The representation of lexical semantic information

CSRP 280

Ann Copestake

Submitted for D.Phil. September 1992

Abstract

This thesis is an investigation of the representation of lexical semantic information from a computational linguistic perspective. An implemented representation language is described which is not specific to lexical semantics, but is based on the use of typed feature structures augmented with default operations. This language, which is formally specified, allows the lexical semantic representations to be tightly integrated with the syntactic component of the lexical sign, capturing generalisations by use of inheritance, while allowing for exceptions with the default mechanism. Default inheritance and default unification are discussed in detail. Grammar rules and lexical rules can be specified in the same formalism and thus the paradigmatic treatment of lexical semantics can be integrated with an account at the syntagmatic level. The use of the language is illustrated with some examples of the representation of verbs, the treatment of logical metonymy and of sense extension. This is followed by a more detailed account of individuation in nominals.

The representation language is designed to allow the lexicon to be highly structured, with relationships between entries expressed in terms of (default) inheritance and lexical rules. The thesis discusses the way in which lexical entries structured by inheritance hierarchies may be derived semi-automatically by partial analysis of definitions in machine readable versions of conventional dictionaries (MRDs). The use of MRDs illustrates the practicality of constructing lexicons with detailed semantic information for use in natural language processing and provides a set of real examples with which to demonstrate the applicability of the representations described.

Contents

Acknowledgements

I am very grateful to my supervisor, Gerald Gazdar, for all his help and advice, and especially for encouraging me while I was writing this thesis. I owe a great deal to Ted Briscoe, for inspiring much of this research as well as for his comments on various drafts.

Most of the work on this thesis was carried out while I was employed at the University of Cambridge Computer Laboratory on the ACQUILEX project (Esprit BRA-3030), which involved the Universities of Cambridge, Pisa, and Amsterdam, University College Dublin, the Universitat Politècnica de Catalunya in Barcelona and Cambridge University Press. I have benefitted very considerably from discussions with my colleagues on ACQUILEX and with members of the NL group at the Computer Laboratory. In particular, much of this research would have been impossible without John Carroll, Valeria de Paiva, Antonio Sanfilippo and Piek Vossen, with whom I have collaborated most closely. I am also very grateful to Bob Carpenter and Dick Crouch, for their comments and advice on particular aspects of this work and to my examiners for their helpful remarks.

Finally, I would like to thank John Bowler for his continuous help and support, and my parents for their encouragement and understanding.

Chapter 1

Introduction

1.1 Lexical semantics

This thesis is an investigation of one approach to the representation of semantic information in the lexicon, from a computational linguistic perspective. I will try to show how some aspects of lexical semantics can be represented formally, within a unification-based framework, in a way which integrates with syntax and compositional semantics, and how the results of this can be applied in the construction of large-scale lexical knowledge bases. To begin with, I will discuss the problem of delimiting lexical semantic information, since there is no consensus as to what semantic information should be regarded as lexical, as opposed to commonsense or real-world knowledge.

If it were possible to represent word meaning by decomposition into a limited set of primitives, which were related according to a simple well-defined language, then complete lexical representation of word meaning would, in principle, be relatively straightforward. Lexical representations would consist of combinations of primitives and real world knowledge would be stated entirely in terms of the primitives. However, it is now generally accepted that this is not possible, and the arguments against attempting to represent word meaning by complete decomposition into a small set of primitives are too well known for detailed discussion here to be necessary. Pulman (1983) gives a full account of the problems. Even the standard example of *bachelor* poses difficulties when a precise and complete account of its meaning is attempted and most words are much more resistant to decomposition. Of the current proponents of such an approach, Wierzbicka is perhaps the only one who has attempted to apply it on a large scale (e.g. Wierzbicka, 1988), but although she claims a limited set of primitives, these look like English words and are used in a way which make use of the variations in meaning which are possible with the corresponding word. They are combined, to give English-like sentences, without formal definition of the rules governing composition. Furthermore, for any particular example, it is possible to argue that part of the meaning has been omitted, in exactly the same way as it is possible to show that *bachelor* is not precisely equivalent to *unmarried man*.

For these reasons it is generally accepted that rather than attempting to decompose meaning completely into primitives, it is more productive to talk about relationships between senses (e.g. Cruse, 1986). We could relate *bachelor* to *man* and *unmarried*, and *man* to *human* and so on. However, this in itself gives no solution to the problem of lexical representation, since there is no way of telling when we have a complete representation of meaning (even assuming that the concept of a complete relational description of meaning is itself coherent). Furthermore, the relational approach provides no basis for distinction between representation of lexical and non-lexical knowledge. If there is no principled distinction, it appears that in order to provide an adequate linguistic account, we might have to represent any sort of information about the real world. But it is methodologically important to distinguish between linguistic and non-linguistic representation, even though the two have to be interrelated so that linguistic utterances can be interpreted as having some connection with the real world. We want to avoid the situation where linguistic representation is dependent on scientific knowledge about the world. For example, the treatment of the linguistic behaviour of mass terms should not depend on whether or not matter is actually composed of atoms. From a computational perspective, we wish to provide a testable constrained theory, and a formal representation language, and to avoid problems which arise in knowledge representation which do not have a linguistic dimension.

It could be claimed that, in practice, it is sufficient to adopt an empirical approach and to take, for example, dictionary definitions as the basis for lexical representation, on the grounds that they implicitly or explicitly contain the information about the meaning of a word which is necessary for the reader to know in order to use it appropriately. Take for example the dictionary definition for *rabbit* from the Longman Dictionary of Contemporary English (LDOCE, Procter 1978):

rabbit^1 **1** [C] a common type of small long-eared animal of the HARE family that lives in a hole (BURROW) in the ground [1]

It is possible to demonstrate that any of this information about rabbits could be involved in arriving at the correct interpretation of some utterance, in that it might be used in lexical or structural disambiguation, for example. Thus, if we know that rabbits have long ears, we can structurally disambiguate the following:

(1) The vet examined the rabbit and the dog with the short pointed ears.

However any arbitrary information known to both speaker and hearer might be used in this way. For example, if both speaker and hearer know that rabbits do not require Vitamin C in their diet, but that guinea pigs do, there is no structural ambiguity in:

(2) The vet examined the rabbit and the guinea pig with dietary Vitamin C deficiency.

Clearly, this information is highly specialised and other information can be assumed to be relatively generally known, however, there seems to be a continuum between the sort of information found in dictionary definitions and specialised knowledge. Thus, even if there is a distinction between knowledge essential to the ability to use a word appropriately, and knowledge necessary to make true statements, it is very unlikely that all the information to be found in dictionary definitions is essential, even in the case of learners' dictionaries such as LDOCE. A speaker does not need the information that a rabbit has long ears in order to use the word appropriately. In fact, the only information given in the dictionary definition which clearly fits this criterion is that a rabbit is an animal. [2]

A better way to approach the problem of the representation of word meaning is to start from the null hypothesis that all that the lexicon contains (for open class vocabulary) are pointers connecting the phonological or orthographic representation of the word with its real world denotation. We then have to establish criteria for providing further information about word meaning, which will ensure that the additions have linguistic motivation. This is essentially the approach which was adopted by Dowty (1979) who investigated the representation of word meaning in Montague Grammar. Decomposing closed class words, such as determiners, is quite standard in Montague Grammar, for example Dowty *et al.* (1981) give the following translation for **every**:

$$\lambda P \lambda Q \Lambda x [P\{x\} \rightarrow Q\{x\}]$$

The reason for doing this decomposition is to demonstrate the logical behaviour of the closed class vocabulary items. Lexical decomposition is a convenient but theoretically non-essential description technique for achieving this in Montague Grammar. Since the intermediate meaning representation

[1] Unless otherwise specified all dictionary definitions referred to here are from LDOCE. When quoting LDOCE definitions, I follow the typographic convention used there that a superscript refers to a distinct entry and an ordinary numeral to a sense within an entry. Letters in square brackets (e.g. [C]) indicate grammar codes, count in this case. Small capitals in the definitions indicate words which were not in the defining vocabulary.

[2] I do not want to suggest that lexical semantic information cannot or should not be used in disambiguation, merely that arguments based on disambiguation cannot be used to discriminate between lexical and non-lexical information.

is unnecessary, the translation could instead be stated directly in terms of conditions on the possible models.

We can then ask whether items of the open class vocabulary should also be decomposed, and if so, how it should be done. Dowty (1979) provides a detailed account of verb aspect using a partial decomposition analysis. Verbs are represented as a combination of stative predicates modified by the primitives, CAUSE, BECOME and DO. This allows entailment relationships to be represented between words and sentences. For example, by decomposing **kill** into CAUSE BECOME ¬**alive**′ we get the (loosely paraphrased) entailment (3b) from (3a):

(3) a John killed Bill.
    b John did something which caused a change in state such that Bill was not alive.

The important point about this is not the particular final state that Bill ends up in, but that this pattern of decomposition is characteristic of the class of accomplishment verbs, which have a particular range of linguistic properties, which Dowty could explicate in terms of the formal behaviour of the primitives.

Dowty states that the aim of decomposition is not only to allow entailment relationships to be conveniently specified, but also to capture what linguists feel to be significant paradigmatic and syntagmatic relationships among classes of word meanings. One eventual aim of this, according to Dowty, is to limit the notion of possible word meaning, since Montague's theory essentially admits all conceivable possibilities in terms of possible world referents. In general, Dowty argues that syntagmatic evidence is more important than paradigmatic evidence, because syntagmatic contrasts are far fewer in number and because compositional semantics plays a much more basic role in developing a semantic theory such as Montague's than does word semantics. So the arguments which lead to the particular form of decomposition of verbs which Dowty adopts are based on syntagmatic and morphological criteria, not observations about the way word meaning can vary paradigmatically, which underlay most of the earlier work on semantic primitives.

Dowty's (1979) work is important in many ways, in particular because of its formal precision, but it has not been generally taken up as a basis for lexical semantics. To a certain extent, this may be due to limitations in Montague's intensional logic. There is no notion of a defeasible inference, which is arguably necessary for the treatment of the imperfective paradox. There are also cases where predicates which should be intensionally equivalent apparently give rise to sentences with different truth conditions, when in identical contexts (see e.g. Chierchia and Turner, 1988). But I think that there are more important non-logical problems. The theory gives no concept of overall lexical structure and tends to lead to splitting of word senses. The strict conditions on the origin of ambiguity in Montague Grammar lead to a characteristic pattern of explanation which involves assuming the existence of ambiguous senses which are related by meaning postulates. For example, Dowty (1979:260f) has to assume that either the verb or the adverb is ambiguous to account for the 'external' and 'internal' readings of sentences such as (4a) ((4b) and (4c) indicate the two readings).

(4) a John closed the door again.
    b John has performed the action of closing the door at least once before.
    c John has brought it about that the door is again in a closed state, although he need not have closed it on a previous occasion.

Dowty prefers the adverbial polysemy account, but this requires that all the adverbs concerned have two senses, related by meaning postulates such as (5).

(5) $\Lambda x \Lambda P \Lambda p \Box[\mathbf{again}_2{}'(\hat{y}[P\{y\}\ \mathrm{CAUSE}\ \ \mathrm{BECOME}\,{}^{\vee}p])(x) \leftrightarrow$
$[P\{y\}\ \mathrm{CAUSE}\ \ \mathrm{BECOME}\ \mathbf{again}_1{}'(p)]]$

In many cases the assumption of polysemy seems dubious, but there is the further problem that meaning postulates are such a general device that any relationship between senses could be expressed. Meaning postulates in Montague Grammar have exactly the same formal status as any other relationship of logical entailment. Furthermore, Dowty has no way of generalising over the class of adverbs concerned — the meaning postulate has to be stated individually in each case.

Dowty's goals were in some ways much more limited than those of many lexical semanticists. He does consider derivational morphology and zero-derivation (or conversion) as well as standard grammar rules, but does not attempt to characterise the class of lexical items to which they apply, except in terms of the usual syntactic categories. Once this is taken as a goal, considerably more fine-grained analyses are needed, and it is also necessary to assume that semantic properties can constrain the application of rules. Approaches to lexical semantics which do take this sort of data into consideration have, on the whole, been descriptive rather than formal. For example, Levin (1992) categorises a very large range of verbs into semantic classes, which differ in their behaviour with respect to particular alternations. Mel'čuk and his coworkers (e.g. Mel'čuk and Zholkovsky, 1988) also investigate this sort of data as well as collocational information. However most of the formal work on lexical semantics has been concentrated on other areas, in particular the use of thematic roles (see, for example, Dowty, 1989, 1991; Sanfilippo, 1990).

Within computational linguistics and natural language processing (NLP), many approaches to the representation of word meaning have been advocated, without any emergence of a dominant paradigm. For example, Wilks proposed a relatively sophisticated theory of semantic primitives (e.g. Wilks, 1975), while many other authors followed Quillian (1968) in using semantic networks to implement a relational model. Collections of papers such as Pustejovsky and Bergler (1992) and Saint-Dizier and Viegas (1992) illustrate the diversity of current approaches. Some researchers, for example Hobbs (1987), assume that there is no distinct level of lexical semantic representation, and that pragmatic inference on real world knowledge alone is sufficient. Essentially this represents the computational version of the null hypothesis introduced above, and the argument against it will be made with respect to particular examples in various places in this thesis. Until quite recently most work on NLP interpretation systems has assumed limited domains, small lexicons with minimal ambiguity, and a very constrained range of possibilities for the interpretation of utterances, which limits the conclusions that can be drawn about the general plausibility of particular processing strategies.

However there is now a much greater realisation that the lexicon has to be considered as more than a simple list of senses. Many current linguistic theories, such as HPSG, place a much greater emphasis on the lexicon and lexical organisation and also allow complex interactions between syntactic and semantic representations. These provide more congenial mechanisms for representing generalisations about the interrelationship of syntax, compositional semantics and lexical semantics than were available in earlier theories. The availability of machine readable dictionaries (MRDs), and the needs of NLP, have also concentrated attention on the lexicon as a whole, as a structured object, rather than as a list of distinct word senses. Pustejovsky's approach to lexical semantics is based on the assumption that the lexicon should be seen as a highly structured, *generative* device. Since his work most closely influences my approach, I will discuss it in some detail, concentrating on Pustejovsky (1991a), which provides the most detailed exposition so far published of the concept of the generative lexicon.

1.2 The Generative Lexicon

In this section I will describe Pustejovsky's approach to lexical semantics, concentrating in particular on those aspects which are directly relevant for material discussed in subsequent chapters. Pustejovsky argues that lexical decomposition should be generative rather than involving exhaustive decomposition into a fixed set of primitives. He illustrates this with reference to the following example sentences:

(6) The door is closed.
The door closed.
John closed the door.

The word *closed* can be minimally decomposed as introducing an *opposition* of terms, *closed* and *not-closed*. Such a minimal analysis is generative because it operates on the predicate already provided by the word. (Pustejovsky relates this to Aristotle's *principle of opposition*.)

Lexical meaning is to be captured in four levels of representation:

1. Argument Structure
2. Event Structure
3. Qualia Structure
4. Inheritance Structure

Pustejovsky (1991a) has little to say about argument structure, but assumes that it not only describes the way in which arguments of verbs map onto syntactic expressions, but that it is highly structured independent of syntax, in the way proposed by Grimshaw (1990). It thus forms a necessary part of the semantic characterisation of a lexical item, but is not in itself sufficient. Event structure, which is discussed in detail in Pustejovsky (1991b), allows the semantics of words, phrases and sentences to be described in terms of sorted events; states (e^s), processes (e^p) or transitions (e^t), which may themselves have a structure in terms of subevents — for example, a transition may be decomposed into a process followed by a state. I will not discuss argument structure and event structure here, since the more novel features of Pustejovsky's approach are described in terms of qualia structure and inheritance structure.[3]

1.2.1 Qualia structure

The most significant part of Pustejovsky's work, from my current perspective, is the considerable emphasis that he puts on the representation of nominals and the way in which this interacts with the verbal representation. The *qualia structure* of a lexical item consists of the four roles, which are defined below, along with the possible values that they may assume (reproduced from Pustejovsky 1991a:426–427):

Constitutive Role the relation between an object and its constituents or proper parts:

- Material
- Weight
- Parts and component elements

Formal Role that which distinguishes the object within a larger domain:

- Orientation
- Magnitude
- Shape
- Dimensionality
- Color
- Position

Telic Role purpose and function of the object:

- Purpose that an agent has in performing an act
- Built-in function or aim that specifies certain activities

Agentive Role factors involved in the origin or "bringing about" of an object:

[3] I will show in Section 4.1 how equivalent information to argument structure and event structure has been encoded in the representation language developed here.

- Creator
- Artifact
- Natural Kind
- Causal Chain

Pustejovsky in fact introduces qualia structure as relating to the principle of opposition exemplified by the verb *close*. However, in this paper, he subsequently only refers to the qualia structure of nominals, and the description above relates to (concrete) nominals more closely than to verbs. Oppositions for verbs are expressed in Pustejovsky (1991a) by logical forms such as:

(7) Lexical Semantics for *escape*:
$\lambda x \lambda e^t \exists e^p, e^s[escape(e^t) \wedge act(e^p) \wedge confined(e^p) \wedge agent(e^p, x) \wedge \neg confined(e^s) \wedge object(e^s, x)]$

This could be regarded as an amalgamation of the different parts of the lexical semantic description. However, in what follows I will assume that only nouns have qualia structure, and that this interacts with verb argument structure.

Pustejovsky argues that, rather than assuming that nominals just behave as passive objects when they combine with verbs, we should treat them as being as active in the semantics as the verb itself is. He refers to this behaviour as *cocompositionality.* He considers the examples:

(8) a John baked the potato.
b John baked the cake.

which contrast in that (8a) refers to a change of state of an existing entity, and denotes a process, but (8b) implies that an entity is actually being created and denotes a transition (ignoring the possibility of buying frozen or part-baked cakes, etc).

(9) a John baked a potato for 90 minutes.
b ? John baked a cake for 90 minutes.

Traditionally the only way to handle this is to assume two lexical entries for *bake*, since the two senses although related, are clearly distinct. However in Pustejovsky's treatment we can take this as an example of *logical polysemy*, where the semantics of the noun can cause a shift in event type and thus there is no need for distinct lexical entries. This is to be achieved by specifying in the agentive role of *cake* that it is an artifact created by a process of *baking.*

Further examples of cocomposition involve *logical metonymy.* For example, verbs such as *enjoy* and *begin* normally select semantically for an event. Under this view, an example such as the following involves metonymy, because the referent of the complement is an event rather than an object:

(10) Mary enjoyed the book.

Pustejovsky proposes that these examples be dealt with as involving a semantic *type coercion.* The sentence is interpreted as

(11) Mary enjoyed some event associated with the book.

In fact the lexical entry for *book* describes possible associated events; for example the telic (purpose) role of the qualia structure has a value equivalent to *reading.* When combined with *enjoy*, type coercion occurs, because *enjoy* selects for a process rather than an object, and the particular sort of event which is likely to be involved can be accessed from the qualia structure, which results in a default interpretation for (10) equivalent to:

(12) Mary enjoyed reading the book.

In a marked context, the default might be overridden or blocked, for example if Mary was the name of a goat, it might be inferred that she enjoyed eating the book rather than reading it, but this is assumed to be part of pragmatics rather than lexical semantics. The type coercion still holds but the nature of the event is defeasible.

An event can also be specified as the value for the agentive role, and although *enjoy* preferentially selects for the telic role, *begin* allows either possibility. Thus the agentive role for *book* is *writing*, which gives the two interpretations (13b) and (13c) for (13a):

(13) a John began the book.
b John began reading the book.
c John began writing the book.

In cases where the lexical semantics of the noun would not specify a telic role or where the event specified would not of an appropriate type, the corresponding sentences are odd:

(14) ? Mary enjoyed the dictionary.
? John enjoyed the rock.
? Bill began the tree.

Although *dictionary* does have a telic role, this specifies a non-process event, *refer_to*, which is incompatible with *enjoy*, which is assumed to select for a process. In the other two sentences no telic role is specified at all and thus the event is unknown, and the sentences seem bad.

In Briscoe, Copestake and Boguraev (1990) we described a treatment of logical metonymy following Pustejovsky's account, which was implemented in a graph-unification based language which was a precursor to the one described in this thesis. We also provided evidence from the Lancaster-Oslo/Bergen (LOB) corpus that supports the idea that a default interpretation applies. I will discuss this account of logical metonymy in Section 4.2.

In more general terms, the significance of qualia structure for the work described here is that it gives some indication that a partial decomposition of word meaning into a fixed and limited number of roles can be used to give an account of some interesting aspects of semantics, in a way which avoids the use of multiple lexical entries. Splitting the lexical semantics in this way also allows for an inheritance structure in the lexicon where different parts of qualia structure are inherited independently. Thus it distinguishes some aspects of word meaning as more important than others and provides some account of how the different aspects of word meaning might interact with each other and with syntax. Qualia structure at least potentially provides the beginnings of an account of the lexical semantics of nominals which differentiates their meaning and allows a testable theory to be developed. I will use the notion of qualia as a framework for the description of nominal semantics in the current work.

1.2.2 Inheritance structure

The fourth level of Pustejovsky's treatment of lexical semantics is inheritance structure; which describes how the word is globally related to other concepts in the lexicon. Pustejovsky distinguishes between *fixed* inheritance and *projective* inheritance. In these terms much of the current work concerns fixed inheritance, about which Pustejovsky says little (he refers to Briscoe *et al.* (1990) as explicitly addressing the issue). Here I will briefly discuss Pustejovsky's concept of *projective inheritance* (Pustejovsky 1991a:433–436).

Pustejovsky discusses two sentences:

(15) The prisoner *escaped* last night.
The prisoner *ate* dinner last night.

He notes that the first is intuitively more 'prototypical' than the second but argues that the distinction can be captured within a theory of lexical semantics rather than as a matter of commonsense knowledge. His treatment involves generating the *projective conclusion space* from the predicate, by applying a series of transformations such as negation, temporal precedence, succession and equivalence, and *act* ("an operator adding agency to an argument") to the values of the qualia roles for the predicate and its generalisations. In this case the telic role of the qualia structure for *prisoner* is:

```
Telic: [ confine(y, *x*) & location(*x*,prison) ]
```

Here `*x*` is to be seen as a distinguished variable, representing the entity itself, i.e. the prisoner in this example. By projecting on the Telic role of *prisoner*, he gives the following derivation:

1. leq_G (generalisation): $[confine(y,x) \wedge loc(x, prison)] \Rightarrow confine(y,x)$
2. $\neg$ (negation): $\exists E_1[\neg confine(E_1, y, x)]$
3. $\exists E_2[confine(E_2, y, x)]$
4. $\leq$ (temporal precedence): $E_1 \leq E_2 = T_1$
5. $\leq$ (temporal precedence): $E_2 \leq E_1 = T_2$
6. *act* : $act(x, T_2) =$ "escape"

Some aspects of this derivation are quite unclear to me. I assume that the change of arity in the predicate *confine* is an error, but I do not know how to interpret $E_1 \leq E_2 = T_1$. It seems that the intuitive idea is to equate T_1 with a transition between two states, so there is clearly something missing from the equation as given. However, even given a rational reconstruction of the formalism, it is in principle implausible that the treatment could work, without assuming non-lexical inference. For example, once the transition event between *not-confined* and *confined* is generated the *act* operator is applied to retrieve "escape", but there is no clue as to how this is achieved. Presumably it is not meant to be a commonsense inference because this would void the claim that this is a lexical semantic, as opposed to a commonsense theory. Therefore the connection with the lexical entry for *escape* must be made at this point. Pustejovsky gives the lexical semantics for *escape* as:

(16) $\lambda x \lambda e^t \exists e^p, e^s[escape(e^t) \wedge act(e^p) \wedge confined(e^p) \wedge agent(e^p, x) \wedge \neg confined(e^s) \wedge object(e^s, x)]$

and, again assuming some rational reconstruction, we can assume that a match can be made on the basis of surface form without commonsense reasoning. But this relies on both lexical entries using the particular predicate *confine*. So this either leads to the use of semantic primitives, or to a level of lexical organisation, which has been left undescribed, which allows expressions to be equated which are semantically equivalent with respect to the projective transformations. If we assume the latter, it is unclear whether any constraints could be imposed which make the problem tractable, but it seems implausible that any sort of restricted, inheritance based, representation would be adequate.

I do not believe that projective inheritance can be treated adequately within the unification based representation which I will propose. However it does not seem to me that Pustejovsky has made a case for attempting this level of decomposition in the lexicon. Prototypicality is not correctly predicted even for this example. The projective conclusion space from the formal role of *prisoner* should also include *die* and *kill*, according to Pustejovsky, but

(17) The prisoner died last night.

does not have the same prototypical feel as the sentence with *escape*. Nor does:

(18) The prisoner was in prison last night.

It is very reasonable to claim that a limited number of relationships should be used in determining the conceptual proximity of concepts, but it is not clear what should be lexical about this account.

So although Pustejovsky's work has influenced my approach, and I have adopted (and adapted) several aspects of it, in particular his notion of qualia structure, there are some quite fundamental differences and the current work cannot be seen as an attempt to implement Pustejovsky's ideas as a whole.

1.3 The lexical representation language

LAUREL, the representation language to be described here, is intended to support theories which restrict the description of lexical semantic information to that which can be shown to interact with grammatical realisation and with compositional semantics. It allows specification of such syntagmatic interaction, in order to give a full formal account. To some extent, paradigmatic relationships between words (or more precisely word senses) can also be described in LAUREL. The representation language is thus restricted in terms of the operations defined on it but can also be used to represent syntactic information, grammar rules and so on.

I view formal (computational) lexical semantics as attempting to determine what extra information is needed in the lexicon other than a simple association between a word and a predicate. For example, I shall argue that it is not sufficient to state in the entry for *rabbit* simply that it corresponds to the predicate **rabbit′** and to assume that general non-linguistic inference can account appropriately for the properties of sentences such as:

(19) a Three rabbits hopped around the garden.
b ? John stroked rabbit.
c * One of our rabbit escaped.
d We had rabbit for dinner.

The basic argument which I will use for including lexical semantic information is that it is necessary in order to account for effects on grammatical behaviour, for example that associated with the count/mass distinction. Thus the way in which a word individuates an entity to which it refers is a semantic property, which results in the distinctions in behaviour between count and mass usages of nouns, but it is not completely predictable from real world knowledge of the entity. (I can refer to a pile of small feathers as *feathers* or *down*; in Italian *spaghetti* is count, in English it is mass, and so on.) Furthermore, I shall argue that extended senses, such as the use of *rabbit* as a mass term to denote rabbit meat or fur, are predictable on the basis of lexical semantic information, and that such a level of representation is necessary to make the appropriate linguistic generalisations about such processes, even though they may also have a pragmatic component.

Lexical semantics is thus seen as having a dual role; in representing aspects of word meaning which are purely linguistic, in that they cannot be plausibly derived from real world properties alone, and also in representing real world information in a way in which interactions with syntactic representation can be described. Considerations of interaction with syntax, and compositional semantics, coupled with the assumption that processing lexical semantic information should be efficient, should not involve extensive inference, and should on the whole be described as part of the parsing/generation process, leads to the adoption of a graph unification-based representation language, with limited capacity for inference. Thus, the lexical semantic information is represented in terms of feature structures, rather than predicate calculus expressions, for example. Rather than general inference, inheritance is used to capture regularities that apply over subparts of the lexicon. However, since the sort of phenomena which I wish to treat are most naturally to be described as having exceptions, the representation language must also incorporate the use of defaults. Just as the lexical semantic information represented at the level of individual entries must be consistent with real world knowledge, so also must the semantics of inheritance of that information be consistent with the results of reasoning with that knowledge. But the unification based representation language which I will describe, LAUREL, is quite similar to formalisms assumed in other work on the lexicon. Essentially the hypothesis to be tested is that such a language can be used to represent lexical semantic information in a way which allows interactions with syntactic and morphological components to be described.

