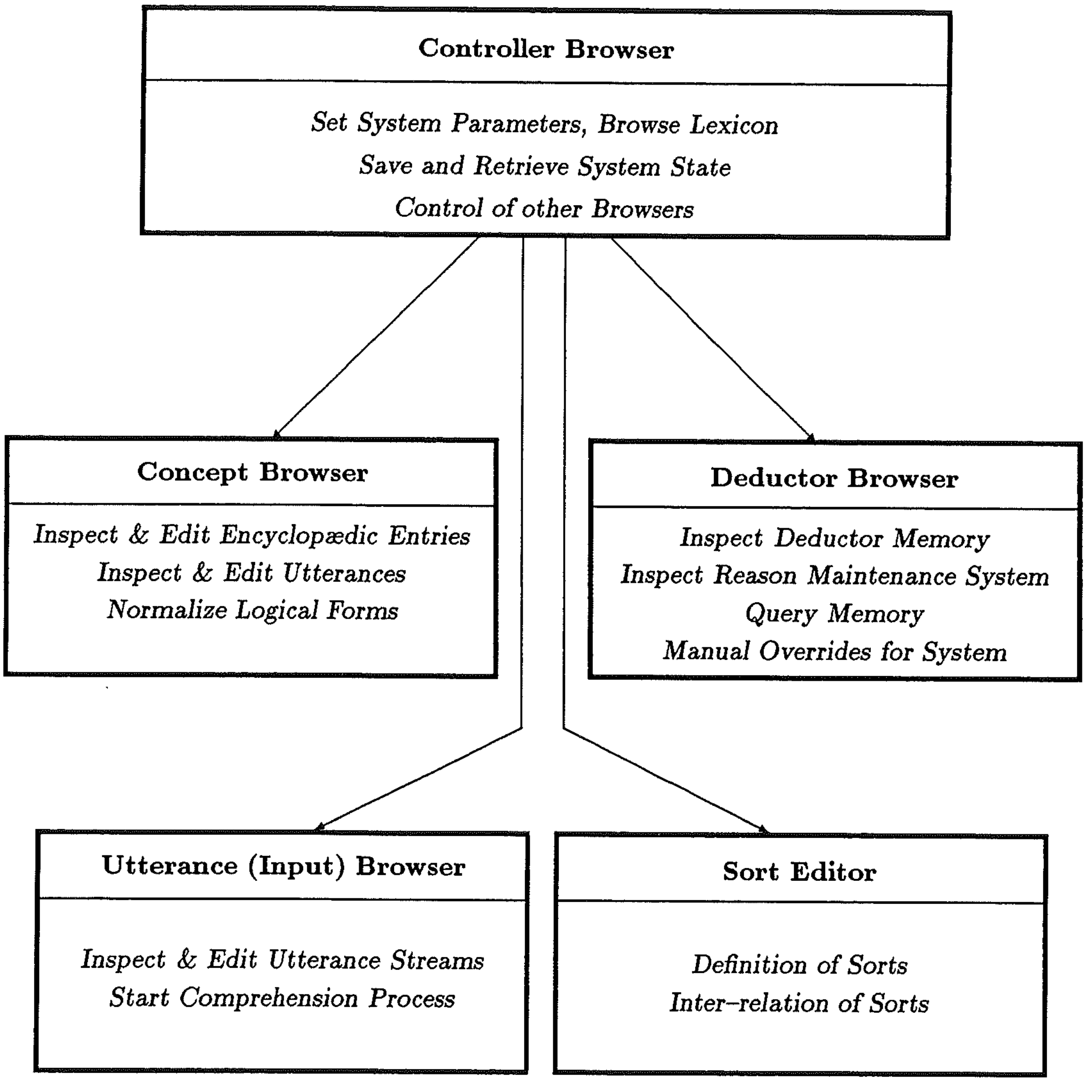



Figure A.1: User Perspective of Crystal

- *delete, undelete* allow entries to be marked or unmarked for (temporary) deletion from the browser.
- *expunge* removes all entries marked for deletion from the browser.

- *erase, dump, undump* provide facilities for saving and erasing encyclopædic information maintained on backing storage.
  - *erase* erases an entry from backing storage.
  - *dump, undump* save/load an encyclopædic entry to/from backing storage.
- *deductor* invokes the browser on the deductive device, and *load* downloads encyclopædic information into the deductor memory.
- *find* and *view* respectively locate and create editor/browsers on particular encyclopædic entries.
- *edit sorts* invokes the sort editor.
- *cleanup* and *exit* respectively dump any unsaved information (including the current system state) and exit the system. *exit* automatically performs *cleanup*.

The lower menu toggles the states of various system flags:

- *print format* indicates whether or not sorts are to be printed.
- *sorts* toggles the use of sorts.
- *abduction* toggles the use of abduction.
- *autoloading* controls whether encyclopædic entries should be automatically loaded from backing storage when required by the context expansion process.

The top window displays status and error messages. Another window can appear above this which can accept typed input when necessary.

### A.1.2 Concept Browsers

A typical concept browser is shown in figure A.3.

The browser has two display modes, showing either "original" or "converted" clauses. Original clauses are as initially input (after macro expansion) and converted clauses are the normalised clauses that are actually used by the deductive device.

Each row of the main display area contains a single assumption or assumption schema. The right-hand menu provides the following operations:

- *display clauses* Either redraws the browser or sets the display mode.
- *add clause* creates a window in which to input a new assumption. The assumption is input in Cambridge Polish notation.
- *edit clause* allows the original or converted clause to be edited in a separate window, in Cambridge Polish form, using a line editor.
- *rename* allows variables, functions, constants and identifiers to be selectively renamed.
- *delete, undelete, expunge* allows clauses to be marked or unmarked for deletion and ultimately removed from the browser, if so desired.
- *dump clauses* saves the entire entry to backing storage.
- *undump clauses* appends the contents of another encyclopædic entry onto the current one.
- *exit* removes the concept browser from the screen.

The top window is an error/status message window. The clause input/editing window appears above this window when necessary.

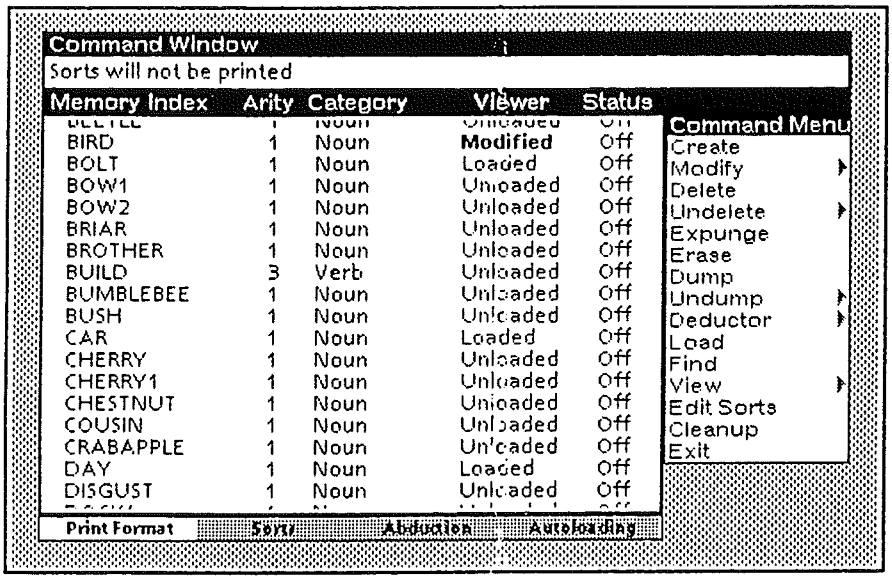



Figure A.2: The Controller Browser

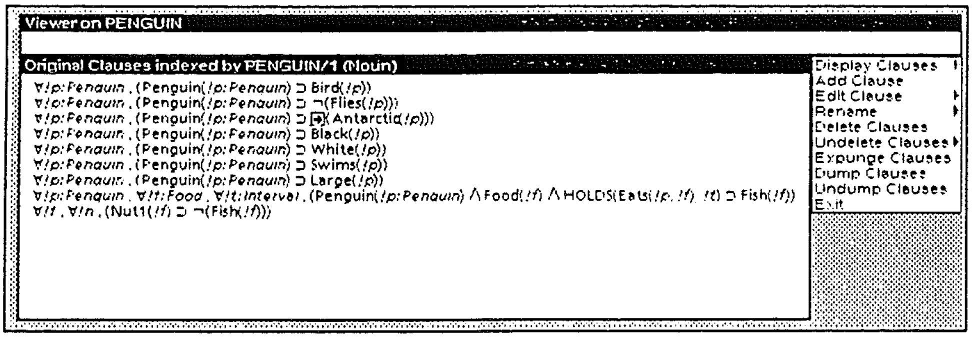



Figure A.3: A Concept Browser for "Penguin"

### A.1.3 Utterance Browser

An utterance browser provides all of the facilities of a concept browser, but has an additional bottom menu, as shown in figure A.4. Each row of the browser contains successive utterance representations; all of the rows constitute a discourse.

The bottom menu supports the following functions:

- *log* controls whether a log is to be kept of the current processing sequence. This log can be opened, closed or displayed. It records the various senses tried out, the abductive guesses asserted, and the Relevance of each interpretation.
- *copy, move* allow individual utterance representations to be copied or moved from one location to another using the mouse.
- *deductor* invokes the deductor browser.
- *clear deductor* clears the deductor memory.
- *process sentence* processes an individual utterance representation according to the algorithm outlined in chapter 4.
- *automatic* processes a sequence of utterance representations.

### A.1.4 Sort Editor

The sort editor maintains the sort hierarchy described in section 5.4. The editor has a main display window which illustrates the hierarchy and a menu which provides some basic functions on the hierarchy. An additional pop-up menu can be invoked to provide some extra functions.

The sort editor is shown in figure A.5. The vertices in the main window represent sorts, the arcs constitute the subsumption relation.

The pop-up menu provides the following functions:

- *add sort*: this adds a new sort to the hierarchy. The user is prompted for the parents of the new sort in a separate window. The sort is then checked for conformance to the rules described in section 5.4.4. If the sort is acceptable, it is added to the hierarchy at the current mouse position. Arcs from the parents to the new sort indicate its relative position in the hierarchy.
- *delete sort* removes a sort and all relations to that sort from the sort hierarchy.
- *add link* adds a subsumption link between two nodes (if it conforms to the rules).
- *delete link* removes a link from the subsumption hierarchy.
- *move node* allows a vertex and all associated arcs to be moved from one position to another.

The right-hand menu provides the following functions:

- *clear hierarchy* maintains the graph, but removes all knowledge of the sortal information from the unifier.
- *set hierarchy* provides the unifier with knowledge of the hierarchy.
- *save hierarchy* saves the hierarchy to backing storage.
- *load hierarchy* loads the hierarchy from backing storage.
- *reconstruct hierarchy* reorganises the hierarchy to provide a less tangled display, if possible.
- *quit* removes the sort editor from the screen.

The top window is an error/status window.

### A.1.5 Deductor Browser

The deductor browser is illustrated in figure A.6. The main display window presents the contents of the deductor memory and the significant aspects of its supporting RMS. There are two display modes: one where each row contains the name of each node and its propositional content, the other (illustrated) additionally shows parts of each node's justification list and consequences. Nodes that are "out" are displayed with a line crossed through the appropriate row.

The top menu includes facitilies for manually clearing the memory, taking the inference closure, highlighting the implicatures, and inputting an utterance representation.

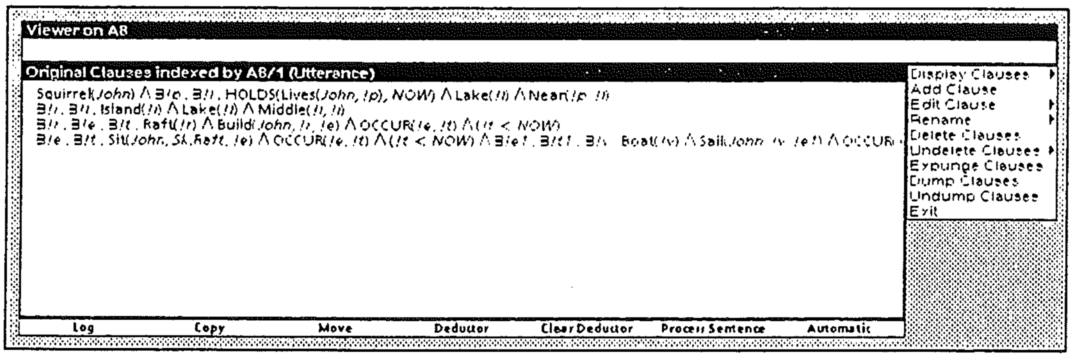



Figure A.4: An Utterance Browser

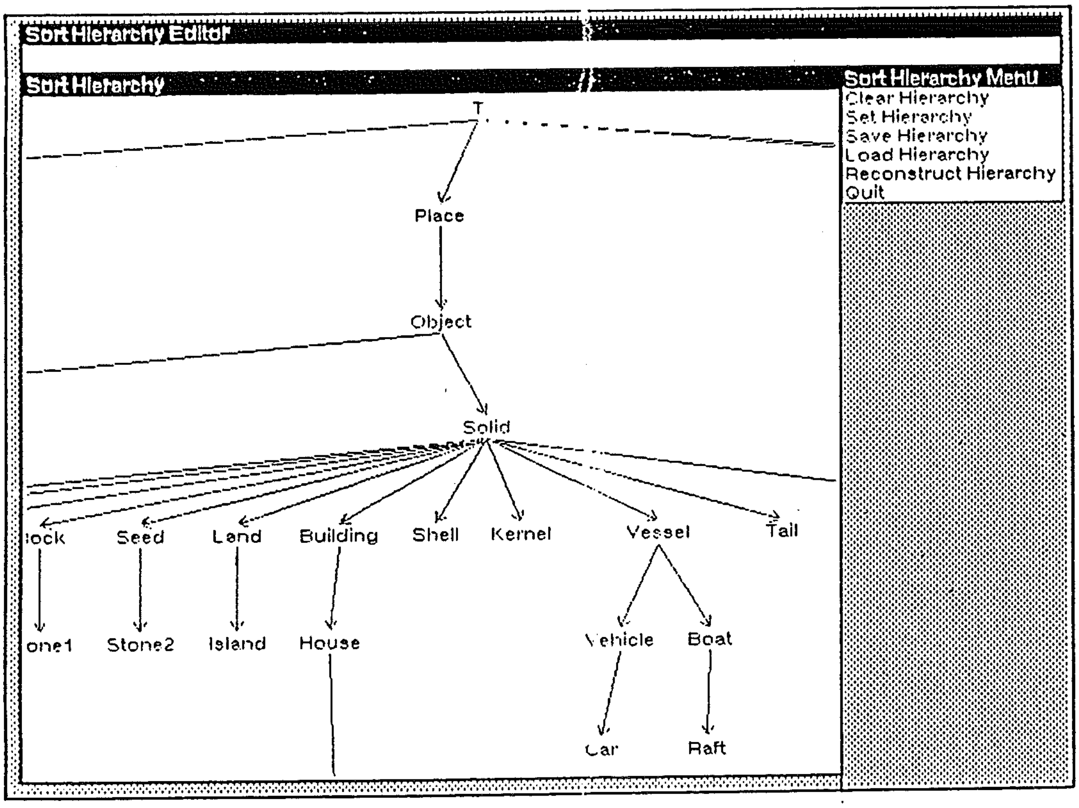



Figure A.5: The Sort Editor

Above this menu is a status display, which maintains running count of significant system variables, such as Relevance, information, effort and number of expansions.

The lower menu, entitled "additional items" provides various querying facitilies. Queries are input as terms or assumptions; they are implemented as a simple scan of the database to find a match (with the optional assistance of the abduction process). Answers to queries are either in the form of graphs that illustrate how a node was derived, or in the form of a list of items which match the query term.

A typical graph is shown in figure A.7. Vertices in the graph represent RMS nodes, arcs represent justifications. Reverse field vertices are "out" nodes. Smooth arcs represent parts of an "in" list, dashed lines represent part of an "out" list. Any vertex can be queried to provide information about its justifications and content. Graphs are particularly useful for illustrating long derivations and illustrating connectivity between utterances. It was apparent from inspection of the graphs that many of the connections between successive utterances were non-demonstrative (i.e. connections to frame or script-like assumptions or definite references).

Other querying facilities include the invocation of a system inspector on an RMS node to examine it in more detail, and a browser on the equivalences present in the equality sub-system. This browser optionally indicates which definite references are bound or unbound.

The menu entitled "RMS state functions" provides some simple facilities for manipulating the interpretation queue using simple stack operations.

The menu entitled "summary" provides facilities for creating summaries. These are lists of significant items stored in the deductor memory. The summaries generally contain the root and leaf nodes of the "forest" contained in the deductor memory, together with abductive assertions and default inferences.

The RHS menu allows certain system parameters, such as the minimum Relevance threshold, to be changed.

## A.2 The Implementation of Crystal

Crystal is written in Interlisp on a Xerox Daybreak. It consists of 17 modules[1] that provide the following functionality:

1. *Browsers:* control of system browsers.
2. *Contradiction:* performs dependency-directed backtracking.
3. *Controller:* supports the controller browser.
4. *Deductor:* the inference controller.
5. *Equality:* the equality sub-system.
6. *Normalise:* the normalisation process.
7. *Query:* provides querying functions.
8. *Relevance:* the main control process.
9. *RMS:* the reason maintenance process.
10. *Setup:* provides initialisation functions.
11. *Sorts:* supports the sort editor and provides the "Subsumes" and "Compatible" functions.
12. *Interface:* the interface process.
13. *Unify:* the abductive unifier.
14. *Utterance:* adds extra functionality to concept browsers to make them utterance browsers.
15. *Viewer1:* supports concept browsers.
16. *Viewer2:* supports concept browsers.
17. *Windows:* extra windowing functions.

The functions occupy around 0.5 Mbytes of memory. However, the core of the system, ignoring browsers, windows and querying functions is much smaller.

The lexicon is stored in 150 separate files (one for every word sense), each containing around eight assumptions and assumption schemas.

[1]These are just files of functions and other data, but I have treated them as if they were strict modules.