1.3.1 An example of representation in LAUREL

As an introductory example consider the following lexical entry for rabbit:

```
rabbit L_1_1
   <> < lex-individual <>
   < QUALIA > < animal_L_1_1 <>.
```

This lexical entry is intended to correspond to the LDOCE entry given earlier and repeated here:

rabbit^1 **1** [C] a common type of small long-eared animal of the HARE family that lives in a hole (BURROW) in the ground

The notation will be explained fully in the next chapter: `<` and `>` are used to delimit paths of zero or more features, and `<` is used to stand for default inheritance. The orthography "rabbit" is implicitly connected with the predicate **rabbit_L_1_1**. The inheritance from `lex-individual` means that the sign as a whole is stated to have a combination of syntactic, formal semantic and lexical semantic properties associated with individual denoting nouns. The lexical semantic properties of noun signs are associated with the feature QUALIA and thus further information is inherited from the lexical semantic portion of the entry for *animal*^{1} *1*. From this it follows that the overall lexical semantic type of the entry is **animal**. [4]

The lexical semantic information is described in terms of its qualia structure. However, my use of qualia structure in these descriptions should not be taken as corresponding in detail to Pustejovsky's. LAUREL is a typed feature structure language, and thus the values of attributes are organised into a type hierarchy; in this case the type **animal** is located under **creature** and is disjoint from **human**. The types determine which features can appear in the feature structure — for example, feature structures of type **noun-sign** will have the feature QUALIA, but feature structures of type **animal** will not. Although the lexical semantic type of an entry will be correlated with its denotation, the structure of the hierarchy is determined by linguistic rather than scientific considerations and types such as **animal** should be taken as mnemonic labels — they are not intended to correspond completely with English words.

From this lexical entry the typed feature structure shown in Figure 1.1 is produced automatically in LAUREL. The way that this expansion happens will be discussed in detail in Chapter 2, but in this case no defaults are overridden. Here, I will informally discuss why various aspects of the entry should follow from the simple specification given above.

1. In the sense being defined *rabbit* will behave as a count noun in English. In Chapter 5, I will discuss more precisely how the mass/count distinction is represented, but informally the important point is that count nouns are taken to have a 'built-in' specification of an individual, whereas mass nouns do not. All words which inherit their lexical semantics from **animal** will have such a specification.

2. The entry also indicates that the morphologically unmarked form will denote an ordinary singular entity, rather than a plural entity or a group. A normal plural, such as *rabbits*, would have the value **true** for the semantic feature PLMOD, which stands for plural modifier, and a value **false** for QUANT which stands for quantised (cf Krifka, 1987). The value for relative form in the qualia structure would be **plural**. Again this is discussed in detail in Chapter 5. Note that the type **animal** does not distinguish between singular, plural and group entities; words such as *herd* and *brood* for example are group denoting.

[4] From now on I will use following conventions: types are represented in boldface font, features in capitals, identifiers for lexical entries and other feature structures in lowercase in typewriter font (e.g. `animal_L_1_1`). A box round a type in displayed feature structure diagrams indicates that that portion of the feature structure is not shown. Boxed numerals in feature structure diagrams are used to indicate reentrancy. When lexical entries are intended to correspond with LDOCE senses I will use an identifier of the form `orthography_L_homonym-number_sense-number`. Occasionally I will use an identifier which corresponds to the conflation of two LDOCE senses: in this case the conflated senses are joined by `+` (e.g. `animal_L_1_1+2`). Although I will discuss the automatic extraction of information from LDOCE in the later chapters of this thesis, I will relate the lexical entries to LDOCE senses throughout, where possible, to provide some grounding for the sense distinctions.

$$\left[\begin{array}{l}
\textbf{lex-count-noun}\\
\text{ORTH} = \textbf{rabbit}\\
\text{CAT} = \boxed{\textbf{noun-cat}}\\
\text{SEM} = \left[\begin{array}{l}
\textbf{obj-noun-formula}\\
\text{IND} = \boxed{0} \left[\begin{array}{l}
\textbf{obj}\\
\text{AGR} = \left[\begin{array}{l}\textbf{agr}\\ \text{NUM} = \textbf{sg}\end{array}\right]
\end{array}\right]\\
\text{PRED} = \textbf{rabbit\_L\_1\_1}\\
\text{ARG1} = \boxed{0}\\
\text{PLMOD} = \textbf{false}\\
\text{QUANT} = \textbf{true}
\end{array}\right]\\
\text{QUALIA} = \left[\begin{array}{l}
\textbf{animal}\\
\text{AGENTIVE} = \left[\begin{array}{l}\textbf{agentive}\\ \text{ORIGIN} = \textbf{basic}\end{array}\right]\\
\text{TELIC} = \boxed{\textbf{verb-sem}}\\
\text{FORM} = \left[\begin{array}{l}\textbf{nomform}\\ \text{ABSOLUTE} = \textbf{real-form}\\ \text{RELATIVE} = \textbf{individual}\end{array}\right]\\
\text{CONSTITUENCY} = \left[\begin{array}{l}\textbf{nomconst}\\ \text{PARTICLES} = \textbf{string}\end{array}\right]\\
\text{OBJECT-INDEX} = \boxed{0}\\
\text{PROPERTIES} = \left[\begin{array}{l}\textbf{creature-properties}\\ \text{STATE} = \textbf{solid\_a}\\ \text{SEX} = \textbf{gender}\end{array}\right]
\end{array}\right]
\end{array}\right]$$

Figure 1.1: Expanded lexical entry for rabbit

3. From the lexical entry it is possible to predict some other senses of *rabbit*. All count nouns denoting physical objects can also be used in a mass sense, denoting some stuff derived from that object. In the particular case of animal denoting nouns, more specific senses are also possible, in particular a sense denoting flesh or meat and one denoting skin or fur. Furthermore, animal denoting nouns can also be used metaphorically to denote humans with some particular characteristics. In section 4.3 I discuss how the lexical aspects of such processes may be represented in LAUREL.

4. Since rabbits are of type **animal** and **animal**s can have a specified sex it follows that in English it is possible to refer to individual rabbits as *he* or *she*, when the sex is known or assumed, or as *it*. This behaviour contrasts with words with type **human**, where the gender neutral pronoun is not *it*, and with plants, which even when they occur in male and female forms are not referred to as *he* or *she*.

Pustejovsky (1991a) apparently assumes that qualia structure is a level of representation which is quite distinct from real world knowledge, but does not explicitly address the issue of how the two might correlate. My own view is that in many cases the information in the qualia structure can be seen as making explicit in the lexical representation some information that would also be needed for real world inference. Thus the entry for *doe* in the sense corresponding to *female rabbit* would specify `< SEX > =` **female**, this would be justified because:

$$\forall x[\mathbf{doe}'(x) \Rightarrow \mathbf{female}'(x)]$$

However a straightforward equivalence between specification in the qualia structure and implicational relationships in the extension of the corresponding predicate will not always hold, in particular because some features in the qualia structure may correspond to information that only defeasibly holds. For example, the qualia structure contains information about the purpose of an entity (the telic role), but although the purpose of books in general is to be read, individual books may not be intended to be read. Furthermore, lexical semantic structure cannot be completely determined by the extension, but is relative to the particular word, hence there would be no requirement that the lexical semantic structure for *buy* was equivalent to that for *sell* even if all selling events were necessarily buying events.

Thus the specification of *rabbit* as an **animal** is justified by various aspects of its grammatical behaviour. Given a lexical semantic representation motivated in this way, verbs can be marked for selectional restrictions/preferences in order to take advantage of the semantic types, as will be described in section 4.1. Thus we can predict that sentences such as the following will be marked:

(20) ? The rabbit was reading a newspaper.
? The rabbit flowed down the hill.

As illustrated earlier, the lexical representation of components of meaning cannot be justified simply on the basis of specifying the oddity of such sentences, because there is no limit on such information. Thus the granularity of selectional restrictions suggested here will not distinguish between the acceptability of:

(21) The rabbit bit John.
The budgerigar bit John.
The newborn rabbit bit John.
The rabbit with no teeth bit John.

The type system could in principle be made more fine-grained to allow for the distinction between *rabbit* and *budgerigar*, but the system suggested here could not deal with selectional restrictions which relied on properties of the noun which could be affected by modifiers such as *newborn*. In all these cases, however, the oddity can be accounted for completely adequately by the properties of rabbits and budgerigars, without any need to invoke a specifically lexical component.

However there are cases of *collocational restrictions* where a lexical component has to be postulated to account for acceptability. For example, *grill* and *toast* are synonymous in terms of the action they denote, but bread is always referred to as being toasted rather than grilled. It is normal to describe a *performance* as *flawless*, *immaculate* or *impeccable* but while *impeccable behaviour* is usual, *immaculate behaviour* or *flawless behaviour* seem odd (Cruse 1986:270–285). In other cases it is not clear whether or not the effect is lexical. There is a sense of *worry*, meaning to attack, which is only usual when dogs or wolves are the agent. There is a clear difference in normality between the following sentences when that sense of *worry* is assumed:

(22) That dog worries sheep.
? That lion worries sheep.

However it is not clear whether this arises because dogs and wolves have a characteristic mode of attack, distinct from that of lions, or whether there is a purely collocational effect here (and it is perfectly possible that for some speakers the effect is collocational whereas for other it is not). At the other end of the spectrum, there is no clearcut distinction between idioms and highly restricted collocations; for example, *raze* is nearly always used with *to the ground*, but other uses are not impossible, e.g. *to ground level*. Although the treatment of idioms and collocational restrictions are important aspects of lexical representation, they lie outside the scope of this work.

1.4 Machine readable dictionaries

As I illustrated above, it is unlikely that dictionaries can be taken as a basis for determining what a lexical semantic representation should comprise. However, machine readable dictionaries (MRDs) are useful tools for evaluating generalisations about lexical semantic behaviour and can also be used as sources of data for instantiation of lexical semantic representations in construction of NLP lexicons.

Dictionaries are useful sources for linguistic research, because of their breadth of coverage, which makes it possible to investigate general patterns of lexicalisation. Levin (1985) has argued that alternations in the subcategorisation patterns of verbs are (partly) predictable from their semantic class and argument structure; the extensive classification of alternation patterns in Levin (1992) was compiled with the aid of MRDs. Boguraev and Briscoe (1988) demonstrated the use of LDOCE to characterise verbs semantically as Object Raising, Equi and so on; information which was not explicitly specified in the grammar coding scheme. Much of the recent work on MRDs has concentrated on LDOCE; Boguraev and Briscoe (1989) is a collection entirely devoted to work using it. This is partly because of its use of complex encoding schemes which make it possible to acquire a considerable amount of data without processing the actual text of the definitions. However even in LDOCE only a relatively small amount of the potential semantic information was codified and the semantic codes, which did not form part of the printed dictionary, are sometimes incomplete and inconsistent. The other feature of LDOCE which has made it a popular tool is the use of a restricted defining vocabulary, which makes processing definitions considerably more tractable.

MRDs are quite straightforwardly usable in applications such as spelling checking, text-to-speech synthesis, and so on, where relatively limited lexical information is needed. But the long-term goal of much work on computational lexicography is the construction of lexicons which can be used in applications such as natural language interfaces and machine translation systems, which require far more detailed lexicons. For such systems to be genuinely useful, large lexicons are needed. Even in cases such as database front ends, where the domain of the application is highly restricted, a practical natural language interface must be able to cope with an extensive vocabulary, in order to respond helpfully to a user who lacks domain knowledge, for example. For applications such as interfaces to large-scale knowledge based systems, summarising and so on, large lexicons are clearly essential. Acquisition of such information is a serious bottleneck in building NLP systems, and MRD sources can be used to semi-automatically acquire at least some of the syntactic and semantic information needed. An overview of the use of MRDs in NLP is given in the introduction to Boguraev and Briscoe (1989).

The use of MRDs as research tools and as sources of data for NLP lexicons complement each other to a considerable extent. For example, although it is possible to investigate verb alternations by querying a database produced from an essentially unanalysed dictionary to find all cases of distinctions between senses with characteristic definition patterns such as “to cause to”, this is time consuming and somewhat unreliable. In the long term, it would be much better to construct a lexical knowledge base which explicitly represents the alternation pattern and the semantic class. It would then be possible to test the theory on a corpus, either automatically or semi-automatically. Such a lexical knowledge base would also provide a source from which to build lexicons for various NLP applications.

Earlier work on extracting and representing syntactic information from MRDs includes the work done on the Alvey Tools lexicon project (Carroll and Grover, 1989) in which a large scale lexicon was produced semi-automatically from LDOCE using a feature and unification based representation. There has been considerable discussion and some implementation of lexical knowledge bases for the representation of semantic information extracted from MRDs (e.g. Boguraev and Levin, 1990; Wilks *et al.*, 1989). However the knowledge representation languages assumed are rarely described formally; typically a semantic network or a frame representation has been suggested, but the interpretation and functionality of the links has been left vague. Work on ACQUILEX, of which this thesis represents a part, has shown that large scale fragments of a lexical knowledge base represented in a formally specified language can be constructed semi-automatically. I will discuss some of the results of the ACQUILEX project as a whole in section 7.2. Indeed, from the point of view of computational lexicography, my description of the lexical representation language and its use in the representation of nominals can be seen as an attempt to establish a formal background with which to approach the extraction of information from MRDs.

Even from a more general perspective, however, the use of MRDs is an important part of this work. Since the approach to computational lexical semantics proposed here does involve representing fairly detailed semantic information in the lexicon, it is important to demonstrate that the overhead involved in acquiring the information does not make it infeasible for practical natural language processing. In the longer term it will only be possible to claim that the decomposition into qualia structure is appropriate if it is shown that there is a consistent approach which works on substantial fragments of the lexicon. Even in the short term, however, the use of MRDs to some extent serves as a test of the approach. The LAUREL representation language depends heavily on the use of inheritance, and noun definitions, at least, implicitly encode taxonomies (Amsler, 1981) which can be used to derive inheritance hierarchies for lexical semantic information. Processing the definitions to extract taxonomies, and to instantiate the lexicon in other ways, involves producing partial lexical semantic representations for the more important phrases, and thus the definitions themselves constitute a test corpus. Most importantly, automatic extraction of semantic information from MRDs demonstrates that a particular semantic distinction is made implicitly or explicitly in dictionaries and this provides some independent evidence that the classification adopted is plausible.

I will discuss the use of MRDs to derive lexical semantic representations semi-automatically in Chapters 6 and 7. However the use of MRDs has influenced other aspects of this work. The techniques for lexical semantic representation are in some respects geared to the representation of information which has been derived semi-automatically from MRDs, and the motivation for developing the treatment of individuation discussed in Chapter 5 was partly the frequency with which individuation shifting constructions occur in the dictionary definitions of nouns. Furthermore I will frequently use LDOCE sense distinctions as a basis for building representations since these are an independent, if sometimes arbitrary, source.

1.5 Outline of following chapters

The remainder of this thesis can be considered as having three parts. In the next two chapters I describe the lexical representation language in detail, discussing first the use of typed feature structures and then default inheritance. In Chapter 4 I give some examples of the use of LAUREL in the representation of lexical semantic information. These examples are essentially illustrative, intended to demonstrate how the formal machinery described may be used in representation. In Chapter 5 I give a more detailed and explicit account of individuation in nominals, discussing some aspects of the representation of mass, plural and group denoting nominals. The next two chapters discuss dictionary definitions, both to describe their practical utility as sources of the hierarchical structure of the lexicon, and also as a test of the treatment of individuation described in chapter 5.

Chapter 2

The lexical representation language

In the introduction I stated as a hypothesis that lexical semantic information could be represented using a unification-based language similar to that used in much current work in computational linguistics. In this chapter I introduce such a language, LAUREL, and discuss its non-default aspects. [1]

2.1 Introduction

As I mentioned in the previous chapter, the adoption of a unification-based approach seems natural, given the goals of integrating lexical semantics with syntax and formal semantics in a computationally tractable language. Unification based approaches also allow the paradigmatic and syntagmatic aspects of representation to be tightly interrelated, which is particularly important because there are close similarities between some lexical processes and others which would traditionally be classed as morphological or grammatical.

LAUREL belongs in the tradition of unification-based language such as PATR-II (Shieber *et al.*, 1983; Shieber, 1986) and FUG (Kay, 1979, 1984) which are not tied to any particular linguistic theory, but are sufficiently flexible that they can be used to implement a range of possible approaches. This makes experimentation much easier, but has the disadvantage that the language itself provides few constraints on the theory: a lexical entry could contain almost anything, for example. Languages such as PATR have no mechanism for stating restrictions that apply globally, but some more recent unification-based languages (see e.g. Calder, 1987; Carpenter, 1990, 1992; Emele and Zajac, 1990; Zajac, in press) allow the specification of a *type system* which gets round this problem to a considerable extent. Using a type system it is possible to specify, for example, that lexical entries have precisely the three components phonology, syntax and semantics, and to limit the possible values of these components. In principle, at least, it is possible to make a clear distinction between the global constraints imposed by the linguistic theory and the particular instantiation embodied by the lexical items which have been encoded. Typing can be used as a way of capturing generalisations. Template mechanisms in untyped languages, such as PATR, allow generalisations in the lexicon to be captured by inheritance, although templates, unlike types, do not restrict the class of linguistic entities. Furthermore, the effect of templates is restricted to the language for description of structures, whereas types are an integral part of the structure, and thus can affect the results of combination of entities in a way which templates cannot.

[1] LAUREL was developed as the lexical representation language (LRL) for the ACQUILEX LKB system, which I implemented. The formalisation of the type feature language, given in section 2.4 was carried out jointly with Valeria de Paiva (see de Paiva, in press) but other unattributed work reported here is my own. The ACQUILEX LKB is a fully implemented system, which has been used by all the groups who worked on ACQUILEX and by some researchers at other sites. A brief section on the ACQUILEX LKB is included in Chapter 7, but many aspects will not be discussed here. Copestake (1992) gives further details.

I suggested above that unification-based approaches allowed tight relationships between the paradigmatic and syntagmatic aspects of representation. This is based on the assumption that the same underlying operation of unification should be used both to capture the paradigmatic structure of the lexicon and to account for syntagmatic combination. The language for description of lexical entries allows structures which correspond to parts of entries, or underspecified entries, to be combined using exactly the same unification operation as that for syntagmatic combination. But this assumption is certainly not necessary. DATR (Evans and Gazdar, 1990) is a language for the description of feature structures which is not itself based on the unification of feature structures. One reason why such a differentiation might be desirable is that it seems that the most succinct and elegant descriptions of many aspects of lexical structure involve the use of default inheritance (see, for example Flickinger, 1987), while it is frequently assumed that combination in the syntagmatic plane should not involve any notion of defeasibility. If this is accepted, the lexical description language necessarily incorporates operations which are not allowed in the combination of lexical entries. I will discuss the use of defaults in lexical representation in more detail in the next chapter.

However, there are advantages in retaining a close correspondence between paradigmatic and syntagmatic representations. The interface between DATR and a parser is non-trivial (see Kilbury *et al.*, 1991) because DATR does not incorporate a notion of reentrancy. If the benefits of typing are accepted, it is desirable that the type system operates over both paradigmatic and syntagmatic components. Finally, the theoretical division between the lexicon and the rest of the system is not completely clearcut, particularly when extended senses and derivational morphology are considered, as will be discussed in Chapter 4. Minimising the distinction means that the representation language is too general, in that the stipulation that defaults are not used in syntagmatic combination, for example, is somewhat arbitrary, but it does allow such issues to be investigated more easily, and is thus appropriate in a relatively 'theory-neutral' language.

2.2 LAUREL

LAUREL can be viewed as an augmentation of a typed graph-based unification formalism with default inheritance; default inheritance is formalised in terms of default unification of feature structures (e.g. Carpenter, 1991). The typed feature structure language is based on Carpenter's (1990, 1992) work, although there are some significant differences. The type system can be regarded as a way of providing (non-default) inheritance, combined with error-checking. The notion of types, and features appropriate for a given type, gives some of the functionality of frame representation languages, such as KL-ONE (Brachman and Schmolze 1985); in particular, classification of a feature structure is possible.

The typed feature structure language is augmented with a default inheritance mechanism. This can be used to organise the lexicon in a completely user-defined way, to allow morphological or syntactic information to be concisely specified, for example, as has been done with DATR and other systems (for example, Russell *et al.*, 1991; Krieger and Nerbonne, in press). However, much of the motivation behind the formalisation of default inheritance in LAUREL comes from the concept of using inheritance between lexical entries to structure the lexicon along a semantic dimension.

Thus the operations that LAUREL supports are (default) inheritance, (default) unification and rule application (which can be formalised in terms of inheritance and unification). The type system provides the non-default inheritance mechanism and constrains default inheritance. Lexical rules can be used as a further component of lexical structure, but are also constrained by the type system.

Compared with many recent unification based formalisms, LAUREL is in some ways an impoverished language. It does not support disjunction of feature structures, negation or implication. Although it is based on Carpenter's work, it does not incorporate inequalities nor the extensional/intensional distinction which he describes. Some of these extensions may well turn out to be desirable in lexical semantic representation, but the aim here is to start with a relatively straightforward language and to extend it in response to problems which may arise.

2.3 Untyped feature structures

In this section I briefly recapitulate some of the basic definitions of untyped feature structures, including subsumption, unification and generalisation. This description closely follows Carpenter (1991), however I will use a notation in which more specific entities (feature structures and, later on, types) are uniformly described as being lower in some hierarchy.

A feature structure is either basic, in which case it must be an *atomic value*, or it is complex, in which case it provides values for one or more *features*. These values in turn are either atomic or complex feature structures. The set Feat of features and set Atom of atomic values are assumed to be finite. Feature structures can be defined in terms of labelled finite-state automata following Kasper and Rounds (1986, 1990).

Definition 1 (Feature Structure) *A feature structure is a tuple* $F = \langle Q, q_0, \delta, \alpha \rangle$ *where:*

- Q*: a finite set of nodes*
- $q_0 \in Q$*: the root node*
- $\delta : \mathsf{Feat} \times Q \to Q$ *: the partial feature value function (transition function)*
- $\alpha : Q \to \mathsf{Atom}$ *: the partial atomic value function*

with the following constraints:

Connectedness *There must be some* path *from the root to every node.*
Paths are defined as sequences of zero or more features. Let $\mathsf{Path} = \mathsf{Feat}^*$ *and* ϵ *be the empty path. The definition of the transition function can be extended to paths by taking* $\delta(\epsilon, q) = q$ *and* $\delta(f \cdot \pi, q) = \delta(\pi, \delta(f, q))$*. Then the connectedness condition can be expressed as stating that there must be a path* π *such that* $q = \delta(\pi, q_0)$ *for all* $q \in Q$

Atomic values *Only nodes without features can be atomic values so that if* $\alpha(q)$ *is defined then* $\delta(f, q)$ *is undefined for every* $f \in \mathsf{Feat}$ (But note that not necessarily all of the nodes without features are labelled with atomic values.)

Acyclicity *The resulting graph is acyclic in that there is no path* π *and non-empty path* π' *such that* $\delta(\pi, q_0) = \delta(\pi \cdot \pi', q_0)$.
(Although it is not necessary to rule out cycles in untyped or typed feature structures doing so makes some definitions simpler and the implementation more straightforward and so the acyclicity condition was adopted in LAUREL.)

In Pereira and Shieber (1984) the atomic value function α is required to be one-to-one; that is atoms are treated as *extensional* or non-copyable (see Carpenter, 1992:35–36). This requirement is not adopted here and so all feature structures are treated as intensional. Thus the feature structures shown in Figure 2.1 are both possible and distinct, whereas under an extensional treatment only the first is allowed (although, because there is no ambiguity, it would be usual to write it without making the reentrancy explicit in the AVM notation, as in the matrix for the intensional example). Although this distinction is not particularly important in practice when using untyped feature structures and non-default operations, since the structures are equivalent with respect to unification, there is a difference in behaviour with typed feature structures, since atomic values may be further specialised, and, arguably, there should also be a difference with default unification on untyped feature structures (see Section 3.1). Carpenter (1992:124f) discusses extensionality and intensionality and generalises the concept so that all types, not just atomic values, may be either intensional or extensional. However, for simplicity, the treatment adopted in LAUREL is purely intensional.

2.3.1 Subsumption

Feature structures can be regarded as being ordered by information content — a feature structure is said to *subsume* another if the latter carries extra information. Subsumption can be formalised

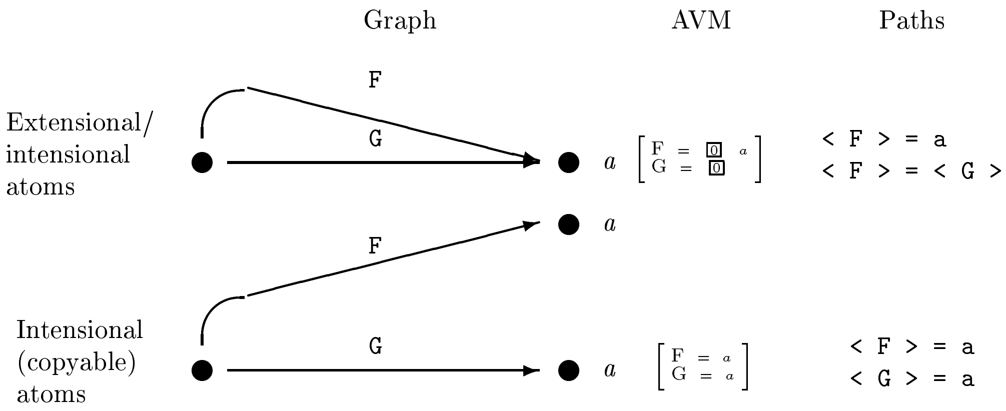


Figure 2.1: Extensional vs. intensional behaviour of atoms

by assuming that a pair of sets is associated with each feature structure which determine the path equivalences that hold and the atomic values assigned to paths. Let $\equiv_F$ be the equivalence relation induced between paths by the structure sharing in F and $\mathcal{P}_F$ be the partial function induced by F which maps paths in F to atomic values.

Definition 2 (Abstract Feature Structure) *If $F = \langle Q, q_0, \delta, \alpha \rangle$ is a feature structure, we let $\equiv_F \subseteq \mathsf{Path} \times \mathsf{Path}$ and $\mathcal{P}_F : \mathsf{Path} \rightarrow \mathsf{Atom}$ be such that:*

- *(Path Equivalence)*
  $\pi \equiv_F \pi'$ if and only if $\delta(\pi, q_0) = \delta(\pi', q_0)$
- *(Path Value)*
  $\mathcal{P}_F(\pi) = \sigma$ if and only if $\alpha(\delta(\pi, q_0)) = \sigma$.

The pair $\langle \mathcal{P}, \equiv_F \rangle$ is called the abstract feature structure corresponding to F.

A feature structure F subsumes another feature structure F' if and only if the information in F is contained in the information in F'; that is, if F' provides at least as much information about path values and structure sharing as F. The abstract feature structure corresponding to F is sufficient to determine its information content and thus subsumption.