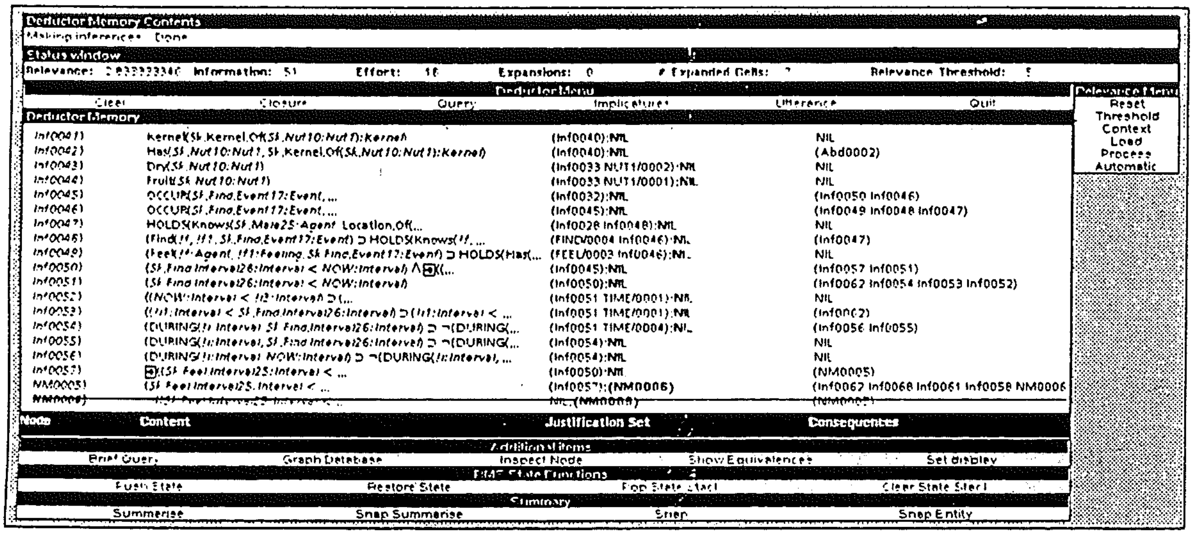



Figure A.6: The Deductor Browser

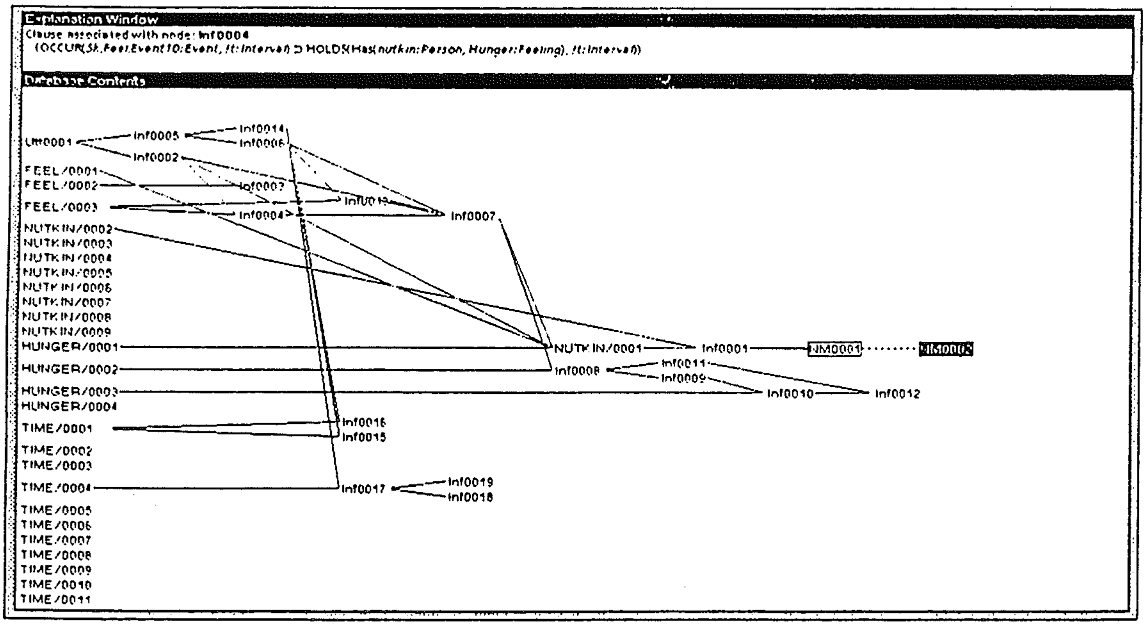



Figure A.7: A Graph of RMS Nodes Supporting the Deductor Memory

# Appendix B

# A Detailed Example

A very simple detailed example is provided here in order to demonstrate the working of Crystal and point out some salient features that have been discussed in the course of this thesis. The discourse fragment in this example is similar to that used in chapters 1, 3 and 4.

I begin by considering the following two utterances, but will subsequently extend and modify them:

1. Nutkin felt hungry.
2. He found a nut.

The second utterance contains an example of lexical ambiguity with the word "nut", which has two senses: Nut1 and Nut2. Nut1 concerns the fruit, Nut2 concerns the metal object used with bolts. It also contains a simple definite reference that must be resolved.

As explained in chapter 4, such ambiguities are handled automatically. I will provide details of the output for the first utterance and both senses of the second utterance. For each case I will show the different means for calculating Relevance discussed in chapter 5. I will also illustrate some of the difficulties encountered with the current implementation of Crystal.

When I illustrate the contents of an encyclopædic entry or the deductor memory, the order in which the assumptions are displayed on the page is the order in which they are found in the system. All of the terms in used in Crystal's knowledge representation are sorted, however I have omitted the sortal information from this appendix for perspicuity.

## B.1 First Utterance

Initially, the deductor memory will be empty. The only information present is the set of meta-rules in the logical memory (shown in appendix D).

Processing begins with the input of the logical form of the first utterance into the deductive device. This will be marked with contextual status "Utterance".

$$\begin{aligned}&\mathrm{Feel}(nutkin, Hunger, Sk.Feel.Event10) \wedge \\ &\quad \mathrm{OCCUR}(Sk.Feel.Event10, Sk.Feel.Interval25) \wedge \\ &\quad (Sk.Feel.Interval25 < NOW)\end{aligned} \tag{B.1}$$

After this, the necessary encyclopædic information concerning the concepts in the utterance is loaded (but skölem constants and functions cannot have encyclopædic entries). The assumptions loaded from memory are all assigned a contextual status of "Memory". The assumptions are loaded in the order that they appear in the utterance representation.

The first set of assumptions to be loaded into the deductive memory will be that about the concept "Feel". Its encyclopædic entry is shown below. Note that (B.2), (B.3) and (B.4) are typical examples of universal assumption schemas. I have made the simplification for the purpose of simple stories that any animate agent is a person.

$$\mathrm{Feel}(!f, !f1, !f2) \supset \mathrm{Person}(!f) \tag{B.2}$$

$$\mathrm{Feel}(!f, !f1, !f2) \supset \mathrm{Feeling}(!f1) \tag{B.3}$$

$$\text{Feel}(!f, !f1, !f2) \wedge \text{OCCUR}(!f2, !t) \supset \text{HOLDS}(\text{Has}(!f, !f1), !t) \tag{B.4}$$

Broadly, this information states that "Feel" is a three argument predicate. The first argument of which is a person, the second a feeling, and the third an event. If the event occurs over an interval, the person has the stated feeling for that period. Much as it is not shown explicitly, the sortal information in (B.2) and (B.3) is duplicated by the "Person" and "Feeling" predications. This is because the explicit inferences result in contextual implications which should be counted as information; sortal information cannot ever be used to derive contextual information[1].

The next entry to be loaded is that for the constant "nutkin".

$$\text{Person}(nutkin) \tag{B.5}$$

$$\mathcal{D}\text{Squirrel}(nutkin) \tag{B.6}$$

$$\text{Squirrel}(!s) \supset \text{Animal}(!s) \tag{B.7}$$

$$\text{Squirrel}(!s) \supset \text{Small}(!s) \tag{B.8}$$

$$\text{Squirrel}(!s) \supset \text{Furry}(!s) \tag{B.9}$$

$$\begin{aligned}&\text{Squirrel}(!s) \supset \text{Tail}(\text{Sk.Tail.Of}(!s)) \wedge \\ &\quad \text{Furry}(\text{Sk.Tail.Of}(!s)) \wedge \text{Has}(!s, \text{Sk.Tail.Of}(!s))\end{aligned} \tag{B.10}$$

$$\text{Squirrel}(!s) \supset \text{Red}(!s) \vee \text{Grey}(!s) \tag{B.11}$$

$$\text{Squirrel}(!s) \wedge \text{Nut1}(!n) \supset \text{Likes}(!s, !n) \tag{B.12}$$

$$\text{Squirrel}(!s) \supset \text{Mammal}(!s) \tag{B.13}$$

I have made the assumption that entities called Nutkin are usually squirrels. This, of course, is very culture dependent, but is the kind of information that should be part of an encyclopædic entry. This information about Nutkin has been encoded as the default assumption schema, (B.6), which states that entities denoted by *nutkin* are typically squirrels. I have made the somewhat implausible assumption that proper names are unique. A better implementation of names would use a binary "Name" predicate which relates a term denoting an entity with its proper name. The information about "Nutkin" would then be associated with the proper name, "Nutkin", rather than with the term denoting the object.

I have also assumed that information about squirrels becomes immediately available when "Nutkin" is mentioned. In order to effect this, I have included the entry for "squirrel" in the entry for Nutkin. This includes the facts that squirrels are small, furry animals with a furry tail. I have also included the information that squirrels like nuts. In fact, this piece of information is incomplete. When we say "squirrels like nuts" we mean "squirrels like to eat nuts". This might be better realised as a script-like prototypical event where a squirrel eats a nut and enjoys it. This would be better than (B.12) because it postulates a possible event, whereas (B.12) asserts that squirrels like *all* nuts.

Next, the encyclopædic information for the concept "Hunger" is loaded. If any of these clauses are already in the deductor memory, they are not added, but are merely used to further justify the information already present. Checking for the presence of assumptions is performed by the interface, which uses the subsumption mechanism described in section 4.5.1. The encyclopædic information includes an existential assumption schema, (B.14). This schema makes explicit the food that somebody would like to eat when they are hungry: "Sk.Food.Of($!t, !a$)". The name was generated automatically by Crystal during the normalisation process by noticing that it was first predicated as "food".

$$\text{HOLDS}(\text{Has}(!p, Hunger), !t) \supset \text{Person}(!p)$$

$$\begin{aligned}&\text{HOLDS}(\text{Has}(!a, Hunger), !t) \supset \text{Food}(\text{Sk.Food.Of}(!t, !a)) \wedge \\ &\quad \text{HOLDS}(\text{Wants}(!a, \text{Food.Of}(!a)))\end{aligned} \tag{B.14}$$

[1]See chapter 5 for a full explanation.

$$\text{Food}(!f) \wedge \text{HOLDS}(\text{Wants}(!a, !f), !t) \wedge \text{HOLDS}(\text{Has}(!a, !f), !t) \supset$$
$$\text{Eats}(!a, !f, \text{Sk.Eats.Event.Of}(!t, !f, !a)) \wedge$$
$$\text{OCCUR}(\text{Sk.Eats.Event.Of}(!t, !f, !a)) \tag{B.15}$$

$$\text{Eats}(!a, !f, !e) \wedge \text{OCCUR}(!e, !t) \wedge (!t <!t1) \supset$$
$$\mathcal{D}\text{HOLDS}(\neg\text{Has}(!a, \mathit{Hunger}), !t1) \tag{B.16}$$

This information states that hungry things are people. When they are hungry, they want food. If they have this food, they eat it. After they eat it, they are typically not hungry any more. This information only holds during the period in which they are hungry. Clearly, I have made a massive simplification regarding the notion of "desiring".

The system loads temporal information whenever one of Allen's predicates is contained in the set of concepts contained in the utterance. The following information concerning time is therefore loaded. These are the weakened axioms of the reified temporal logic as described in section 5.5.1[2]:

$$(!i1 <!i2) \wedge (!i2 <!i3) \supset (!i1 <!i3)$$

$$\text{MEETS}(!i1, !i2) \wedge \text{DURING}(!i2, !i3) \supset \text{OVERLAPS}(!i1, !i3) \vee \text{DURING}(!i1, !i3) \vee \text{MEETS}(!i1, !i3)$$

$$\text{OVERLAPS}(!i1, !i2) \supset$$
$$\text{DURING}(\text{Sk.Overlap.Interval.Of}(!i2, !i1), !i1) \wedge$$
$$\text{DURING}(\text{Sk.Overlap.Interval.Of}(!i2, !i1), !i2)$$

$$\text{MEETS}(!i1, !i2) \supset \neg(\text{BEFORE}(!i1, !i) \wedge \text{BEFORE}(!i, !i2))$$

$$\text{MEETS}(!i1, !i2) \vee (!i1 <!i2) \supset \neg(\text{DURING}(!i, !i1) \wedge \text{DURING}(!i, !i2))$$

$$\text{POINT}(!i) \supset \neg\text{OVERLAPS}(!i, !i1)$$

$$\text{POINT}(!i) \supset \neg\text{OVERLAPS}(!i1, !i)$$

$$\text{POINT}(!i1) \wedge \text{POINT}(!i2) \supset \neg\text{MEETS}(!i1, !i2)$$

$$\text{POINT}(!i) \supset \neg\text{DURING}(!i1, !i)$$

$$\text{HOLDS}(!p \wedge !q, !i) \supset \text{HOLDS}(!p, !i) \wedge \text{HOLDS}(!q, !i)$$

$$\text{HOLDS}(!p \vee !q, !i) \supset (\text{POINT}(!i1) \supset (\text{DURING}(!i1, !i) \supset (\text{HOLDS}(!p, !i1) \vee \text{HOLDS}(!q, !i1))))$$

$$\text{HOLDS}(\neg !p, !i) \supset \neg\text{HOLDS}(!p, !i)$$

$$\text{HOLDS}((\exists !x.\, !p), !i) \supset \text{POINT}(!i1) \wedge \text{DURING}(!i1, !i) \supset \text{HOLDS}(!p, !i1)$$

Note that many of these assumptions are loaded unnecessarily because the only concepts mentioned in the utterance representation are "HOLDS", "OCCUR" and "<". This means that unnecessary effort is being expended in loading them into the deductive device. However, it is not important for the discourses shown here because it is present in all of the senses of the utterance, incurring the same effort premium in each. A way of avoiding unnecessary effort is to fragment the theory so that only the germane parts are loaded, ie. when indexed by keywords. This problem of expending unnecessary effort has arisen due to the fact that the the temporal logic is not of the same status as the encyclopædic information associated with other concepts in the system (see section 5.5.1).

[2] In fact, I found this temporal information provided here too weak to handle some of the examples enumerated in appendix E, and strengthened it so that it could recognise that $p$ and $\neg p$ cannot both hold at the same time. The only difference that this would make to the current examples is that more temporal inferences would result in each context derived from an utterance.

### B.1.1 Inferencing

The next step taken is to start the inference closure operation. On its termination, the deductive device will contain the inferences enumerated below, as well as the utterance representation and the loaded encyclopædic information. Contextual implications are marked with a "†" and simple implications are marked with a "◇". They are superscripted with the number of times their associated node has been implicated via different chains, if this number is greater than one.

$$\text{Squirrel}(\mathit{nutkin}) \tag{B.17}$$

$$\text{Mammal}(\mathit{nutkin}) \tag{B.18}$$

$$\text{Nut1}(!n) \supset \text{Likes}(\mathit{nutkin}, !n) \tag{B.19}$$

$$\text{Red}(\mathit{nutkin}) \vee \text{Grey}(\mathit{nutkin}) \tag{B.20}$$

$$\begin{aligned}&\text{Tail}(\text{Sk.Tail.Of}(\mathit{nutkin}))\wedge\\&\quad \text{Furry}(\text{Sk.Tail.Of}(\mathit{nutkin})) \wedge\\&\quad \text{Has}(\mathit{nutkin}, \text{Sk.Tail.Of}(\mathit{nutkin}))\end{aligned} \tag{B.21}$$

$$\text{Tail}(\text{Sk.Tail.Of}(\mathit{nutkin})) \tag{B.22}$$

$$\begin{aligned}&\text{Furry}(\text{Sk.Tail.Of}(\mathit{nutkin}))\wedge\\&\quad \text{Has}(\mathit{nutkin}, \text{Sk.Tail.Of}(\mathit{nutkin}))\end{aligned} \tag{B.23}$$

$$\text{Has}(\mathit{nutkin}, \text{Sk.Tail.Of}(\mathit{nutkin})) \tag{B.24}$$

$$\text{Furry}(\mathit{nutkin}) \tag{B.25}$$

$$\text{Small}(\mathit{nutkin}) \tag{B.26}$$

$$\dagger^2 \diamond^2 \text{Person}(\mathit{nutkin}) \tag{B.27}$$

$$\text{Feel}(\mathit{nutkin}, \mathit{Hunger}, \mathit{Sk.Feel.Event}10) \tag{B.28}$$

$$\dagger \diamond \text{Feeling}(\mathit{Hunger})$$

$$\dagger \diamond \text{OCCUR}(\mathit{Sk.Feel.Event}10, !t) \supset \text{HOLDS}(\text{Has}(\mathit{nutkin}, \mathit{Hunger}), !t)$$

$$\text{OCCUR}(\mathit{Sk.Feel.Event}10, \mathit{Sk.Feel.Interval}25) \wedge (\mathit{Sk.Feel.Interval}25 < \mathit{NOW}) \tag{B.29}$$

$$\text{OCCUR}(\mathit{Sk.Feel.Event}10, \mathit{Sk.Feel.Interval}25) \tag{B.30}$$

$$\dagger^2 \text{HOLDS}(\text{Has}(\mathit{nutkin}, \mathit{Hunger}), \mathit{Sk.Feel.Interval}25) \tag{B.31}$$

$$\begin{aligned}&\dagger \diamond \text{Food}(\text{Sk.Food.Of}(\mathit{Sk.Feel.Interval}25, \mathit{nutkin}))\wedge\\&\quad \text{HOLDS}(\text{Wants}(\mathit{nutkin},\\&\qquad \text{Sk.Food.Of}(\mathit{Sk.Feel.Interval}25, \mathit{nutkin})), \mathit{Sk.Feel.Interval}25)\end{aligned} \tag{B.32}$$

$$\dagger \diamond \text{Food}(\text{Sk.Food.Of}(\mathit{Sk.Feel.Interval}25, \mathit{nutkin})) \tag{B.33}$$

$$\begin{aligned}&\dagger \diamond \text{HOLDS}(\text{Wants}(!a, \text{Sk.Food.Of}(\mathit{Sk.Feel.Interval}25, \mathit{nutkin})), !t)\wedge\\&\quad \text{HOLDS}(\text{Has}(!a, \text{Sk.Food.Of}(\mathit{Sk.Feel.Interval}25, \mathit{nutkin})), !t) \supset\\&\quad \text{Eats}(!a, \text{Sk.Food.Of}(\mathit{Sk.Feel.Interval}25, \mathit{nutkin}),\\&\qquad \text{Sk.Eats.Event.Of}(!t, \text{Sk.Food.Of}(\mathit{Sk.Feel.Interval}25, \mathit{nutkin}), !a)) \wedge\\&\quad \text{OCCUR}(\text{Sk.Eats.Event.Of}(!t, \text{Sk.Food.Of}(\mathit{Sk.Feel.Interval}25, \mathit{nutkin}), !a), !t)\end{aligned} \tag{B.34}$$

$\dagger \diamond$ HOLDS(Wants($nutkin$,
Sk.Food.Of($Sk.Feel.Interval25, nutkin$)), $Sk.Feel.Interval25$) (B.35)

$\dagger$ HOLDS(Has($nutkin$, Sk.Food.Of($Sk.Feel.Interval25, nutkin$)), $Sk.Feel.Interval25$) $\supset$
[Eats($nutkin$, Sk.Food.Of($Sk.Feel.Interval25, nutkin$),
Sk.Eats.Event.Of($Sk.Feel.Interval25$,
Sk.Food.Of($Sk.Feel.Interval25, nutkin$), $nutkin$))] $\wedge$
[OCCUR(Sk.Eats.Event.Of($Sk.Feel.Interval25$,
Sk.Food.Of($Sk.Feel.Interval25, nutkin$), $nutkin$), $Sk.Feel.Interval25$)] (B.36)

$\dagger \diamond$ Feel($!f, !f1, Sk.Feel.Event10$) $\supset$ HOLDS(Has($!f, !f1, Sk.Feel.Interval25$))

($Sk.Feel.Interval25 < NOW$)

$\dagger \diamond$ ($NOW <!i3$) $\supset$ ($Sk.Feel.Interval25 <!i3$)

$\dagger \diamond$ ($!i1 < Sk.Feel.Interval25$) $\supset$ ($!i1 < NOW$)

$\dagger \diamond \neg$(DURING($!i, Sk.Feel.Interval25$) $\wedge$ DURING($!i, NOW$))

The first inference that takes place is (B.17), which occurs immediately after the the encyclopædic entries are added because of the default assumption schema (B.6). However, it is not a contextual implication because it has been derived solely from assumptions of status "Memory". Because of the depth-first inference strategy, (B.17) is immediately used to "instantiate" the information about squirrels contained in the encyclopædic entry for "nutkin" using modus ponens. The instantiations (B.18) to (B.26), are not contextual implications because they also were only derived from sources of status "Memory".

After this, the first conjunct of the utterance representation, (B.28), is asserted. Since this is merely an analytic implication derived from representation (B.1) of status "Utterance"(using and elimination), it is also not a contextual effect. However, this conjunct is then immediately used to instantiate the encyclopædic information about "feeling", which will be regarded as contextual implications because the status of the assumptions use to derive them will be "Memory" and "Utterance". The first contextual implication made is the reinforcement of (B.27). This is again reinforced at a later stage by the information contained in the encyclopædic entry for "hunger".

Next, the remainder of the logical form of the utterance, (B.29), is asserted, followed by the second conjunct, (B.30) using "and-elimination". The information that follows from it is also asserted. The remaining conjunct of the utterance is similarly reduced and processed.

The only other node to be inferred more than once is (B.31). This has been inferred from two paths beginning at the same node, (B.4). Generally, assumption schema with with conjuncts preceding a material implication yield two separate inference paths to their conclusion if all of the antecedent conjuncts are satisfied (see chapter 5). This results in two non-simple implications, although intuitively we expect one (I suggest a principled way of avoiding this in chapter 9).

One other non-simple contextual implication is inferred, (B.36).

### B.1.2 Effort Calculations

There is no advantage in considering parsing or input effort in this example (see section 6.4.1).

Given calculations using the "simple inference" strategy, loading effort $E_c$ is constant, which I will assume to be 1 for convenience. Otherwise, loading effort is calculated as follows.

- The loading effort for "feel" is 3 units, one for each assumption schema.
- For "hunger", if the assumptions were broken down as much as possible, the effort would be 6 units. However, the existential schemas (B.14) and (B.15) are not as compact as they could be because they have a conjunction on the right-hand side of the material implication. Because of this, each one will deliver three contextual implications, rather than one, in the cases where modus ponens is successfully applied. Using the approach suggested in section 6.4.2, the total is therefore adjusted to take this into account. Thus, the loading effort is 8 units.
- Similarly, the loading effort for "nutkin" is 13 units, and for "time" is 17 units.

Thus, the total loading effort $E_l = 41$.

### B.1.3 Contextual Information

#### Simple Implications Approach

The three non-simple contextual implications are (B.36) and (B.31) (twice), which can be paraphrased as "if Nutkin has some food during the period that he is hungry, he will eat it" and "Nutkin has hunger during the period that he is hungry". Note that (B.33) and (B.35) are simple implications because they are analytical inferences derived from (B.32). Considering the utterance, these seem reasonable contextual effects. It can be noted that, even when there is no preceding context (as is the case for this present example), it is still possible to obtain contextual effects. This demonstrates that RT is not just a theory of how assumptions link back to previous contexts.

If every justification is to be counted, then the information measured this way, $I_{js} = 3$.

If only nodes are counted, then the information $I_{ns} = 2$.

#### Effort Approach

There are two possible measures of information here: one that counts every justification, $I_{jl}$, and another which only counts new nodes, $I_{nl}$. Both of these can be calculated by inspection:

$$I_{jl} = 15$$

and

$$I_{nl} = 13$$

$I_{nl}$ discounts the two extra justifications attributed to (B.27) and (B.31).

### B.1.4 Relevance

The Relevance values are only intended to be used comparatively, and are only shown here to illustrate how the calculations are made. Therefore, I shall comment on them only briefly.

#### Simple Implications Approach

Let the Relevance Factor using simple implications and counting multiple justifications be $\mathcal{R}_{js}$. Then:

$$\mathcal{R}_{js} = \frac{I_{js}}{E_c} = 3$$

Let the Relevance Factor using simple implications and counting only nodes be $\mathcal{R}_{js}$. Then:

$$\mathcal{R}_{ns} = \frac{I_{ns}}{E_c} = 2$$

#### Effort Approach

Let the Relevance Factor using loading effort and counting multiple justifications be $\mathcal{R}_{jl}$. Then:

$$\mathcal{R}_{jl} = \frac{I_{jl}}{E_l} = \frac{15}{41} = 0.366$$

Let the Relevance Factor using loading effort and counting new nodes only be $\mathcal{R}_{nl}$. Then:

$$\mathcal{R}_{nl} = \frac{I_{nl}}{E_l} = \frac{13}{41} = 0.317$$

Of course, these values are not very useful at this point because there are no other candidate interpretations to compare this one with. The only significant feature of these figures is that the initial value of Relevance is low which can be attributed to the fact that the utterance occurs at the very beginning of the discourse (see chapter 6.5.2).

## B.2 Second Utterance

The second utterance is represented thus:

$$\begin{aligned}&\text{Male}(Sk.Male25) \wedge \text{IT}(Sk.Male25) \wedge \text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event17) \wedge\\ &\quad \text{Nut}(Sk.Nut10) \wedge \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge\\ &\quad (Sk.Find.Interval26 < NOW)\end{aligned} \tag{B.37}$$

The context expansion process is aware that "Nut" represents an ambiguity because "Nut1" and "Nut2" have the same categories (noun, arity 1) in the conceptual memory (see section 5.3.4).

Each assumption in the previous context, $C_1$, comprising the utterance, encyclopædic entries and all related inferences, is now marked as having contextual status "context". It is also saved on the interpretation queue for later restoration when processing the second sense of the utterance. The first sense is then processed in this context and the results, $C_{2a}$, pushed onto the interpretation queue for possible restoration. The context associated with the first utterance, $C_1$, is then restored from the queue and the second sense of the second utterance processed, yielding context $C_{2b}$. If $C_{2b}$ is more Relevant than $C_{2a}$, it replaces it on the interpretation queue. Otherwise, $C_{2b}$ is discarded. The top context (the most Relevant interpretation) of the interpretation queue replaces the contents of the deductive device (if it is not currently present). The result is that the interpretation queue keeps track of each successive context and the most Relevant interpretation is in the deductive device (see chapter 4).

## B.3 Second Utterance ($1^{st}$ Sense)

Everything in the deductive device is marked as having contextual status "Context" before the logical form of the second sense of the second utterance utterance is entered. Its representation is:

$$\begin{aligned}&\text{Male}(Sk.Male25) \wedge \text{IT}(Sk.Male25) \wedge \text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event17) \wedge\\ &\quad \text{Nut1}(Sk.Nut10) \wedge \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge\\ &\quad (Sk.Find.Interval26 < NOW)\end{aligned} \tag{B.38}$$

as before, it is of contextual status "Utterance". The memory entries for the concepts within this utterance are then loaded with contextual status "memory". Note that there is no need to load "Time" as it is already present.

Encyclopædic information for concept "Male" is next loaded, with contextual status "Memory".

$$\text{Male}(!m) \supset \text{Person}(!m) \tag{B.39}$$

$$\text{Male}(!m) \supset \neg\text{Female}(!m)$$

$$\text{Male}(!m) \supset \neg\text{Pregnant}(!m, !e) \tag{B.40}$$

This states that males are people that are not female and will never be pregnant (in fact, (B.40) expresses the slightly stronger fact that that there cannot be an event of a male being pregnant, whether it occurs or not).

The ambiguous "Nut" is loaded as "Nut1" (in this interpretation):

$$\text{Nut1}(!n) \supset \text{Fruit}(!n)$$

$$\text{Nut1}(!n) \supset \text{Dry}(!n)$$

$$\text{Nut1}(!n) \supset \text{Kernel}(\text{Sk.Kernel.Of}(!n)) \wedge \text{Has}(!n, \text{Sk.Kernel.Of}(!n)) \tag{B.41}$$

$$\begin{aligned}&\text{Nut1}(!n) \supset \text{Shell}(\text{Sk.Shell.Of}(!n)) \wedge\\ &\quad \text{Hard}(\text{Sk.Shell.Of}(!n)) \wedge \text{Has}(!n, \text{Sk.Shell.Of}(!n))\end{aligned} \tag{B.42}$$

$$\text{Nut1}(!n) \supset \text{Food}(!n)$$

This states that nuts are dry fruit that have kernels and hard shells. They are food. This is typical, frame-like information, especially (B.41) and (B.42).

Finally, encyclopædic information for concept "Find" is loaded:

$$\text{Find}(!f, !f1, !f2) \supset \text{Person}(!f) \tag{B.43}$$

$$\text{Find}(!f, !f1, !f2) \supset \text{Object}(!f1)$$

$$\text{Find}(!f, !f1, !f2) \supset \text{Accident}(!f2)$$

$$\text{Find}(!f, !f1, !f2) \wedge \text{OCCURS}(!f2, !t) \supset \text{HOLDS}(\text{Knows}(!f, \text{Location.Of}(!f1)), !t) \tag{B.44}$$

Broadly, finding involves a person discovering the location of an object by accident.

### B.3.1 Inferencing

The closure is now taken with the deductor memory containing the utterance, the freshly loaded encyclopædic entries and the previous context. The following is inferred:

$$\dagger\,\text{Find}(!f, !f1, Sk.Feel.Event10) \supset \\ \text{HOLDS}(\text{Knows}(!f, \text{Location.Of}(!f1)), !t) \tag{B.45}$$

$$\text{Male}(Sk.Male25)$$

$$\dagger \diamond \neg\text{Pregnant}(Sk.Male25, !e)$$

$$\dagger \diamond \neg\text{Female}(Sk.Male25)$$

$$\dagger\,\mathcal{D}(Sk.Male25 = nutkin) \tag{B.46}$$

$$\dagger\,(Sk.Male25 = nutkin) \tag{B.47}$$

$$\dagger^{5} \diamond^{2}\,\text{Person}(Sk.Male25) \tag{B.48}$$

$$\text{IT}(Sk.Male25) \wedge \text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event.17) \wedge \\ \text{Nut1}(Sk.Nut10) \wedge \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge \\ (Sk.Find.Interval26 < NOW)$$

$$\text{IT}(Sk.Male25) \tag{B.49}$$

$$\text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event17) \wedge \text{Nut1}(Sk.Nut10) \wedge \\ \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge (Sk.Find.Interval26 < NOW)$$

$$\text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event17)$$

$$\dagger \diamond \text{Accident}(Sk.Find.Event17)$$

$$\dagger \diamond \text{Object}(Sk.Nut10)$$

$$\dagger \diamond \text{OCCUR}(Sk.Find.Event17, !t) \supset \text{HOLDS}(\text{Knows}(Sk.Male25, \text{Location.Of}(Sk.Nut10)), !t)$$

$$\text{Nut1}(Sk.Nut10) \wedge \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge \\ (Sk.Find.Interval26 < NOW)$$

$$\text{Nut1}(Sk.Nut10)$$

$$\dagger\,\mathcal{D}(Sk.Nut10 = \text{Sk.Food.Of}(Sk.Feel.Interval25, nutkin)) \tag{B.50}$$

$$\dagger\,(Sk.Nut10 = \text{Sk.Food.Of}(Sk.Feel.Interval25, nutkin)) \tag{B.51}$$

$$\dagger^{2} \diamond \text{Food}(Sk.Nut10) \tag{B.52}$$

$\dagger \diamond$ Shell(Sk.Shell.Of($Sk.Nut10$)) $\wedge$ Hard(Sk.Shell.Of($Sk.Nut10$))$\wedge$
Has($Sk.Nut10$, Sk.Shell.Of($Sk.Nut10$))

$\dagger \diamond$ Shell(Sk.Shell.Of($Sk.Nut10$))

$\dagger \diamond$ Hard(Sk.Shell.Of($Sk.Nut10$)) $\wedge$ Has($Sk.Nut10$, Sk.Shell.Of($Sk.Nut10$))

$\dagger \diamond$ Hard(Shell.Of($Sk.Nut10$))

$\dagger \diamond$ Has($Sk.Nut10$, Sk.Shell.Of($Sk.Nut10$))

$\dagger \diamond$ Kernel(Sk.Kernel.Of($Sk.Nut10$)) $\wedge$ Has($Sk.Nut10$, Sk.Kernel.Of($Sk.Nut10$))

$\dagger \diamond$ Kernel(Sk.Kernel.Of($Sk.Nut10$))

$\dagger \diamond$ Has($Sk.Nut10$, Sk.Kernel.Of($Sk.Nut10$))

$\dagger \diamond$ Dry($Sk.Nut10$)

$\dagger \diamond$ Fruit($Sk.Nut10$)

OCCUR($Sk.Find.Event17$, $Sk.Find.Interval26$) $\wedge$ ($Sk.Find.Interval26 < NOW$)

$\dagger$ Eats(!$a$, !$f$, $Sk.Find.Event17$) $\supset$ $\mathcal{D}$HOLDS($\neg$Has(!$a$, $Hunger$, $NOW$)) (B.53)

OCCUR($Sk.Find.Event17$, $Sk.Find.Interval26$)

$\dagger^2$ HOLDS(Knows($Sk.Male25$, Location.Of($Sk.Nut10$)), $Sk.Find.Interval26$) (B.54)

$\dagger \diamond$ Find(!$f$, !$f1$, $Sk.Find.Event17$) $\supset$
HOLDS(Knows(!$f$, Location.Of(!$f1$)), $Sk.Find.Interval26$) (B.55)

$\dagger$ Feel(!$f$, !$f1$, $Sk.Find.Event17$) $\supset$
HOLDS(Has(!$f$, !$f1$), $Sk.Find.Interval26$) (B.56)

($Sk.Find.Interval26 < NOW$) (B.57)

$\dagger \diamond$ ($NOW <$!$i3$) $\supset$ ($Sk.Find.Interval26 <$!$i3$) (B.58)

$\dagger$ (!$i1 < Sk.Find.Interval26$) $\supset$ (!$i1 < NOW$) (B.59)

$\dagger \neg$(DURING(!$i$, $Sk.Find.Interval26$) $\wedge$ DURING(!$i$, $NOW$)) (B.60)

The first inference is obtained from the encyclopædic information and the previous context: (B.45) is derived from (B.30) and (B.44). This is a problematic inference, meaning something like "if Sk.Feel.Event10 is a finding event, then the agent knows the location of the object involved". However, it can be argued that "finding" and "feeling" are intuitively distinct events, so this assumption if meaningless as it is vacuously true (the LHS of the implication is always false). This inference could be blocked by sorting events. Indeed, one version of the sort hierarchy allows this. However, I have not sorted events here, partially to illustrate these inadequate inferences, and partially because I do not feel that events should necessarily have the same status as material objects (or even abstractions such as "hunger"). As evidence to support this opinion, I found that, in practice, it was very difficult to define a plausible taxonomy of events (see section 5.4 for more detail about sorting assumptions). This discussion also applies to the inferences (B.53) and (B.56). However, (B.56) is not so critical because it is only a simple implication.

Consider how the hypothesis (B.46) is generated. The abductive unifier attempts to unify (B.48) and (B.27). This initially fails, but, since these terms have "Compatible" sorts, (B.46) is asserted, followed by (B.47), since it is consistent to do so. This means that (B.48) and (B.27) unify. There are now two alternatives:

1. Add justifications for (B.48) to (B.27) and not bother asserting (B.48) at all.
2. Separately assert (B.48).

The first alternative would have the advantage that the appropriate consequences of the rejustified node would become contextual implications and no further duplications of inferences have to be made. The second alternative has the advantage that the hypothesis can be justified by the two nodes (B.48) and (B.27); if (B.48) should be made "out", then so would the hypothesis (B.46) and its believed repercussions. However, in the first case, the justification sets of the two nodes would be united; unless the justifications for both nodes became invalid, the hypothesis would remain "in". Thus, the second alternative has to be taken.

As a result of asserting (B.48) separately, the inferences from the previous context that justified (B.27) will also justify (B.48). Thus, it has 5 contextual implications consisting of:

- The three justifications for (B.27) (each now including the node (B.46) in their inlists).
- The two justifications that arise from the fact that finders and males are people ((B.43) and (B.39)).

Of these, only the former are non-simple implications. An advantage of allowing this rejustification is that connecting into previous contexts results in extra contextual effects[3]. Thus, such connections are favoured as to asserting something independently. This offers one explanation as to why it is better to use definite references rather than a full specification. Note also, that it is now assumed that Nutkin is male: *nutkin* has inherited all the properties of *Sk.Male25* because of the support of the equality sub-system (see section 8.4.5).

The next assertion of interest is (B.49). On being passed to the interface, *Sk.Male25* is added to the reference and "IT" lists before the assumption is asserted. This is used after the inference closure in conjunction with the equality sub-system to check that all references have been resolved. Resolution is interpreted as binding the referring constant to one or more other constants. In this case, it has already been achieved by the time (B.47) has been asserted. For other examples, I asserted all "IT" (and "REF") predicates at the beginning of the logical form of the utterance so that a reference would be asserted as such before a hypothesis concerning its equality with another object can be asserted. The fact that an object may be part of a definite reference should be established as early as possible because this information may be used in the culprit selection phase of dependency-directed backtracking when an inconsistency is discovered; referring expressions are assumed to be more difficult to give up than non-referring expressions.