Definition 3 (Subsumption) *F subsumes F', written $F' \sqsubseteq F$, if and only if:*

- *$\pi \equiv_F \pi'$ implies $\pi \equiv_{F'} \pi'$*
- *$\mathcal{P}_F(\pi) = \sigma$ implies $\mathcal{P}_{F'}(\pi) = \sigma$*

Thus F subsumes F' if and only if every piece of information in F is contained in F'. For untyped feature structures, $\top$ is defined to be the single node feature structure with no atomic value assigned. Thus $\pi \equiv_\top \pi'$ if and only if $\pi = \pi' = \epsilon$ and $\mathcal{P}(F)$ is undefined everywhere. Note that $F \sqsubseteq \top$ for every feature structure F. $\top$ is used in attribute-value matrices to denote the lack of any known value.

Feature structures which only vary in the identity of their nodes are said to be *alphabetic variants*. Thus if $F \sqsubseteq F'$ and $F' \sqsubseteq F$, then F and F' are alphabetic variants and we write $F \sim F'$. The definitions of operations on feature structures such as unification are not sensitive to any distinction between alphabetic variants.

2.3.2 Unification

Unification corresponds to conjunction of information, and thus can be defined in terms of subsumption, which is a relation of information containment. The unification of two feature structures is defined to be the most general feature structure which contains all the information in both of the feature structures.

Definition 4 (Unification) *The unification $F \sqcap F'$ of two feature structures F and F' is taken to be the greatest lower bound of F and F' in the collection of feature structures ordered by subsumption.*

Thus $F \sqcap F' = F''$ if and only if $F'' \sqsubseteq F$, $F'' \sqsubseteq F'$ and for every F''' such that $F''' \sqsubseteq F$ and $F''' \sqsubseteq F'$ it is also the case that $F''' \sqsubseteq F''$.

2.3.3 Generalisation

Generalisation is the opposite of unification in which the lowest upper bound of two feature structures is taken. The generalisation of two feature structures is defined to be the most specific feature structure which contains only information found in both feature structures.

Definition 5 (Generalisation) *The generalisation $F \sqcup F'$ of two feature structures is defined to be their greatest lower bound in the subsumption ordering.*

Thus $F \sqcup F' = F''$ if and only if $F \sqsubseteq F''$, $F' \sqsubseteq F''$ and for every F''' such that $F \sqsubseteq F'''$ and $F' \sqsubseteq F'''$ then $F'' \sqsubseteq F'''$. Although unification corresponds to conjunction, generalisation does not correspond to disjunction but is more like information intersection. In languages which support it, disjunction of feature structures will give a result which is either more specific than that produced by generalisation (defined as an operation which produces a non-disjunctive feature structure) or equal to it.

2.4 Typed feature structures

As stated earlier the approach to typed feature structures adopted here is based on that of Carpenter (1992), but whereas Carpenter describes both appropriateness conditions and constraints on types, here the system is simplified and a notion of constraint alone is used, which is, however, less powerful than that employed by Carpenter in that disjunction is not allowed. Constraints in LAUREL do support type checking and classification in much the same way as Carpenter's appropriateness conditions do and can alternatively be regarded as a generalisation of the notion of appropriateness conditions. (The systems are compared in more detail in section 2.4.5.) A type system defined in LAUREL can be described as having three components; a fixed finite set of features Feat, a fixed type hierarchy $\langle \mathsf{Type}, \sqsubseteq \rangle$ with a finite set of types, and a constraint function C which associates a constraint feature structure with every type which determines which feature structures are *well-formed*.

2.4.1 The type hierarchy

The definition of a type hierarchy is almost identical to that used by Carpenter but as before I follow the reverse convention on ordering, so that more specific types are described as being lower in the hierarchy. The type hierarchy $\langle \mathsf{Type}, \sqsubseteq \rangle$ defines a partial order (notated $\sqsubseteq$, "is more specific than") on the types and specifies which types are *consistent*. A set of types is said to be consistent if the members of the set share a common subtype. That is the subset $S \subseteq \mathsf{TYPE}$ is consistent iff there is some t_0 in TYPE such that $t_0 \sqsubseteq t$ for any t in S. The significance of this is that only feature structures with mutually consistent types can be unified. Two types which are unordered in the hierarchy are assumed to be inconsistent unless a common subtype has been explicitly specified. It is a condition on the type hierarchy that every *consistent* set of types $S \subseteq \mathsf{TYPE}$ must have a unique greatest lower bound or meet (notation $\sqcap S$). The unique greatest lower bound condition on consistent types allows feature structures to be typed deterministically; if two feature structures of types $\mathbf{a}$ and $\mathbf{b}$ are unified the type of the result will be $\mathbf{a} \sqcap \mathbf{b}$, which must be unique if it exists. If $\mathbf{a} \sqcap \mathbf{b}$ does not exist unification fails. Thus in the fragment of a type hierarchy shown in Figure 2.2 **artifact** and **physical** are consistent; **artifact** $\sqcap$ **physical** $=$ **art_phys**. I will use a very simple type system at this point for ease of exposition.

Because the type hierarchy is a partial order it has properties of reflexivity, transitivity and anti-symmetry (from which it follows that the type hierarchy cannot contain cycles). Note that

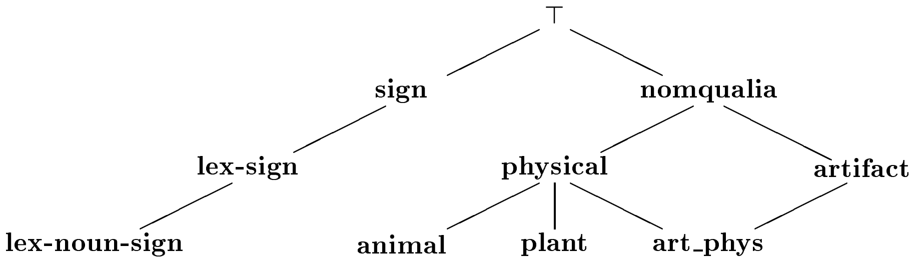


Figure 2.2: A fragment of a type hierarchy

the empty set is (vacuously) consistent, as for any t_0 in TYPE it satisfies the condition that $t_0 \sqsubseteq t$ for all t's in the empty set. The maximal element $\top$ of $\langle \mathsf{TYPE}, \sqsubseteq \rangle$ is defined as the meet of the empty set, $\top = \sqcap\emptyset$. This element $\top$ is such that $t \sqsubseteq \top$ for any t in TYPE. Thus the join operation on the type hierarchy is total.

The meet $\sqcap$ operation can be made total by adding the join of the empty-set $\perp = \sqcup\emptyset$ to $\langle \mathsf{TYPE}, \sqsubseteq \rangle$. But even if $\perp$ is added to make $\langle \mathsf{TYPE}, \sqsubseteq \rangle$ a lattice, this lattice need not be distributive. For example, given the type hierarchy in Figure 2.2 (**animal** $\sqcup$ **plant**) $\sqcap$ **artifact** = **art_phys** but (**animal** $\sqcap$ **artifact**) $\sqcup$ (**plant** $\sqcap$ **artifact**) = $\perp$. Although meet will be taken as corresponding to conjunction, join does not correspond to disjunction. (The lattice could be completed so that join was equivalent to disjunction, see section 2.5.) There is thus a symmetry with the behaviour of feature structures, where generalisation and disjunction are not equivalent. Nothing in what follows depends on whether the existence of a bottom element in the type hierarchy, $\perp$, is assumed or not. I use $\perp$ as a notational convenience to indicate inconsistency, just as $\perp$ is used to indicate unification failure for feature structures.

2.4.2 Typed feature structures

Initially I will define the set $\mathcal{F}$ of typed feature structures to include both those which are, and those which are not, well-formed with respect to a particular type system. A typed feature structure is defined as a tuple $F = \langle Q, q_0, \delta, \alpha \rangle$ where the only significant difference from the definition in the untyped case is that instead of α being a partial atomic value function it is a total node typing function, $\alpha\colon Q \rightarrow \mathsf{TYPE}$. Thus every node has a type.

Definition 6 (Typed Feature Structure) *A typed feature structure is a tuple $F = \langle Q, q_0, \delta, \alpha \rangle$ where:*

- *Q: a finite set of (connected, acyclic) nodes*
- *$q_0 \in Q$: the root node*
- *$\delta : \mathsf{Feat} \times Q \rightarrow Q$: the partial feature value function*
- *$\alpha : Q \rightarrow \mathsf{Type}$: the total node typing function*

with connectedness and acyclicity being defined as in the untyped case.

Consider the following example of a feature structure:

$$F_1 = \begin{bmatrix} \textbf{artifact} & \\ \text{TELIC} = & \begin{bmatrix} \textbf{verb-sem} \\ \text{IND} = \boxed{0}\ \textbf{eve} \\ \text{PRED} = \textbf{drink\_1} \\ \text{ARG1} = \boxed{0} \\ \text{ARG2} = \textbf{obj} \\ \text{ARG3} = \textbf{obj} \end{bmatrix} \end{bmatrix}$$

As every feature structure has a unique initial node, q_0, the type of a feature structure can be said to be the type of its initial node, that is:

Definition 7 (Type of a feature structure) *If $F = \langle Q, q_0, \delta, \alpha \rangle$ then $Typeof(F) = \alpha(q_0)$.*

Thus the type of feature structure F_1 is **artifact**.

The definition of subsumption of typed feature structures is very similar to that for untyped feature structures, with the additional proviso that the ordering must be consistent with the ordering on their types. The symbol $\sqsubseteq$ ("is-more-specific-than", "is-subsumed-by") is thus overloaded to express subsumption of feature structures as well as the ordering on the type hierarchy. Thus if F_1 and F_2 are feature structures of types t_1 and t_2 respectively, then $F_1 \sqsubseteq F_2$ only if $t_1 \sqsubseteq t_2$. If two feature structures are identical except for their types then the subsumption ordering on the feature structures will be equivalent to the ordering on their types.

Definition 8 (Abstract Typed Feature Structure) *If $F = \langle Q, q_0, \delta, \alpha \rangle$ is a feature structure, we let $\equiv_F \subseteq \mathsf{Path} \times \mathsf{Path}$ and $\mathcal{P}_F : \mathsf{Path} \rightarrow \mathsf{Type}$ be such that:*

- *(Path Equivalence)*
  $\pi \equiv_F \pi'$ *if and only if* $\delta(\pi, q_0) = \delta(\pi', q_0)$
- *(Path Value)*
  $\mathcal{P}_F(\pi) = t$ *if and only if* $\alpha(\delta(\pi, q_0)) = t$.

Definition 9 (Subsumption of typed feature structures) *F subsumes F', written $F' \sqsubseteq F$, if and only if:*

- $\pi \equiv_F \pi'$ *implies* $\pi \equiv_{F'} \pi'$
- $\mathcal{P}_F(\pi) = t$ *implies* $\mathcal{P}_{F'}(\pi) = t'$ *and* $t' \sqsubseteq t$

Unification of typed feature structures is defined in the same way as for untyped feature structures, that is the unification of two typed feature structures will be their greatest lower bound in the subsumption ordering. Since if F and F' are feature structures of types t and t' respectively, their unification $F \sqcap F'$ has to have type $t \sqcap t'$, unification will fail if $t \sqcap t'$ does not exist. Generalisation can also be defined in the same way as for untyped feature structures, that is as the lowest upper bound of two feature structures in the subsumption ordering.

Thus the type hierarchy makes a finer grained differentiation of feature structures possible, because the ordering is dependent on the type hierarchy as well as the order of untyped feature structure subsumption. For example, if F_1 and F_2 are the two feature structures given below, then $F_2 \sqsubset F_1$ assuming that **e-sentient** $\sqsubset$ **obj**:

$$F_1 = \begin{bmatrix} \textbf{artifact} & \\ \text{TELIC} = & \begin{bmatrix} \textbf{verb-sem} \\ \text{IND} = \boxed{1}\ \textbf{eve} \\ \text{PRED} = \textbf{drink\_1} \\ \text{ARG1} = \boxed{1} \\ \text{ARG2} = \textbf{obj} \\ \text{ARG3} = \textbf{obj} \end{bmatrix} \end{bmatrix} \quad F_2 = \begin{bmatrix} \textbf{artifact} & \\ \text{TELIC} = & \begin{bmatrix} \textbf{verb-sem} \\ \text{IND} = \boxed{1}\ \textbf{eve} \\ \text{PRED} = \textbf{drink\_1} \\ \text{ARG1} = \boxed{1} \\ \text{ARG2} = \textbf{e-sentient} \\ \text{ARG3} = \textbf{obj} \end{bmatrix} \end{bmatrix}$$

This is in itself useful; for example it allows selectional restrictions to be encoded as sorts in a way which is extremely cumbersome with untyped feature structures (cf. Moens *et al.*, 1989). However the type system also specifies a set of *well-formed* feature structures, which satisfy a constraint function, and this gives the system a functionality which includes the properties of inheritance, error-checking and classification. It is at this point that LAUREL begins to differ significantly from Carpenter's system.

2.4.3 Constraints

The constraint function C defines the set of feature structures $\mathcal{WF}$ which are well-formed with respect to the type system. In LAUREL every type must have exactly one associated feature structure which acts as a constraint on all feature structures of that type; by subsuming all well-formed feature structures of that type. The constraint also defines which features are *appropriate* for a particular type; in a well-formed feature structure each node must have all the features appropriate to its type and no others. Constraints are inherited by all subtypes of a type, but a subtype may introduce new features (which will be inherited as appropriate features by all its subtypes). A constraint on a type is a well-formed feature structure of that type; all constraints must therefore be mutually consistent.

For example, considering some of the types shown in Figure 2.2, the constraint associated with the type **artifact** might be:

$$\left[\begin{array}{ll} \textbf{artifact} & \\ \text{TELIC} = & \left[\begin{array}{l} \textbf{verb-sem} \\ \text{IND} = \boxed{0}\ \textbf{eve} \\ \text{PRED} = \textbf{string} \\ \text{ARG1} = \boxed{0} \\ \text{ARG2} = \textbf{obj} \\ \text{ARG3} = \textbf{obj} \end{array}\right] \end{array}\right]$$

This constraint states that any feature structure of type **artifact** must have a feature structure of type **verb-sem** as the value for its TELIC feature. This constraint must be consistent with the constraints on all of its subparts, i.e. it must itself be a well-formed feature structure. In this case, for example, the constraint on **verb-sem** is:

$$\left[\begin{array}{l} \textbf{verb-sem} \\ \text{IND} = \boxed{0}\ \textbf{eve} \\ \text{PRED} = \textbf{string} \\ \text{ARG1} = \boxed{0} \\ \text{ARG2} = \textbf{obj} \\ \text{ARG3} = \textbf{sem} \end{array}\right]$$

and thus the constraint on **artifact** is well-formed, given that **obj** $\sqsubseteq$ **sem**.

The type **physical** might have constraint:

$$\left[\begin{array}{l} \textbf{physical} \\ \text{FORM} = \textbf{form} \end{array}\right]$$

Here **form** is an atomic type, and has no appropriate features, the constraint on **form** is simply the atomic feature structure [**form**]. The constraint on **art_phys** will contain information inherited from both parents, thus:

$$\left[\begin{array}{ll} \textbf{art\_phys} & \\ \text{FORM} = \textbf{form} & \\ \text{TELIC} = & \left[\begin{array}{l} \textbf{verb-sem} \\ \text{IND} = \boxed{0}\ \textbf{eve} \\ \text{PRED} = \textbf{string} \\ \text{ARG1} = \boxed{0} \\ \text{ARG2} = \textbf{obj} \\ \text{ARG3} = \textbf{obj} \end{array}\right] \end{array}\right]$$

Given that **mass** is an atomic subtype of **form**, and that **e-sentient** is an atomic subtype of **obj**, the feature structure below is well-formed since it contains all the appropriate features and no inappropriate ones, it is subsumed by the constraints on its type and all its substructures are well-formed.

$$\left[\begin{array}{ll} \textbf{art\_phys} & \\ \text{FORM} = \textbf{mass} & \\ \text{TELIC} = & \left[\begin{array}{l} \textbf{verb-sem} \\ \text{IND} = \boxed{0}\ \textbf{eve} \\ \text{PRED} = \textbf{string} \\ \text{ARG1} = \boxed{0} \\ \text{ARG2} = \textbf{e-sentient} \\ \text{ARG3} = \textbf{obj} \end{array}\right] \end{array}\right]$$

Formally, the notion of appropriate features is defined as follows:

Definition 10 (Appropriate features) *If $C(t) = \langle Q, q_0, \delta, \alpha\rangle$ then the appropriate features of t are defined as $Appfeat(t) = Feat(\langle F, q_0\rangle)$ where $Feat(\langle F, q\rangle)$ is defined to be the set of features labelling transitions from the node q in some feature structure F i.e. $f \in Feat(\langle F, q\rangle)$ such that $\delta(f, q)$ is defined.*

We can then define the constraint function:

Definition 11 (Constraint function) *The constraint function which associates constraint feature structures with types is given by $C\colon \langle \mathsf{TYPE}, \sqsubseteq\rangle \rightarrow \mathcal{F}$.*
This must satisfy the following conditions:

Monotonicity *Given types t_1 and t_2 if $t_1 \sqsubseteq t_2$ then $C(t_1) \sqsubseteq C(t_2)$*

Type *For a given type t, if $C(t)$ is the feature structure $\langle Q, q_0, \delta, \alpha\rangle$ then $\alpha(q_0) = t$.*

Compatability of constraints *For all $q \in Q$ the feature structure $F' = \langle Q', q, \delta, \alpha\rangle \sqsubseteq C(\alpha(q))$ and $Feat(q) = Appfeat(\alpha(q))$.*

Maximal introduction of features *For every feature $f \in \mathsf{FEAT}$ there is a unique type $t = Maxtype(f)$ such that $f \in Appfeat(t)$ and there is no type s such that $t \sqsubset s$ and $f \in Appfeat(s)$. The maximal appropriate value of a feature $Maxappval(f)$ is the type t such that if $C(Maxtype(f)) = \langle Q, q_0, \delta, \alpha\rangle$ then $t = \alpha(\delta(f, q_0))$*

The compatability condition implies that no constraint feature structure $C(t) = F$ can strictly contain a feature structure of type t or any subtype of t. That is, if F is given by $\langle Q, q_0, \delta, \alpha\rangle$, then for all non-initial nodes $q \in Q$ such that $q \neq q_0$ the type of the node $\alpha(q) \not\sqsubseteq t$. If such a node existed it would have to be the initial node of a feature structure $F|_q$ which was more specific than F, i.e. $F|_q \sqsubseteq F$, and would therefore itself have to contain such a node, and so on. Thus such a constraint could only be satisfied by a cyclic or infinite structure, and we disallow both of these possibilities. This condition does not, however, rule out recursive structures such as lists, because the type **list** can be defined to have two subtypes **empty-list** and **nonempty-list**, where the former has no appropriate features and the latter has two, HEAD which can take any value, and TAIL, which will take a value of type **list**. The type **nonempty-list** does not violate the compatability condition, since the structure can be terminated.

Definition 12 (Well-formed feature structures) *We say that a given feature structure $F = \langle Q, q_0, \delta, \alpha\rangle$ is a well-formed feature structure iff for all $q \in Q$, we have that $F' = \langle Q', q, \delta, \alpha\rangle \sqsubseteq C(\alpha(q))$ and $Feat(q) = Appfeat(\alpha(q))$.*

From these definitions it can be seen that all constraint feature structures are themselves well-formed feature structures.

Since the type system gives a concept of a well-formed feature structure, it follows that non-well-formed feature structures can be detected, allowing error checking. Typing also allows for a form of classification; a feature may only be introduced as appropriate at one point in the type hierarchy (and will be inherited as an appropriate feature by all subtypes of that type); it follows from this that there is a unique maximal type for any set of features, and therefore an untyped feature structure can always be typed deterministically. For example, assuming the type system introduced above, the attribute value specification:

```
< FORM > = mass
< TELIC : ARG2 > = e-sentient
```

would be expanded out into the following feature structure:

$$\left[\begin{array}{l} \textbf{art\_phys} \\ \text{FORM} = \textbf{mass} \\ \text{TELIC} = \left[\begin{array}{l} \textbf{verb-sem} \\ \text{IND} = \boxed{0}\ \textbf{eve} \\ \text{PRED} = \textbf{string} \\ \text{ARG1} = \boxed{0} \\ \text{ARG2} = \textbf{e-sentient} \\ \text{ARG3} = \textbf{obj} \end{array}\right] \end{array}\right]$$

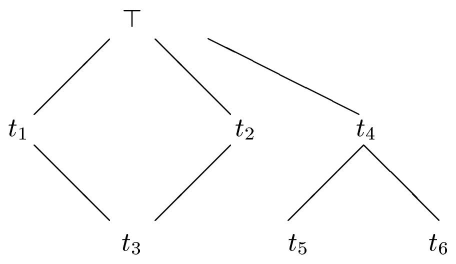
(Full details of the feature structure description language are given in section 2.6.1, below.) The type of the feature structure is determined automatically; since the features FORM and TELIC are specified, its type has to be **art_phys** (or some subtype of that type). The procedure for making a feature structure well-formed, WF, involves recursively unifying the constraints of the maximal types of the subparts of the feature structure; since we disallow cyclic feature structures, this operation terminates straightforwardly at the atomic feature structures.

2.4.4 Unification and well-formedness

The definition of subsumption for well-formed feature structures is exactly the same as that for typed feature structures. The result of well-formed unification of well-formed feature structures,

$\langle \mathcal{WF}, \sqsubseteq \rangle$, is defined to be the greatest lower bound in the subsumption ordering of well-formed feature structures and is thus itself well-formed. This does however mean that the algorithm for unification involves an additional step, because it is potentially necessary to unify in the constraint feature structure associated with the meet of the types of the feature structures being unified.

If F_1 and F_2 are well-formed feature structures of types t_1 and t_2 respectively, then $F_1 \sqcap F_2$, if it exists, has type $t_1 \sqcap t_2$. Since F_1 and F_2 are well-formed, in particular we know that $F_1 \sqsubseteq C(t_1)$ and $F_2 \sqsubseteq C(t_2)$. Thus if F_1 and F_2 are consistent, $F_1 \sqcap F_2 \sqsubseteq C(t_1) \sqcap C(t_2)$. But to be well-formed $F_1 \sqcap F_2$ has to satisfy $F_1 \sqcap F_2 \sqsubseteq C(t_1 \sqcap t_2)$ and $C(t_1 \sqcap t_2)$ might be more specific than $C(t_1) \sqcap C(t_2)$. Consider the following example of a type hierarchy:



Assume that the types t_4, t_5 and t_6 are atomic (i.e. they have constraints $[t_4]$, $[t_5]$ and $[t_6]$, respectively) and the constraints on types t_1, t_2 and t_3 are:

$$C(t_1) = \begin{bmatrix} t_1 \\ \mathrm{F} = t_4 \end{bmatrix} \quad C(t_2) = \begin{bmatrix} t_2 \\ \mathrm{G} = \top \end{bmatrix} \quad C(t_3) = \begin{bmatrix} t_3 \\ \mathrm{F} = t_5 \\ \mathrm{G} = \top \\ \mathrm{H} = \top \end{bmatrix}$$

We then have: $C(t_1) \sqcap C(t_2) = \begin{bmatrix} t_3 \\ \mathrm{F} = t_4 \\ \mathrm{G} = \top \end{bmatrix}$

Then $t_3 = t_1 \sqcap t_2$ but $C(t_3) \sqsubset C(t_1) \sqcap C(t_2)$

Consider the following well-formed feature structures, F_1, F_2 and F_3:

$$F_1 = \begin{bmatrix} t_1 \\ \mathrm{F} = t_4 \end{bmatrix} \qquad F_2 = \begin{bmatrix} t_2 \\ \mathrm{G} = \top \end{bmatrix} \quad F_3 = \begin{bmatrix} t_1 \\ \mathrm{F} = t_6 \end{bmatrix}$$

$$F_1 \sqcap F_2 = \begin{bmatrix} t_3 \\ \mathrm{F} = t_4 \\ \mathrm{G} = \top \end{bmatrix}$$

$$F_2 \sqcap F_3 = \begin{bmatrix} t_3 \\ \mathrm{F} = t_6 \\ \mathrm{G} = \top \end{bmatrix}$$

$F_1 \sqcap F_2$ is not a well-formed feature structure of type t_3 as $F_1 \sqcap F_2 \not\sqsubseteq C(t_3)$. To extend it to a well-formed feature structure involves unifying in $C(t_3)$. $F_2 \sqcap F_3$ is also not a well-formed feature structure but it cannot be extended to a well-formed feature structure, because its value for F is inconsistent with the constraint for t_3.

The same situation can arise with Carpenter's appropriateness specifications (see below) as discussed in Carpenter (1992:104f). We could impose a condition of meet preservation on the constraints, analagous to Carpenter's join preservation on appropriateness conditions, but this is over-restrictive. Instead we use the following definition for the operation of well-formed unification, $\sqcap_w$:

Definition 13 (Well-formed unification) *The well-formed unification of F_1 and F_2, $F_1 \sqcap_w F_2$ is given by $F_1 \sqcap F_2 \sqcap C(t_1 \sqcap t_2)$. This will be well-formed if it exists.*

No such complication arises with generalisation, since $F = F' \sqcup F''$ will always be well-formed if F' and F'' are well-formed. From now on, when talking about typed feature structures, I will use the term feature structure to mean well-formed typed feature structure and $\sqcap$ will be used to refer to well-formed unification, $\sqcap_w$, dropping the subscript.

2.4.5 A comparison with Carpenter's system

Rather than specifying typing by associating a feature structure with every type, Carpenter uses appropriateness specifications, which are defined as follows: [2]

Definition 14 (Appropriateness Specification) *An appropriateness specification over the inheritance hierarchy* $\langle \mathsf{TYPE}, \sqsubseteq \rangle$ *and features* $Feat$ *is a partial function* *Approp*: $Feat \times Type \rightarrow Type$ *that meets the following conditions:*

Feature Introduction *For every feature* $f \in \mathsf{FEAT}$ *there is a most general type* $Intro(f) \in Type$ *such that* $Approp(f, Intro(f))$ *is defined.*

Upward Closure / Right Monotonicity *If* $Approp(f,t)$ *is defined and* $t' \sqsubseteq t$ *then* $Approp(f,t')$ *is also defined and* $Approp(f,t') \sqsubseteq Approp(f,t)$

As mentioned above, our constraints can be seen as a generalisation of Carpenter's appropriateness specification in that the two would be equivalent if we added the condition to our constraint function that all constraints be one-level, non-reentrant feature structures (i.e. for all features, f, in $Feat(F, q_0)$, the feature structures $\delta(f, q_0)$ are distinct and atomic). Our monotonicity condition is derived from Carpenter's upward closure (downward closure in our terms) and similarly our maximal introduction of features is derived from Carpenter's feature introduction condition. Our other two conditions are necessary because constraints have themselves to be well-formed feature structures in order to have a consistent system.

Carpenter defines two notions of typing; *well-typedness*, where feature structures may only contain appropriate features with appropriate values, and *total well-typedness* where feature structures must meet the well-typedness conditions and must also contain all the appropriate features. Our notion of well-formedness is thus comparable to Carpenter's *total well-typedness.* As Carpenter shows finite total typing, that is the extension of a well-typed feature structure to a most general finite totally typed feature structure, is only possible under conditions where *Approp* contains no loops; as shown above, an equivalent condition arises from our consistency condition.