Another hypothesis of equality is later made: (B.50). This was guessed from (B.52) and (B.33). (B.51) is consistent and consequently asserted. The guess postulates that the nut is assumed to be the food that Nutkin desires (and will eat, if he finds it). As was the case for (B.48), (B.52) inherits all of the justifications of (B.33) (one in this case). Thus it has one non-simple justification. (B.52) has some repercussions: (B.34) and (B.36). These will be rejustified as contextual implications using (B.51). Thus, two more non-simple implications are made. This amounts to realising that if Nutkin has the nut, he will eat it.

The inferences derived from the final conjunct, (B.58) to (B.60) are non-simple implications. This is because they are derived using a previous concept, "time". It is doubtful whether reference to time in a discourse is "referring to the subject" in the same way that referring to Nutkin or hunger would be (see section 6.4.1). An alternative approach that avoids this would be to raise temporal encyclopædic rules to the meta-level, which would also have the desirable effect of making them analytic rules.

## B.3.2 Effort Calculations

### Effort Approach

- Loading effort for "Nut1": 11 units (3 for (B.41) and 5 for (B.42)).
- Loading effort for "Male": 3 units.
- Loading effort for "Find": 4 units.

Total effort, $E_l = 11 + 3 + 4 = 18$ units.

[3] In fact, using the "simple implications" approach, it is the *only* way of obtaining contextual effects when asserting equality.

### B.3.3 Contextual Information

#### Simple Implication Approach

The numbers refer to number of justifications or nodes that are non-simple contextual implications.

$I_{js} =$

10 single justifications in context $(C_{2a} - C_1) +$
$(3 + 2)$ multiple justifications $+$
2 implications in context $C_1$
$= 17$ units

$I_{ns} =$

12 nodes in context $(C_{2a} - C_1) +$
2 nodes in context $C_1$
$= 14$ units

#### Effort Approach

The numbers refer to the number of justifications or nodes that are contextual implications.

$I_{jl} =$

26 single justifications in context $(C_{2a} - C_1) +$
$(5 + 2 + 2)$ multiple justifications $+$
2 implications in context $C_1$
$= 37$ units

$I_{nl} =$

29 nodes in context $(C_{2a} - C_1) +$
2 nodes in context $C_1$
$= 31$ units

### B.3.4 Relevance

#### Simple Implications Approach

$$\mathcal{R}_{js} = \frac{I_{js}}{E_c} = 17$$

and

$$\mathcal{R}_{ns} = \frac{I_{ns}}{E_c} = 14$$

#### Effort Approach

$$\mathcal{R}_{jl} = \frac{I_{jl}}{E_l} = \frac{37}{18} = 2.056$$

and

$$\mathcal{R}_{nl} = \frac{I_{nl}}{E_l} = \frac{31}{18} = 1.722$$

Note that the Relevance is higher for the second utterance in the discourse (in fact, this is true for both senses) because it has more connection with the previous context, and because one of its concepts ("time") has already been loaded. We can now go on to consider the processing of the second sense of this utterance representation.

## B.4 Second Utterance ($2^{nd}$ Sense)

The second sense of the second utterance involves the other sense of "nut": Nut2.

$$\begin{aligned}&\text{Male}(Sk.Male25) \wedge \text{IT}(Sk.Male25) \wedge \text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event17) \wedge \\ &\quad \text{Nut2}(Sk.Nut10) \wedge \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge \\ &\quad (Sk.Find.Interval26 < NOW)\end{aligned} \tag{B.61}$$

Before this utterance representation is processed, context $C_{2a}$ is pushed onto the interpretation queue. $C_1$ is then restored. The logical form of the second sense of the utterance is loaded and the appropriate encyclopædic entries accessed. Everything is loaded as for the first sense except for "Nut1", for which the entry for "Nut2" is substituted.

$$\text{Nut2}(!n) \supset \text{Metal}(!n)$$

$$\text{Nut2}(!n) \supset \text{Small}(!n)$$

$$\begin{aligned}&\text{Nut2}(!n) \supset \text{Attach}(\text{Sk.Solid22.Of}(!n), \text{Sk.Solid23.Of}(!n), \text{Sk.Attach.Event.Of}(!n)) \wedge \\ &\quad \text{Using}(\text{Sk.Attach.Event.Of}(!n), !n) \wedge \\ &\quad \text{Using}(\text{Sk.Attach.Event.Of}(!n), \text{Sk.Bolt5.Of}(!n)) \wedge \\ &\quad \text{Solid}(\text{Sk.Solid22.Of}(!n)) \wedge \text{Solid}(\text{Sk.Solid23.Of}(!n)) \wedge \\ &\quad \neg(\text{Sk.Solid22.Of}(!n) = \text{Sk.Solid23.Of}(!n)) \wedge \text{Bolt}(\text{Sk.Bolt5.Of}(!n)) \wedge \\ &\quad \text{Adjacent}(\text{Sk.Solid22.Of}(!n), \text{Sk.Solid23.Of}(!n))\end{aligned} \tag{B.62}$$

The above encyclopædic entry states that "Nut2"s are small metal objects. They can be used with bolts to attach two adjacent solids together. "Sk.Attach.Event.Of(!n)" is a potential attaching event that has not yet occurred[4]. (B.62) is therefore a script-like assumption schema. However, it has the advantages that the logical rules of inference can be applied to it (or parts of it).

### B.4.1 Inferencing

After processing, the deductor memory will contain $C_1$ plus:

$$\begin{aligned}&\dagger\,\text{Find}(!f, !f1, Sk.Feel.Event10) \supset \\ &\quad \text{HOLDS}(\text{Knows}(!f, \text{Location.Of}(!f1)), !t)\end{aligned} \tag{B.63}$$

$$\text{Male}(Sk.Male25)$$

$$\dagger \diamond \neg\text{Pregnant}(Sk.Male25, !e)$$

$$\dagger \diamond \neg\text{Female}(Sk.Male25)$$

$$\dagger\,\mathcal{D}(Sk.Male25 = nutkin)$$

$$\dagger\,(Sk.Male25 = nutkin)$$

$$\dagger^5 \diamond^2 \text{Person}(Sk.Male25)$$

$$\begin{aligned}&\text{IT}(Sk.Male25) \wedge \text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event.17) \wedge \\ &\quad \text{Nut2}(Sk.Nut10) \wedge \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge \\ &\quad (Sk.Find.Interval26 < NOW)\end{aligned}$$

$$\text{IT}(Sk.Male25)$$

[4] Nor does it have to. It is a sort of prototypical event (see section 5.5).

$\text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event17) \wedge \text{Nut2}(Sk.Nut10) \wedge$
$\quad \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge (Sk.Find.Interval26 < NOW)$

$\text{Find}(Sk.Male25, Sk.Nut10, Sk.Find.Event17)$

$\dagger \diamond \text{Accident}(Sk.Find.Event17)$

$\dagger \diamond \text{Object}(Sk.Nut10)$

$\dagger \diamond \text{OCCUR}(Sk.Find.Event17, !t) \supset \text{HOLDS}(\text{Knows}(Sk.Male25, \text{Location.Of}(Sk.Nut10)), !t)$

$\text{Nut2}(Sk.Nut10) \wedge \text{OCCUR}(Sk.Find.Event17, Sk.Find.Interval26) \wedge$
$\quad (Sk.Find.Interval26 < NOW)$

$\text{Nut2}(Sk.Nut10)$ (B.64)

$\dagger \diamond \text{Attach}(\text{Sk.Solid22.Of}(Sk.Nut10), \text{Sk.Solid23.Of}(Sk.Nut10), \text{Sk.Attach.Event.Of}(Sk.Nut10)) \wedge$
$\quad \text{Using}(\text{Sk.Attach.Event.Of}(Sk.Nut10), Sk.Nut10) \wedge$
$\quad \text{Using}(\text{Sk.Attach.Event.Of}(Sk.Nut10), \text{Sk.Bolt5.Of}(Sk.Nut10)) \wedge$
$\quad \text{Solid}(\text{Sk.Solid22.Of}(Sk.Nut10)) \wedge \text{Solid}(\text{Sk.Solid23.Of}(Sk.Nut10)) \wedge$
$\quad \neg(\text{Sk.Solid22.Of}(Sk.Nut10) = \text{Sk.Solid23.Of}(Sk.Nut10)) \wedge$
$\quad \text{Bolt}(\text{Sk.Bolt5.Of}(Sk.Nut10)) \wedge$
$\quad \text{Adjacent}(\text{Sk.Solid22.Of}(Sk.Nut10), \text{Sk.Solid23.Of}(Sk.Nut10))$ (B.65)

$\dagger \diamond \text{Attach}(\text{Sk.Solid22.Of}(Sk.Nut10), \text{Sk.Solid22.Of}(Sk.Nut10), \text{Sk.Attach.Event.Of}(Sk.Nut10))$

$\dagger \diamond \text{Using}(\text{Sk.Attach.Event.Of}(Sk.Nut10), Sk.Nut10) \wedge$
$\quad \text{Using}(\text{Sk.Attach.Event.Of}(Sk.Nut10), \text{Sk.Bolt5.Of}(Sk.Nut10)) \wedge$
$\quad \text{Solid}(\text{Sk.Solid22.Of}(Sk.Nut10)) \wedge \text{Solid}(\text{Sk.Solid23.Of}(Sk.Nut10)) \wedge$
$\quad \neg(\text{Sk.Solid22.Of}(Sk.Nut10) = \text{Sk.Solid23.Of}(Sk.Nut10)) \wedge$
$\quad \text{Bolt}(\text{Sk.Bolt5.Of}(Sk.Nut10)) \wedge$
$\quad \text{Adjacent}(\text{Sk.Solid22.Of}(Sk.Nut10), \text{Sk.Solid23.Of}(Sk.Nut10))$

$\dagger \diamond \text{Using}(\text{Sk.Attach.Event.Of}(Sk.Nut10), Sk.Nut10)$

$\dagger \diamond \text{Using}(\text{Sk.Attach.Event.Of}(Sk.Nut10), \text{Sk.Bolt5.Of}(Sk.Nut10)) \wedge$
$\quad \text{Solid}(\text{Sk.Solid22.Of}(Sk.Nut10)) \wedge \text{Solid}(\text{Sk.Solid23.Of}(Sk.Nut10)) \wedge$
$\quad \neg(\text{Sk.Solid22.Of}(Sk.Nut10) = \text{Sk.Solid23.Of}(Sk.Nut10)) \wedge$
$\quad \text{Bolt}(\text{Sk.Bolt5.Of}(Sk.Nut10)) \wedge$
$\quad \text{Adjacent}(\text{Sk.Solid22.Of}(Sk.Nut10), \text{Sk.Solid23.Of}(Sk.Nut10))$

$\dagger \diamond \text{Using}(\text{Sk.Attach.Event.Of}(Sk.Nut10), \text{Sk.Bolt5.Of}(Sk.Nut10))$

$\dagger \diamond \text{Solid}(\text{Sk.Solid22.Of}(Sk.Nut10)) \wedge \text{Solid}(\text{Sk.Solid23.Of}(Sk.Nut10)) \wedge$
$\quad \neg(\text{Sk.Solid22.Of}(Sk.Nut10) = \text{Sk.Solid23.Of}(Sk.Nut10)) \wedge$
$\quad \text{Bolt}(\text{Sk.Bolt5.Of}(Sk.Nut10)) \wedge$
$\quad \text{Adjacent}(\text{Sk.Solid22.Of}(Sk.Nut10), \text{Sk.Solid23.Of}(Sk.Nut10))$ (B.66)

$\dagger \diamond \text{Solid}(\text{Sk.Solid22.Of}(Sk.Nut10))$ (B.67)

$\dagger \mathcal{D}(\text{Sk.Solid22.Of}(Sk.Nut10) = \text{Sk.Solid23.Of}(Sk.Nut10))$ (B.68)

$\dagger \diamond \neg(\text{Sk.Solid22.Of}(Sk.Nut10) = \text{Sk.Solid23.Of}(Sk.Nut10))$ (B.69)

$$\begin{aligned}\dagger \diamond\ & \mathrm{Solid(Sk.Solid23.Of(}Sk.Nut10))\wedge \\ & \neg(\mathrm{Sk.Solid22.Of(}Sk.Nut10) = \mathrm{Sk.Solid23.Of(}Sk.Nut10)) \wedge \\ & \mathrm{Bolt(Sk.Bolt5.Of(}Sk.Nut10)) \wedge \\ & \mathrm{Adjacent(Sk.Solid22.Of(}Sk.Nut10), \mathrm{Sk.Solid23.Of(}Sk.Nut10))\end{aligned}$$

$$\begin{aligned}\dagger \diamond\ & \neg(\mathrm{Sk.Solid22.Of(}Sk.Nut10) = \mathrm{Sk.Solid23.Of(}Sk.Nut10))\wedge \\ & \mathrm{Bolt(Sk.Bolt5.Of(}Sk.Nut10)) \wedge \\ & \mathrm{Adjacent(Sk.Solid22.Of(}Sk.Nut10), \mathrm{Sk.Solid23.Of(}Sk.Nut10))\end{aligned} \tag{B.70}$$

$$\begin{aligned}\dagger \diamond\ & \mathrm{Bolt(Sk.Bolt5.Of(}Sk.Nut10))\wedge \\ & \mathrm{Adjacent(Sk.Solid22.Of(}Sk.Nut10), \mathrm{Sk.Solid23.Of(}Sk.Nut10))\end{aligned}$$

$$\dagger \diamond\ \mathrm{Bolt(Sk.Bolt5.Of(}Sk.Nut10))$$

$$\dagger \diamond\ \mathrm{Adjacent(Sk.Solid22.Of(}Sk.Nut10), \mathrm{Sk.Solid23.Of(}Sk.Nut10))$$

$$\dagger \diamond\ \mathrm{Small(}Sk.Nut10)$$

$$\dagger \diamond\ \mathrm{Metal(}Sk.Nut10)$$

$$\mathrm{OCCUR}(\boldsymbol{Sk.Find.Event17}, \boldsymbol{Sk.Find.Interval26}) \wedge (\boldsymbol{Sk.Find.Interval26} < NOW) \tag{B.71}$$

$$\dagger \mathrm{Eats}(!a, !f, \boldsymbol{Sk.Find.Event17}) \supset \mathcal{D}\mathrm{HOLDS}(\neg \mathrm{Has}(!a, Hunger, NOW)) \tag{B.72}$$

$$\mathrm{OCCUR}(\boldsymbol{Sk.Find.Event17}, \boldsymbol{Sk.Find.Interval26})$$

$$\dagger^2\ \mathrm{HOLDS(Knows(}Sk.Male25, \mathrm{Location.Of(}Sk.Nut10)), \boldsymbol{Sk.Find.Interval26})$$

$$\begin{aligned}\dagger \diamond\ & \mathrm{Find}(!f, !f1, \boldsymbol{Sk.Find.Event17}) \supset \\ & \mathrm{HOLDS(Knows(}!f, \mathrm{Location.Of(}!f1)), \boldsymbol{Sk.Find.Interval26})\end{aligned} \tag{B.73}$$

$$\dagger \mathrm{Feel}(!f, !f1, \boldsymbol{Sk.Find.Event17}) \supset \mathrm{HOLDS(Has(}!f, !f1), \boldsymbol{Sk.Find.Interval26})$$

$$(\boldsymbol{Sk.Find.Interval26} < NOW)$$

$$\dagger (NOW < !i3) \supset (\boldsymbol{Sk.Find.Interval26} < !i3)$$

$$\dagger (!i1 < \boldsymbol{Sk.Find.Interval26}) \supset (!i1 < NOW)$$

$$\dagger \neg(\mathrm{DURING}(!i, \boldsymbol{Sk.Find.Interval26}) \wedge \mathrm{DURING}(!i, NOW))$$

$$\dagger \mathrm{Solid(Sk.Solid23.Of(}Sk.Nut10)) \tag{B.74}$$

The inferencing on the first two conjuncts of the utterance yields similar inferences to those of the first sense (with "Nut2" substituted for "Nut1" in the necessary places). "He" is still guessed to be Nutkin. Inferences diverge with the processing of the third conjunct, (B.64). This results in the script-like assumption, (B.62), being instantiated to (B.65), which is a contextual implication. Despite the apparent proliferation of inferences, most of them are just the results of "and-elimination" being applied to (B.65); they are, therefore, merely simple implications[5].

The next assertion of interest is (B.68). This hypothesises that the two solids that might be attached to each other by *Sk.Nut10*. This guess is made from (B.66) and (B.70) which do not ultimately unify because of the assertion (B.69). However, temporarily:

$$\dagger (\mathrm{Sk.Solid22.Of(}Sk.Nut10) = \mathrm{Sk.Solid23.Of(}Sk.Nut10)) \tag{B.75}$$

[5]Using the effort approach, the loading effort will balance the information.

is asserted. The assumption (B.69) cancels this hypothesis because it is not a default at all, having been derived from (B.65) by a series of "and elimination"s. This results in (B.67), which had been validly justified twice, (because Sk.Solid22.Of(*Sk.Nut10*) and Sk.Solid23.Of(*Sk.Nut10*) were considered equal) being justified only once. The information count is decreased and the effort increased. This is an example of how a hypothesis of equality can be asserted of two Compatible terms, but may ultimately prove to be wrong. In this particular case, the incorrect hypothesis could have been avoided by ensuring that (B.69) was asserted before (B.66) and (B.70). In order to ensure the correct processing order, the consequent conjuncts in (B.62) could be re-ordered[6]. This re-ordering can avoid artificially high Relevance estimates, as well as the artificially low ones that occur in this example; had (B.75) been of status Context, the information count would have been incremented rather than decremented (see section 7.6.1) giving an unreasonably high estimation of the number of contextual effects.

It can be seen that after (B.75) is retracted, it is again necessary to assert (B.74) which was previously assumed to be equal to (B.67). Other than this assumption, the remainder of the assumptions from (B.71) are the same as for context $C_{2a}$.

### B.4.2 Effort Calculations

#### Effort Approach

- Loading effort for "Nut2": 17 units (15 for (B.62)).
- "Male" and "Find" come to 7 units (as shown for $C_{2a}$).

Total effort, $E_l = 24$ units[7].

## B.5 Contextual Information

#### Simple Implication Approach

The numbers refer to number of justifications or nodes that are non-simple contextual implications.

$I_{js} =$

9 single justifications in context $(C_{2b} - C_1) +$
$(3 + 2)$ multiple justifications $+$
$= 14$ units

$I_{ns} =$

11 nodes in context $(C_{2a} - C_1) +$
$= 11$ units

I have omitted hypothesis (B.68) from these calculations.

#### Effort Approach

The numbers refer to the number of justifications or nodes that are contextual implications.

$I_{jl} =$

31 single justifications in context $(C_{2a} - C_1) +$
$(5 + 2)$ multiple justifications $+$
$= 38$ units

[6] This current ordering was left in for illustration only.