2.4.6 Error checking, classification and constraint satisfaction

Error checking and classification with respect to a type system in the system described are computationally efficient, because we disallow cyclic feature structures, but have limitations. One disadvantage is that it is not possible, in general, to enforce cooccurrence restrictions, even of a quite limited sort. For example, suppose we have a type **t** with two features, F and G which both take boolean values, but we wish to state that when F has value **true**, G must have value **false** and vice versa. The nearest we could get to achieving this would be to define two subtypes of **t**, **t1** and **t2** with the following constraints:

$$\begin{bmatrix} \textbf{t1} \\ \text{F} = \textbf{true} \\ \text{G} = \textbf{false} \end{bmatrix} \qquad\qquad \begin{bmatrix} \textbf{t2} \\ \text{F} = \textbf{false} \\ \text{G} = \textbf{true} \end{bmatrix}$$

This does not really achieve the desired result, however. For example consider the following structure:

$$\begin{bmatrix} \textbf{t} \\ \text{F} = \textbf{true} \\ \text{G} = \textbf{true} \end{bmatrix}$$

This is still a well-formed feature structure, despite the fact that it cannot be extended to be a well-formed structure with a type corresponding to that of any leaf node in the type hierarchy.

For a more realistic example of a cooccurrence restriction, consider Sanfilippo's representation of verb semantics in LAUREL (Sanfilippo, in press), which is discussed in detail in Section 4.1. This treatment uses thematic roles and encodes restrictions on arguments of a predicate by sorting the variables. In order to do this a type **theta-formula** is defined to have the following constraint:

[2] This definition is taken from Carpenter (1992:86) but I use the reverse directionality on the type hierarchy for consistency with our definitions.

$$\left[\begin{array}{l} \textbf{theta-formula} \\ \text{IND} = \boxed{1}\ \textbf{eve} \\ \text{PRED} = \textbf{theta-relation} \\ \text{ARG1} = \boxed{1} \\ \text{ARG2} = \textbf{obj} \end{array}\right]$$

To classify psychological predicates, thematic predicates such as **theta-sentient** are used; in this case the second argument to any formula whose predicate is **theta-sentient** should denote a sentient entity; i.e. if the value of PRED is **theta-sentient** then the value of ARG2 is **e-sentient**. We would like to encode all the valid subtypes of **theta-formula**, e.g. **theta-sentient-formula**, with constraint:

$$\left[\begin{array}{l} \textbf{theta-sentient-formula} \\ \text{IND} = \boxed{1}\ \textbf{eve} \\ \text{PRED} = \textbf{theta-sentient} \\ \text{ARG1} = \boxed{1} \\ \text{ARG2} = \textbf{e-sentient} \end{array}\right]$$

But this is inadequate, as before, because we still have possible well-formed feature structures which violate the cooccurrence restrictions, such as:

$$\left[\begin{array}{l} \textbf{theta-formula} \\ \text{IND} = \boxed{1}\ \textbf{eve} \\ \text{PRED} = \textbf{theta-sentient} \\ \text{ARG1} = \boxed{1} \\ \text{ARG2} = \textbf{e-plant} \end{array}\right]$$

(Here I assume that **e-plant** $\sqcap$ **e-sentient** $= \bot$.)

This seems undesirable; the type system is assumed to be complete, in that, for example, the meet of two compatible types has to be explicitly specified, so intuitively we might expect such a feature structure to be ill-formed in some sense. We refer to a feature structure which can be extended to a well-formed structure where every type is a leaf type (cf. the HPSG concept of a sort-resolved feature structure (Pollard and Sag, in press) and also Zajac's notion of a ground feature structure (Zajac, in press)) as 'ultimately well-formed', and in theory we could enforce such cooccurrence restrictions by checking for ultimate well-formedness.

Definition 15 *(Ultimate well-formedness) A feature structure F is said to be ultimately well-formed iff it subsumes some well-formed type-resolved feature structure F', where a feature structure $\langle Q, q_0, \delta, \alpha \rangle$ is type-resolved if the types of all its nodes are leaf types, that is for all nodes $q \in Q$ there is no type t such that $t \sqsubset \alpha(q)$*

Note that an equivalent result applies to Carpenter's appropriateness specifications; a well-typed feature structure in his system cannot necessarily be extended to a well-typed sort-resolved feature structure. Carpenter leaves the extension of type inference so that it can cope with such cases as an open problem (see Carpenter, 1992:144f).

It is not possible to check for such examples efficiently in general, because to do so would involve, in effect, attempting to fully classify the feature structure with respect to the type system. We would have to consider not only the immediate subtypes of a type, but their subtypes, and so on. The classification process described above, in section 2.4.3, is limited in that the procedure only takes account of the top level features in a structure and not their values. (Again, the same is true of Carpenter's system.) Even if the only subtype of **binary-formula** which had a value for PRED which was compatible with **theta-relation** was **theta-formula**, the following feature structure would be classified as a **binary-formula** rather than a **theta-formula**:

$$\left[\begin{array}{l} \textbf{top} \\ \text{IND} = \boxed{1}\ \textbf{entity} \\ \text{PRED} = \textbf{theta-relation} \\ \text{ARG1} = \boxed{1} \\ \text{ARG2} = \textbf{obj} \end{array}\right]$$

The concept of satisfying ultimate well-formedness is related to to the more general notion of constraint resolution within a typed system (see Zajac, in press). Carpenter discusses constraint resolution (Carpenter, 1992: 227f) in detail. In his formalisation, only sort-resolved feature structures will be produced as solutions to a constraint system, if the convention is adopted that part of the constraint associated with a type is the disjunction of all its immediate subtypes. This forces a solution to contain only maximally specific types, because anything with a non-maximal type will not satisfy the description.

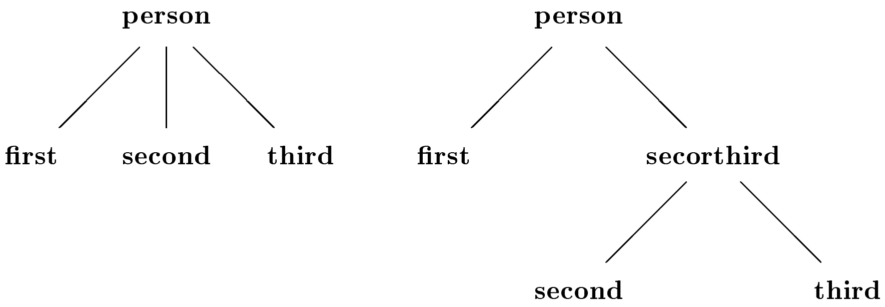


Figure 2.3: Adding types to the hierarchy

Allowing full classification to be invoked when lexical entries are being created may be a practical option which potentially has considerable advantages in allowing augmentation of automatically acquired information, however neither full classification nor ultimate well-formedness checking is currently implemented in the LKB system.[3] Zajac (in press) discusses implementations of constraint resolution. In practice, solving general disjunctive constraints turns out to be responsible for much of the computation time, and this would be avoided in LAUREL. However we currently avoid full classification and do not make use of it during parsing, which is based on the use of rules rather than constraint satisfaction.

2.5 Extensions to the language

A type system has essentially to be fully defined before lexical entries can be built. This causes obvious problems with respect to atomic types representing orthography and predicate names for example, where it is unrealistic to assume that the complete set can be known in advance. To get round this any string is allowed as a valid type; all strings are assumed to be subtypes of the predefined atomic type **string**, but to be unordered with respect to one another. Particular features such as ORTH, which are specified as having value **string**, thus in effect take arbitrary string values.[4]

Although many feature structure based languages allow disjunctive feature structures, this is avoided in LAUREL. Arbitrary disjunction can result in a computationally intractable system and it is not clear whether it is in fact necessary, given that the type system can be set up in a way which allows degrees of underspecification. However the lowest common supertype (join) of the types might be more general than the disjunction, in which case a new type might have to be created to get the required level of specification. For example if the type **person** was defined to have the subtypes **first**, **second**, **third**, a new type could be inserted in the hierarchy in order to express the equivalent of the disjunction **second** or **third** (see Figure 2.3).

The effect of the non-equivalence of disjunction and join may in fact be exploited to allow a more precise specification of a language. For example, Pollard and Sag (in press) give a description of the inflection of German adjectives such as *klein*, in which the following values for case are given as possibilities: **nom**, **acc**, **gen**, **dat**, **nom** ∨ **acc**, **gen** ∨ **dat**, *unspecified*. Encoding this in the type system, as shown in Figure 2.4, rather than making use of disjunction, would directly express the restriction that only these values were available and that the following were not: **nom** ∨ **gen**, **acc** ∨ **dat**, **nom** ∨ **acc** ∨ **gen**, **nom** ∨ **acc** ∨ **dat**, **nom** ∨ **gen** ∨ **dat**, **acc** ∨ **gen** ∨ **dat**.

However, in LAUREL, a limited form of disjunction is supported; particular atomic types may be specified as allowing disjunction with an effect which is formally similar to creating all possible

[3] An incomplete check for ultimate well-formedness is implemented, which catches the straightforward examples like that given above; this is optionally invoked when expanding lexical entries.

[4] It would be necessary to extend LAUREL for morphophonemic representation; there are several possible techniques, but the most interesting approach might be based on Bird's (1992) suggestions for integration of non-linear phonology and HPSG.

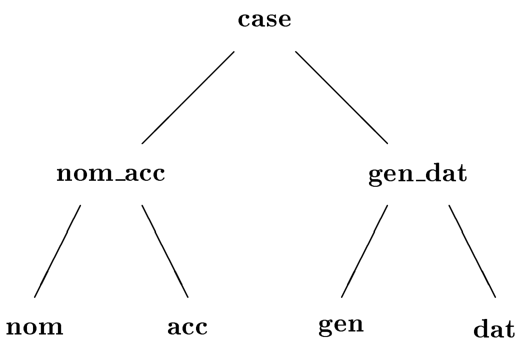


Figure 2.4: Restricting the possibilities for case specification

additional join types; this is expressed as a list of values, for example, (**second third**). For atomic types, completing some fragment of the type hierarchy under join gives the equivalent of disjunction (cf. Carpenter 1992;17f). The approach of completing the type hierarchy could be extended to complex feature structures but would, however, give a finer grain of generalisation rather than disjunction. In order to have a well-defined notion of constraint under this operation the following extra condition on constraints could be added:

Generalisation of constraints The constraint on a type t should be equal to the generalisation of the set of constraints of all its subtypes.
Let $t_1, t_2 \ldots t_n$ be the set of all types which are subtypes of t. Then $C(t) = \sqcap C(t_1), C(t_2) \ldots C(t_n)$.

From the usual monotonicity condition it follows that $\sqcap C(t_1), C(t_2) \ldots C(t_n) \sqsubseteq C(t)$ but it is desirable to avoid the situation where $\sqcap C(t_1), C(t_2) \ldots C(t_n) \sqsubset C(t)$, given that the type hierarchy is complete, because it would correspond to the situation where information which should be true of all feature structures of type t was being specified redundantly, at multiple points in the type hierarchy.

This condition would probably be over-restrictive during development of a type system. However, if we did introduce it, we could automatically complete the type lattice under the operation $\sqcap$ and incorporate new types with complex constraints which were the generalisation of the constraints on their subtypes. But this only removes the effect that $t_1 \vee t_2$ might be more specific than $t_1 \sqcap t_2$; the non-equivalence of disjunction and generalisation still holds, as it does for untyped feature structures. This has not been implemented, because it does not seem likely that a compromise between generalisation and disjunction would give particularly perspicuous representations.

The motivation for avoiding disjunction is not just computational efficiency, but also that the use of disjunction in a description can be seen as an indication that there is something rather arbitrary about the representation. We should perhaps distinguish between two uses of disjunction. The first could be characterised as disjunction for structure sharing, where one disjunctive feature structure is specified rather than two individual structures because a considerable amount of information is equivalent in both. For example, rather than specifying different lexical entries for *worry* in the senses most obviously involved in the following sentences:

(1) The greenhouse effect worries Mary.
That dog worries sheep.

there might be a temptation, given an approach to the lexicon which minimised semantic information, to use a single lexical entry with a disjunctive specification of the predicate **worry1′** $\vee$ **worry2′**. This would be an inappropriate use of disjunction at the descriptive level, given a theory of the lexicon where different senses were regarded as distinct linguistic entities, because implementational considerations would be determining the representation. Practical problems would arise in trying to extend the lexicon to contain semantic information, for example. In LAUREL a succinct description of feature structures which have structure in common is available through the inheritance mechanisms, which would potentially allow an implementation to use structure sharing in a transparent way. In this particular case, both entries would be specified as having the type appropriate for transitive verbs, which would allow a parser to efficiently determine the possibility of packing the lexical entries (or to avoid expansion of the lexical entries until required).

Alternatively disjunction can be used to capture the behaviour of some single linguistic entity. Whether general disjunction is really required depends on the predictability of the alternation; if there is little regularity and the disjuncts are not predictable then general disjunction is needed because of the essentially static nature of the type system. However, if only certain sets of values are possible in principle, then the type system can be used to capture this, as illustrated above. Furthermore, if we are dealing with information in two distinct feature structures, and attempting to form a coherent description from that information, generalisation may sometimes be more appropriate than disjunction, because generalisation of feature structures will give results which reflect in specificity the compatability of the feature structures. Thus it seems plausible that disjunction might be avoided in the representation and potentially worthwhile to try and do so. I will thus only make use of limited atomic disjunction in the representations which follow.

2.6 Description language

I will split the discussion of the LAUREL description language into two sections. The first covers the *local* language which is quite similar to the PATR-II path descriptions. The formalisation of such a language in the typed case is discussed by Carpenter (1992). The second section describes the *non-local* description language, which allows the definition of feature structures in terms of other feature structures. This is more distinctive to LAUREL, although it has parallels in many other unification-based languages, such as the template mechanism in PATR-II. The default mechanism operates purely at the level of non-local descriptions — I will introduce default inheritance in this chapter, but the detailed discussion and definition is to be found in Chapter 3.

2.6.1 Local descriptions

The syntax of the description language which is used to define feature structures in LAUREL is based on the path notation used in PATR, with the modification that : is used to separate features. However, in a notation for a typed system, it must be possible to specify the types of non-terminal nodes in the feature structure; in LAUREL this may either be done by making the type the value of the relevant path or, for compactness, by specifying the type in the path (as an optional value, before the feature).[5] For example:

< **lex-count-noun** QUALIA : **physical** TELIC : ARG2 > = **e-sentient**

is equivalent to:

<> = **lex-count-noun**
< QUALIA > = **physical**
< QUALIA : TELIC : ARG2 > = **e-sentient**

Instead of specifying the empty path in a path value specification, the type name alone can be used:

lex-count-noun
< QUALIA > = **physical**
< QUALIA : TELIC : ARG2 > = **e-sentient**

Type specifications consist of the name of the type followed by a bracketed list indicating the immediate parents. This is followed, if appropriate, by a local description of a feature structure which forms the constraint on the type when unified with the constraints on the parent types. Figure 2.5 shows a description of a tiny type system which I will use in subsequent examples in this chapter and the next.

[5] To make it easier to read the descriptions I will keep the same typographic conventions as in the rest of the text, that is types in bold font, features in capitals and feature structure identifiers in lower case typewriter font. Case and font distinctions are not, however, significant in the language.

```
top ().

boolean (top)
   (OR true false).

string (top).

sign (top)
   < ORTH > = string.

lex-sign (sign).

lex-noun-sign (lex-sign)
   < COUNT > = boolean
   < QUALIA > = nomqualia.


rule (top)
   < 0 > = sign
   < 1 > = sign.

grammar-rule (rule)
   < 2 > = sign.

lexical-rule (rule)
   < 0 > = lex-sign
   < 1 > = lex-sign
   < 0 : ORTH > = < 1 : ORTH >.


nomqualia (top).

physical (nomqualia)
   < FORM > = form.

form (top)
   (OR mass individual plural).

animal (physical)
    < SEX > = gender.

gender (top)
   (OR male female).

plant (physical).

artifact (nomqualia)
   < TELIC > = verb-sem.

art_phys (physical artifact).


verb-sem (top)
   < IND > = eve
   < PRED > = string
   < ARG1 > = < IND >
   < ARG2 > = obj
   < ARG3 > = sem.

sem (top).

eve (sem).

obj (sem).

e-sentient (obj).
```

Figure 2.5: Illustrative type system

The following BNF description gives the syntax of the type specification language:

```
Type specification ->
          typename Parents [Comment][Constraint].
Comment -> " string "
Parents -> ( typename* )
Constraint -> Path_spec_list | Enumeration
Path_spec_list -> Path_spec*
Path_spec -> EPath = Typevalue | Path = Path
Typevalue -> typename | ( typename+ )
Epath -> Path | <>
Path -> < Type_F_pair_list >
Type_F_pair_list -> Type_Feature_pair
          | Type_Feature_pair :  Type_F_pair_list
Type_Feature_pair -> [typename] feature
Enumeration -> ( OR typename+ )
```

Disjunction is indicated by bracketed lists of atomic types as values for a path. Enumeration is syntactic sugar for enumerated atomic types. For example:

> **boolean (top) (OR true false).**

expands out into the equivalent of:

> **boolean (top).**
> **true (boolean).**
> **false (boolean).**

The descriptions can be taken as formulas which must be satisfied by the feature structures they describe, as was done in Pereira and Shieber (1984). The logic of the description language used here is identical to that described by Carpenter (1992:52f) with the exception that we do not allow disjunctive descriptions (and thus the algorithm for deciding the satisfiability of a formula has the same order of complexity as unification). The following definition is a slightly modified version of the one given by Carpenter:

Definition 16 (Descriptions) *The set of* descriptions *over the poset* $\langle \mathsf{TYPE}, \sqsubseteq \rangle$ *of types and the collection* FEAT *of features is the least set* DESC *such that:*

- $t \in \mathsf{DESC}$ *if* $t \in \mathsf{TYPE}$
- $\pi : \phi \in \mathsf{DESC}$ *if* $\pi \in \mathsf{FEAT}^*$ *and* $\phi \in \mathsf{DESC}$
- $\pi_1 \doteq \pi_2 \in \mathsf{DESC}$ *if* $\pi_1, \pi_2 \in \mathsf{FEAT}^*$
- $\phi \wedge \psi \in \mathsf{DESC}$ *if* ϕ *and* $\psi \in \mathsf{DESC}$

A description of the form $\pi : \phi$ applies to objects whose value for the path π satisfies ϕ. A description of the form $\pi \doteq \pi'$ means that the value at the end of path π is token identical to that at the end of path π'.

The LAUREL path notation for local descriptions can be mapped directly into the syntax used by Carpenter.

LAUREL	Carpenter
t	t
`<> =` **t**	t
`< f1 : f2 ... fn > =` **t**	$\pi : t$
`< f1 : f2 ... fn > = < g1 : g2 ... gn >`	$\pi \doteq \pi'$

where π is the path f1, f2 ...fn and π' is the path g1, g2 ...gn. A list of LAUREL path specifications in a description are to be taken as conjoined.

Thus the results given by Carpenter concerning the satisfiability of a description also apply to LAUREL:

Definition 17 (Satisfaction) *The* satisfaction relation *relates the collection of feature structures $\mathcal{F}$ and the set of descriptions* DESC*. It is the least relation '$\models$' such that, if F is the feature structure $\langle Q, q_0, \delta, \alpha\rangle$ and $\phi \in \mathsf{DESC}$*

- $F \models t$ *if* $t \in \mathsf{TYPE}$ *and* $\alpha(q_0) \sqsubseteq t$
- $F \models \pi{:}\,\phi$ *if* $F@\pi \models \phi$
- $F \models \pi_1 \doteq \pi_2$ *if* $\delta(q_0, \pi_1) = \delta(q_0, \pi_2)$
- $F \models \phi \wedge \psi$ *if* $F \models \phi$ *and* $F \models \psi$

where if $F = \langle Q, q_0, \delta, \alpha\rangle$*,* $F@\pi$ *is* $\delta(\pi, q_0)$*.*

Definition 18 *Consider the set of all feature structures that satisfy a certain description ϕ, that is $Sat(\phi) = \{F \in \mathcal{F} | F \models \phi\}$. If ϕ is a formula in* DESC*, ϕ is said to be* satisfiable *if there exists a feature structure F that satisfies it, that is the set $Sat(\phi)$ is not empty.*

As Carpenter shows we have the usual results for the satisfaction relation, $\models$:

Monotonicity If ϕ is a formula then if $F_1 \models \phi$ and $F_2 \sqsubseteq F_1$ then $F_2 \models \phi$.

Most general satisfier For every satisfiable formula ϕ, there is a unique most general feature structure $MGSat(\phi)$ that satisfies it.

Description For any feature structure in $\mathcal{F}$ there is a description $\mathsf{Desc}(F)$ such that

$$F \sim MGSat(\mathsf{Desc}(F))$$

2.6.2 Non-local descriptions

The flexibility of LAUREL is enhanced by allowing feature structures to be described in terms of other feature structures. Particular feature structures may have identifiers associated with them; feature structures representing complete lexical entries are identified by a combination of orthography plus sense information, while lexical and grammar rules also have associated names, and in general any feature structure may be defined with an associated identifier. The term *psort* will be used to refer to any such named structure; the significance of this is the use of psorts in the description of other feature structures.[6] The function FS returns the feature structure associated with a psort identifier. Although formally this scheme is very similar to the use of templates, at least when the non-default operations are considered, its particular significance is that it allows relationships between word senses (and rules) to be encoded.

Feature structure descriptions are thus not just local, and relationships between feature structures may be set up in a variety of ways. In general terms a path can be specified as having a value which has a particular relationship with a feature structure, which is either some specified part of a psort, or of some combination of psorts. Ordinary path/value equations can be seen as a special case of this, where the feature structure involved is the constraint on the value type. LAUREL supports the following relationships: [7]

= Local description (as above)
=* Equality (with a type constraint)
== Equality (with a psort)
<= Non default inheritance from a psort (subsumption)
< Default inheritance from a psort

[6] The term psort is arbitrary here and should not be taken as having any connection with any other use of the term. It originally stood for "pseudo-sort", but it is probably not very helpful to think of psorts as being like sorts in any strict sense.

[7] In the first version of the language only default inheritance from psorts was supported. Non-default inheritance and equality constraints were added later, in response to particular representation issues.

Operators available to combine feature structures are:

/\ Unification
\/ Generalisation
+ Lexical rule application (see section 2.6.4, below)

The complete syntactic description for lexical entries is as follows:

```
Lexentry -> LexID PPath_spec+ .
LexID -> orth sense-id
PPath_spec -> EPath < FS | EPath <= FS | EPath == FS
          | EPath =* Typevalue | Path_spec | typename
FS -> psortname Epath | CFS Epath
CFS -> ( FS \/ FS ) | ( FS /\ FS )
          | ( CFS + rulename ) | ( psortname + rulename )
Path_spec -> EPath = Typevalue | Path = Path
Typevalue -> typename | ( typename+ )
Epath -> Path | <>
Path -> < Type_F_pair_list >
Type_F_pair_list -> Type_Feature_pair
          | Type_Feature_pair :  Type_F_pair_list
Type_Feature_pair -> [typename] feature
```

`orth`, `sense-id`, `typename`, `psortname`, `rulename` and `feature` are terminal symbols.

The description language for rules and general psorts differs from that of lexical entries only in that the identifier is not split.

The system allows the global specification of path values in terms of the identifier component which apply to all lexical entries. This avoids the need to specify the values for orthography, predicate identity and so on individually for each lexical entry. In the following discussion I will use a mutually consistent set of 'toy' lexical entries to illustrate the language. These make use of the types shown in the left column of Figure 2.5 and of the global path specification:

`< ORTH > = [orth]`

where `[orth]` will be instantiated by a string type corresponding to the orthography for the particular lexical entry.

The following lexical entry uses no non-local descriptions, so the expansion involved in its evaluation to produce the psort `animal_1` is due to the type system.

animal 1
`< COUNT > =` **true**
`< QUALIA > =` **animal**
`< QUALIA : FORM > =` **individual**.

$$\texttt{animal\_1} = \begin{bmatrix} \textbf{lex-noun-sign} & \\ \text{ORTH} = \textbf{"animal"} & \\ \text{COUNT} = \textbf{true} & \\ \text{QUALIA} = & \begin{bmatrix} \textbf{animal} \\ \text{FORM} = \textbf{individual} \\ \text{SEX} = \textbf{gender} \end{bmatrix} \end{bmatrix}$$

From now on I will drop the use of double quotes to mark string types in the feature structure diagrams, since it will be clear from the context whether a type is a string or not.

The simplest of the non-local relationships conceptually is non-default inheritance, (notated `<=`), which allows a feature structure to be described as inheriting information from a psort. For example, we could define the feature structure corresponding to the lexical entry for *sheep* as inheriting its qualia structure from `animal_1`.

sheep 1
`< COUNT > =` **true**
`< QUALIA > <= animal_1 < QUALIA >.`

$$\texttt{sheep\_1} = \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{ORTH} = \textbf{sheep} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \begin{bmatrix} \textbf{animal} \\ \text{FORM} = \textbf{individual} \\ \text{SEX} = \textbf{gender} \end{bmatrix} \end{bmatrix}$$

Values may be further instantiated, thus the following entry for ewe_1 inherits its qualia structure from sheep_1, but specifies the SEX to be **female**:

ewe 1
  < QUALIA : SEX > = **female**
  < QUALIA > <= sheep_1 < QUALIA >.

$$\texttt{ewe\_1} = \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{ORTH} = \textbf{ewe} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \begin{bmatrix} \textbf{animal} \\ \text{FORM} = \textbf{individual} \\ \text{SEX} = \textbf{female} \end{bmatrix} \end{bmatrix}$$

Non-default inheritance is simply implemented by unification of the feature structure with a copy of the relevant part of the psort.

The default inheritance relationship (notated as <) allows values to be overridden, thus we could (albeit somewhat perversely) specify ram_1 as inheriting information by default from ewe_1, but override the value for SEX:

ram 1
  < QUALIA : SEX > = **male**
  < QUALIA > < ewe_1 < QUALIA >.

$$\texttt{ram\_1} = \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{ORTH} = \textbf{ram} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \begin{bmatrix} \textbf{animal} \\ \text{FORM} = \textbf{individual} \\ \text{SEX} = \textbf{male} \end{bmatrix} \end{bmatrix}$$

Non-default inheritance can be seen as producing a feature structure which is the minimal satisfier of the union of the set of local descriptions given in the attribute value language with the set of descriptions of which the psort is the minimal satisfier. Default inheritance, in contrast, produces a feature structure which is the minimal satisfier of the union of the set of local descriptions with non-conflicting descriptions from the psort. Deciding how to define the non-conflicting descriptions is a complex problem, which I leave to the next chapter.

We also allow a feature structure to be specified as being identical (modulo alphabetic variance) to a psort (notated ==) or to a constraint on a type (notated =*).[8] I will discuss these together, since they behave very similarly, although the =* descriptions are local. The main significance of these *equality constraints* is that they allow default inheritance of information to be blocked by stating that the value for some feature is known to be underspecified (the rationale behind this is discussed in the next chapter in Section 3.4). For example, both the descriptions given below expand out into the feature structure shown, with an underspecified value of < QUALIA : SEX >, despite inheriting information from a psort where this value is specified:

lamb 1
  < QUALIA : SEX > =* **gender**
  < QUALIA > < ewe_1 < QUALIA >.

lamb 2
  < QUALIA : SEX > == sheep_1 < QUALIA : SEX >.
  < QUALIA > < ewe_1 < QUALIA >.

[8] Earlier versions of the language used == for non-default inheritance.

$$\texttt{lamb\_1}, \texttt{lamb\_2} = \left[\begin{array}{l} \textbf{lex-noun-sign} \\ \text{ORTH} = \textbf{lamb} \\ \text{COUNT} = \textbf{boolean} \\ \text{QUALIA} = \left[\begin{array}{l} \textbf{animal} \\ \text{FORM} = \textbf{individual} \\ \text{SEX} = \textbf{gender} \end{array}\right] \end{array}\right]$$

The non-default inheritance relationship can be seen as a constraint that the daughter feature structure is subsumed by the psort feature structure; the equality relationship corresponds to a mutual subsumption constraint. We can formalise the notion of satisfaction of an equality description as follows:

Definition 19 (Feature structure equality constraint) *A feature structure* $F = \langle Q, q_0, \delta, \alpha \rangle$ *satisfies a feature structure equality constraint* $\pi == F'$ *iff* $F@\pi \sim F'$

Definition 20 (Type equality constraint) *A feature structure* $F = \langle Q, q_0, \delta, \alpha \rangle$ *satisfies a type equality constraint* $\pi =* t$ *iff* $F@\pi \sim C(t)$

Psort feature structures may also be combined by unification and generalisation. Suppose that the lexical entry for `plant_1` is the following:

plant 1
  < COUNT > = **true**
  < QUALIA > = **plant**
  < QUALIA : FORM > = **individual**.

$$\texttt{plant\_1} = \left[\begin{array}{l} \textbf{lex-noun-sign} \\ \text{ORTH} = \textbf{plant} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \left[\begin{array}{l} \textbf{plant} \\ \text{FORM} = \textbf{individual} \end{array}\right] \end{array}\right]$$

We can then build a lexical entry for `albino_1` which inherits information from the generalisation of `animal_1` and `plant_1`:

albino 1
  < > <= (animal_1 < > \/ plant_1 < >) < >.

$$\texttt{albino\_1} = \left[\begin{array}{l} \textbf{lex-noun-sign} \\ \text{ORTH} = \textbf{albino} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \left[\begin{array}{l} \textbf{physical} \\ \text{FORM} = \textbf{individual} \end{array}\right] \end{array}\right]$$

Finally, psorts may be transformed by lexical rule application. Lexical rules are feature structures of type **lexical-rule**, which has the constraint:

$$\left[\begin{array}{l} \textbf{lexical-rule} \\ 0 = \textbf{lex\_sign} \\ 1 = \textbf{lex\_sign} \end{array}\right]$$

Application of a particular lexical rule simply involves unification of the input of the psort with the input part of the lexical rule, indicated by the path `<1>`, and returns the instantiated output of the rule, given by the path `<0>`. Lexical rules are discussed in more detail in Section 2.6.4, below. The rule `plural` is as follows:

plural
  lexical-rule
  < 0 > = **lex-noun-sign**
  < 1 > = **lex-noun-sign**
  < 0 : COUNT > = **true**
  < 1 : COUNT > = **true**
  < 0 : QUALIA : FORM > = **plural**
  < 1 : QUALIA : FORM > = **individual**.

$$\texttt{plural} = \begin{bmatrix} \textbf{lexical-rule} & \\ 0 = & \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{ORTH} = \boxed{0}\ \textbf{string} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \begin{bmatrix} \textbf{physical} \\ \text{FORM} = \textbf{plural} \end{bmatrix} \end{bmatrix} \\ 1 = & \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{ORTH} = \boxed{0} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \begin{bmatrix} \textbf{physical} \\ \text{FORM} = \textbf{individual} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Applying the rule to the psort `sheep_1` we get the following feature structure as output:

$$\begin{bmatrix} \textbf{lex-noun-sign} \\ \text{ORTH} = \textbf{sheep} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \begin{bmatrix} \textbf{physical} \\ \text{FORM} = \textbf{plural} \end{bmatrix} \end{bmatrix}$$

Thus the lexical entry below is equivalent to specifying that the qualia structure is inherited from this feature structure:

flock 1
`<  > < ( sheep_1 + plural ) < >.`

$$\texttt{flock\_1} = \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{ORTH} = \textbf{flock} \\ \text{COUNT} = \textbf{true} \\ \text{QUALIA} = \begin{bmatrix} \textbf{physical} \\ \text{FORM} = \textbf{plural} \end{bmatrix} \end{bmatrix}$$

2.6.3 Evaluation of descriptions

The process of evaluation can be described as having two stages:

1. Construction of three components from the descriptions: a non-default feature structure, a default feature structure and a series of equality constraints.

2. Combination of these components to form the final entry.

A number of constraints on non-local descriptions are needed, in order that evaluation can be carried out. The first is that there are no cycles — that is that the induced hierarchy of psorts is a partial order. Because of the non-cyclicity condition, expansion of psorts will always terminate, at psorts which are defined using local descriptions alone. Thus the right hand side of the descriptions can be evaluated by expanding the feature structures for the psorts, and combining the specified structures by unification and generalisation.

So we have the following possibilities for local and non-local descriptions where F is the expanded right hand side feature structure.

- t
- $\pi = t$ (i.e. path value specification)
- $\pi_1 = \pi_2$ (i.e. path equivalence specification)
- $\pi =* t$
- $\pi <= F$
- $\pi == F$
- $\pi < F$

Since the first three cases were covered in the previous section, I will define the evaluation in terms of a feature structure F which corresponds to the most general satisfier of the set of these local descriptions, and a set of non-local descriptions, Φ.

Definition 21 (Non default feature structure evaluation) *The non-default feature structure is $AddN(F, \Phi)$ where:*
$AddN(F, \pi <= F') = F[\pi \sqcap F']$
$AddN(F, \pi == F') = F[\pi \sqcap F']$
$AddN(F, \pi = * t) = F[\pi \sqcap C(t)]$
$AddN(F, \pi < F') = F[\pi \sqcap C(TypeOf(F'))]$
where the notation $F[\pi \sqcap F']$ is used to indicate the result of unifying F's π value with F'.

Definition 22 (Default feature structure evaluation) *The default feature structure given by a description Φ is $AddD(\top, \Phi)$ where:*
$AddD(F, \pi < F') = F[\pi \sqcap F']$
$= F$ *for all other types of description.*

The incorporation of information into the non-default feature structure from the default description arises from the constraint that defaults must be consistent with the type hierarchy.

Definition 23 (Consistency of defaults with type hierarchy) *If the description of a psort, p, contains a default description, $\pi < F$, then $TypeOf(FS(p)@\pi) \sqsubseteq TypeOf(F)$*

The default descriptions are further constrained to be mutually consistent, which is why they can be unified together to form a single feature structure:

Definition 24 (Mutual consistency of defaults) *If the description of a psort, p, contains the default descriptions, $\pi < F$ and $\pi' < F'$ then these are said to be mutually consistent if $\top[\pi \sqcap F] \sqcap \top[\pi' \sqcap F'] \neq \bot$*

Thus we define the final feature structure in terms of a well-formed non-defeasible feature structure, a series of equality constraints and a single defeasible feature structure.

Definition 25 (Evaluation of descriptions) *The feature structure F associated with the identifier p where there is a description Φ is given by $FS(p) = \langle WF(F'), C\rangle \overset{<}{\sqcap}_e F''$, where F' is the non-defeasible feature structure given by $AddN(\top, \Phi)$, C is the set of equality constraints and F'' is the default feature structure, given by $AddD(\top, \Phi)$.*

The definition of the operator $\overset{<}{\sqcap}_e$ (default unification in the presence of equality constraints) is given in the next chapter. Intuitively it corresponds to incorporating the maximal information from the default feature structure which is consistent with the constraints imposed both by the non-defeasible information and the equality constraints. The final feature structure must be consistent with the equality constraints.

Various applications of these descriptions will be found in subsequent chapters. Although introducing psorts as well as types may seem unnecessarily complex, given that the type mechanism allows inheritance, there are compelling reasons for doing so. The first point is that a notion of defaults is essential to allow lexical generalisations to be captured, but typing and defaults are incompatible notions. Types are used to enforce an organisation on the lexicon; if the typing scheme can be overridden then none of the advantages of a typed system over a straight inheritance system hold. Error checking and classification both require that information associated with types is non-defeasible (as does constraint resolution). Non-default inheritance from psorts also turns out to be desirable because of its flexibility. The type system is taken to be complete, and various conditions are imposed on it, such as the greatest lower bound condition, which ensure that deterministic classification is possible. However this condition is quite difficult to comply with, and where no introduction of features is involved it is much easier to define inheritance in terms of psorts. In implementational terms a type scheme should be relatively static; any alterations may affect a large amount of data and checking that the scheme as a whole is still consistent is a non-trivial process. In contrast, adding psorts need not involve significant recomputation. The design of the description language was essentially motivated by the desire to use it to capture semantic relationships between lexical items, in particular the hyponymy relationship, which allows a fine-grain of inheritance that it is inappropriate to describe in the type system.

2.6.4 Rules

The rule mechanism involves no further extensions to LAUREL, but indicates how typing and inheritance may be applied to feature structures other than lexical entries. Grammar and lexical rules are typed feature structures, which represent relationships between two or more signs.

The expanded constraint for the type **rule** is:

$$\left[\begin{array}{lll}\textbf{rule} & & \\ 0 & = & \textbf{sign} \\ 1 & = & \textbf{sign}\end{array}\right]$$

All rules are feature structures of **rule** or some subtype of **rule**. Thus all rules have to have the features 0 and 1 which must both have values which are of type **sign**. Further features may be introduced as normal on subtypes of **rule**, but in the grammar treatment considered here all rules will be unary or binary:

$$\left[\begin{array}{lll}\textbf{grammar-rule} & & \\ 0 & = & \textbf{sign} \\ 1 & = & \textbf{sign} \\ 2 & = & \textbf{sign}\end{array}\right]$$

Rules can be regarded as a means of generating new signs; if a series of signs, $F_1, F_2 \ldots F_n$ can be unified with the feature structures at the end of the paths `<1>`, `<2>` ...`<n>` in the rule, then the feature structure at the end of the path `<0>` is a new sign. Linear order of constituents can be specified independently, but here I will always assume that linear order matches numerical order. A parser can operate by applying rules, either bottom-up according to the description above, or top-down, where an input sign can generate a series of output signs. Alternatively rules can be regarded statically, as expressing the relationship between existing signs.

As mentioned in section 2.6.2, lexical rules can be defined as feature structures of type **lexical-rule** which is a subtype of **rule**:

$$\left[\begin{array}{lll}\textbf{lexical-rule} & & \\ 0 & = & \textbf{lex\_sign} \\ 1 & = & \textbf{lex\_sign}\end{array}\right]$$

The notation for lexical rule application in the description language given above is not essential but simplifies the description which would otherwise involve an intermediate psort. For example:

```
( feather_1 + plural )
```

would be equivalent to

```
( plfeather < 0 >)
```

where `plfeather` was set up as follows:

```
plfeather
   < > <= plural < >
   < 1 > <= feather_1 < >.
```

This treatment is general enough to encode a variety of approaches to the notion of rule. Theoretical constraints on rules are assumed to be encoded in the type system. The particular instantiation of LAUREL which I will describe in Chapter 4, makes use of a UCG style categorial framework which combines three grammar rules (forward application, backward application and wrapping) with a series of unary and lexical rules for type shifting operations, but other treatments could be encoded. The parser makes no distinction between lexical and non-lexical rules, although lexical rules, as defined above, will not unify with any phrasal signs if these have a type which is incompatible with **lex_sign**.

The use of a rule-based approach may seem somewhat incongruous in a typed feature structure system, since most recent work has assumed the use of constraint-resolution. The basic LAUREL language could be used in a constraint based system, along the same lines as those described by Carpenter (1992) and by Zajac (in press) for example (although it might not be practical to do this without adding disjunction). However I chose not to do this for a variety of reasons. In practical terms it is much simpler to implement a rule-based parser than a general constraint solving mechanism and, currently, explicitly rule-based systems can be made reasonably efficient much more easily. Especially with an approach to grammar where very few distinct rules are encoded, it is relatively straightforward to map a constraint based approach into a rule-based one, given that rules are themselves typed feature structures, since principles such as the HPSG Head Feature Principle can be defined over rules, rather than phrasal signs. Pollard and Sag (1987) make use of an external concept of lexical rule rather than using the general constraint mechanism, but Krieger and Nerbonne (in press) demonstrate that this is not an essential assumption for morphology within a typed feature structure framework. However, in Chapter 4, I will make extensive use of type-shifting unary rules to model both non-lexical and lexical processes, including 'zero-derivation', and it is not clear how these should be modelled within a purely constraint based approach.

Chapter 3

Default inheritance and default unification

The formalisation of defaults in LAUREL is based on the assumption that the use of defaults is confined to the paradigmatic level of description. Lexical entries and rules can be described in a language which makes use of defaults, but no notion of defeasibility is carried to the syntagmatic plane where individual instances of signs are combined. The motivation for the use of default inheritance in the lexicon has been discussed by Flickinger (1987) and Gazdar (1987, 1990). Essentially it seems as though it is necessary to have a notation which admits defaults in some way, in order to capture regularity of structure in the lexicon, since there are exceptions to many of the generalisations which one would like to state. The most straightforward examples are morphological, for example past participle formation by suffixation by *ed* can be described as a regular process which is blocked for verbs such as *hold* by the presence of the irregular form *held*, but syntactic and semantic exceptions can also be treated by defaults. For some recent examples of the use of defaults within unification based representations see Russell *et al.* (1991), Krieger and Nerbonne (1991), Cahill and Evans (1990).

However, the need for defaults is also acute in lexical semantics, where many generalisations have exceptions. Consider, for example, the connection between number agreement and semantic plurality of nominals in English, which will be discussed more extensively in Chapter 5. There are lexical exceptions to the generalisation that these correlate. One class of exceptions are pair nouns such as *scissors*, *trousers*, *binoculars*[1]. There are also individual examples of nouns which take either singular or plural agreement, although semantically they behave as singulars, such as *gallows*. Without going into the details of an analysis here, it seems intuitively that to account for these patterns a treatment of defaults will be needed which allows for exceptions both at the level of classes and of individual items.

I shall also make extensive use of default inheritance between lexical entries (rather than specially defined classes), in particular lexical entries which stand in a hyponymy relationship. As will be seen, this significantly influences the design of the default language. For example, *book* has hyponyms such as *autobiography*, *novel* and *dictionary*. I assume that the lexicon is structured in a way which reflects the hyponymy relationship, so that the lexical entries for *autobiography*, *novel*, *dictionary* will inherit lexical semantic information, encoded in the qualia structure, from *book*. But such inheritance must be defeasible. Although *book*, *autobiography* and *novel* all have their telic roles instantiated to *read*, this is inappropriate for *dictionary* which should have a telic role corresponding to *refer_to*. Furthermore, hyponyms of *dictionary*, such as *lexicon* (in the more generally used sense), will inherit the modified telic role of *refer_to* rather than read. Thus, as is usual, the inheritance of lexical information should be taken as obeying the 'Penguin Principle'

[1] Although there have been proposals that these be treated as semantically plural, I will argue in Chapter 5 that this is not reasonable.

familiar from inheritance networks in AI.[2] However, in this case, the nodes on the network do not correspond to classes which are set up by the linguist in order that the appropriate information is inherited, but instead correspond to full lexical entries which stand in a particular semantic relationship. Furthermore, as I will show in Chapters 6 and 7, these hyponymy relationships may be acquired semi-automatically from MRDs. This essentially precludes the use of complex hierarchical inheritance schemes which allow the linguist to determine the order in which defaults are overridden in case of conflicts, because the organisation of the hierarchy is not determined by the linguist.

So for example, assume that the following is part of the lexical entry for book_L_1_1 (i.e. book_L_1_1 is the name of the psort, where the code L_1_1 refers to the sense number in LDOCE).

$$\texttt{book\_L\_1\_1} = \begin{bmatrix} \textbf{lex-noun-sign} & \\ \text{QUALIA} = & \begin{bmatrix} \textbf{art\_phys} & \\ \text{TELIC} = & \begin{bmatrix} \textbf{verb-sem} \\ \text{PRED} = \textbf{read\_L\_1\_1} \end{bmatrix} \\ \text{FORM} = \textbf{individual} & \end{bmatrix} \end{bmatrix}$$

I have assumed the type system shown in the previous chapter, but I have omitted some features such as IND and COUNT from the diagram. The following path specifications make the lexical entries defined inherit their qualia structure from book_L_1_1:

```
autobiography L_0_0
   < QUALIA > < book_L_1_1 < QUALIA >.

dictionary L_0_1
   < QUALIA > < book_L_1_1 < QUALIA >
   < QUALIA : TELIC : PRED > = "refer_to_L_0_2" .

lexicon L_0_0
   < QUALIA > < dictionary_L_0_1 < QUALIA > .
```

autobiography_L_0_1 thus has the same values as book_L_1_1 for both TELIC and FORM. dictionary_L_0_1 will inherit the value **individual** for the feature FORM but the value of the TELIC role, which is specified as **refer_to_L_0_2**, overrides that which would be inherited from book_L_1_1 (**read_L_1_1**). lexicon_L_0_0 inherits its value for the telic role from dictionary_L_0_1 rather than from book_L_1_1.

$$\texttt{lexicon\_L\_0\_0} = \begin{bmatrix} \textbf{lex-noun-sign} & \\ \text{QUALIA} = & \begin{bmatrix} \textbf{art\_phys} & \\ \text{TELIC} = & \begin{bmatrix} \textbf{verb-sem} \\ \text{PRED} = \textbf{refer\_to\_L\_0\_2} \end{bmatrix} \\ \text{FORM} = \textbf{individual} & \end{bmatrix} \end{bmatrix}$$

I will return to the representation of hyponymy and discuss this example in more detail in Chapter 6. The main purpose of this chapter is to consider the formal definition of default inheritance. Default inheritance in LAUREL involves default unification and a large part of this chapter will involve a discussion of various definitions of default unification. It will become apparent that default inheritance in feature structure languages does relate to more general notions of default inheritance in AI. This is of particular relevance for the discussion of defaults in lexical semantics, since, as argued in Chapter 1, to the extent that lexical semantics does involve real world knowledge, the properties of inheritance in the representation should be compatible with the properties of inheritance in descriptions of commonsense knowledge.

Default inheritance in LAUREL is defined in terms of the non-local description language, discussed in the previous chapter. As I mentioned there non-default inheritance is implemented

[2] The Penguin Principle states that when there is a conflict between defaults, the default which applies to a more specific class should apply in preference to one which applies to a more general class. The eponymous example is:
Birds can normally fly.
All penguins are birds.
Penguins cannot normally fly.
Tweety is a penguin.
Therefore conclude (non-monotonically) that Tweety cannot fly.

by unification of the feature structure with a copy of the relevant part of the psort. The default inheritance mechanism is similar except that default unification, rather than ordinary unification is involved. I will use $A \overset{<}{\sqcap} B$ to indicate default unification where A is non-default. The effect of default unification is that incompatible values for attributes are ignored, rather than causing unification failure. For example:

$$F = \left[\begin{array}{l} \textbf{lex-noun-sign} \\ \mathrm{ORTH} = \textbf{string} \\ \mathrm{COUNT} = \textbf{boolean} \\ \mathrm{QUALIA} = \left[\begin{array}{l} \textbf{animal} \\ \mathrm{FORM} = \textbf{individual} \\ \mathrm{SEX} = \textbf{male} \end{array}\right] \end{array}\right] \quad F' = \left[\begin{array}{l} \textbf{lex-noun-sign} \\ \mathrm{ORTH} = \textbf{ewe} \\ \mathrm{COUNT} = \textbf{true} \\ \mathrm{QUALIA} = \left[\begin{array}{l} \textbf{animal} \\ \mathrm{FORM} = \textbf{form} \\ \mathrm{SEX} = \textbf{female} \end{array}\right] \end{array}\right]$$

$$F \sqcap F' = \bot$$

$$F \overset{<}{\sqcap} F' = \left[\begin{array}{l} \textbf{lex-noun-sign} \\ \mathrm{ORTH} = \textbf{ewe} \\ \mathrm{COUNT} = \textbf{true} \\ \mathrm{QUALIA} = \left[\begin{array}{l} \textbf{animal} \\ \mathrm{FORM} = \textbf{individual} \\ \mathrm{SEX} = \textbf{male} \end{array}\right] \end{array}\right]$$

Since psorts may themselves inherit information, default inheritance in effect operates over a hierarchy of psorts and the ordering on the psort hierarchy gives us an ordering on defaults (since cycles in the inheritance ordering are prohibited). However once information is incorporated into a feature structure no distinction between inherited and non-inherited information is maintained and a psort will be fully expanded before it is used: inheritance thus operates top-down.

Our variant of default unification is restricted by the type system so that a feature structure will not inherit any information from a feature structure of incompatible type. However, I will begin by discussing default unification of untyped feature structures.

3.1 Default unification of untyped feature structures

There are now a considerable number of descriptions of non-monotonic operations on feature structures. Shieber (1986) described overwriting in PATR-II: this allows particular atomic values of feature structures to be non-monotonically altered. Kaplan's (1987) outline of priority union is more similar to default unification. It does apply to non-atomic feature structures but does not allow for cases where the feature structures are reentrant, which can give rise to conflicts between default information. Recent definitions which do consider general feature structures have been put forward by Bouma (1990a,b), Calder (1991), Carpenter (1991, in press), Russell *et al.* (1991) and van den Berg and Prüst (1991). The issues are discussed in detail in Carpenter (1991); here I wish to review his proposal and to extend the discussion slightly in order to illustrate the varieties of default unification which have been proposed.

Default unification is defined so that when a non-default feature structure is unified with a default feature structure only values in the default structure which do not conflict with values in the non-default structure are incorporated. It should ideally have the following properties:

1. $A \overset{<}{\sqcap} B \sqsubseteq A$
   Default unification adds information monotonically to the non-default.
   Clearly it should not be possible to remove non-default information, and all definitions of default unification of which I am aware do meet this criterion.

2. if $A \sqcap B \neq \bot$ then $A \sqcap B = A \overset{<}{\sqcap} B$
   Default unification behaves like ordinary unification in the cases where ordinary unification would succeed.
   Intuitively ordinary unification should correspond to the case where the default feature structure is totally compatible with the non-default structure.

3. $A \overset{<}{\sqcap} B \neq \bot$
   Default unification never fails.

This is a property which could be taken as definitional for default unification. However, as shown later, some versions of default unification drop this condition (or put preconditions on default unification such that it cannot be applied to some pairs of feature structures) in order to meet the other requirements.

4. Default unification returns a single result, deterministically.
   It seems desirable not to introduce non-determinism into the system. Multiple results are awkward both from the point of view of implementation and usability.

It is, in general, necessary that default unification be implementable with reasonable efficiency and it is highly desirable that it give results which are intuitively plausible to users.

The examples of default unification given earlier are unproblematic. However there are cases where there are conflicts between parts of the default information, because of reentrancy in the default or in the non-default feature structure, as in the following examples:

(1) a $\begin{bmatrix} \mathrm{F} & = & \boxed{1} \\ \mathrm{G} & = & \boxed{1} \\ \mathrm{H} & = & \top \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} \mathrm{F} & = & a \\ \mathrm{G} & = & b \\ \mathrm{H} & = & c \end{bmatrix}$

b $\begin{bmatrix} \mathrm{F} & = & a \\ \mathrm{G} & = & \top \\ \mathrm{H} & = & \top \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} \mathrm{F} & = & \boxed{1} & b \\ \mathrm{G} & = & \boxed{1} & \\ \mathrm{H} & = & c & \end{bmatrix}$

As shown in section 3.1.2 the various definitions of default unification have been proposed are not in agreement about the result of default unification for such examples. The difficulty in defining default unification is to exclude the possibility of the result depending on the order in which individual parts of the default feature structure are unified with the non-default feature structure. All the definitions which I will discuss here exclude such order dependence, but they do so in different ways.

3.1.1 Definitions of default unification

In many ways I regard Carpenter's (1991) definition of skeptical default unification as the ideal; it meets all the conditions enumerated above and has a definition which can be simply paraphrased as "incorporate the maximal consistent information from the default". I repeat Carpenter's definitions here (as usual with order reversed)[3].

Definition 26 (Credulous Default Unification) *The result of credulously adding the default information in G to the strict information in F is given by:*

$$F \overset{<}{\sqcap}_c G = \{F \sqcap G' \mid G \sqsubseteq G' \textit{ and } G' \textit{ is maximally specific such that } F \sqcap G' \textit{ is defined}\}$$

Credulous default unification gives multiple results, for example:

(2) a $\begin{bmatrix} \mathrm{F} & = & \boxed{1} \\ \mathrm{G} & = & \boxed{1} \\ \mathrm{H} & = & \top \end{bmatrix} \overset{<}{\sqcap}_c \begin{bmatrix} \mathrm{F} & = & a \\ \mathrm{G} & = & b \\ \mathrm{H} & = & c \end{bmatrix} = \left\{ \begin{bmatrix} \mathrm{F} & = & \boxed{1} & a \\ \mathrm{G} & = & \boxed{1} & \\ \mathrm{H} & = & c & \end{bmatrix}, \begin{bmatrix} \mathrm{F} & = & \boxed{1} & b \\ \mathrm{G} & = & \boxed{1} & \\ \mathrm{H} & = & c & \end{bmatrix} \right\}$

b $\begin{bmatrix} \mathrm{F} & = & a \\ \mathrm{G} & = & \top \\ \mathrm{H} & = & \top \end{bmatrix} \overset{<}{\sqcap}_c \begin{bmatrix} \mathrm{F} & = & \boxed{1} & b \\ \mathrm{G} & = & \boxed{1} & \\ \mathrm{H} & = & c & \end{bmatrix} = \left\{ \begin{bmatrix} \mathrm{F} & = & \boxed{1} & a \\ \mathrm{G} & = & \boxed{1} & \\ \mathrm{H} & = & c & \end{bmatrix}, \begin{bmatrix} \mathrm{F} & = & a \\ \mathrm{G} & = & b \\ \mathrm{H} & = & c \end{bmatrix} \right\}$

Skeptical default unification results in only the information which is common to all the credulous results being retained.

Definition 27 (Skeptical Default Unification) *The result of skeptical default unification is the generalisation of credulous default unification:* $F \overset{<}{\sqcap}_s G = \bigsqcup(F \overset{<}{\sqcap}_c G)$

[3] Calder (1991) independently defined priority union in a way which is equivalent to Carpenter's definition of credulous default unification. A notion of maximal incorporation of information which is very similar to Carpenter's has also been proposed by van den Berg and Prüst (1991). However in their paper the specific application to feature structures uses a similar process of normalisation to that of Bouma (see below), which results in reentrant structures with specified values being treated as equivalent to the non-reentrant case, where the paths terminate in distinct identical values.

(3) a $\begin{bmatrix} \text{F} = \boxed{1} \\ \text{G} = \boxed{1} \\ \text{H} = \top \end{bmatrix} \overset{<}{\sqcap}_s \begin{bmatrix} \text{F} = a \\ \text{G} = b \\ \text{H} = c \end{bmatrix} = \begin{bmatrix} \text{F} = \boxed{1} \\ \text{G} = \boxed{1} \\ \text{H} = c \end{bmatrix}$

b $\begin{bmatrix} \text{F} = a \\ \text{G} = \top \\ \text{H} = \top \end{bmatrix} \overset{<}{\sqcap}_s \begin{bmatrix} \text{F} = \boxed{1} \quad b \\ \text{G} = \boxed{1} \\ \text{H} = c \end{bmatrix} = \begin{bmatrix} \text{F} = a \\ \text{G} = \top \\ \text{H} = c \end{bmatrix}$

In order to compare Carpenter's definition with some of the other varieties of default unification which have been proposed, I will formalise them all in terms of successively unifying pieces of information carried by the default into the non-default feature structure while taking account of possible conflicts (cf. Russell *et al.*, 1991). A critical notion here is that of "pieces of information"; I will make use of a general notion of decomposition of a feature structure $Decomp(F)$ into component pieces of information. This must meet the following criterion (if default unification is to have the property of being equivalent to ordinary unification in the cases where that would succeed):

$\sqcap(Decomp(F)) = F$
F is equal to the unification of all the information in its decomposition.