[7] I am choosing to ignore the effort incurred by reason maintenance as it could have been avoided by re-ordering. Incurring needless effort could be argued to be fixing the results.

$I_{nl} =$

33 nodes in context $(C_{2a} - C_1) +$
$= 33$ units

### B.5.1 Relevance

#### Simple Implications Approach

$$\mathcal{R}_{js} = \frac{I_{js}}{E_c} = 14$$

and

$$\mathcal{R}_{ns} = \frac{I_{ns}}{E_c} = 11$$

#### Effort Approach

$$\mathcal{R}_{jl} = \frac{I_{jl}}{E_l} = \frac{38}{24} = 0.125$$

and

$$\mathcal{R}_{nl} = \frac{I_{nl}}{E_l} = \frac{33}{24} = 1.375$$

### B.5.2 Sense Selection

Using all forms of measurement, the first sense of the utterance is more Relevant than the second sense. It is therefore restored as the preferred sense in the deductive device by popping the top element off the interpretation queue.

It can be observed that counting each justification as information results in a bigger difference between the Relevance Factors $\mathcal{R}_{jl}$ for the two contexts $C_{2a}$ and $C_{2b}$. This generally occurs between different senses because the inherent redundancy in natural language results in contextual inferences being "reinforced" many times over.

The crucial difference between $C_{2a}$ and $C_{2b}$ is that $C_{2a}$ allows another contextual implication which connects the utterance with the context: it assumes that the nut may be the food that Nutkin wants to eat. For all other examples, I will just show such critical inferences which distinguish the utterances.

### B.5.3 Ordering of Utterances

I explained in section 7.6.3 that it was possible for Crystal to automatically add default assertions that successive events follow each other. I have not illustrated this feature in the transcript so far. However, since the inferences obtained are common to both interpretations of the senses, I shall present the differences which are obtained when this feature is enabled.

The second utterance would have another conjunct on its rightmost side that states:

$$\cdots \wedge \mathcal{D}(Sk.Feel.Interval25 < Sk.Find.Interval26)$$

This will result in the assertion of the following guess at the end of the deductor memory:

$$(Sk.Feel.Interval25 < Sk.Find.Interval26)$$

This is consistent, and results in a series of inferences that include the following:

$$(Sk.Find.Interval26 <!i3) \supset (Sk.Feel.Interval25 <!i3)$$

$$(!i1 < Sk.Feel.Interval25) \supset (!i1 < Sk.Find.Interval26)$$

$$\neg(\mathrm{DURING}(!i, Sk.Feel.Interval25) \wedge \mathrm{DURING}(!i, Sk.Find.Interval26))$$

These should be not be contextual effects, since they were derived from the utterance. Unfortunately, due to the use of a reified temporal logic, they do provide contextual effects.

However, in situations where these inferences are overridden, the cancellation should be treated as a contextual effect. Crystal will handle this correctly by making appropriate checks when a node goes "out" during reason maintenance (see chapter 7.6.1).

## B.6 Obtaining The Other Sense of "Nut": Nut2

Consider another simple discourse:

- Nutkin is a squirrel.
- One day, he found a bolt.
- Afterwards, Nutkin found a nut.

This might be translated into:

$$\text{Squirrel}(\boldsymbol{nutkin}) \tag{B.76}$$

$$\begin{aligned}&\text{Day}(\boldsymbol{Sk.Day}002) \wedge \text{IT}(\boldsymbol{Sk.Male}26) \wedge \text{Male}(\boldsymbol{Sk.Male}26) \wedge \\ &\quad \text{Find}(\boldsymbol{Sk.Male}26, \boldsymbol{Sk.Bolt}05, \boldsymbol{Sk.Find.Event}17) \wedge \text{Bolt}(\boldsymbol{Sk.Bolt}05) \\ &\quad \text{OCCUR}(\boldsymbol{Sk.Find.Event}17, \boldsymbol{Sk.Find.Interval}27) \wedge (\boldsymbol{Sk.Find.Interval}27 < \boldsymbol{NOW}) \wedge \\ &\quad \text{DURING}(\boldsymbol{Sk.Find.Interval}27, \boldsymbol{Sk.Day}002)\end{aligned} \tag{B.77}$$

$$\begin{aligned}&\text{OCCUR}(\boldsymbol{Sk.After.Event}12, \boldsymbol{Sk.After.Interval}07) \wedge \\ &\quad \text{IT}(\boldsymbol{Sk.After.Event}12) \wedge \text{IT}(\boldsymbol{Sk.After.Interval}07) \wedge \\ &\quad (\boldsymbol{Sk.Before.Interval}03 < \boldsymbol{Sk.After.Interval}07) \wedge \text{Nut}(\boldsymbol{Sk.Nut}11) \\ &\quad \text{Find}(\boldsymbol{nutkin}, \boldsymbol{Sk.Nut}11, \boldsymbol{Sk.Find.Event}18) \wedge \\ &\quad \text{OCCUR}(\boldsymbol{Sk.Find.Event}18, \boldsymbol{Sk.Before.Interval}03) \wedge \\ &\quad (\boldsymbol{Sk.Before.Interval}03 < \boldsymbol{NOW})\end{aligned} \tag{B.78}$$

During the course of the processing of these utterances, it is determined that:

$$(\boldsymbol{Sk.Male}26 = \boldsymbol{nutkin})$$

$$(\boldsymbol{Sk.Before.Interval}03 = \boldsymbol{Sk.Find.Interval}27)$$

$$(\boldsymbol{Sk.After.Event}12 = \boldsymbol{Sk.Find.Event}17)$$

The guesses about intervals are made from the "<" information. The event guess is made from the "OCCUR" information.

The sense of the utterance that contains concept "Nut2" invokes a script similar to that of (B.62):

$$\begin{aligned}&\text{Bolt}(!b) \supset \text{Attach}(\text{Sk.Solid24.Of}(!b), \text{Sk.Solid25.Of}(!b), \text{Sk.Attach.Event77.Of}(!b)) \wedge \\ &\quad \text{Using}(\text{Sk.Attach.Event77.Of}(!b), !b) \wedge \\ &\quad \text{Using}(\text{Sk.Attach.Event.Of}(!b), \text{Sk.Nut26.Of}(!b)) \wedge \\ &\quad \text{Solid}(\text{Sk.Solid24.Of}(!b)) \wedge \text{Solid}(\text{Sk.Solid25.Of}(!b)) \wedge \\ &\quad \neg(\text{Sk.Solid24.Of}(!b) = \text{Sk.Solid25.Of}(!b)) \wedge \text{Nut2}(\text{Sk.Nut26.Of}(!b)) \wedge \\ &\quad \text{Adjacent}(\text{Sk.Solid24.Of}(!b), \text{Sk.Solid25.Of}(!b))\end{aligned} \tag{B.79}$$

This will be instantiated with "Sk.Bolt05". Subsequently, the following inferences are made:

$$\boldsymbol{Sk.Nut}11 = \text{Sk.Nut26.Of}(\boldsymbol{Sk.Bolt}05) \tag{B.80}$$

$$\text{Sk.Solid24.Of}(\boldsymbol{Sk.Bolt}05) = \text{Sk.Solid22.Of}(\text{Sk.Nut26.Of}(\boldsymbol{Sk.Bolt}05))$$

$$\text{Sk.Solid25.Of}(\boldsymbol{Sk.Bolt}05) = \text{Sk.Solid23.Of}(\text{Sk.Nut23.Of}(\boldsymbol{Sk.Bolt}05))$$

$$\text{Sk.Bolt5.Of}(\text{Sk.Nut26.Of}(\boldsymbol{Sk.Bolt}05)) = \boldsymbol{Sk.Bolt}05$$

(B.80) equates the "Nut2" of the second utterance with the nut associated with the bolt of the first utterance. This results in all of the inferences associated with this object becoming non-simple implications. The two subsequent guesses equate the objects being attached to each other in the instantiations of the "nut" and "bolt" scripts. The final equality equates the bolt of the first utterance with the anonymous bolt of the second utterance. Clearly, the second interpretation will result in a higher Relevance factor due to these connections as they either overcome the loading effort or result in more non-simple implications, depending on the measuring method selected. Informally, this interpretation has been selected because it can be seen how the nut and bolt might be used together.

## B.7 Continuations of the First Discourse

The discourse given in the first section has been extended as follows:

1. Nutkin felt hungry.
2. He found a nut.
3. He took the nut.
4. Nutkin is a penguin.

The third utterance is intended to illustrate how Relevance considerations can disambiguate "nut" by virtue of the fact that the concept has already been mentioned. The fourth utterance shows how Relevance can be achieved by updating extant beliefs using the RMS.

One modification I made to the encyclopædic memory before processing this sequence was to add more information about "Nut1". "Nut1" was strengthened so that it was known that "Nut1"s are not fish. This is a rather specific piece of information that is only used to help detect inconsistencies; a better way of detecting them would be to use the sort hierarchy (which already contains this information implicitly). Thus, the entry for Nut1 includes:

$$\mathrm{Nut1}(!n) \supset \neg\mathrm{Fish}(!n) \tag{B.81}$$

The entry for penguin is:

$$\begin{aligned}
&\mathrm{Penguin}(!p) \supset \mathrm{Bird}(!p)\\
&\mathrm{Penguin}(!p) \supset \neg\mathrm{Mammal}(!p)\\
&\mathrm{Penguin}(!p) \supset \neg\mathrm{Flies}(!p)\\
&\mathrm{Penguin}(!p) \supset \mathcal{D}\mathrm{Antarctic}(!p)\\
&\mathrm{Penguin}(!p) \supset \mathrm{Black}(!p)\\
&\mathrm{Penguin}(!p) \supset \mathrm{White}(!p)\\
&\mathrm{Penguin}(!p) \supset \mathrm{Swims}(!p)\\
&\mathrm{Penguin}(!p) \supset \mathrm{Large}(!p)\\
&\mathrm{Penguin}(!p) \wedge \mathrm{Food}(!f) \wedge\\
&\qquad \mathrm{Eats}(!p, !f, !e) \supset \mathrm{Fish}(!f)
\end{aligned} \tag{B.82}$$

i.e. Penguins are large, black and white, non-flying birds that swim. They are typically Antarctic and eat only fish (as food).

The entry for "Take" includes:

$$\mathrm{Take}(!t1, !t2, !e) \wedge \mathrm{OCCUR}(!e, !t) \supset \mathrm{Has}(!t1, !t2, !t)$$

### B.7.1 Processing Results

#### Third Utterance

The logical form is:

$$\begin{aligned}
&\mathrm{REF}(Sk.Nut13) \wedge \mathrm{Take}(\boldsymbol{nutkin}, Sk.Nut13, Sk.Take.Event02) \wedge\\
&\qquad \mathrm{Nut}(Sk.Nut13) \wedge \mathrm{OCCUR}(Sk.Take.Event02, Sk.Feel.Interval25)
\end{aligned} \tag{B.83}$$

Note that the translation of "the" uses "REF" rather than "IT".

Processing the third utterance disambiguates the "Nut". "Nut1" is chosen because it requires no loading effort and can abductively unify with the previous nut:

$$(Sk.Nut10 = Sk.Nut13) \tag{B.84}$$

It is then possible to draw the inference:

$$\begin{aligned}
&\mathrm{Eats}(\boldsymbol{nutkin}, \mathrm{Sk.Food.Of}(Sk.Feel.Interval25, \boldsymbol{nutkin}),\\
&\qquad \mathrm{Sk.Eats.Event.Of}(Sk.Feel.Interval25,\\
&\qquad\qquad \mathrm{Sk.Food.Of}(Sk.Feel.Interval25, \boldsymbol{nutkin}), \boldsymbol{nutkin}))
\end{aligned} \tag{B.85}$$

And the related inference that the above event will "OCCUR". These assert that Nutkin will eat the nut. They are both derived from the script-like assumption schema, (B.36).

## Fourth Utterance

The utterance representation for the fourth utterance is:

$$\text{Penguin}(nutkin) \tag{B.86}$$

Much as this may not appear to be very Relevant, the utterance becomes Relevant because it incurs belief revision of the assumptions already present in the deductor memory. In other words, the utterance forces us to update parts of the existing context, rather than just augmenting it. From the entry for penguins, it is inferred that Nutkin is not a mammal. This results in an inconsistency, because it was assumed earlier that Nutkin *was* a mammal (B.18). The RMS is invoked, and (B.17) ("Squirrel($nutkin$)") rejected as the culprit. This is a contextual effect, as are the subsequent retractions of its believed repercussions (which include all assertions about squirrels), because they depend on that node being "in".

It is also inferred that the food that Nutkin eats must be a fish, due to rule (B.82):

$$\text{Fish}(Sk.Nut10) \tag{B.87}$$

but in the previous (and therefore in the current) context we have that

$$\neg\text{Fish}(Sk.Nut10)$$

because of rule (B.81)[8]. An inconsistency node is asserted.

This results in the invocation of the dependency-directed backtracking component of the RMS. It is decided that (B.84) and (B.50) are "nogood". The latter is made the culprit because it is not a reference, but it is otherwise the closest asserted node chronologically.

The rejection of this hypothesis results in subsequent rejection of (B.81), causing more information gain. The inconsistency then disappears. In effect, Crystal has decided that, being a penguin, Nutkin cannot possibly eat the nut, *Sk.Nut10*, because he only eats fish.

[8] This rule is rather contrived. It would have been better to detect such inconsistencies using the sort hierarchy. The same is true for the penguin/squirrel conflict.

# Appendix C

# Crystal's RMS

In this appendix, I provide a glossary of terminology concerning Doyle's RMS, followed by details of the RMS implementation within Crystal (see chapters 7, 8 and 9 for further information concerning its use).

## C.1 Glossary of RMS Terminology

In this section, I provide a glossary of terminology associated with Doyle's RMS. I have divided it into three sections. The first defines attributes associated with a typical RMS node. The second defines notions derived from the terms in the first section. The final section defines various ways of describing the validity of a justification.

### C.1.1 Properties of a Node

**Justification:** An ordered pair, consisting of an *inlist* and an *outlist*. A node is "in" if all of the nodes in one of its justification's inlist are "in" and the nodes in its outlist are "out".

**Justification Set:** The set of all current justifications, whether valid or not, for a particular node.

**Support Status:** One of "in", "out", or "nil".

- "in" means that the node is currently believed because it has at least one valid justification in its justification set.
- "out" means that the node is not currently believed because it does not have any valid justifications in its justification set.
- "nil" or "indeterminate" means that the node is currently undergoing the process of belief revision. This is a value used by the reason maintenance system rather than the separate inference engine.

**Supporting Justification:** A single valid justification which makes the node "in".

**Supporting Nodes:** Depending on whether a node is "in" or not:

*In:* The union of the nodes in the inlist and outlist of the supporting justification — the antecedents.

*Out:* For each justification in the justification set, an "in" node from the outlist or an "out" node from the inlist.

### C.1.2 Terms Derived from Properties of Nodes

**Antecedents:** The union of the nodes in the inlist and outlist of a supporting justification.

**Foundations:** The transitive closure of the antecedents of a node, i.e. all nodes which are used to prove that the particular node is "in".

**Ancestors:** The transitive closure of the supporting nodes of a particular node, i.e. all nodes that might ultimately affect the node's support status.

**Consequences:** All the nodes which mention that node in their justification set.

**Affected Consequences:** Just those consequences of a node which mention the node in their supporting nodes.

**Believed Consequences:** The "in" consequences of a node that have that node as an antecedent — the affected consequences that are "in".

**Repercussions:** The transitive closure of the affected consequences of a node.

**Believed Repercussions:** The transitive closure of the believed consequences of a node.

### C.1.3 Definitions Concerning Justifications

The following are true of a justification if it satisfies the constraints in their definition.

**Well-Founded-Valid:** All the nodes in the inlist are "in", and all those in the outlist are "out".

**Well-Founded-Invalid:** At least one node in the inlist is "out" or at least one node in the outlist is "in".

**Not-Well-Founded-Valid:** All nodes in the inlist are "in" and all nodes in the outlist are not "in" (i.e. "out" or "indeterminate").

**Not-Well-Founded-Valid:** The justification cannot be classified by any of the previous descriptions.

Justifications are always checked for well-founded-validity or well-founded-invalidity first. If this fails, a constraint relaxation algorithm is applied in order to determine a node's support status using the weaker definitions. In Crystal, which uses simple defaults incrementally, this never proved to be necessary.

## C.2 Crystal's RMS Structures

Nodes in Crystal's RMS were structures with the following fields:

**Clause:** This contains the actual assumption associated with the node.

**Support Status:** This will be "in", "out", or "indeterminate". "in" means "currently believed", "out" means "not currently believed", and "indeterminate" means "being considered by the reason maintenance process". When a node changes its support status, its repercussions' support statuses must be re-evaluated in case their justifications have changed. Each repercussion is temporarily marked as "indeterminate" to show they have not yet been re-evaluated by the reason maintenance process.

**Justification Set:** This is a set of all of the potential reasons (justifications) for believing the node should be "in". Only one of these, the *supporting justification*, is used to determine the node's support status.

**Supporting Nodes:** If the node is "in", it is the set of nodes that make up the supporting justification. If the node is "out", then for each justification, it is an "in" node from its outlist, or an "out"node from its inlist.

**Consequences:** This is the set of nodes that mention the node in their justification set.

All clauses have the following fields:

**Content:** This is the logical form itself.

**Context Type:** This is one of *context, contextual implication, memory or utterance.* The information is required for the determination of contextual implications as explained in chapter 5.

All justifications have the following fields:

**Inlist:** The list of nodes that need to be "in" to make the justification valid.

**Outlist:** The list of nodes that need to be "out" to make the justification valid.

**Reason:** The reason for this justification, for example "modus ponendo ponens" or "contradiction".

## C.3 Graphical Browsers

The justifications, consequences and clauses of nodes are used by the querying system to build graphical representations of the inferential structures contained within the deductive device (see appendix A and figure A.7).

When graphs are built, *all* of the valid justifications, rather than just the supporting justification, are used to form arcs. This is because every valid justification counts as a contextual effect.

Nodes are given helpful identifiers to help the user immediately identify how the node originated (this information is not used by the system). This is particularly useful when examining graphical representations. Memory entries are always named after the associated concept (e.g. cougar/001), utterances are always named "utt/nnn", where *nnn* increases with subsequent utterances in a discourse stream, normal inferences are similarly named "inf/nnn", non-monotonic inferences "nm/nnn" and hypotheses, "abd/nnn".

The graphs display only the node identifiers and their "in" and "out" links (obtained from the justifications or consequences). The querying system optionally highlights the nodes that are "in" and those that are contextual implications. Further information about a node can be obtained by clicking the mouse on a particular node to produce a menu. Selection of these menu items allows information about the contents of the node, its context type, and its justifications to be inspected.

The main utility of the graphs is to provide a quick visual summary of the contents of the deductive device at any time, and also to show the connectivity of successive utterances in a discourse, which is a central notion of Relevance.