The first case to consider is where decomposition is into the minimal atomic units of information. Carpenter's definition of atomic feature structures is as follows:

Definition 28 (Atomic Feature Structure) *A feature structure is* atomic *if it is of one of the following two forms:*

- (Path Value)
  the feature structure contains a single path assigned to an atomic value.
- (Path Sharing)
  the feature structure contains only a pair of (possibly identical) paths which are shared.

The function At when applied to a feature structure F gives the set of atomic units:

$At(F) = \{F \sqsubseteq F' \mid F' \text{ atomic}\}$

Carpenter shows that At meets the criterion for the decomposition function
(i.e. $\sqcap(At(F)) = F$).

A definition of default unification in terms of incorporation of atomic feature structures can thus be given which is equivalent to Carpenter's skeptical default unification:

Definition 29 (Skeptical Default Unification)
$F_1 \overset{<}{\sqcap} F_2 = F_1 \sqcap \bigsqcap\{F \in At(F_2) \mid F_1 \sqcap F \neq\bot$ *and there is no* F' *such that* $F_2 \sqsubseteq F'$ *and* $F' \sqcap F_1 \neq\bot$ *and* $F' \sqcap F_1 \sqcap F =\bot\}$

The intuitive basis for this definition is to consider successively adding the minimal (atomic) units of information from the default into the non-default. In the cases where there are conflicts, such as the examples just given, this would give different results depending on the order in which the atomic feature structures were added in. To produce the equivalent of credulous default unification we would do the addition once for each possible ordering of default atomic feature structures (and remove duplicates). The definition above is equivalent to skeptical default unification because only information which is consistent with all possible orderings is added. It is thus obvious that the complexity of the algorithm as described is exponential since checking for all possible F' would involve creating the unification of each member of the power-set of $At(F_2)$. An illustration of the type of feature structure which exhibits this worst case behaviour is shown in example 5, below.

The option taken by Russell *et al.* (1991) is to keep the tractable (near-linear) behaviour of ordinary unification by defining default unification in such a way that it fails under the circumstances where there are conflicts in the default information. In terms of the definition above $At(F)$ is split into $PE(F)$, the set of path equivalence specifications, and $PV(F)$ the set of path value specifications. If the reentrant part of the default unifies with the non-default, and the reentrant part of the non-default unifies with the default, no conflicts can arise in the default information. Thus Russell *et al.* have:

Definition 30 (Russell et al. default unification)

$$\begin{aligned} F_1 \overset{<}{\sqcap} F_2 &= \bot \ \textit{if} \ \sqcap\mathrm{PE}(F_2) \sqcap F_1 = \bot \\ &\qquad \textit{or} \ \sqcap\mathrm{PE}(F_1) \sqcap F_2 = \bot \\ &= F_1 \sqcap \bigsqcap\{F \in \mathit{At}(F_2) \mid F_1 \sqcap F \neq\bot\} \ \textit{otherwise} \end{aligned}$$

The ACQUILEX LKB's original default unification algorithm also made use of a distinction between reentrant and non-reentrant atomic feature structures. The definition used was:

Definition 31 (ACQUILEX LKB original)
$F_1 \overset{<}{\sqcap} F_2 = F_1 \sqcap \bigsqcap\{F \in \mathit{At}(F_2) \mid F_1 \sqcap F \neq\bot$ and for all conflicting F' such that $F_2 \sqsubseteq F'$ and $F' \sqcap F_1 \neq\bot$ and $F' \sqcap F_1 \sqcap F =\bot$, F "takes precedence over" $F'\}$
Where F takes precedence over F' iff F is a specification of path equivalence ($F \in \mathrm{PE}(F_1)$) and F' contains at least one path value specification.

Thus a precedence order was introduced between path equivalence and path value specifications. This was actually done because the linguists involved in designing the LRL expressed a preference for a behaviour where reentrancy took precedence over values; for the application to automatic acquisition of information from MRDs in particular this was desirable because reentrancy is usually set up manually, see e.g. Sanfilippo (in press), whereas values are more likely to be acquired automatically. However it was found in practice that reentrancy tended to be set up in the type system, and was thus, in effect, non-default. This is considered further below.

This approach to default unification can, in practice, be implemented considerably more efficiently than Carpenter's, although its worst case behaviour is still exponential. Initially the re-entrant parts of the default feature structure can be unified individually with the non-default, and it is only necessary to consider conflicts that arise in the reentrant set. Thus the exponential term involves only the path equivalence specifications, and since typically there are many fewer path equivalence specifications than path value specifications the implementation is not unreasonably slow. Furthermore this is the worst case behaviour; it is usually possible to split $\mathrm{PE}(F)$ into sets which are guaranteed not to interact. The procedure then reduces to one of default unifying a tree-structured feature structure with a non-default reentrant structure. There are still possible conflicts, of the type in example 3 below (which would cause unification failure by Russell *et al.*'s definition). However it is possible to allow for these with a near-linear algorithm by storing the original non-default value in the feature structure representation at reentrant points as unification proceeds, and reverting to it if a conflict arises. In effect, what this relies on is that if the non-default feature structure is the only reentrant one, all conflicts are localised.

However this definition still seems unsatisfactory, even though it meets all the criteria we enumerated at the beginning of this section. The worst case complexity is exponential, the implementation is awkward, and the behaviour can be obscure. A better compromise for the ACQUILEX LKB seemed to be to specify that inheritance of information about reentrancy is non-defeasible, and that default unification fails in the case where the non-default feature structure and the reentrant part of the default feature structure do not unify.

Definition 32 (ACQUILEX LKB)
$F_1 \overset{<}{\sqcap} F_2 = F_3 \sqcap \bigsqcap\{F \mid F \in PV(F_2)$ and $F_1 \sqcap F \neq\bot$ and there is no $F' \in PV(F_2)$ such that $F' \sqcap F_3 \neq\bot$ and $F' \sqcap F_3 \sqcap F =\bot\}$
where $F_3 = F_1 \sqcap \bigsqcap PE(F_2)$

Such a definition, where default unification involves filling in values, and expanding, rather than modifying, existing feature structure skeletons, is relatively simple to understand.[4] It does, however, violate the third criterion that I gave earlier, that default unification never fail. In practice this definition will be adequate if reentrancy is not seen as defeasible, which turned out to be the case for the ACQUILEX LKB system, where the path equivalence specifications were always set up in the type system.

[4] It also avoids the rather complex behaviour of Carpenter's definition with respect to the difference between specified and unspecified paths, see Carpenter (1991). However I will not discuss that further here, since default unification of totally typed features structures, as used in LAUREL, avoids the problem in any case.

Bouma's (1990a,b) treatment of default unification can also be described in terms of addition of pieces of information from the default structure to the non-default structure. However I will not attempt to reformalise it in these terms here, since the mechanism by which conflicts are excluded is complex and Bouma's definition is lengthy. Instead I will give examples of the behaviour of Bouma's approach below, which illustrate that it does not conform to all of the original criteria.

3.1.2 Examples of default unification

The following examples illustrate the differences in behaviour between the definitions that have been proposed, by Bouma (1990a,b), Carpenter (1991), Calder (1991), Russell *et al.* (1991). I have limited these examples to the cases where the default and non-default structures have the same features, since this discussion is essentially a precursor to the description of default unification in a totally typed system.

(1)
$$\begin{bmatrix} \text{F} = a \\ \text{G} = \top \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} \text{F} = b \\ \text{G} = c \end{bmatrix} = \begin{bmatrix} \text{F} = a \\ \text{G} = c \end{bmatrix}$$

This is the simplest case of default unification; the conflicting information in the default is ignored, but the non-conflicting information is incorporated.

(2)
$$\begin{bmatrix} \text{F} = \boxed{1} \\ \text{G} = \boxed{1} \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} \text{F} = a \\ \text{G} = \top \end{bmatrix} = \begin{bmatrix} \text{F} = \boxed{1} \\ \text{G} = \boxed{1} \end{bmatrix}$$
Bouma
$$= \begin{bmatrix} \text{F} = \boxed{1}\ a \\ \text{G} = \boxed{1} \end{bmatrix}$$
other definitions

I include this example to illustrate that Bouma's definition of default unification does not meet the second criterion, since it gives a different result from ordinary unification. This arises because in Bouma's approach any pieces of information in the default which potentially conflict with each other are excluded, and Bouma's algorithm treats the values of F and G as being in potential conflict. In practice this behaviour is presumably not apparent since Bouma proposes that ordinary unification is attempted before default unification.

(3)
$$\begin{bmatrix} \text{F} = \boxed{1} \\ \text{G} = \boxed{1} \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} \text{F} = a \\ \text{G} = b \end{bmatrix} = \left(\begin{bmatrix} \text{F} = \boxed{1}\ a \\ \text{G} = \boxed{1} \end{bmatrix}, \begin{bmatrix} \text{F} = \boxed{1}\ b \\ \text{G} = \boxed{1} \end{bmatrix} \right)$$
Calder
Carpenter
(credulous)

$$= \begin{bmatrix} \text{F} = \boxed{1} \\ \text{G} = \boxed{1} \end{bmatrix}$$
Bouma
Carpenter
(skeptical)
ALKB

$$= \bot$$
Russell *et al.*

Here the presence of reentrancy in the non-default means that the two default values are in conflict. A credulous definition will return multiple values; skeptical definitions return only that information which is common to all the credulous results.

(4)
$$\begin{bmatrix} \text{F} = a \\ \text{G} = \top \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} \text{F} = \boxed{1}\ b \\ \text{G} = \boxed{1} \end{bmatrix} = \left(\begin{bmatrix} \text{F} = \boxed{1}\ a \\ \text{G} = \boxed{1} \end{bmatrix}, \begin{bmatrix} \text{F} = a \\ \text{G} = b \end{bmatrix} \right)$$
Calder
Carpenter
(credulous)

$$= \begin{bmatrix} \text{F} = a \\ \text{G} = \top \end{bmatrix}$$
Carpenter
(skeptical)

$$= \begin{bmatrix} \text{F} = \boxed{1}\ a \\ \text{G} = \boxed{1} \end{bmatrix}$$
ALKB

$$= \begin{bmatrix} \text{F} = a \\ \text{G} = b \end{bmatrix}$$
Bouma

$$= \bot$$
Russell *et al.*

In example 3 there is no basis for deciding which of the conflicting information in the default structure should be incorporated. However in example 4 I would claim that there is a basis for distinguishing between the two pieces of default information which could be potentially incorporated but which are in conflict with each other, since one involves a path equivalence specification and the other a path value specification;

```
< F > = < G >
< G > = b
```

As described above, in the ACQUILEX LKB a decision was made to prefer specifications of equivalence to specifications of values. Bouma's result arises because he normalises default structures, in a way which gives:

```
< F > = b
< G > = b
```

for the default structure in example 4, whereas Carpenter's decomposition function *At* would give:

```
< F > = b
< G > = b
< F > = < G >
```

Bouma's normalisation function thus does not meet my criterion for the decomposition function, since the result of reunifying the pieces would be:

$$\begin{bmatrix} \mathrm{F} = b \\ \mathrm{G} = b \end{bmatrix}$$

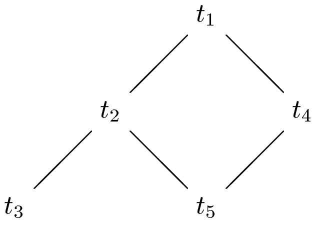
Bouma justifies this on the basis that this structure has equivalent behaviour with respect to unification to the original default structure. This is a reasonable position to take if feature structure values are defined to be extensional (see section 2.4.5), but it does not seem natural for an intensional treatment, such as adopted in LAUREL, nor does it extend naturally to the default unification of typed feature structures, where atomic values are not necessarily maximally specific.

$$
\begin{array}{llll}
 & \begin{bmatrix} \mathrm{F} = \top \\ \mathrm{G} = a \\ \mathrm{H} = b \\ \mathrm{J} = \top \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} \mathrm{F} = \boxed{1} \\ \mathrm{G} = \boxed{1} \\ \mathrm{H} = \boxed{1} \\ \mathrm{J} = \boxed{1} \end{bmatrix} & = \text{(4 possibilities)} & \text{Calder} \\ & & & \text{Carpenter} \\ & & & \text{(credulous)} \\
(5) & & = \begin{bmatrix} \mathrm{F} = \top \\ \mathrm{G} = a \\ \mathrm{H} = b \\ \mathrm{J} = \top \end{bmatrix} & \begin{array}{l} \text{Carpenter (skeptical)} \\ \text{ALKB (old)} \end{array} \\
 & & = \begin{bmatrix} \mathrm{F} = \boxed{1} \\ \mathrm{G} = a \\ \mathrm{H} = b \\ \mathrm{J} = \boxed{1} \end{bmatrix} & \text{Bouma} \\
 & & = \bot & \begin{array}{l} \text{Russell } \textit{et al.} \\ \text{ALKB (current)} \end{array}
\end{array}
$$

Here the conflict in the default information is entirely between specifications of reentrancy. This example illustrates why default unification involving maximal incorporation of information has exponential complexity; the atomic feature structure `< F > = < J >` conflicts only with a particular combination of the other atomic feature structures from the default.

3.2 Default unification of typed feature structures

Consideration of typed feature structures further increases the possible definitions of default unification. I have not attempted to even approximate to a definition which corresponds to incorporating the maximal amount of information carried by the types in a feature structure, although Carpenter's definition could be extended in this way. There are two main reasons for this; first, as mentioned earlier, in LAUREL default inheritance is constrained by using the type system. Furthermore, maximal incorporation of typed information can give results which are quite unintuitive. Consider the following type system:



We assume the atomic types **true** and **false** to be subtypes of **bool** and that the constraints on the other types are:

$$C(t_1) = \begin{bmatrix} t_1 \\ \text{F} = \textbf{bool} \end{bmatrix} \; C(t_2) = \begin{bmatrix} t_2 \\ \text{F} = \textbf{bool} \\ \text{G} = \textbf{bool} \end{bmatrix} \; C(t_3) = \begin{bmatrix} t_3 \\ \text{F} = \textbf{bool} \\ \text{G} = \textbf{true} \end{bmatrix}$$

$$C(t_4) = \begin{bmatrix} t_4 \\ \text{F} = \textbf{bool} \\ \text{H} = \textbf{bool} \end{bmatrix} \; C(t_5) = \begin{bmatrix} t_5 \\ \text{F} = \textbf{true} \\ \text{G} = \textbf{bool} \\ \text{H} = \textbf{true} \end{bmatrix}$$

This gives the following result for a definition of default unification based on unification of the non-default structure with the maximally informative well-formed feature structures which subsume the default, followed by generalisation of the results:

(6) $$\begin{bmatrix} t_4 \\ \text{F} = \textbf{bool} \\ \text{H} = \textbf{bool} \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} t_3 \\ \text{F} = \textbf{bool} \\ \text{G} = \textbf{true} \end{bmatrix} = \begin{bmatrix} t_5 \\ \text{F} = \textbf{true} \\ \text{G} = \textbf{true} \\ \text{H} = \textbf{true} \end{bmatrix}$$

The result seems wrong, because information which is not carried by either feature structure is appearing in the result. The effect arises because t_3 and t_4 are incompatible types and the maximally informative well-formed feature structure which subsumes the default and unifies with the non-default is:

(7) $$\begin{bmatrix} t_2 \\ \text{F} = \textbf{bool} \\ \text{G} = \textbf{true} \end{bmatrix}$$

This is identical to the default structure, except that its type is t_2, the immediate parent of the type of the default, t_3. Types t_2 and t_4 are compatible, but their greatest lowest bound is type t_5, which introduces extra constraints, and thus we obtain the additional information in the result of default unification.

The situation becomes even more complex if we add the further information to the default about the value of F even though this does not conflict with the non-default:

(8) $$\begin{bmatrix} t_4 \\ \text{F} = \textbf{bool} \\ \text{H} = \textbf{bool} \end{bmatrix} \overset{<}{\sqcap} \begin{bmatrix} t_3 \\ \text{F} = \textbf{false} \\ \text{G} = \textbf{true} \end{bmatrix} = \begin{bmatrix} t_4 \\ \text{F} = \textbf{bool} \\ \text{H} = \textbf{bool} \end{bmatrix}$$

This result arises because there are two maximally specific feature structures which subsume the default, and well-formed unify with the non-default:

(9) $$\begin{bmatrix} t_2 \\ \text{F} = \textbf{bool} \\ \text{G} = \textbf{true} \end{bmatrix}, \begin{bmatrix} t_1 \\ \text{F} = \textbf{false} \end{bmatrix}$$

To work out the result we generalise over the result of unifying the non-default structure with each of these:

(10) $$\begin{bmatrix} t_5 \\ \text{F} = \textbf{true} \\ \text{G} = \textbf{true} \\ \text{H} = \textbf{true} \end{bmatrix} \sqcup \begin{bmatrix} t_4 \\ \text{F} = \textbf{false} \\ \text{H} = \textbf{bool} \end{bmatrix}$$

These results seem unintuitive; adopting this approach leads to a situation where working out the results of default unification involves considering the interaction of the information ordering given by the type hierarchy with that of the feature structure system.

Thus in LAUREL a definition was used in which the type system constrains default inheritance. In both the examples above, since t_3 and t_4 are incompatible types, default unification would simply return the non-default feature structure. In general information which is carried by any feature structure which is part of the default is incorporated if it is fully compatible with the non-default (i.e. unifies with the relevant part), but is only ever partially incorporated (i.e. default unified) if its type is the same as, or more general than, the non-default.

This can be formalised in terms of a decomposition function (TypeDecomp) which differs from the atomic decomposition function in that it does not split up the feature structures completely. Only parts of the feature structure which are fully type compatible with the non-default structure are split; TypeDecomp is thus defined relative to the non-default structure. In what follows reentrancy is regarded as non-defeasible.

Definition 33 (Decomposition) *If $F_1 = \langle Q_1, q_0, \delta_1, \alpha_1\rangle$ and $F_2 = \langle Q_2, q_0', \delta_2, \alpha_2\rangle$ where F_1 contains no path equivalence specifications then we define the decomposition of F_1 with respect to F_2,* TypeDecomp(F_1, F_2)*, as follows:*
If $\alpha_2(q_0') \sqsubseteq \alpha_1(q_0)$ then TypeDecomp(F_1, F_2) *is the set of all feature structures, $F_n = \langle Q_n, q_0'', \delta_n, \alpha_n\rangle$ such that:*

- $\alpha_n(q_0'') = \alpha_1(q_0)$
- *If F_1 is non-atomic (i.e.* $\mathrm{Feat}_0(F_1)$ *is non-empty), then $\delta_n(q_0'', f)$ for some $f \in$* $\mathrm{Feat}_0(F_1)$ *is a member of* TypeDecomp$(\delta_1(q_0, f), \delta_2(q_0', f))$ *(where because of the type compatability condition $\delta_2(q_0', f)$ must be defined) and $\delta_n(q_0'', f')$ for all $f' \neq f$ is undefined*

If $\alpha_2(q_0') \not\sqsubseteq \alpha_1(q_0)$ then TypeDecomp$(F_1, F_2) = \{F_1\}$

Note that the members of TypeDecomp(F_1, F_2) are not necessarily well-formed feature structures by this definition.

Typed default unification is then defined as follows:

Definition 34 (Typed default unification) *$F_1 \overset{<}{\sqcap} F_2 = F_3 \sqcap \bigsqcap\{F \mid F \in D(F_2)$ and $F_1 \sqcap F \neq \perp$ and there is no $F' \in D(F_2)$ such that $F' \sqcap F_3 \neq \perp$ and $F' \sqcap F_3 \sqcap F = \perp\}$*
where $F_3 = F_1 \sqcap \bigsqcap PE(F_2)$ and $D(F_2)$ is TypeDecomp$(\sqcap PV(F_2), F_1)$

This is the definition used in LAUREL when no equality constraints are involved. For example, given:

$$F_1 = \begin{bmatrix} \textbf{t1} & \\ \text{F} = & \begin{bmatrix} \textbf{t2} \\ \text{g} = \textbf{t5} \\ \text{h} = \top \end{bmatrix} \\ \text{J} = & \begin{bmatrix} \textbf{t3} \\ \text{g} = \textbf{t5} \\ \text{h} = \top \end{bmatrix} \end{bmatrix} ; F_2 = \begin{bmatrix} \textbf{t1} & \\ \text{F} = & \begin{bmatrix} \textbf{t2} \\ \text{g} = \textbf{t6} \\ \text{h} = \textbf{t7} \end{bmatrix} \\ \text{J} = & \begin{bmatrix} \textbf{t4} \\ \text{g} = \textbf{t6} \\ \text{h} = \textbf{t7} \end{bmatrix} \end{bmatrix} ; \textbf{t5} \sqcap \textbf{t6} = \perp \text{ and } \textbf{t3} \not\sqsubseteq \textbf{t4}$$

then:

$$F_1 \overset{<}{\sqcap} F_2 = \begin{bmatrix} \textbf{t1} & \\ \text{F} = & \begin{bmatrix} \textbf{t2} \\ \text{g} = \textbf{t5} \\ \text{h} = \textbf{t7} \end{bmatrix} \\ \text{J} = & \begin{bmatrix} \textbf{t3} \\ \text{g} = \textbf{t5} \\ \text{h} = \top \end{bmatrix} \end{bmatrix}$$

since TypeDecomp$(\sqcap PV(F_2), F_1) =$

$$\left\{ \begin{bmatrix} \textbf{t1} & \\ \text{F} = & \begin{bmatrix} \textbf{t2} \\ \text{g} = \textbf{t6} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textbf{t1} & \\ \text{F} = & \begin{bmatrix} \textbf{t2} \\ \text{h} = \textbf{t7} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textbf{t1} & \\ \text{J} = & \begin{bmatrix} \textbf{t4} \\ \text{g} = \textbf{t6} \\ \text{h} = \textbf{t7} \end{bmatrix} \end{bmatrix} \right\}$$

The following definition is for the combination of a non-default and a default feature structure in the presence of equality constraints:

Definition 35 (Typed default unification with equality constraints) *The default unification of a feature structure F_1 with a default feature structure F_2 given a set of equality constraints, C, is given by $\langle F_1, C\rangle \overset{<}{\sqcap}_e F_2$*
$= F_3 \sqcap \bigsqcap\{F \mid F \in D(F_2)$ and $F_1 \sqcap F \neq \perp$ and for all $\phi \in C$, $F_1 \sqcap F$ satisfies ϕ, and there is no $F' \in D(F_2)$ such that $F' \sqcap F_3 \neq \perp$ and $F' \sqcap F_3 \sqcap F = \perp\}$
where $F_3 = F_1 \sqcap \bigsqcap PE(F_2)$ and $D(F_2)$ is TypeDecomp$(\sqcap PV(F_2), F_1)$

where the definition of satisfaction of an equality constraint repeated from Section 2.6.2 is:

Definition 36 (Feature structure equality constraint) *A feature structure* $F = \langle Q, q_0, \delta, \alpha \rangle$ *satisfies a feature structure equality constraint* $\pi == F'$ *iff* $F@\pi \sim F'$

Definition 37 (Type equality constraint) *A feature structure* $F = \langle Q, q_0, \delta, \alpha \rangle$ *satisfies a type equality constraint* $\pi =* t$ *iff* $F@\pi \sim C(t)$

The motivation for equality constraints is considered in detail in Section 3.4.

3.3 Inheritance hierarchies and multiple inheritance

Clearly, since default unification is non-monotonic, the order in which a sequence of default unification operations are carried out will be significant. Carpenter (1991) discusses inheritance based on templates and inheritance over an explicitly defined hierarchy. Inheritance in LAUREL is similar to Carpenter's template scheme in that the inheritance hierarchy is implicit, but unlike it in that complete feature structures rather than descriptions are related.

One way of allowing inheritance to operate over a hierarchy would be to modify the definition of default unification, to order the information units in a way which corresponded to the inheritance hierarchy (compare the way in which we defined preference of path equivalence specifications to path value specifications). This would give a definition which, in effect, ignored the level of granularity given by the complete feature structures. Clearly this is not any more computationally feasible than the formulation of default unification which we gave originally. Conflicts can arise, not just from reentrancy, but also from multiple inheritance conflicts of the 'Nixon diamond' type, where there is no ordering between defaults to allow resolution. Again we could produce variant definitions; if reentrancy is regarded as non-defeasible for example, all the reentrant atomic feature structures could be unified first and if that succeeded the non-reentrant structures could be considered in groups according to their priority. Essentially definitions along these lines give a skeptical, 'bottom-up', inheritance scheme, but there are a whole range of possibilities, which give more or less intuitive results.

The definition used in LAUREL was intended to give behaviour which did respect the integrity of the feature structures, which was relatively simple to understand and which could be used with data automatically acquired from MRDs (at the cost of some flexibility in ordering defaults). The semantics of the non-local description language leads to inheritance operating top-down over whole feature structures; that is a psort will be fully expanded with inherited information before it is used for default inheritance. Once a psort is expanded, no distinction is retained between inherited and locally specified information. As Carpenter explains top-down and bottom-up definitions give differing behaviour since default unification is non-associative. The particular example that Carpenter uses does not have non-associative behaviour under our definition of default unification, but there are other cases which have non-associative behaviour under both definitions. Consider an example lexical entry file:

```
A 1
   < F > = < G > .

B 1
   < F > = true
   < G > = true
   < > < A_1 < >.

C 1
   < F > = false
   < > < B_1 < >.
```

We have the following results for the value of `C_1` (assuming a typing scheme such that F and G are appropriate features for all lexical entries with value **bool**):

$\begin{bmatrix} \text{F} = \boxed{1}\ \textbf{false} \\ \text{G} = \boxed{1} \end{bmatrix}$	LKB default unification, top-down	(both versions)
$\begin{bmatrix} \text{F} = \textbf{false} \\ \text{G} = \textbf{true} \end{bmatrix}$	LKB default unification, bottom-up	(old version)
$\perp$		(new version)
$\begin{bmatrix} \text{F} = \textbf{false} \\ \text{G} = \textbf{boolean} \end{bmatrix}$	Carpenter's skeptical default unification, top-down	
$\begin{bmatrix} \text{F} = \textbf{false} \\ \text{G} = \textbf{true} \end{bmatrix}$	Carpenter's skeptical default unification, bottom-up	

If we were to view default inheritance in terms of individual units of information being asserted at various points in an inheritance hierarchy, top-down inheritance can thus result in information which is asserted at a higher level being preferred over information asserted at the lower level.

Thus the main use of default inheritance in LAUREL is as a relationship between coherent parts of fully expanded lexical entries and from this the top-down behaviour follows. This is far more efficient than bottom-up inheritance would be, for this application, since the expanded psort can be cached. Again, in practice, top-down, bottom-up distinctions in behaviour arise with very low frequency.

As shown in section 2.6.2, all psorts from which information is to be inherited are unified before default unification takes place, thus removing any order dependency in the description language. The decision to restrict multiple default inheritance to the cases where the information inherited is consistent was determined by the use of semi-automatically acquired data. A fundamental point is that we cannot decide on an appropriate way of resolving conflicts in multiple inheritance without knowing what type of conflicts actually arise. Given that automatic extraction of information from MRDs is inevitably error prone, and that lexicographers' definitions are frequently not mutually consistent, we expected that most conflicts would be due to errors in the extraction process, or to inadequate definitions. Thus disallowing multiple inheritance conflicts is reasonable as an initial position. This at least gives the user the option of manually editing the lexical entries in order to get the desired behaviour, whereas any approach which did not signal the presence of conflicts would not.