## C.4 Simple Queries

Crystal can illustrate derivations using the RMS structures present in the deductive device. The existence and support status of any node can be explained by tracing back through its ancestors. A typical explanation is:

Animal (*cougar002*) (**inf/004**)
    was derived by *modus ponendo ponens* from **utt/003** and **cougar/001**.
        Cougar(*cougar002*) (**utt/003**) is an utterance.
        Cougar(!*c*) $\supset$ Animal(!*c*) (**cougar/001**) is an
            encyclopædic entry.

It is ensured that cyclic explanations are avoided, by noting which of the foundations have already been explained, and ensuring that these explanations are not repeated.

## C.5 Alternative Implementations

The reason maintenance and dependency-directed backtracking algorithms described in this thesis belong to a class of systems known as *justification-based truth maintenance systems* (JTMS). However, there are alternative approaches, the most notable of which is De Kleer's *assumption-based truth maintenance system* or ATMS (deKleer, 1986a). I shall briefly justify my decision to employ a JTMS as opposed to an ATMS.

### C.5.1 The ATMS

A problem with the JTMS is its inefficiency in exploring multiple extensions. The ATMS maintains all of the possible extensions simultaneously (deKleer, 1984)[1]. Thus, switching extensions becomes a trivial matter. Defaults can also be implemented within the system (deKleer, 1986c). Another possible implementation of Crystal could have maintained all of the possible extensions and measured their Relevance, and then selected the most apt. The main problems with adopting this approach are:

1. There will be too many extensions for the ATMS to handle efficiently, especially with the introduction of hypotheses of equality (see chapter 8).
2. The Relevance of an utterance representation may vary wildly in various extensions, being very low in some and very high in others; it would be foolish to maintain all of these extensions needlessly.

[1]This paper also highlights various other deficiencies of reason maintenance systems. Some of them are fundamental.

3. Maintaining all possible extensions seems a poor model of human reasoning from a Relevance viewpoint; the number of extensions increases with successive utterances. Intuitively, the converse should occur since utterances are effectively constraining the possible interpretations.
4. The semantics of the ATMS is less well documented than that of the JTMS. It is not clear as to how it ties up with the various non-monotonic formalisms discussed in this thesis.

The problem of having too many extensions has been partially solved by allowing the ATMS to maintain a few extensions and backtrack through the rest when necessary (deKleer, 1986b). This introduces the further problem of how to decide which subset of extensions should be maintained. However, the backtracking ATMS remains an interesting alternative to the JTMS.

### C.5.2 Other Consistency Maintenance Systems

Most other systems are variants on the JTMS or the ATMS. A good survey of these can be found in deKleer (1986d).

# Appendix D

# A Sample Set of Logical Rules

The following are the logical (meta-) rules employed by Crystal's deductive device.

## D.1 Analytic Rules

**Identifier:** And Elimination (a)
**Trigger:** $!A \wedge !B$
**Inference:** $!A$

**Identifier:** And Elimination (b)
**Trigger:** $!A \wedge !B$
**Inference:** $!B$

## D.2 Synthetic Rules

**Identifier:** Conjunctive Modus Ponens (a)
**Trigger 1:** $(!A \wedge !B) \supset !C$
**Trigger 2:** $!A$
**Inference:** $!B \supset !C$

**Identifier:** Conjunctive Modus Ponens (b)
**Trigger 1:** $(!A \wedge !B) \supset !C$
**Trigger 2:** $!B$
**Inference:** $!A \supset !C$

**Identifier:** Disjunctive Modus Ponens (a)
**Trigger 1:** $(!A \vee !B) \supset !C$
**Trigger 2:** $!A$
**Inference:** $!C$

**Identifier:** Disjunctive Modus Ponens (b)
**Trigger 1:** $(!A \vee !B) \supset !C$
**Trigger 2:** $!B$
**Inference:** $!C$

**Identifier:** Modus Tollendo Ponens (a)
**Trigger 1:** $!A \vee !B$
**Trigger 2:** $\neg !A$
**Inference:** $!B$

**Identifier:** Modus Tollendo Ponens (b)
**Trigger 1:** $!A \vee !B$
**Trigger 2:** $\neg !B$
**Inference:** $!A$

**Identifier:** Modus Ponendo Ponens
**Trigger 1:** $!A \supset !B$
**Trigger 2:** $!A$
**Inference:** $!B$

**Identifier:** Contraposition
**Trigger 1:** $!A \supset !B$
**Trigger 2:** $\neg !B$
**Inference:** $\neg !A$

**Identifier:** Inconsistency
**Trigger 1:** $!A$
**Trigger 2:** $\neg !A$
**Inference:** INCONSISTENT$(!A, \neg !A)$

# Appendix E

# Brief Annotated Examples

In this appendix, I have collected together some examples that Crystal processes in order to illustrate its performance and capabilities. Rather than provide the contents of the deductive device after each discourse fragment is processed, I just point out the inferences and features of significance. In the final section, I discuss a way of extending Crystal's coverage to handle more aspects of Gricean generalised conversational implicature.

The following example proved to be somewhat controversial:

> Tweety, the bird, felt hungry. She found a nut.
> Tweety is a penguin. She found a fish.

This discourse was the one that I originally intended as the elaborated example in appendix B.

My understanding of this discourse is that, at first, it seems reasonable that Tweety may eat the nut, which I assume to be food. I also realise that Tweety must be female. However, after I am told that Tweety is a penguin, I realise that she cannot eat the nut because she only eats fish. I also revise my "mental image" of Tweety, changing it from a stereotypical flying bird, to a stereotypical non-flying penguin. In the final utterance, I believe that Tweety ultimately finds something which she may eat.

Unfortunately, other people found this discourse less easy to understand, largely because they did not have the same intuitions about birds and nuts. In SW's terms, it was not mutually manifest that birds like to eat nuts or that penguins do not (or, at least, rarely have the opportunity to) eat nuts. However, since the encyclopædic memory contains representations of some of *my* understanding of these concepts, Crystal unsurprisingly drew similar conclusions to those that I did. This is totally in accord with my understanding of RT. I assume that when I produce an utterance, I do not have access to the mutual cognitive environment, but predict its contents by reflecting on my own cognitive environment and considering my knowledge of the audience. It follows that I can make mistakes about the audience's beliefs and consequently generate utterances which are meaningless to them.

The significant aspects of Crystal's processing of this utterance are as follows. The first utterance results in the default inference that Tweety can fly. It is also assumed that there is some food that she would like to eat. In the next utterance, it is guessed that Tweety is female. This inference is drawn from the fact that Tweety must be a person, because only people feel hungry, and the fact that females are people. Two versions of the second utterance are examined, one with the sense of "nut" as the food, the other with the sense of "nut" as the metal artefact. The former is preferred because it links better with the context; it is hypothesised that the nut may be the food that Tweety would like to eat. This expectation is proved incorrect by the next utterance. Tweety, being a penguin, does not eat nuts. This utterance becomes Relevant by cancelling the default flying information and the food-eating hypothesis. Note that the sense of "nut" is not revised, which is in accordance with my own intuitions. In the final utterance, the fish found by Tweety is hypothesised to be the food that she may eat.

The next example illustrates how the ordering of constituents in the utterance representation is significant.

> Olga took the bucket to her car.
> She washed it.

Normally, "bucket" will be the point of focal stress, and will consequently be the last constituent processed. Therefore, the first utterance has the logical form:

$$\text{Take}(Olga, Sk.Bucket2, Take.Event) \cdots \wedge \text{Bucket}(Sk.Bucket2)$$

This results in the guess that "it" is the bucket. However, part of the knowledge for "washing" is that people do not wash buckets. Therefore, the guess of equality is retracted, and subsequently the guess is made that "her car" is the referent of "it".

If stress is placed on "her car", this becomes the last constituent of the representation, resulting in it being picked up as the first compatible co-referent on the IT list. This means that this reading is more Relevant because it does not expend any effort in reason maintenance.

A further abductive inference generated is that Olga may travel in the car.

The following discourse fragment illustrates how sense disambiguation does not always require a non-empty context.

> Nutkin swallowed the stone of a cherry.
> It was deliberate.

It is realised that "stone" is the stone of a fruit rather than a rock because it yields more inferences when combined with the information about cherries. It is also realised that the fruit associated with the stone frame is the cherry and that the stone associated with the cherry frame is the stone mentioned in the utterance. Furthermore, since the stone is inedible, it is assumed by default that the swallowing is an accident. The next utterance is Relevant because it overrides the default assumption, which is given up in preference to the more "epistemically entrenched" definite reference.

"it" is known to be an event because it has an adverbial modifier. It is therefore easy to find a suitable co-referent.

The following example illustrates a phenomenon similar to retroactive strengthening:

> Jerzy is a bird.
> He lives in a nest.
> One day, Jerzy felt hungry.
> He left his nest.
> He found a nut.
> He took it.
> He ate it.

This is similar to some of the previous examples, but somewhat longer. However, in this example, our expectations are satisfied. In the first utterance, it is guessed that Jerzy can fly. In the next utterance, it is realised that "he" is Jerzy. "he" is also guessed to be the builder of the nest. Next, it is realised that since Jerzy is hungry, he needs to eat some food, after which he will no longer be hungry. In the next utterance, it is hypothesised that "he" is Jerzy, and that the nest is the one mentioned earlier; it follows that "his" means "Jerzy's". When Jerzy goes out, it is assumed that he is not adjacent to his nest. When he finds the nut, it is guessed that it is the food he needs because it will then follow that it fits into the "hunger" script. When he takes the nut, it is deduced from the hypothesis that it fits an eating script that he will eat it during the time that he feels hungry. In the next utterance, Jerzy does indeed eat the nut. This event is abductively guessed to be identical to the prototypical hunger event. This guess is considered to be a contextual effect, and can be seen as a retroactive strengthening of the eating hypothesis; the Relevance of this interpretation increases because the eating hypothesis proved to be correct.

Consider the following:

> Jerzy is a bird.
> He lives in a nest.
> One day, Jerzy felt hungry.
> He went out.
> He found a bolt.
> He found a nut.

This is similar to the previous discourse, except that the bolt primes the system to recognise the other sense of "nut", because it guesses that they both can take part in a prototypical "attaching" event. In fact, Crystal goes as far as to hypothesise that Jerzy might attach his nest to the place where he stands using the nut and bolt. However, it rejects this hypothesis because his nest is not accessible (Crystal has inferred by default from "going out" that Jerzy is no longer adjacent to it). The utterance "he found a bolt" seems a little contrived. However, the same sense of "nut" is obtained with a more interesting discourse when the third utterance is replaced with "he wanted to attach his nest to a tree" and the fifth utterance deleted.

The relative ordering of clauses within a discourse will effect their Relevance, even if they ultimately produce the same information within the deductive device. Consider the following two discourses:

a1. Mary slapped John.

a2. It killed him.

a3. Mary is a gorilla.

and

b1. Mary is a gorilla.

b2. She slapped John.

b3. It killed him.

The sequence prefixed by 'a' is processed as follows. It is assumed that slapping someone does not seriously hurt them if there is no reason to believe otherwise. Next, this default is overridden by (a2). The deductive explanation for this is in the form of (a3); Crystal reasons that if gorillas hit someone, it typically hurts them seriously. Information about killing includes the fact that seriously hurt people die without some reason to believe otherwise. In this discourse, (a2) achieves extra Relevance because it overrides a default.

By contrast, on reading (b2), it can be deduced (by default) that John is seriously hurt and will typically die. The next utterance only serves to reinforce this point. No defaults of type "context" are overridden, and thus there is less new information.

The encyclopædic entries for Mary and John include the assumption that they are typically human names. In the first dialogue, this inference for Mary is overridden relatively late (a3). This results in an erasure of type "context". By contrast, it is immediately stated that Mary is a gorilla in (b1), so the default is immediately overridden, yielding less information (since it can be directly asserted).

Thus, according to Crystal, dialogue "a" is more interesting than dialogue "b" because the last two utterances are more Relevant.

The following sequence illustrates how Relevance can be used to limit context expansion.

Petra the penguin was always happy.
She was sad.

Relevance can *constrain* context expansion by terminating the expansion process when the Relevance becomes too low. Crystal initially asserts that "she" is "Petra", however this immediately leads to an inconsistency, since someone who is sad cannot be happy, and Petra is always happy. This invokes the dependency-directed backtracking mechanism which retracts the abductive guess about the referent of "she". After the inference closure has been taken, "she" does not have a co-referent, so the context is expanded with information about "feelings", "wings", "birds", "flying" etc (i.e. the concepts present in the deductive device). This requires much more effort, but still a referent is not found. At this point, Crystal decides that the Relevance is too low and that "she" must be referring to another person.

Crystal's sense disambiguation mechanism is not entirely satisfactory, as can be seen by the way it handles the following discourse.

Thelonius went into the garden.
He saw a bulb.
He put it in a lamp.

The encyclopædic entry for gardens includes the information that gardens typically contain plants, and the entry for organic bulbs contains a script-like assumption where a bulb, if not obstructed, grows into a plant. Crystal consequently decides that the second utterance refers to an organic bulb because it may grow into a plant in the garden; the abductive unifier matches the plant in the bulb script to the plants typically found in a garden. When Crystal processes the third utterance, there is no reason for it to doubt that it is an organic bulb that Thelonius is putting into the lamp, so no inconsistency arises, and it is assumed that "it" is the organic bulb. However, according to my intuitions, the "bulb" in the previous utterance is an electric light bulb, rather than an organic bulb. Unfortunately, Crystal could not detect this problem, and even if it could, there is presently no way of handling incorrect word-sense choices.

In section 3.3.2, I claimed that it is often necessary to consider all senses of a word during incremental processing of an utterance rather than picking the first one available that makes an interpretation Relevant. The following two discourse fragments illustrate how Crystal can correctly disambiguate the two senses of "admitted" by explicitly calculating and comparing the Relevance of the two possible interpretations.

Consider the first fragment:

Jennifer stole a watch.
A policeman approached her.
Jennifer admitted stealing.

In this example, "admitted" is assumed to be an admonition of guilt about some "wrongdoing". This sense of admitted is only selected because "stealing" is known to be a wrongdoing[1]. This contrasts with SW's idea that word senses can be immediately determined at the point at which they are considered during incremental processing; by contrast, the whole sentence is considered by Crystal in order to select a sense. The other sense is not selected because there is nothing to admit the policeman through, and "stealing" is a wrongdoing event rather than an object to be admitted through something.

Now consider the second fragment:

Jennifer stole a watch.
She was in her house.
A policeman knocked on the door.
Jennifer admitted him into the house.

In this example, the other sense of admitted – to allow to enter – is chosen because it can be connected to the door, which forms a separate "through" argument of the verb, i.e. Jennifer admitted him through the door.

The next three examples illustrate the effect that defaults have on Relevance. The following two example are syntactically very similar, but the first is intuitively more Relevant than the second.

Abelard entered a church.
There is a crypt underneath the church.

Zadok visited a bungalow.
There is a crypt underneath the bungalow.

Considering the above two examples, the second utterance of the first discourse is more Relevant than that of the second. The reason for this is that a frame for churches includes the fact that they typically have crypts. Thus, the abductive unifier can explain why the crypt is underneath the church: it is the crypt that it expects to be there. In the second example, the existence of the crypt is not explained. It is hard to find any other links between the two utterances.

If the last example was extended as follows, the second utterance becomes more Relevant because the abductive unifier guesses that the crypt belongs to the church.

Zadok visited a bungalow.
It is built next to a church.
There is a crypt underneath the bungalow.

The following is a discourse fragment, the last sentence of which is Relevant because the obvious prediction that people tend to make from the first two utterances proves to be wrong.

John felt suicidal.
There was a gun beside him.
The gun had run out of caps.

The second utterance fits into a schema for suicides, because it is assumed to be a weapon by default. Thus, it is assumed that John may kill himself with the gun. The third utterance has two readings: one involving ammunition for a toy gun, the other involving hats. The first of these is selected because it links into the context and cancels the "suicide" hypothesis by the following argument. Since the gun has run out of caps, it can contain caps. If a gun can contain caps, it is assumed that the gun uses caps, and if the gun uses caps, then it is a toy. This overrides the default inference that the gun can be used as a weapon. If the gun cannot be used as a weapon, then the inferences contingent on this are erased by the RMS, including the fact that John will kill himself. The alternative reading does not override the hypothesis and also does not provide many contextual implications; it is rejected because it is less Relevant.

It was explained in section 7.6.3 how events portrayed in successive utterances could be assumed to follow each other by adding an extra conjunct that asserts such facts by default. This conjunct can optionally be added automatically by Crystal. The following discourse illustrates how these default assumptions may be used to help explain how certain utterances becomes Relevant. Consider the following:

[1] It is interesting to note that Longmans does not explicitly mention the fact that stealing is a wrongdoing of any kind.

Rambo went into his house.
He cleaned his poodle.
He went in after he cleaned his poodle.

The utterance browser in which the dialogue was entered automatically adds default inferences that instruct the deductive device to assume that progressive utterances occur in the order in which they are mentioned. However, being defaults, these can be cancelled. In this example, cancellation occurs in the processing of the last utterance. The abductive unifier assumes that the cleaning event in the last utterance is the cleaning event in the second utterance and that the "going in" event is the"going in" event of the second utterance. At this point, it is inferred that the event in the first utterance cannot precede the event in the second, and the default event sequencing assumption is overridden. The purpose of the last utterance was presumably to do just that; it thus becomes more Relevant because it cancels the default of contextual status "Context". If the last utterance had been "He went in before he cleaned his poodle" it would have been less Relevant; even though the cleaning and "going in" events co-refer with those in the second utterance, the default sequencing is not overridden.

The next two examples sketch a way of handling some of Blakemore's semantic directives (see section 7.6.4). The following two examples have been processed by Crystal, but with some manual intervention. I instructed Crystal when to expand the context (using the deductor browser menus described in appendix A), and manually checked for the overriding of defaults or the presence of contradictions which indicated that the goal requested by the semantic directive had been achieved.

The first example illustrates a naïve processing of "but".

Angus is from Edinburgh but he does not have a strong accent.

I began by adding the first conjunct and processed it as if it was a normal utterance representation. Next, I marked the contents of the deductor memory as having contextual status "Context". After this, I added the second conjunct and processed it as if it was a normal utterance representation. I assumed that the pragmatic goal for "but" was that the context should be expanded until one of the following was the case:

1. An assumption of contextual status "Context" overrode a default inference of some other contextual status.
2. A default inference of contextual status "Context" was overridden by an assumption of some other contextual status.
3. An assumption of contextual status "Context" contradicted an assumption of some other contextual status.

In this case, the first conjunct results in the default inference that, because people from Edinburgh typically have strong accents, Angus is assumed to have one. This inference is then marked as having contextual status "Context" and subsequently overridden by the second conjunct. Thus, the goal for "but" has been satisfied. The utterance is Relevant enough, and the context need not be expanded any further.

The next utterance is a similar example, but it requires further context expansion before the pragmatic goal is achieved.

Angus owns a silk dress even though he is a vegan.

"even though" is processed manually as for "but", and provides an identical pragmatic goal. As for the last example, the first conjunct is added and processed, followed by the second[2]. The context at this stage includes the information that:

1. Angus owns a dress (from the utterance).
2. The dress is composed of silk (from the utterance).
3. Silk is produced by silkworms, which are exploited (from the encyclopædic entry for "silk").
4. Angus is a vegan (from the utterance using abductive unification).
5. Vegans do not own anything that is composed of something produced by the exploitation of animals. (from the encyclopædic entry for "vegan").
6. Angus does not own anything composed of something produced by the exploitation of animals (by deduction).

[2]I assume the input processes translate "silk dress" into "dress made of silk". This is not a totally legitimate action for an input process (which has no access to encyclopædic knowledge).

The interface does not indicate that a contradiction has been found, because the deductive device does not yet contain the information that silkworms are animals.

The context is then expanded to include all of the concepts in the deductor memory, including "silkworm", "produces", "exploited", "material", "clothes" etc. Of course, the Relevance drops hugely. However, a contradiction is then found: Angus cannot own anything composed of silk and he owns a silk dress. The role of "even though" is to force the deductive device to find an appropriate contradiction or override a default, as long as it is Relevant to attempt this. If a contradiction had not been found at this stage, the Relevance is too low to merit another expansion.

Another example of a word whose semantics employs a directive concerning pragmatic processing is "before". Consider the following two separate utterances:

> Shylock dropped his wallet before he left the ghetto.

> Before he left the ghetto, Shylock dropped his wallet.

The two examples above exhibit factual readings of "before". In the second utterance, a cataphoric reference is successfully resolved. The processing order is such that the inferences derived from first conjunct are explored before the second.

In the first example, the second conjunct is translated as "if he left the ghetto, then it happened after he dropped his wallet". A further autoepistemic directive is added as the final conjunct in order to assume that "he left the ghetto", if possible. This assumption is found to be consistent, so a factual reading is obtained. In the second example, the ordering of the conjuncts are reversed; the first conjunct from the first example is asserted last. In this case, changing the order makes no difference to the final result.

In both cases, the pronominal reference is resolved because the interface checks for an abductive match when adding a conjunct. In the first utterance representation, the abductive unifier resolves the reference because, from the name "Shylock", it is inferred that Shylock is a person, and so is the "he" who left. The unifier then guesses that the two people are identical. The same thing occurs in the second example, "he" is first inferred to be a person because "people leave". Next it is inferred that Shylock denotes a person. The unifier guesses these to be the same.

Additionally, the entry for Shylock includes the information that he typically lives in a ghetto. It is consequently abductively assumed that "the ghetto" is "the ghetto Shylock lives in".

In contrast to the previous two utterances, the order of the conjuncts is important for the following:

> The car stopped before it hit the tree.

> Before it hit the tree, the car stopped.

These two examples have already been discussed in considerable detail in section 7.6.4. The examples are processed as for those using "before" above. In both cases, adding the fact that the car hit the tree results in an inconsistency. However, the different orderings of the constituents result in the rejection of different assumptions. In the first case, "the car hit the tree" is rejected and its negation brought "in" because it is already known that the car has stopped. This results in a counterfactual reading. In the second case, it is consistent to add the information that the car hit the tree. However, a contradiction occurs when it is discovered that the car was stationary. In this case, the culprit selection heuristics, explained in section 7.5.2, reject a different default because it has been applied more recently: the assumption that something that stops at time $t$ is also stationary at time $t+1$. It is effectively assumed that the car must have started moving again, producing a factual reading.

The next example illustrates how an utterance, similar in syntactic structure to the first utterance of the previous example, is processed to give a factual interpretation.

> The car stopped before the cyclone started.

In this case, the factual reading is obtained. Crystal does not make the inference that perhaps the car stopped to avoid any danger. This inference is arguably desirable, but I found it hard to encode the required information in a plausible manner. I could have added the information that cyclones are dangerous to vehicles in the entry for cyclone, and that typically cars stop under dangerous conditions under the entry for cars, but this seemed a little too artificial.

The following also yields a factual reading:

> Before he summoned a demon, Tim lit some candles.

However, the reading is made even more Relevant, because a script for summoning demons includes lighting candles. Thus, it is assumed abductively that Tim lit the candles in order to summon the demon.

The next example illustrates a property of abductive equality.

Edwina was a shark.

One day, she looked for some fish.

When the first utterance is processed, information about sharks is loaded, which includes the fact that sharks eat fish in order to survive. The fish in the second utterance are abductively assumed to be those in the first, i.e. Edwina is assumed to be looking for fish to eat in order to survive. We can observe that the script for "sharks eating fish" has been enriched; it has been instantiated and extended in such a way that it now states that Edwina is looking for fish to eat in order to survive.

The next example illustrates various features of the reference resolution mechanism within Crystal.

Pepys entered the inn.

He asked for the roast ox.

The significant inference made here is that Pepys asks the innkeeper for the ox. As explained in chapter 8, the reference resolution process operates using the same mechanism as that which performs enrichment (although pronominal references are restricted to co-refer with terms on the IT list). The roast ox matches some anonymous food which the innkeeper can serve. Furthermore, the addressee of the asking is assumed to be the innkeeper (the information about "innkeepers" is asserted after information about Pepys; furthermore, people typically do not ask themselves things).

In the following discourse, Relevance is used to explain how a particular reading is obtained because it requires less loading effort. Effectively, Relevance considerations often favour "sticking to the subject". Consider the following:

Zebedee drank water from a spring.

After that, he came across a dirty spring.

In the first utterance, the "spring" is assumed to be a pool of water rather than the coiled metal or the season. This is because its contents, some water, will unify with the water that Zebedee drank. The first utterance has a logical form that includes a "from" predicate, which takes the drinking event and the object of that action.

In section 5.5.1, I discussed how verbs modified by adjuncts were given meaning using assumption schemas: arguments become part of the main predicate, and the meaning of adjuncts are provided in separate assumption schemas. A good example of the separation of arguments and adjuncts can be seen in the encyclopædic entry for "drink". This entry not only includes information about drinking, but also includes information about drinking modified by "from", etc. So, for example, there will be assumption schemas that simply state that "drinking" involves an agent ingesting a fluid. However, the will also be a more specific assumption schema for "drinking from", which states that the containing area holds the fluid, and that it is not empty during the entire period of drinking, i.e.

$$\begin{aligned}
&\mathrm{Drinks}(!h, !w, !e) \wedge \mathrm{From}(!e, !s) \wedge \mathrm{OCCUR}(!e, !t) \supset \\
&\quad \mathrm{Area}(!s) \wedge \mathrm{Contains}(!s, !w) \wedge \\
&\quad (\mathrm{DURING}(!t1, !t) \supset \neg\mathrm{HOLDS}(\mathrm{Empty}(!s), !t1))
\end{aligned} \tag{E.1}$$

The processing of the second utterance representation results in the abductive guess that "he" is Zebedee and that the spring has the same sense as in the first utterance, since this requires less loading effort because it is already present in the deductor memory.

There are two readings for the following utterance. SW suggest that the "limb" reading is the most Relevant. I argued in chapter 6 that the "height" reading is better because the other results in effort-consuming inconsistency resolution. Crystal also produces this interpretation.

My son has grown another foot.

Only the sense of "foot" as a distance is acceptable, because the entry for "foot" states that humans cannot have more than two feet (and typically they have two). This reading results in an inconsistency, which would require a revision of an encyclopædic entry to resolve and therefore vast amounts of effort. On the other hand, for the alternative reading, it is assumed that it is possible to grow a foot in height. This reading is therefore preferred because it requires less effort to be understood.

The following illustrates how the sense of a particular utterance depends on whether it is uttered in isolation (an empty context) or in an existing (non-empty) context, and how Crystal is not presently equipped to handle this in a satisfactory manner. Consider the following:

George Best walked to the ball.

Encyclopædic information about the concept "*george.best*" is loaded. This includes information about footballers and football. Thus, the reading of "ball" as football is preferred, because it fits in with a script-like assumption concerning football playing. Now consider a somewhat different context:

George Best went to a dance.
He walked to the ball.

"Dancing" is the last processed concept in the first utterance. Several possible dancing events are instantiated, including discos and balls. An assumption schema that is instantiated by this event states that "going dancing involves dancing at a ball". The "ball" in the second utterance abductively unifies with the ball in the first.

The other reading is equally as Relevant since "the ball" can be abductively unified with the ball that George Best might play football with. Thus, Crystal makes an arbitrary choice based on the Relative frequency of the word sense (determined by the ordering of the senses in the encyclopædic memory): the football reading is chosen (against my own intuitions). A better method of distinguishing these readings would be to place a cost on abductive unification that occurs when an assumption is added via the interface: it could be assumed that more effort is required to abductively unify an assumption with assumptions derived chronologically earlier than with those derived relatively recently. Thus, in the above case, the "dance" reading would be obtained, since "dance" was processed after "George Best" and footballs.

## E.1 Generalised Conversational Implicature

In order to handle arbitrary Gricean generalised conversational implicature, Crystal would need to be extended so that the surface form of utterances is explicitly asserted as part of their semantic representation. Crystal would then include information about inferences that can be drawn from the use of words like "some". Consider:

Some computer scientists wear anoraks. (171)

This has the associated generalised conversational implicature:

Not all computer scientists wear anoraks. (172)

In order to draw inferences such as (172), it is necessary to have a rule in the entry for "$\exists$" that states that if the word "some" was used in an utterances, then, if it is consistent to believe so, assert the "not all" form.

Much as this was not implemented in Crystal, it is a straightforward extension.

## References


Allen, James F. and Hayes, Patrick J. (December 1987). *Moments and Points in an Interval-Based Temporal Logic.* TR 180, Dept. CS and Philosophy, University of Rochester, Rochester, NY 14627.

Allen, James F. and Kautz, Henry A. (1985). A Model of Naive Temporal Reasoning. In Hobbs, Jerry R. and Moore, Robert C. (eds.), *Formal Theories of the Commonsense World*, chapter 7, pages 251–268, Ablex, NJ.

Allen, James F. (1984). Towards a General Theory of Action and Time. *Artificial Intelligence*, 23:123–154.

Alshawi, Hiyan (1987). *Memory and Context for Language Interpretation.* Cambridge University Press.

Appelt, Douglas and Konolige, Kurt (1988). A Practical Nonmonotonic Theory for Reasoning about Speech Acts. In *26th Annual Meeting of the Association of Computational Linguists*, pages 170–178, State University of New York at Buffalo, Buffalo, New York, USA.

Appelt, E., Douglas (1990). A Theory of Abduction Based on Model Preference. In *Proc. AAAI Symposium on Abduction*, pages 67–71.

Bar-Hillel, Yehoshua (1964). *Language and Information.* Addison-Wesley Publishing Company, London.

Barwise, Jon and Perry, John (1983). *Situations and Attitudes. A Bradford Book*, MIT Press, London.

Blakemore, Diane (1987). *Semantic Constraints on Relevance.* Basil Blackwell, New York.

Brachman, R. J. and Levesque, H. J. (1984). The Tractability of Subsumption in Frame-Based Description Languages. In *Proc. AAAI 84*, pages 34–37, Austin, Texas.

Brachman, Ronald J., Fikes, Richard E., and Levesque, Hector J. (May 1983a). *Krypton: A Functional Approach to Knowledge Representation.* FLAIR Technical Report 16, Fairchild Laboratory for Artificial Intelligence, Palo Alto, CA.

Brachman, Ronald J., Fikes, Richard E., and Levesque, Hector J. (August 1983b). Krypton: Integrating Terminology and Assertion. In *Proc. AAAI 83*, pages 31–35, Washington DC.

Brewka, Gerhard (1987). The Logic of Inheritance in Frame Systems. In *Proc. IJCAI 87*, pages 483–488, Milan.

Buchanan, Bruce G. and Duda, Richard O. (August 1982). *Principles of Rule-Based Expert Systems.* Technical Report, Stanford University, Stanford, CA.

Buchanan, B. G. and Shortcliffe, E. H. (1984). *Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA.

Carberry, Sandra (1986). User Models: The Problem of Disparity. In *Proceedings of COLING 86*, pages 29–34, Bonn.

Carnap, R. (1956). *Meaning and Necessity.* University of Chicago Press, Chicago, Illinois.

Carnap, R. (1962). *Logical Foundations of Probability.* University of Chicago Press, Chicago, Illinois.

Charniak, Eugene and Goldman, Robert (1988). A Logic for Semantic Interpretation. In *Proc. ACL 88.*

Charniak, Eugene and Goldman, Robert (March 1989a). Plan Recognition in Stories and in Life. Draft paper.

Charniak, Eugene and Goldman, Robert (1989b). A Semantics for Probabilistic Quantifier-Free First-Order Languages, with Particular Application to Story Understanding. In *Proc. IJCAI 89.*

Charniak, E., Gavin, M., and Hendler, J. (1982). *Frail 2.1, some documentation.* Brown University, Dept. CS, Providence, RI.

Charniak, Eugene, Riesbeck, Christopher K., McDermott, Drew V., and Meehan, James R. (1987). *Artificial Intelligence Programming.* Lawrence Erlbaum, Hillsdale, New Jersey, second edition.

Charniak, E. (1972). *Towards a Model of Children's Story Comprehension.* Technical Report 226, MIT Artifical Intelligence Laboratory Monographs, Cambridge, Mass.

Charniak, Eugene (1986). A Neat Theory of Marker Passing. In *Proc. AAAI 86*, pages 584–588.

Charniak, Eugene (April 1988). Motivation Analysis, Abductive Unification and Non-Monotonic Equality. *Artificial Intelligence*, 34(3):275–295.

Chomsky, N. (1957). *Syntactic Structures.* Mouton, The Hague.

Clark, H. and Marshall, C. (1981). Definite Reference and Mutual Knowledge. In Joshi, A., Webber, B., and Sag, I. (eds.), *Elements of Discourse Understanding*, Cambridge University Press, Cambridge, England.

Cohn, Anthony G. (1987). A More Expressive Formulation of Many Sorted Logic. *Journal of Automated Reasoning*, 3(2):113–200.

Collins, A., Brown, J. S., and Larkin, E. M. (1980). Inference in Text Understanding. In Spiro, R. J., Bruce, B. C., and Brewer, W. F. (eds.), *Theoretical Issues in Reading Comprehension*, pages 385–407, Lawrence Erlbaum Associates, Hillsdale, N. J.

Cooper, Gregory F. (1990). The Computational Complexity of Probabilistic Inference Using Belief Networks. *Artificial Intelligence*, 42(2–3):393–405.

Crouch, R. and Poznański, V. (1990). The Semantics and Pragmatics of Counterfactual *Before.* Submitted to ACL, 1991.

Crouch, R. (1990). The Temporal Properties of English Modals and Conditionals. PhD Thesis in Preparation.

Davis, Randall and Buchanan, Bruce G. (August 1977). Meta-Level Knowledge: Overview and Applications. In *Proc. IJCAI 77*, pages 920–927, Cambridge, MA.

Davis, Randall, Buchanan, Bruce, and Shortcliffe, Edward (1977). Production Rules as a Representation for a Knowledge-Based Consultation Program. *Artificial Intelligence*, 8(1):15–45.

de Kleer, J. (1984). Choices Without Backtracking. In *Proc. AAAI 84*, pages 79–85, Austin, Texas.

de Kleer, J. (1986a). An Assumption-Based Truth Maintenance System. *Artificial Intelligence*, 28:127–162.

de Kleer, J. (1986b). Back to Backtracking: Controlling the ATMS. In *Proc. Fifth AAAI*, pages 910–917.

de Kleer, Johan (1986c). Extending the ATMS. *Artificial Intelligence*, 28:163–196.

de Kleer, Johan (1986d). Problem Solving with the ATMS. *Artificial Intelligence*, 28:197–224.

Delgrande, James P. (1988). An Approach to Default Reasoning Based on a First-Order Conditional Logic: Revised Report. *Artificial Intelligence*, 36(1):63–90.

Dempster, A. P. (1968). A Generalisation of Bayesian Inference. *Journal of the Royal Statistical Society*, 30(2):205–247. Series B.

Dowty, D. R., Peters, S., and Wall, R. (1981). *Introduction to Montague Semantics.* Reidel, Dordrecht.

Dowty, David R. (1979). *Word Meaning and Montague Grammar.* Volume 7 of *Texts and Studies in Linguistics and Philosophy*, D. Reidel Publishing Company, Dordrecht, Holland.

Doyle, J and Wellman, M. P. (1989). Impediments to Universal Preference-Based Default Theories. In *Proc. First International Conference on Knowledge Representation and Reasoning*, Toronto.

Doyle, Jon (1979). A Truth Maintenance System. *Artificial Intelligence*, 12:232–272.

Doyle, Jon (1983). The Ins and Outs of Reason Maintenance. In *Proc. IJCAI 83*, pages 349–351, Karlsruhe, Germany.

Dubois, Didier and Prade, Henri (1985). Combination and Propagation of Uncertainty with Belief Functions. In *Proc. IJCAI 85*, pages 111–113, LA, California.

Dubois, D. and Prade, H. (1987). An Introduction to Possibilistic and Fuzzy Logics. In Smets, P., Mamdani, A., Dubois, D., and Prade, H. (eds.), *Non-Standard Logics for Automated Reasoning*, Academic Press.

Elkan, Charles (1990). A Rational Reconstruction of Nonmonotonic Truth Maintenance Systems. *Artificial Intelligence*, 43(2):219–234.

Etherington, David W., Mercer, Robert E., and Reiter, Raymond (1984). On the Adequacy of Predicate Circumscription for Closed-World Reasoning. In *Proc. AAAI Workshop on Nonmonotonic Reasoning*, pages 70–81, Mohonk House, New Paltz, New York.

Etherington, David W. (1987a). Relating Default Logic and Circumscription. In *Proc. 10th IJCAI*, pages 489–494, Milan, Italy.

Etherington, W., David (1987b). A Semantics for Default Logic. In *Proc. Tenth International Joint Conference on Artificial Intelligence*, pages 495–498, Milan, Italy.

Fenstad, J. E., Halvorsen, P. K., Langhom, T, and van Benthem, J. (1987). *Situations, Language and Logic.* Reidel, Dordrecht.

Fikes, Richard E. and Nilsson, Nils J. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2:189–208.

Fodor, J. A., Garrett, M., Walker, E., and Parkes, C. (1980). Against Definitions. *Cognition*, 8(3):263–367.

Fodor, J. A. (1983). *The Modularity of Mind.* MIT Press, Cambridge, Mass.

Frisch, Alan M., Allen, James F., and Giuliano, Mark (1983). An Overview of the HORNE Logic Programming System. *SIGART Newsletter*, (84):27–29.