3.4 Defaults and equality constraints

The issues that arise in describing default inheritance over feature structures are closely related to those that arise in AI in default inheritance hierarchies (see for example Touretzky *et al.* (1987)) and in non-monotonic logic. Thus the multiple results for credulous default unification are analogous to the multiple extensions which will arise in the presence of conflicts such as the infamous 'Nixon Diamond', and the concept that more specific information should be preferred corresponds to the 'Penguin Principle'. The semantics given for defaults in DATR (Evans and Gazdar, 1989b) is loosely based on that given by Moore (1985) for auto-epistemic logic and similar formalisations could be attempted for default operations on feature structures. For example, a situation where defaults were partially ordered but where conflicts in defaults of equal priority led to inconsistency, could be described in similar terms to hierarchical auto-epistemic logic (HAEL; Konolige 1987).

I will return briefly to the connection between defaults in the lexicon and non-monotonic logic in Chapter 8. Here I would like to describe one issue which arose when using the default inheritance scheme in the ACQUILEX LKB, which is far less frequently discussed than, for example, multiple inheritance conflicts, but which has general relevance for default inheritance systems, and non-monotonic logics which attempt to capture the 'Penguin Principle'. It arises because we may wish to override inherited information not by giving a conflicting value for the attribute, but by giving a value which is less specified or only partially conflicts. This is the motivation for the introduction of equality constraints.

As an illustrative example assume that substance_1 is a psort of type **physical**:

```
substance 1
   < QUALIA > = physical
   < QUALIA : FORM > = mass.
```

We can override the value for FORM for a more specific psort by specifying `< QUALIA : FORM > =` **individuated**, for example. However suppose we want to create the lexical entry for `cake_1` and that this should inherit information from `substance_1`. In this case we might want to override the information that `< QUALIA : FORM > =` **mass** without specifying a new value, in order to allow both for mass and count usages of *cake*. But if we specify

```
cake 1
   < QUALIA > < substance_1 < QUALIA >
   < QUALIA : FORM > = form.
```

default unification by any of the definitions suggested gives:

$$\texttt{cake\_1} = \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{QUALIA} = \begin{bmatrix} \textbf{physical} \\ \text{FORM} = \textbf{mass} \end{bmatrix} \end{bmatrix}$$

Clearly we cannot simply claim that the lexical entry:

```
cake 1
   < QUALIA > < substance_1 < QUALIA >.
```

should give

$$\texttt{cake\_1} = \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{QUALIA} = \begin{bmatrix} \textbf{physical} \\ \text{FORM} = \textbf{mass} \end{bmatrix} \end{bmatrix}$$

whereas

```
cake 1
   < QUALIA > < substance_1 < QUALIA >
   < QUALIA : FORM > = form.
```

should give

$$\texttt{cake\_1} = \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{QUALIA} = \begin{bmatrix} \textbf{physical} \\ \text{FORM} = \textbf{form} \end{bmatrix} \end{bmatrix}$$

given that the type entry was

```
physical (nomqualia)
   < FORM > = form.
```

since the expansion of the non-default feature structure would result in the same thing in both cases.

A related problem arose when constructing entries automatically as part of the ACQUILEX project. For example consider the following definitions from the Italian Garzanti Dictionary:[5]

cacio latte di vacca, pecora o capra cagliato, salto e seccato in forma; formaggio

marzolino cacio fatto con latte di pecora o di bufala

Schematically the LKB entries can be represented as:

```
cacio
   < SOURCE > = (cow sheep goat).


marzolino
   < > < cacio < >
   < SOURCE > = (sheep buffalo).
```

[5] My thanks to Elisabetta Marinai and Simonetta Montemagni for bringing these examples to my attention.

where disjunction is indicated by bracketing a list of types. But (default) unification results in SOURCE having a value **sheep** in `marzolino` whereas the desired result seems obviously to be (**sheep buffalo**).

These sort of examples also appear in general default reasoning, and the same problems can arise in any system if conflicts are necessary to block inheritance. For example, the following are reasonable statements:

> Usually Quakers are pacifists.
> Nominal Quakers (i.e. people who are technically members of the Society of Friends but who are not now active members) may or may not be pacifists.
> Nixon is a nominal Quaker.

Intuitively we should be able to make some inferences about Nixon by virtue of the fact that he is technically a Quaker (even if only that his name is on some list), but although we would not expect most other properties of Quakers to hold, we have no grounds for stating conflicting values for these properties for the class of nominal Quakers. Thus, given the information above we should only conclude by default:

> Nixon may or may not be a pacifist.

We might also expect information about current political conviction to be able to set the property of pacifism without there being any conflict, thus allowing one possible way of avoiding the Nixon Diamond.

Intuitively it seems that both in this context, and in that of dictionary definitions, a statement about a class or an individual entity with respect to some property (e.g. type of milk used, pacifism) should be taken to be *maximally specific*, in that it obeys Grice's first sub-maxim of Quantity and is "the strongest, or most informative, that can be made in the situation" (Levinson, 1983). The most informative default statements about some properties of some classes may have to be weaker than valid statements about their superclasses, if defaults have a probabilistic interpretation in even the very loose sense that:

> Usually Xs are Ys.

has a paraphrase:

> If all that is known about something is that it is an X (and all inferences from this) then if you assume that it is a Y you will be right significantly more than half of the time.

There must be some subclass Z of X for which neither of the following statements are true:

> Usually Zs are Ys.
> Usually Zs are not Ys.

It may be that this is not a situation that arises frequently with natural classes and natural properties in a taxonomic setting. However, if this is the case, it implies that natural classes have extra semantic properties besides that of being a subset of the superclass.

There thus seem to be good technical reasons for wanting to allow this inference pattern in a representation language, and some intuitive justification for its validity. But given a non-monotonic logic which represents statements such as "Usually Quakers are pacifists" as:

$$Q > P$$

where one can conclude this from:

$$Q \Rightarrow P$$

we cannot represent the statement about nominal Quakers simply as:

$$N > (P \vee \neg P)$$

and expect the desired effect to arise, since this will be true for any N and P, not just nominal Quakerism and pacificism.

The application of the default to the N has to be explicitly blocked. In a non-monotonic logic such as that of Delgrande (1988), this can be done by asserting the negation of the individual default statements $N > P$ and $N > \neg P$. Since this logic adopts the Penguin Principle, this prevents application of the default $Q > P$. I would claim that this captures the intuition that the disjunct is maximally informative. Thus we actually need to state:

$\neg(N > P)$ (It is not the case that nominal Quakers are usually pacifists)
$\neg(N > \neg P)$ (It is not the case that nominal Quakers are usually not pacifists)

In general, for disjunctive statements, such as:

Marzolino is usually made from sheep or buffalo milk.

uttered in a context where maximal informativeness can be assumed, we have to assert the negation of the individual default statements:

$M > (S \vee B)$
$\neg(M > S)$
$\neg(M > B)$

(where $\neg(B \wedge S)$). This will prevent the application of a default such as $C > S$ where $M \Rightarrow C$.

I introduced equality constraints into LAUREL in order to achieve an equivalent effect. Equality up to alphabetic variance of a path with the constraint on a type is represented by `=*` in the path notation. Thus we have:

```
cake 1
   < QUALIA > < substance_1 < QUALIA >
   < QUALIA : FORM > =* form.
```

which expands into the desired result:

$$\texttt{cake\_1} = \begin{bmatrix} \textbf{lex-noun-sign} \\ \text{QUALIA} = \begin{bmatrix} \textbf{physical} \\ \text{FORM} = \textbf{form} \end{bmatrix} \end{bmatrix}$$

Some other applications of equality constraints will be seen in subsequent chapters.

3.5 A summary of the main features of LAUREL

As I described in the previous chapter, the basic data structure used in LAUREL are typed feature structures very similar to those proposed by Carpenter. I have simplified Carpenter's scheme slightly, to remove the distinction between appropriateness specifications and constraints, so that constraints on types are themselves just well-formed feature structures. What is more novel is the notion of a psort, because this allows inheritance to operate from objects such as lexical entries and lexical rules. The significance of this is that it allows the lexicon to be structured not only in terms of predefined types or templates, but also by encoding relationships between (parts of) lexical entries. The built-in relationships are non-default and default inheritance and equality, but by defining lexical rules it is also possible to describe lexical entries in terms of more complex transformations to other entries. Psorts are themselves well-formed feature structures, rather then descriptions of feature structures. This distinction is important when we consider default inheritance, as I discussed in section 3.3. Psorts are not explicitly organised into a hierarchy but it is necessary to impose a non-cyclicity condition on the induced hierarchy of psorts in order to allow evaluation.

The use of defaults in lexical description languages raises a number of issues. I have discussed the problems involved at arriving at a definition of default unification and given an overview of some of the possible alternatives. I proposed a definition which is computationally efficient and meets most of the other desiderata and extended this to a typed framework, in a way which results in the types constraining defaults. The conditions on default inheritance from psorts mean that multiple inheritance is only allowed when the inherited information is mutually consistent. This is restrictive, but avoids the complication of having to introduce an explicit mechanism for the resolution of conflicts. Practical experience with LAUREL so far suggests that the consistent multiple inheritance condition is not overrestrictive and that the non-associative behaviour of default unification and the potential complications involving reentrant structures are not significant problems when using the language.

One extension to LAUREL which was prompted by experience of its use is the specification of equality of feature structures, defined as mutual subsumption. This quite straightforwardly allows the possibility of overriding inherited information with explicit statements of lack of information. This is a potentially useful facility in a lexical description language of this type because lexical entries themselves represent partial information and are necessarily underspecified.

Thus, using LAUREL, the lexicon is defined and constrained by the predefined types, but is also more loosely structured by inheritance relationships between entries and by the relationships encoded by lexical rules. The extent to which these various elements contribute to the description of any particular theory can vary. The motivations for the differences between LAUREL and other typed unification based formalisms were considerations of the representation of lexical semantics and the relationships between lexical entries, and also the use of MRDs to construct lexical entries (semi-)automatically. Some of these issues have been alluded to in the last two chapters but I will now turn to describing such use of LAUREL in considerably more detail, using more linguistically motivated examples.

Chapter 4

Use of the representation language

LAUREL is a very general language which makes a variety of techniques available for the representation of linguistic information. In this chapter I will illustrate its application to several problems in lexical semantic representation. The linguistic basis for much of this is controversial and I will not attempt to do full justice to all the issues involved here, but I will instead concentrate on the representation problems. I have, however, made some attempt to integrate the fragments described, in that the lexical entries are consistent with each other and make use of a single type system and grammar, which will also be the basis for the treatment of individuation discussed in Chapter 5. I adopt a categorial style grammatical framework, following Sanfilippo's treatment of the verb type system (Sanfilippo, in press), discussed below, but I augment this with an extensive range of unary type-shifting rules, as discussed in subsequent sections.

4.1 Verb representation and psychological predicates

Sanfilippo's treatment of the verb type system in the ACQUILEX LKB provides a good introduction to the way in which syntactic and semantic information can be integrated and to the use of the type system to encode generalisations about lexical behaviour. The approach taken is sign-based (Pollard and Sag, 1987), words and phrases are represented as (typed) feature structures in which orthographic, syntactic and semantic information are simultaneously represented. Unlike HPSG, the grammar is based on a categorial approach (Unification Categorial Grammar (UCG); Zeevat *et al.*, 1987). The semantic description uses Dowty's (1989) account of thematic information within a neo-Davidsonian treatment of verb semantics (Sanfilippo, 1990), further augmented by classifications based on those discussed by Jackendoff (1990) and Levin (1992). I will describe the type system rather informally by giving some examples of lexical entries and fragments of the type hierarchy; a fuller description is given in Sanfilippo (in press). The classification of psychological predicates is discussed in Sanfilippo and Poznanski (1992), which also describes the semi-automatic acquisition of the lexical entries from information encoded in LDOCE and the Longman Lexicon.

This sign shown in Figure 4.1 has a category value which will allow it to combine with an object NP.[1] The grammar rule involved is shown in Figure 4.2. In fact in Sanfilippo's system, NPs are type raised, so the verbal sign instantiates the value of `< CAT : ACTIVE >` in the type raised NP, for example the sign shown in Figure 4.3. The category of a type-raised NP is, in effect, `X | (X | NP)` where the directionality of the categorial slash (`< CAT : DIRECTION >`) is acquired from the argument sign, hence the somewhat counter-intuitive specification of `< CAT : DIRECTION >` as backward in Figure 4.1 and the reentrancy between `< CAT : DIRECTION >`and `< CAT : ACTIVE : CAT : DIRECTION >` in Figure 4.3. The phrasal sign corresponding to *scare John* is shown in Figure 4.4.

The semantics of the verb is introduced by the feature SEM. Feature structures can be used to

[1] In this figure and some of the subsequent examples, I have omitted some features which are not relevant to the discussion in order to make the description clearer.

$$\left[\begin{array}{l}\textbf{strict-trans-sign}\\ \text{ORTH} = \textbf{scare}\\ \text{CAT} = \left[\begin{array}{l}\textbf{strict-trans-cat}\\ \text{RESULT} = \left[\begin{array}{l}\textbf{strict-intrans-cat}\\ \text{RESULT} = \boxed{\textbf{sent-cat}}\\ \text{DIRECTION} = \textbf{forward}\\ \text{ACTIVE} = \boxed{\textbf{np-sign}}\end{array}\right]\\ \text{DIRECTION} = \textbf{backward}\\ \text{ACTIVE} = \boxed{\textbf{dir-obj-np-sign}}\end{array}\right]\\ \text{SEM} = \left[\begin{array}{l}\textbf{strict-trans-sem}\\ \text{IND} = \boxed{0}\ \textbf{eve}\\ \text{PRED} = \textbf{and}\\ \text{ARG1} = \left[\begin{array}{l}\textbf{verb-formula}\\ \text{IND} = \boxed{0}\\ \text{PRED} = \boxed{1}\ \textbf{scare\_L\_1\_1}\\ \text{ARG1} = \boxed{0}\end{array}\right]\\ \text{ARG2} = \left[\begin{array}{l}\textbf{binary-formula}\\ \text{IND} = \boxed{0}\\ \text{PRED} = \textbf{and}\\ \text{ARG1} = \boxed{2}\left[\begin{array}{l}\textbf{p-agt-formula}\\ \text{IND} = \boxed{0}\\ \text{PRED} = \textbf{p-agt-neg-cause}\\ \text{ARG1} = \boxed{0}\\ \text{ARG2} = \textbf{obj}\end{array}\right]\\ \text{ARG2} = \boxed{3}\left[\begin{array}{l}\textbf{p-pat-formula}\\ \text{IND} = \boxed{0}\\ \text{PRED} = \textbf{p-pat-neg-affected-emotive}\\ \text{ARG1} = \boxed{0}\\ \text{ARG2} = \textbf{obj}\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]$$

Figure 4.1: Lexical entry for *scare*

$$\left[\begin{array}{l}\textbf{grammar-rule}\\ 0 = \left[\begin{array}{l}\textbf{sign}\\ \text{ORTH} = \left[\begin{array}{l}\textbf{complex-orth}\\ \text{ORTH1} = \boxed{0}\ \textbf{orth}\\ \text{ORTH2} = \boxed{1}\ \textbf{orth}\end{array}\right]\\ \text{CAT} = \boxed{2}\ \textbf{cat}\\ \text{SEM} = \boxed{3}\ \textbf{sem}\end{array}\right]\\ 1 = \boxed{4}\left[\begin{array}{l}\textbf{sign}\\ \text{ORTH} = \boxed{0}\\ \text{CAT} = \left[\begin{array}{l}\textbf{complex-cat}\\ \text{RESULT} = \textbf{cat}\\ \text{DIRECTION} = \textbf{backward}\\ \text{ACTIVE} = \left[\begin{array}{l}\textbf{sign}\\ \text{ORTH} = \textbf{orth}\\ \text{CAT} = \textbf{cat}\\ \text{SEM} = \textbf{sem}\end{array}\right]\end{array}\right]\\ \text{SEM} = \textbf{sem}\end{array}\right]\\ 2 = \left[\begin{array}{l}\textbf{sign}\\ \text{ORTH} = \boxed{1}\\ \text{CAT} = \left[\begin{array}{l}\textbf{complex-cat}\\ \text{RESULT} = \boxed{2}\\ \text{DIRECTION} = \textbf{direction}\\ \text{ACTIVE} = \boxed{4}\end{array}\right]\\ \text{SEM} = \boxed{3}\end{array}\right]\end{array}\right]$$

Figure 4.2: Grammar rule for backward application

$$\left[\begin{array}{l}
\textbf{reg-raised-np-sign} \\
\text{ORTH} = \textbf{john} \\
\text{CAT} = \left[\begin{array}{l}
\textbf{raised-np-cat} \\
\text{RESULT} = \boxed{0}\ \textbf{cat} \\
\text{DIRECTION} = \boxed{1}\ \textbf{direction} \\
\text{ACTIVE} = \left[\begin{array}{l}
\textbf{sign} \\
\text{ORTH} = \textbf{orth} \\
\text{CAT} = \left[\begin{array}{l}
\textbf{complex-cat} \\
\text{RESULT} = \boxed{0} \\
\text{DIRECTION} = \boxed{1} \\
\text{ACTIVE} = \left[\begin{array}{l}
\textbf{sign} \\
\text{ORTH} = \textbf{orth} \\
\text{CAT} = \boxed{\textbf{basic-cat}} \\
\text{SEM} = \left[\begin{array}{l}
\textbf{binary-formula} \\
\text{IND} = \textbf{entity} \\
\text{PRED} = \textbf{logical-pred} \\
\text{ARG1} = \textbf{sem} \\
\text{ARG2} = \boxed{2}\ \textbf{obj}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\text{SEM} = \boxed{3} \left[\begin{array}{l}
\textbf{formula} \\
\text{IND} = \boxed{4}\ \textbf{entity} \\
\text{PRED} = \textbf{logical-pred} \\
\text{ARG1} = \textbf{sem}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\text{SEM} = \left[\begin{array}{l}
\textbf{binary-formula} \\
\text{IND} = \boxed{4} \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \left[\begin{array}{l}
\textbf{unary-formula-entity-arg1} \\
\text{IND} = \boxed{2} \\
\text{PRED} = \boxed{5}\ \textbf{john\_1} \\
\text{ARG1} = \boxed{2}
\end{array}\right] \\
\text{ARG2} = \boxed{3}
\end{array}\right]
\end{array}\right]$$

Figure 4.3: Feature structure for a type raised NP

$$\left[\begin{array}{l}
\textbf{sign} \\
\text{ORTH} = \left[\begin{array}{l}
\textbf{complex-orth} \\
\text{ORTH1} = \textbf{scare} \\
\text{ORTH2} = \textbf{john}
\end{array}\right] \\
\text{CAT} = \left[\begin{array}{l}
\textbf{strict-intrans-cat} \\
\text{RESULT} = \boxed{\textbf{sent-cat}} \\
\text{DIRECTION} = \textbf{forward} \\
\text{ACTIVE} = \boxed{\textbf{np-sign}}
\end{array}\right] \\
\text{SEM} = \left[\begin{array}{l}
\textbf{binary-formula} \\
\text{IND} = \boxed{0}\ \textbf{eve} \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \left[\begin{array}{l}
\textbf{unary-formula-entity-arg1} \\
\text{IND} = \boxed{1}\ \textbf{obj} \\
\text{PRED} = \textbf{john\_1} \\
\text{ARG1} = \boxed{1}
\end{array}\right] \\
\text{ARG2} = \left[\begin{array}{l}
\textbf{strict-trans-sem} \\
\text{IND} = \boxed{0} \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \left[\begin{array}{l}
\textbf{verb-formula} \\
\text{IND} = \boxed{0} \\
\text{PRED} = \textbf{scare\_L\_1\_1} \\
\text{ARG1} = \boxed{0}
\end{array}\right] \\
\text{ARG2} = \left[\begin{array}{l}
\textbf{binary-formula} \\
\text{IND} = \boxed{0} \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \boxed{2} \left[\begin{array}{l}
\textbf{p-agt-formula} \\
\text{IND} = \boxed{0} \\
\text{PRED} = \textbf{p-agt-neg-cause} \\
\text{ARG1} = \boxed{0} \\
\text{ARG2} = \textbf{obj}
\end{array}\right] \\
\text{ARG2} = \left[\begin{array}{l}
\textbf{p-pat-formula} \\
\text{IND} = \boxed{0} \\
\text{PRED} = \textbf{p-pat-neg-affected-emotive} \\
\text{ARG1} = \boxed{0} \\
\text{ARG2} = \boxed{1}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

Figure 4.4: Feature structure for *scare John*

express semantic constructions which can be interpreted equivalently to expressions of the lambda calculus in most cases (Moore, 1989), but arguments are instantiated via unification, not function application. The semantic representation adopted in UCG is InL (Indexed Language, Zeevat *et al*, 1987). The semantics for *scare* shown in Figure 4.1 can be expressed in a linearised notation in InL as follows:

$$[e][\mathbf{scare\_L\_1\_1}(e) \wedge \mathbf{p\text{-}agt\text{-}neg\text{-}cause}(e, x) \wedge \mathbf{p\text{-}pat\text{-}neg\text{-}affected\text{-}emotive}(e, y)]$$

Here the index e corresponds to the instance of the **eve** type shown in Figure 4.1. The two occurrences of **obj** in Figure 4.1 correspond to the variables x and y in the linearised notation. As I mentioned in section 2.3, under an intensional interpretation of atomic values these instances are (potentially) distinct. For clarity of exposition, I omit references to indices of subformulae and expand out the nested conjunctions when they share the same index. The semantics for the type raised NP shown in Figure 4.3 can be represented as:

$$[a][\mathbf{john\_1}(y) \wedge [a]A]$$

Here $[a]A$ stands for the formula that translates the argument to which the NP will be applied. The semantics for *scare John* is:

$$\begin{aligned}[e][\mathbf{john\_1}(y) \wedge \mathbf{scare\_L\_1\_1}(e) \wedge \mathbf{p\text{-}agt\text{-}neg\text{-}cause}(e, x) \\ \wedge\ \mathbf{p\text{-}pat\text{-}neg\text{-}affected\text{-}emotive}(e, y)]\end{aligned}$$

The correspondence between the argument of **john_1** and the second argument of **p-pat-neg-affected-emotive** is set up by the category values (not shown in full in the figures). From now on I will gloss some of the feature structure diagrams with the linearised notation to make it easier to decipher the semantic representation.

The predicates **p-agt-neg-cause** and **p-pat-neg-affected-emotive** in the representation for *scare* illustrate a novel aspect of Sanfilippo's treatment; they can be thought of as further instantiations of Dowty's proto-roles **p-agt** and **p-pat**. This allows the representation of psychological predicates to reflect their semantic classification according to the following parameters (see, for example, Jackendoff (1990) and Levin (1992)):

- The stimulus argument is realised as object and experiencer as subject (e.g. *admire, experience, fear*).
- The experiencer is realised as object and stimulus as subject (e.g. *delight, interest, scare*).
- The affect is positive (*admire, delight*), neutral (*experience, interest*) or negative (*fear, scare*).

One point about this representation is that roles appear as types rather than features. This contrasts with the HPSG approach, where arguments are associated with verbs by features which are essentially specific to the verb, e.g. SCARER and SCARED. This can only be achieved if a new type is introduced for each verb, with those features specified as appropriate for the type. This does not pose any formal problems for the typed feature structure language, but practically it causes difficulties since it makes the type system very large, and means it has to be augmented for every new lexical entry.[2] It also makes it impossible to state generalisations about verb argument association such as those discussed by Sanfilippo (in press).

Essentially, the verb type hierarchy consists of two distinct systems, one for the syntax and one for the semantics, which are brought together in the sign types. For example, the type **strict-trans-sign** has the constraint shown in Figure 4.5, which brings together the types **strict-trans-cat** and **strict-trans-sem**. Parts of the syntax and semantics hierarchies are shown in Figures 4.6 and 4.7. Here **x-sign-x-cat** is an intermediate type which includes all verbs with more than one argument, see Sanfilippo (in press).

Regular alternations can be encoded by lexical rule. For example the causative-inchoative alternation, which is represented as shown in Figure 4.8, allows sentences such as *John scares*

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{strict-trans-sign}}\\
\text{ORTH} = & \textbf{orth}\\
\text{CAT} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{strict-trans-cat}}\\
\text{RESULT} = & \left[\begin{array}{l}
\textbf{strict-intrans-cat}\\
\text{RESULT} = \boxed{\textbf{sent-cat}}\\
\text{DIRECTION} = \textbf{forward}\\
\text{ACTIVE} = \boxed{\textbf{np-sign}}
\end{array}\right]\\
\multicolumn{2}{l}{\text{DIRECTION} = \textbf{backward}}\\
\multicolumn{2}{l}{\text{ACTIVE} = \boxed{\textbf{dir-obj-np-sign}}}
\end{array}\right]\\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{strict-trans-sem}}\\
\multicolumn{2}{l}{\text{IND} = \boxed{0}\ \textbf{eve}}\\
\multicolumn{2}{l}{\text{PRED} = \textbf{and}}\\
\text{ARG1} = & \left[\begin{array}{l}
\textbf{verb-formula}\\
\text{IND} = \boxed{0}\\
\text{PRED} = \boxed{1}\ \textbf{string}\\
\text{ARG1} = \boxed{0}
\end{array}\right]\\
\text{ARG2} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{binary-formula}}\\
\multicolumn{2}{l}{\text{IND} = \boxed{0}}\\
\multicolumn{2}{l}{\text{PRED} = \textbf{and}}\\
\text{ARG1} = \boxed{2} & \left[\begin{array}{l}
\textbf{p-agt-formula}\\
\text{IND} = \boxed{0}\\
\text{PRED} = \textbf{p-agt}\\
\text{ARG1} = \boxed{0}\\
\text{ARG2} = \textbf{obj}
\end{array}\right]\\
\text{ARG2} = \boxed{3} & \left[\begin{array}{l}
\textbf{p-pat-formula}\\
\text{IND} = \boxed{0}\\
\text{PRED} = \textbf{p-pat}\\
\text{ARG1} = \boxed{0}\\
\text{ARG2} = \textbf{obj}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

$$[e][\text{PRED}(e) \wedge \textbf{p-agt}(e,x) \wedge \textbf{p-pat}(e,y)]$$

Figure 4.5: Constraint for the type **strict-trans-sign**

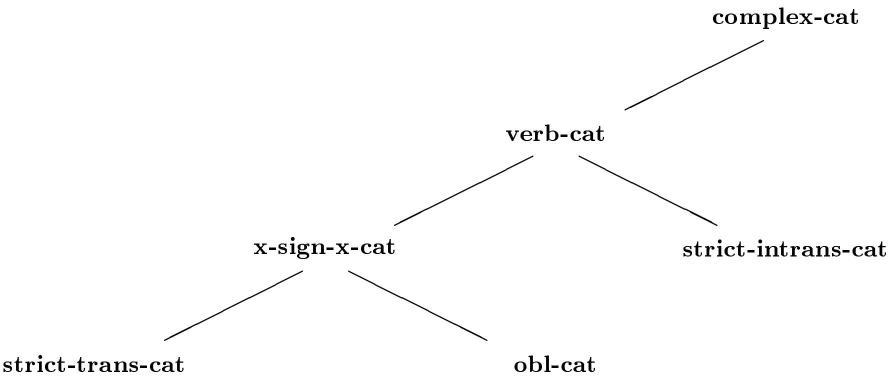


Figure 4.6: Part of the syntactic type hierarchy for verbs

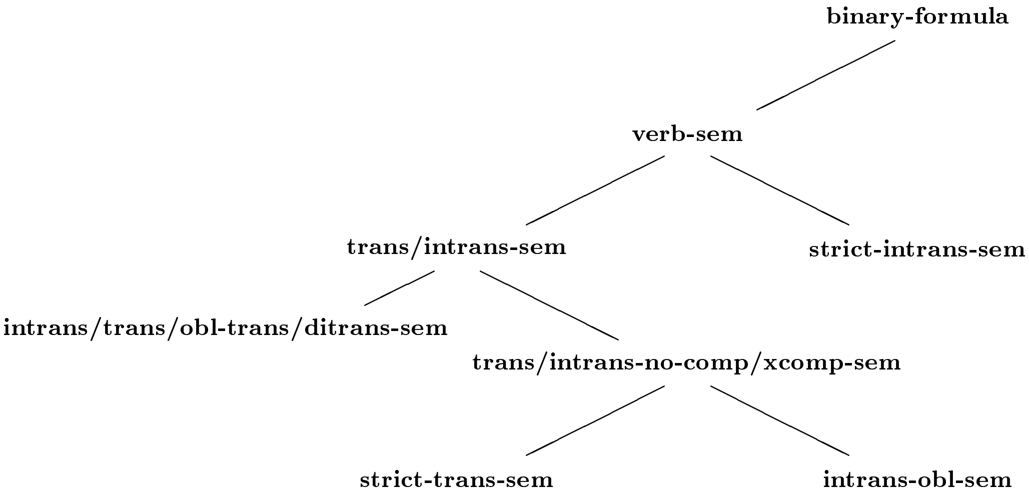


Figure 4.7: Part of the semantic type hierarchy for verbs

easily to be parsed. In a full grammar, non-lexical unary rules are also needed to supplement the categorial combination rules — for type raising mass and plural nouns to NPs without a determiner, for example.