Frisch, Alan M. (August 1986). *Knowledge Retrieval as Specialised Inference.* PhD thesis, CS Dept, University of Rochester.

Frisch, Alan M. (May 1989). A General Framework for Sorted Deduction: Fundamental Results on Hybrid Reasoning. In *Proc. First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto.

Galliers, Julia Rose (1990). *Belief Revision and a Theory of Communication.* Technical Report 193, University of Cambridge, Computer Lab., Cambridge, UK.

Gärdenfors, Peter and Makinson, David (1988). Revisions of Knowledge Systems using Epistemic Entrenchment. In *Proc. 2nd Conference on Theoretical Reasoning about Knowledge*, pages 83–95, Pacific Grove, California.

Garvey, T., Lowrance, J., and Fischler, M. (1981). An Inference Technique for Integrating Knowledge from Disparate Sources. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 319–325, Morgan Kaufmann, Vancouver, BC.

Gazdar, G. and Good, D. (1982). On a Notion of Relevance. In Smith, N. V. (ed.), *Mutual Knowledge*, chapter 2, pages 88 – 100, Academic Press, London.

Gazdar, G. (1979). *Pragmatics: Implicature, Presupposition and Logical Form.* Academic Press, New York.

Gelfond, Michael and Lifschitz, Vladimir (1989). Compiling Circumscriptive Theories into Logic Programs. In *Proc. 2nd Int. Workshop on Nonmonotonic Reasoning*, Springer.

Genesereth, Michael R. and Nilsson, Nils J. (1987). *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann Publishers, Inc., Los Altos, CA 94022.

Goldblatt, R. (1987). *Logics of Time and Computation. CSLI Lecture Notes*, CSLI.

Goldman, Robert P. and Charniak, Eugene (1990). *A Probabilistic Approach to Text Understanding.* Technical Report CS-90-13, Brown University, Dept. CS, Providence, Rhode Island 02912.

Goldman, Robert P. (May 1987). Probabilistic Chaining for Motivation Analysis.

Grice, H. P. (1957). Meaning. *Philosophical Review*, 67.

Grice, H. P. (1975). Logic and Conversation. In Cole, P. and Morgan, J. (eds.), *Syntax and Semantics 3: Speech Acts*, pages 41–58, Academic Press, New York.

Harnish, Robert M. (1979). Logical Form and Implicature. In Bach, Kent and Harnish, Robert M. (eds.), *Linguistic Communication and Speech Acts*, pages 313–391, MIT Press, Cambridge, Mass.

Hayes-Roth, Frederick, Waterman, Donald A., and Lenat, Douglas B. (1983). *Building Expert Systems.* Addison-Wesley Publishing Company, Reading, Mass.

Hayes, Patrick J. (1979). The Logic of Frames. In Metzing, D. (ed.), *Frame Conceptions and Text Understanding*, Walter de Gruyter and Co., Berlin.

Hayes, Patrick (1984). The Second Naive Physics Manifesto. In Hobbs, Jerry R. and Moore, Robert C. (eds.), *Formal Theories of the Commonsense World*, chapter 1, Ablex, N.J.

Hayes, Patrick J. (1985). Naive Physics I: Ontology for Liquids. In Hobbs, Jerry R. and Moore, Robert C. (eds.), *Formal Theories of the Commonsense World*, chapter 3, pages 71–107, Ablex, NJ.

Hayes, Patrick J. (1987). Short Time Periods. In *Proc. 10th IJCAI*, pages 981–983, Milan, Italy.

Hirschberg, Julia Bell (1985). *A Theory of Scalar Implicature.* PhD thesis, University of Pennsylvania.

Hobbs, Jerry R. and Martin, Paul (1987). Local Pragmatics. In *Proc. IJCAI 87*, pages 520–523, Milan, Italy.

Hobbs, Jerry R., Croft, William, Davies, Todd, Edwards, Douglas, and Laws, Kenneth (July 1987). The Tacitus Commonsense Knowledge Base. Rough Draft from SRI International.

Hobbs, Jerry R., Stickel, Mark, Martin, Paul, and Edwards, Douglas (1988). Interpretation as Abduction. In *Proc. 26th ACL*, pages 95–103.

Hobbs, Jerry. (1988). Interpretation as Abduction. Seminar given to Cambridge University AI group, UK.

Hobbs, Jerry R. (1990). Personal Communication.

Horn, L. (1984). Toward a New Taxonomy for Pragmatic Inference: Q-based and R-based Implicature. In Schiffrin, D. (ed.), *Meaning, Form and Use in Context: Linguistic Applications*, Georgetown University Press.

Hughes, G. E. and Cresswell, M. J. (1968). *An Introduction to Modal Logic.* Methuen and Co, London, UK.

Johnson-Laird, P. N. (1983). *Mental Models.* Cambridge University Press, Cambridge, UK.

Joshi, A. K., Webber, B., and Weischedel, R. (July 1984). Preventing False Inferences. In *Proceedings of COLING 84*, Stanford CA.

Joshi, Aravind K., Webber, Bonnie L., and Weischedel, Ralph M. (1986). *Some Aspects of Default Reasoning in Interactive Discourse.* Technical Report, Dept. Computer and Information Science, University of Pennsylvania, Philadelphia.

Kamp, H. (1981). A Theory of Truth and Semantic Representation. In Groenendijk, J. A. G., Stokhof, M., and Janssen, T. (eds.), *Formal Methods in the Study of Language*, Mathematic Centre Tracts 135, Mathematisch Centrum, Amsterdam.

Katz, J. J. (1972). *Semantic Theory.* Harper and Row, New York.

Kautz, Henry and Allen, James (1986). Generalized Plan Recognition. In *Proc. AAAI 86*, pages 32–37, Philadelphia, PA.

Konolige, Kurt (1988a). Hierarchic Autoepistemic Theories for Nonmonotonic Reasoning. In *Proc. AAAI 88*, pages 439–443, St. Paul, Minnesota.

Konolige, Kurt (1988b). On the Relation between Default and Autoepistemic Logic. *Artificial Intelligence*, 35(3):343–382.

Konolige, Kurt (1989). On the Relationship Between Autoepistemic Logic and Circumscription: Preliminary Report. In *Proc IJCAI 89.*

Kowalski, R. A. and Sergot, M. J. (1986). A Logic-Based Calculus of Events. *New Generation Computing*, 4:67–95.

Kyburg, Henry E. (1987a). Bayesian and Non-Bayesian Evidential Updating. *Artificial Intelligence*, 31:271–293.

Kyburg, Henry E. (1987b). *Probabilistic Inference.* Technical Report 230, University of Rochester, Computer Science, Rochester, NY.

Levesque, H. J. (1990). All I Know: A Study in Autoepistemic Logic. *Artificial Intelligence*, 42(2–3):263–309.

Levinson, S. (1983). *Pragmatics.* Cambridge University Press, Cambridge, England.

Levinson, S. (1985). Minimization and Conversational Inference. Draft copy.

Levinson, Stephen C. (1987). Implicature Explicated? Material for Horn and Levinson's Pragmatics Seminar, given at the Stanford Linguistic Institute.

Lewis, D. (1972). General Semantics. In Davidson, D. and Harman, G. (eds.), *Semantics of Natural Language*, Reidel, Dordrecht.

Lifschitz, Vladimir (1985). Computing Circumscription. In *Proc IJCAI 85*, pages 121–127, Los Altos, CA.

Lifschitz, Vladimir (1986). Pointwise Circumscription: Preliminary Report. In *Proc. AAAI 86*, pages 406–410, Philadelphia.

Longmans (1987). *Longman Dictionary of Contemporary English.* UK, second edition.

Loui, Ronald Prescott (1987). *Theory and Computation of Uncertain Inference and Decision.* Technical Report 228, University of Rochester, Dept CS, Rochester, NY.

Lukaszewicz, W. (1985). Two Results on Default Logic. In *Proc. 9th IJCAI*, pages 459–461, Los Angeles, CA.

Lyons, J. (1977). *Semantics.* Volume 1 and Volume 2, Cambridge University Press, Cambridge, England.

McAllester, David A. (1980). *An Outlook on Truth Maintenance.* AI Memo 551, MIT AI Lab, Cambridge, Mass.

McCarthy, John (1980). Circumscription — A Form of Nonmonotonic Reasoning. *Artificial Intelligence*, 13.

McCarthy, John (1984). Applications of Circumscription to Formalising Common Sense Knowledge. In *AAAI Workshop on Nonmonotonic Reasoning*, pages 295–323, Mohonk House, New Paltz, NY.

McDermott, Drew and Doyle, Jon (1980). Non-Monotonic Logic I. *Artificial Intelligence*, 13:41–72.

McDermott, Drew (1982a). Nonmonotonic Logic II: Nonmonotonic Modal Theories. *Journal of the Association for Computing Machinery*, 29(1):33–57.

McDermott, Drew (1982b). A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science*, 6:101–155.

McDermott, D. (1986a). Tarskian Semantics, or No Notation Without Denotation! In Grosz, Barbara J., Sparck Jones, Karen, and Webber, Bonnie Lynn (eds.), *Readings in Natural Language Processing*, Morgan Kaufmann, Los Altos, California.

McDermott, Drew (1986b). *A Critique of Pure Reason.* Research Report 480, Yale University.

Mercer, R. E. and Reiter, R. (1982). The Representation of Presuppositions using Defaults. In *Proc. Canadian Soc. Computational Studies of Intelligence*, pages 103–107.

Milward, David R. (May 1986). *Inference Using Resolution and Many Sorted Logic.* Computer Lab., Diploma Project, University of Cambridge, UK.

Minker, Jack and Perlis, Donald (1984). Protected Circumscription. In *Proc. Workshop on Nonmonotonic Reasoning*, pages 337–343, Mohonk House, New Paltz, NY.

Minsky, Marvin (1981). A Framework for Representing Knowledge. In Haugeland, J. (ed.), *Mind Design*, pages 95–128, MIT Press, Cambridge, MA.

Moens, Marc and Steedman, Mark (July 1987). Temporal Ontology in Natural Language. In *Proc. 25th ACL, Stanford*, pages 1–7, Stanford, California.

Moore, Robert C. (1982). The Role of Logic in Knowledge Representation and Commonsense reasoning. In *Proc. AAAI 82*, Pittsburgh, PA.

Moore, R. (1985a). A Formal Theory of Knowledge and Action. In Hobbs, J. and Moore, R. (eds.), *Formal Theories of the Commonsense World*, Reidel, Norwood, NJ.

Moore, Robert C. (1985b). Semantical Considerations on Nonmonotonic Logic. *Artificial Intelligence*, 25.

Moore, Robert C. (1988). Autoepistemic Logic. In Smets, P., Mamdani, E. H., Dubois, D., and Prade, H. (eds.), *Non-Standard Logics for Automated Reasoning*, Academic Press.

Nagel, E. (1961). *The Structure of Science.* Routledge, London.

Nebel, Bernhard (April 1988). Computational Complexity of Terminological Reasoning in BACK. *Artificial Intelligence*, 34(3):371–383.

Nilsson, Nils J. (1982). *Principles of Artificial Intelligence.* Springer-Verlag, Berlin.

Nilsson, N. J. (1986). Probabilistic Logic. *Artificial Intelligence*, 28(1):71–87.

Norvig, Peter (1987). Inference in Text Understanding. In *Proc. 6th AAAI*, pages 561–565, Seattle, Washington.

O'Rorke, Paul (1983). Reasons for Beliefs in Understanding: Applications of Non-Monotonic Dependencies to Story Understanding. In *Proc. AAAI 83*, pages 306–309, Washington DC.

Partee, Barbara H. (1984). Nominal and Temporal Anaphora. *Linguistics and Philosophy*, 7:243–286.

Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo.

Pereira, Fernando C. N. and Pollack, Martha E. (1988). An Integrated Framework for Semantic and Pragmatic Interpretation. In *Proc 26th ACL*, Buffalo, NY.

Poole, David (1988). A Logical Framework for Default Reasoning. *Artificial Intelligence*, 36(1):27–47.

Provan, G. M. (1988). *Complexity Analysis of Truth Maintenance Systems, with Applications to High-Level Vision.* PhD thesis, University of Oxford, Oxford, England.

Przymusinski, T. C. (1989). An Algorithm to Compute Circumscription. *Artificial Intelligence*, 38(1):49–73.

Pulman, S. G. (1983). *Word Meaning and Belief.* Croom Helm, London.

Quillian, M. Ross (1967). Word Concepts: A Theory and Simulation of some Basic Semantic Capabilities. *Behavioral Science*, 12:410–430.

Reichgelt, Han (1986). *Semantics for Reified Temporal Logic.* DAI Research Paper 299, Dept. of AI, University of Edinburgh, 80 South Bridge, Edinburgh.

Reinhart, T. (1983). *Anaphora and Semantic Interpretation.* Croom Helm.

Reiter, Raymond (1980). A Logic for Default Reasoning. *Artificial Intelligence*, 13.

Robinson, J. A. (1965). A Machine-Oriented Logic based on the Resolution Principal. *Journal of the Association for Computing Machinery*, 12(1):23–41.

Ruspini, E. H. (1986). *The Logical Foundations of Evidential Reasoning.* Technical Note 46, SRI International, Menlo Park, California.

Schank, R. C. and Abelson, R. (1977). *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Schank, Roger C. and Rieger III, Charles J. (1974). Inference and the Computer Understanding of Natural Language. *Artificial Intelligence*, 5(4):373–412.

Schank, R. and Riesbeck, C. (1981). *Inside Computer Understanding.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Schank, Roger C. (1979). Interestingness: Controlling Inferences. *Artificial Intelligence*, 12:273–297.

Schank, Roger C. (1982). *Dynamic Memory.* Cambridge University Press.

Schiffer, S. (1972). *Meaning.* Clarendon Press, Oxford.

Shafer, G. (1976). *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, New Jersey.

Shannon, C. and Weaver, W. (1949). *The Mathematical Theory of Communication.* University of Illinois Press, Urbana.

Shoham, Y. (1986). Reified Temporal Logics: Semantical and Ontological Considerations. In *Proc 7th ECAI*, pages 390–397.

Shoham, Yoav (1987). Nonmonotonic Logics: Meaning and Utility. In *Proc. 10th IJCAI*, pages 388–393, Milan, Italy.

Shoham, Y. (1988). Chronological Ignorance: Experiments in Nonmonotonic Temporal Reasoning. *Artificial Intelligence*, 36(3):279–331.

Sidner, C. L. (1979). *Towards a Computational Theory of Definite Anaphora Comprehension In English Discourse.* PhD thesis, MIT, Cambridge, MA.

Smets, Philippe and Hsia, Yen Teh (1990). Default Reasoning and the Transferable Belief Model. Submitted to the 6th Conference on Uncertainty in AI, Boston, 1990.

Smets, Philippe and Kennes, Robert (1990). The Tranferable Belief Model. Personal Communication.

Smith, N. V. (1982). *Mutual Knowledge.* Academic Knowledge, London, NW1.

Sperber, D. and Wilson, D. (1982a). Mutual Knowledge and Relevance in Theories of Comprehension. In Smith, N. (ed.), *Mutual Knowledge*, chapter 2, pages 61–131, Academic Press, London.

Sperber, Dan and Wilson, Deirdre (1982b). Reply to Gazdar and Good. In Smith, N. V. (ed.), *Mutual Knowledge*, chapter 2, pages 101–110, Academic Press, London.

Sperber, Dan and Wilson, Deirdre (1986). *Relevance.* Basil Blackwell Ltd, 108 Cowley Road, Oxford, OX4 1JF, UK.

Stallman, R. M. and Sussman, G. J. (1977). Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis. *Artificial Intelligence*, 9:135–196.

Stickel, Mark E. (1985). Automated Deduction by Theory Resolution. In *Proc. IJCAI 85*, pages 455–458, Los Angeles, CA.

Stickel, Mark E. (1988). A Prolog-Like Inference System for Computing Minimum Cost Abductive Explanations in Natural Language Interpretation. Forthcoming.

Strawson, P. (1971). *Logico-Linguistic Papers.* Methuen, London.

Thagard, Paul R. (1978). The Best Explanation: Criteria for Theory Choice. *The Journal of Philosophy*, 76–92.

Tong, Richard M., Shapiro, Daniel G., Dean, Jeffrey S., and McCune, Brian P. (1983). A Comparison of Uncertainty Calculi in an Expert System for Information Retrieval. In *Proc IJCAI 83*, pages 194–197, Karlsruhe, W. Germany.

Touretzky, David S., Horty, John F., and Thomason, Richmond H. (1987). A Clash of Intuitions: The Current State of Nonmonotonic Multiple Inheritance Systems. In *Proc. 10th IJCAI*, pages 476–482, Milan, Italy.

Touretzky, D. S. (1986). *The Mathematics of Inheritance Systems.* Morgan Kaufmann, Los Altos, CA.

Tsang, Edward P. K. (1987). Time Structures for AI. In *Proc. 10th IJCAI*, pages 456–461, Milan, Italy.

Turner, Raymond (1984). *Logics for Artificial Intelligence.* Ellis Horwood LTD, Chichester.

Van Benthem, J. F. A. K. (1982). *The Logic of Time.* Reidel.

Vilain, Marc (August 1985). The Restricted Langauge Architecture of a Hybrid Reasoning System. In *Proc. IJCAI 85*, pages 547–551, LA, California.

Weyhrauch, Richard W. (1985). Prolegomena to a Theory of Mechanised Formal Reasoning. In Brachman, Ronald J. and Levesque, Hector J. (eds.), *Readings in Knowledge Representation*, chapter 4, pages 309–328, Morgan Kaufmann, Los Altos, CA 94022.

Wilensky, Robert W. (1978). *Understanding Goal-Based Stories.* Computer Science Research Report, Yale University, New Haven, CT.

Wilks, Yorick (1975). An Intelligent Analyzer and Understander of English. *Communications of the ACM*, 18(5):264–274.

Wilson, D. and Sperber, D. (1986). Inference and Implicature. In Travis, C. (ed.), *Meaning and Interpretation*, Basil Blackwell, Oxford.

Wilson, D. and Sperber, D. (1987). An Outline of Relevance Theory. Notes from UCL, London.

Wilson, Deirdre and Sperber, Dan (1988). On Defining Relevance. In Grandy, Richard E. and Warner, Richard. (eds.), *Philosophical Grounds of Rationality*, chapter 9, pages 243–259, Clarendon Paperbacks, Oxford, UK.

Woods, William A. (1975). What's in a Link? Foundations for Semantic Networks. In Bobrow, D. G. and Collins, A. M. (eds.), *Representation and Understanding: Studies in Cognitive Science*, pages 35–82, Academic Press, New York.

Zadeh, Lofti A. (1986). A Simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination. *The AI Magazine*, 7(2):85–90.

Zadrożny, Włodek W. (August 1987). Intended Models, Circumscription and Commonsense Reasoning. In *Proc. IJCAI 87*, pages 909–916, Milan, Italy.