This description of verbs therefore illustrates how detailed lexical semantic information may be encoded in LAUREL and integrated with a syntactic account. I will not attempt to discuss the linguistic issues involved, but a few problematic areas of the analysis as it stands should be mentioned, since they affect the way in which I have utilised Sanfilippo's type system. The thematic roles **p-agt** and **p-pat** and their subtypes not only specify properties of the arguments of a verb but are used to associate the arguments with the event which the verb denotes. Prepositions may also act in this way, but no further thematic roles are defined and there is no way of associating predicative or sentential arguments with the event, which means that the logical form for sentences such as *John enjoyed reading the book* will be incomplete. I have left this issue unresolved, since there is no straightforward solution which is in keeping with the rest of the analysis and since none of the work described in subsequent sections depends significantly on the particular treatment adopted, or indeed on any aspects of the neo-Davidsonian approach.

The type system is also somewhat inelegant with respect to the treatment of non-lexical signs, in that the feature structure assigned to the phrase *scares John* for example, which was shown in Figure 4.4, is not an ultimately well-formed feature structure as defined in section 2.4.6. It cannot be a **strict-intrans-sign**, for example, because the semantic specification is not compatible with **strict-intrans-sem**. Although ultimate well formedness is not enforced in LAUREL, it is a desirable property, since it allows a constraint-based approach to parsing. In this case, the connection between the syntax and the semantics is very loose if ultimate well-formedness is not enforced, since there is no other restriction which prevents a sign with the category value **strict-trans-cat** occurring with a semantics of **raised-np-sem**, for example.

In Sanfilippo's system, CAT specifies only the CAT of the result and not the whole sign; the semantics and orthography are instantiated by the grammar rules. Thus, even if the appropriate non-lexical sign types were specified, this would not result in such types actually being given to the phrasal signs produced during a parse, unless classification were applied during parsing, with its attendant problems of computational expense. In order to specify the syntactic/semantic relationship more tightly and to have meaningful phrasal signs, CAT : RESULT would have to specify the entire result sign, and the inheritance of the orthography and semantics which are currently specified in the grammar rule, should be part of a basic type **complex-sign**, as shown in

[2] As discussed in Chapter 2, string types are introduced to avoid this being necessary in LAUREL, but to implement the HPSG approach where new features must also be introduced, genuine incremental update of the type system would be needed.

$$
\left[\begin{array}{l}
\textbf{lexical-rule}\\
0 = \left[\begin{array}{l}
\textbf{lex-sign}\\
\text{ORTH} = \boxed{0}\ \textbf{orth}\\
\text{CAT} = \left[\begin{array}{l}
\textbf{strict-intrans-cat}\\
\text{RES.} = \left[\begin{array}{l}
\textbf{sent-cat}\\
\text{CAT-TYPE} = \textbf{sent}\\
\text{M-F.} = \left[\begin{array}{l}
\textbf{sent-m-feats}\\
\text{REG-MORPH} = \boxed{1}\ \textbf{boolean}\\
\text{VFORM} = \boxed{2}\ \textbf{vform}\\
\text{COMP-FORM} = \boxed{3}\ \textbf{comp-form}\\
\text{PRT} = \boxed{4}\ \textbf{string}\\
\text{DIA.} = \left[\begin{array}{l}\textbf{strict-intrans-dia.}\\ \text{PRT-ALT} = \boxed{5}\ \textbf{prt-or-obl-alt}\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\text{DIRECTION} = \textbf{forward}\\
\text{ACTIVE} = \boxed{\textbf{np-sign}}
\end{array}\right]\\
\text{SEM} = \left[\begin{array}{l}
\textbf{binary-formula}\\
\text{IND} = \boxed{6}\ \textbf{eve}\\
\text{PRED} = \textbf{and}\\
\text{ARG1} = \boxed{7}\ \left[\begin{array}{l}\textbf{verb-formula}\\ \text{IND} = \boxed{6}\\ \text{PRED} = \textbf{string}\\ \text{ARG1} = \boxed{6}\end{array}\right]\\
\text{ARG2} = \boxed{9}\ \left[\begin{array}{l}\textbf{p-pat-formula}\\ \text{IND} = \boxed{6}\\ \text{PRED} = \textbf{p-pat}\\ \text{ARG1} = \boxed{6}\\ \text{ARG2} = \textbf{obj}\end{array}\right]
\end{array}\right]
\end{array}\right]\\
1 = \left[\begin{array}{l}
\textbf{strict-trans-sign}\\
\text{ORTH} = \boxed{0}\\
\text{CAT} = \left[\begin{array}{l}
\textbf{strict-trans-cat}\\
\text{RES.} = \left[\begin{array}{l}
\textbf{strict-intrans-cat}\\
\text{RES.} = \left[\begin{array}{l}
\textbf{sent-cat}\\
\text{CAT-TYPE} = \textbf{sent}\\
\text{M-F.} = \left[\begin{array}{l}
\textbf{sent-m-feats}\\
\text{REG-MORPH} = \boxed{1}\\
\text{VFORM} = \boxed{2}\\
\text{COMP-FORM} = \boxed{3}\\
\text{PRT} = \boxed{4}\\
\text{DIA.} = \left[\begin{array}{l}\textbf{strict-trans-dia.}\\ \text{PRT-ALT} = \boxed{5}\\ \text{TRANS-ALT} = \textbf{caus-inch}\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\text{DIRECTION} = \textbf{forward}\\
\text{ACTIVE} = \boxed{\textbf{np-sign}}
\end{array}\right]\\
\text{DIRECTION} = \textbf{backward}\\
\text{ACTIVE} = \boxed{\textbf{dir-obj-np-sign}}
\end{array}\right]\\
\text{SEM} = \left[\begin{array}{l}
\textbf{strict-trans-sem}\\
\text{IND} = \boxed{6}\\
\text{PRED} = \textbf{and}\\
\text{ARG1} = \boxed{7}\\
\text{ARG2} = \left[\begin{array}{l}
\textbf{binary-formula}\\
\text{IND} = \boxed{6}\\
\text{PRED} = \textbf{and}\\
\text{ARG1} = \boxed{11}\ \left[\begin{array}{l}\textbf{p-agt-formula}\\ \text{IND} = \boxed{6}\\ \text{PRED} = \textbf{p-agt}\\ \text{ARG1} = \boxed{6}\\ \text{ARG2} = \textbf{obj}\end{array}\right]\\
\text{ARG2} = \boxed{9}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Rule input: $[e][\text{PRED}(e) \wedge \textbf{p-agt}(x) \wedge \textbf{p-pat}(y)]$
Rule output: $[e][\text{PRED}(e) \wedge \textbf{p-pat}(y)]$

Figure 4.8: Lexical rule for the causative-inchoative alternation (some feature and type names have been abbreviated).

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{complex-sign}} \\
\text{ORTH} = \boxed{1} & \\
\text{CAT} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{complex-cat}} \\
\text{RESULT} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{sign}} \\
\text{ORTH} = & \left[\begin{array}{l}
\textbf{complex-orth} \\
\text{DIRECTION} = \boxed{0}\ \textbf{direction} \\
\text{ORTH1} = \boxed{1}\ \textbf{orth} \\
\text{ORTH2} = \boxed{2}\ \textbf{orth}
\end{array}\right] \\
\text{CAT} = \boxed{2}\ \textbf{cat} & \\
\text{SEM} = \boxed{3}\ \textbf{sem} &
\end{array}\right] \\
\text{DIRECTION} = \boxed{0} & \\
\text{ACTIVE} = & \left[\begin{array}{l}
\textbf{sign} \\
\text{ORTH} = \boxed{2} \\
\text{CAT} = \boxed{3}\ \textbf{cat} \\
\text{SEM} = \textbf{sem}
\end{array}\right]
\end{array}\right] \\
\text{SEM} = \boxed{3} &
\end{array}\right]$$

Figure 4.9: Alternative specification of complex categorial signs

Figure 4.9. Again I will not pursue this issue here; in the type system used in subsequent sections I have changed CAT : RESULT to specify the entire sign, but this was done just to allow the QUALIA to be an optional component of a sign and the change is essentially insignificant. Conceptually the QUALIA should perhaps be part of the SEM, but this would have involved very significant modifications to the type system.

One aspect of Sanfilippo's system, which is not entirely compatible with the approach to nominal lexical semantics which I adopt, is his treatment of selectional restrictions. Selectional restrictions are used in one form or another by many NLP systems, to improve the efficiency of parsing by removing spurious lexical or structural ambiguity. However, the case for making such specifications in the lexicon in a unification-based system is not clear. As stated in Chapter 1, the resolution of ambiguity by itself cannot be taken as a criterion for inclusion of information in the lexicon, because arbitrarily specialised knowledge can apparently be used for disambiguation.

Sanfilippo adopts a treatment of selectional restrictions essentially similar to that described by Moens *et al.* (1989), where arguments are sorted to encode semantic restrictions. The feature structures shown above have arguments which are specified as **eve** or **obj**, but the full verb type system has a finer grain of specification, with various subtypes of events such as **state**, **proc**, **qua-eve** and objects such as **e-human**, **e-male**, **e-plant**, **qua-plant** and **cum-plant**. The motivation for the classification according to cumulativity is described in Sanfilippo (1990). I adopt an alternative encoding of the cumulative/quantised distinction which is described in Chapter 5. So, for the purposes of the current discussion, I will concentrate on the basic hierarchy of objects. Sanfilippo developed this to coincide with the encoding scheme used in LDOCE, which allowed straightforward automatic acquisition of selectional restrictions on verb arguments. Nouns are assumed to be lexically encoded using the same schema. Thus, given that *rabbit* in the animal sense has type **e-animal** and in the meat sense **e-solid**, and that the subject of *fear* is specified as **e-animate**, *rabbit* will be disambiguated to the animal sense in (1).

(1) The rabbit feared the dog.

A sentence such as (2) will not be accepted by the grammar:

(2) * The book feared the dog.

which seems reasonable given the intuition that there is something highly anomalous about it.

Although such results are desirable, selectional restrictions of this form are over-restrictive, as has been recognised in NLP for many years (see, for example, Wilks 1977). The same arguments against selectional restrictions and in favour of selectional preferences which were made then are equally applicable now. The problems are not just that no interpretation is derived for sentences which are metaphorical or otherwise marked, such as:

(3) My car drinks petrol.

but that normal usages are ruled out, such as:

(4) Most heathers prefer acid soil.

Plants are normally described as *liking*, *enjoying* or *preferring* conditions in which they thrive and *disliking* or *hating* adverse conditions. It is reasonable to describe such uses of psychological predicates as extended, but it does not seem reasonable to suggest that such sentences are marked, although giving totally distinct lexical entries for each such extended usage would be undesirable. In other circumstances, selectional restrictions will not exclude implausible sentences, such as (5) and they cannot be used to prefer the more plausible reading in examples such as (6).

(5) The dead rabbit bit Mary.

(6) The astronomer saw the star with the telescope.

Selectional restrictions by themselves are thus inadequate for disambiguation. Extracting selectional restrictions automatically from LDOCE codes also introduces problems; (7) would be ruled out, for example, because *scare* is specified as having a human object:

(7) The dog scared the rabbit.

Selectional restrictions which are powerful enough to be really useful in disambiguation correspond to default rather than necessary properties of the predicate. It does not seem that there is any good way to treat selectional preferences within a strictly unification based framework. It is sometimes suggested that defaults could be used. However, simply allowing selectional restrictions to be overridden achieves nothing, unless there is some mechanism for flagging the fact that the default was overridden. Furthermore, it is necessary to distinguish between different readings on the basis of the 'seriousness' of the violation of the selectional restrictions. I know of no way to build such functionality into a strictly constraint based approach, even one augmented with the sort of defaults assumed here. It seems more appropriate to regard disambiguation as a distinct process of ordering interpretations produced by the constraint based grammar.

As I described in section 1.3.1, a distinction can be made between selectional restrictions which can be accounted for entirely in terms of the denotation of the predicate, and collocational restrictions which have a strictly lexical dimension. Although there is an intuitive distinction between sentences where selectional restrictions are broken, which seem anomalous, and sentences which happen to be untrue, it has not been shown that it is necessary to attempt an account of this effect which involves explicit representation of selectional restrictions. We might suppose that a sentence which violates selectional restrictions will be obviously untrue, at least to adult native hearers, and furthermore they will know that the speaker knows it to be untrue, and so on. Under this assumption it is actually utterances which may be anomalous rather than sentences. Thus (8a) might be anomalous rather than simply false when uttered by one chemist to another, and (8b) would not be anomalous when uttered by a five year old child, since it is not clearly false to the speaker.

(8) a The atomic number of hydrogen is 2.
b The can-opener bit me.

To make such an account work, it is necessary to have a means of distinguishing between sentences which are 'obviously' untrue, and those which require extensive inference to determine their falsity, but, though lexical selectional restrictions could be regarded as encoding this obvious information, it is not clear that this is the best approach, at least given a strictly constraint-based system, since this gives no mechanism for recovering a non-literal or extended interpretation for the utterance.

Approaches to sense disambiguation within NLP, such as those described by Hirst (1987) and Dahlgren (1988), make use of a combined approach which integrates information about syntax, selectional restrictions and wider general world knowledge. These systems cannot be simply transferred into a unification based framework, because the information sources are not just combined together, but are accessed sequentially, in a way which depends on the result of previous processing, and may involve weighting possibilities. Unlikely readings are not blocked altogether, but dispreferred if another interpretation is available. For example, in Dahlgren's system, which deals with lexical ambiguity, a table of 'fixed and frequent phrases' is consulted first, followed by syntactic tests, followed by commonsense reasoning (which includes the use of selectional restrictions). Fixed phrases alone eliminate a high proportion of the ambiguity.

The only conclusion about selectional restrictions which I think can be drawn, is that there is little point in discussing them in a primarily lexical context rather than as part of a general model of language processing, and that the use of selectional restrictions for disambiguation does not fit into a constraint-based framework in a natural way. For current purposes I will make the assumption that there is some notion of 'core meaning' and that with respect to this selectional restrictions can be taken as obligatory. I will use this to illustrate the sense extension mechanism which I will discuss in section 4.3, since sense extension rules which give rise to metonymic or metaphorical interpretations will sometimes be triggered by violation of selectional restrictions. Derivational rules, such as that describing suffixation with *er*, will also make use of selectional restrictions on the verb to predict the semantic characterisation of the noun. I will make use of distinctions which are encoded in the type system for other reasons, to avoid creating an orthogonal set of primitives which would essentially be arbitrary. I will assume therefore that the verb may specify the lexical semantic structure of its arguments directly.

4.2 Logical metonymy

In section 1.2 I introduced Pustejovsky's concept of logical metonymy which offers an account of the default interpretation of sentences such as (9a) as (9b).

(9) a Mary enjoyed the book.
b Mary enjoyed reading the book.

The conventional approach would be to give separate lexical entries for the various subcategorisation patterns of *enjoy* and to relate these by meaning postulates. But this is unsatisfactory, since it does not capture the general behaviour of this class of verbs, which includes *begin* and *finish*, for example, as well as *enjoy*, and it does not explain why *enjoy* does not seem to be lexically ambiguous. Sentences such as (10) are quite natural, but under quite generally accepted assumptions about lexical ambiguity (Zwicky and Sadock, 1975; Cruse, 1986) only one sense of *enjoy* can be involved here.

(10) Mary enjoys books, television and playing the guitar.

So it is preferable to assume that there is a single lexical entry which obligatorily takes an event as argument, and that noun phrases such as *the book* are coerced into denoting events in the context of verbs such as *enjoy*.

In Pustejovsky's account the basic coercion essentially introduces an underspecified predicate, relating the object to the enjoying event, but this is further specified by the noun, to be an event characteristically associated with it, at least in unmarked examples. Briscoe *et al.* (1990) give corpus data to support the idea that a default interpretation holds in such sentences. Logical metonymies were found to occur quite regularly with predicates such as *enjoy*, *start*, *finish* and *begin*, and default inference based on lexically specified qualia structure could account for the interpretation of most of them. Where lexical recovery of the predicate would fail, the data seemed to support the claim that the context was sufficiently rich that pragmatic inference would be quite directed and constrained. Furthermore, there was some evidence that VP complementation was chosen when default lexical recovery of the predicate would lead to the wrong interpretation. This data therefore gives some support to the predictions made by Pustejovsky's theory, although it is hardly conclusive.

Pustejovsky gives a more detailed account of logical metonymy in terms of type coercion in Pustejovsky (1989), where he relates type coercion to various type shifting operators proposed by, for example, Partee and Rooth (1983), but he does not show exactly how his proposal would work. In particular, Pustejovsky assumes that the type coercion is lexically specified on the verb, but this lexical specification should presumably be absent when the verb combines with a VP complement. In Briscoe *et al.* (1990) we showed an implemented alternative formalisation of logical metonymy in an untyped feature structure language. Essentially the same approach can be described using LAUREL.

Our description relies on the use of unary rules to perform the type shifting and to turn an object denoting NP into one denoting an event associated with that object. For example, the lexical entry for *book* will have the telic role in its qualia structure instantiated by the semantics for *read.* The NP *the book* shown in Figure 4.10 will have the same qualia structure. This cannot be a complement to *enjoy*, shown in Figure 4.11, which is specified as taking an obligatory event argument with a categorial specification appropriate to a verb. The NP *the book* is coerced to an event denoting sign, basically equivalent to *reading the book*, by the unary rule shown in Figure 4.12.[3] However the nature of the event is indicated to be defeasible by the operator, M. This is intended to be paraphrased as 'assume P unless there is evidence to the contrary' and could be taken as equivalent to the M operator in McDermott and Doyle (1980). However, other non-monotonic formalisations are also possible. The semantics for the sign for *enjoy the book* is shown in Figure 4.13. Note that this analysis predicts correctly that the control behaviour of *enjoy* with a type shifted complement exactly parallels that in the normal VP complement case.

Thus, as mentioned earlier, the interpretation of the sentence as involving a particular type of event is defeasible. To repeat an earlier example, if Mary is a goat, we assume that pragmatic effects account for a reinterpretation of (11a) as (11b).

(11) a Mary enjoyed the book.
b Mary enjoyed eating the book.

Note that the default lexical interpretation here could actually be blocked by encoding selectional restrictions on arguments. However, there are other cases where the interpretation of the event as the telic role is blocked, where the blocking itself has to be assumed to be (at least partly) due to pragmatic processing. For example:

(12) ? John enjoyed the book with blank pages.

will not be predicted to be odd on the basis of the purely lexical default which will be instantiated to *read* in the usual way; under the account described here *the book with blank pages* essentially inherits the qualia structure of *book.* In such examples pragmatic inference has to be assumed to be involved in blocking the default, because blocking inheritance of the telic role might involve extensive inference. This account becomes rather more plausible on an assumption of incremental interpretation, especially if pragmatic processing is sufficiently tightly coupled that it is not the sentence interpretation that is overridden, but the inheritance of the telic role *read* by the NP *the book with no pages.* This is consistent with the assumption that the representation of the telic role in the qualia structure is a way of foregrounding real world knowledge so that it is available for linguistic processing, and in this case it is clearly an oversimplification to assume that the qualia structure of the NP is inherited from the head noun. However it is an oversimplification that I will continue to adopt.[4]

[3] The unary rule actually works by instantiating the active slot in the type-raised NP with a verb sign constructed using the semantics of the telic role. The instantiated result is then the output of the lexical rule. The process can thus be regarded as an instantiation rather than a coercion.

[4] I think it is worth noticing the parallels between this, and Pollard and Sag's (in press) comments about the accumulation of appropriacy conditions. Their tentative 'Principle of Contextual Consistency' requires that the contextual assumptions associated with part of an utterance are inherited by the utterance as a whole, and, as they describe, this does not account for the behaviour of presuppositions.

$$
\left[\begin{array}{l}
\textbf{nominal-sign} \\
\text{ORTH} = \left[\begin{array}{l} \textbf{complex-orth} \\ \text{ORTH1} = \textbf{the} \\ \text{ORTH2} = \textbf{book} \end{array}\right] \\
\text{CAT} = \left[\begin{array}{l}
\textbf{raised-np-cat} \\
\text{RESULT} = \boxed{0}\ \boxed{\textbf{sign}} \\
\text{DIRECTION} = \boxed{1}\ \textbf{direction} \\
\text{ACTIVE} = \left[\begin{array}{l}
\textbf{sign} \\
\text{ORTH} = \textbf{orth} \\
\text{CAT} = \left[\begin{array}{l}
\textbf{complex-cat} \\
\text{RESULT} = \boxed{0} \\
\text{DIRECTION} = \boxed{1} \\
\text{ACTIVE} = \left[\begin{array}{l}
\textbf{nominal-sign} \\
\text{ORTH} = \textbf{orth} \\
\text{CAT} = \boxed{\textbf{basic-nominal-cat}} \\
\text{SEM} = \boxed{\textbf{binary-formula}} \\
\text{QUALIA} = \boxed{2}
\end{array}\right]
\end{array}\right] \\
\text{SEM} = \boxed{5}\ \boxed{\textbf{formula}}
\end{array}\right]
\end{array}\right] \\
\text{SEM} = \left[\begin{array}{l}
\textbf{binary-formula} \\
\text{IND} = \boxed{6}\ \textbf{entity} \\
\text{PRED} = \textbf{the\_1} \\
\text{ARG1} = \left[\begin{array}{l}
\textbf{obj-noun-formula} \\
\text{IND} = \boxed{4} \\
\text{PRED} = \textbf{book\_L\_1\_1} \\
\text{ARG1} = \boxed{4} \\
\text{PLMOD} = \textbf{false} \\
\text{QUANT} = \textbf{true}
\end{array}\right] \\
\text{ARG2} = \boxed{5}
\end{array}\right] \\
\text{QUALIA} = \boxed{2}\ \left[\begin{array}{l}
\textbf{artifact\_physical} \\
\text{TELIC} = \left[\begin{array}{l}
\textbf{strict-trans-sem} \\
\text{IND} = \boxed{3}\ \textbf{eve} \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \left[\begin{array}{l}
\textbf{verb-formula} \\
\text{IND} = \boxed{3} \\
\text{PRED} = \textbf{read\_1} \\
\text{ARG1} = \boxed{3}
\end{array}\right] \\
\text{ARG2} = \left[\begin{array}{l}
\textbf{binary-formula} \\
\text{IND} = \boxed{3} \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \left[\begin{array}{l}
\textbf{p-agt-formula} \\
\text{IND} = \boxed{3} \\
\text{PRED} = \textbf{p-agt} \\
\text{ARG1} = \boxed{3} \\
\text{ARG2} = \boxed{\textbf{obj}}
\end{array}\right] \\
\text{ARG2} = \left[\begin{array}{l}
\textbf{p-pat-formula} \\
\text{IND} = \boxed{3} \\
\text{PRED} = \textbf{p-pat} \\
\text{ARG1} = \boxed{3} \\
\text{ARG2} = \boxed{4}\ \boxed{\textbf{obj}}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\text{OBJECT-INDEX} = \boxed{4}
\end{array}\right]
\end{array}\right]
$$

SEM: $[a][\textbf{the}(\textbf{book\_L\_1\_1}(y), [a]A)]$

QUALIA : TELIC: $[e][\textbf{read\_1}(e) \wedge \textbf{p-agt}(e,x) \wedge \textbf{p-pat}(e,y)]$

Figure 4.10: Sign for *the book*. The argument of **book_L_1_1** and the second argument of **p-pat** are coindexed. This is set up in the type system by specifying that the proto-patient in the telic role of an **artifact** is coindexed with the OBJECT-INDEX in the qualia structure. The value of OBJECT-INDEX is coindexed with the value of SEM : IND for nouns of all qualia types.

$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{subj-equi-intrans-vping-sign}}\\
\text{ORTH} = & \textbf{enjoy}\\
\text{CAT} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{intrans-vping-control-cat}}\\
\text{RESULT} = & \boxed{\textbf{sign}}\\
\text{DIRECTION} = & \textbf{forward}\\
\text{ACTIVE} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{sign}}\\
\text{ORTH} = & \textbf{orth}\\
\text{CAT} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{strict-intrans-cat}}\\
\text{RESULT} = & \boxed{\textbf{sent-sign}}\\
\text{DIRECTION} = & \textbf{forward}\\
\text{ACTIVE} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{np-sign}}\\
\text{ORTH} = & \textbf{orth}\\
\text{CAT} = & \boxed{\textbf{np-cat}}\\
\text{SEM} = & \left[\begin{array}{l}
\textbf{theta-formula}\\
\text{IND} = \boxed{\scriptstyle 0}\ \textbf{eve}\\
\text{PRED} = \textbf{theta-relation}\\
\text{ARG1} = \boxed{\scriptstyle 0}\\
\text{ARG2} = \boxed{\scriptstyle 1}\ \boxed{\textbf{obj}}
\end{array}\right]\\
\text{QUALIA} = & \boxed{\textbf{nomqualia}}
\end{array}\right]
\end{array}\right]\\
\text{SEM} = & \boxed{\scriptstyle 2}\left[\begin{array}{l}
\textbf{formula}\\
\text{IND} = \textbf{eve}\\
\text{PRED} = \textbf{logical-pred}\\
\text{ARG1} = \textbf{sem}
\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{p-agt-subj-intrans-xcomp/comp-sem}}\\
\text{IND} = & \boxed{\scriptstyle 3}\ \textbf{eve}\\
\text{PRED} = & \textbf{and}\\
\text{ARG1} = & \left[\begin{array}{l}
\textbf{verb-formula}\\
\text{IND} = \boxed{\scriptstyle 3}\\
\text{PRED} = \textbf{enjoy\_1}\\
\text{ARG1} = \boxed{\scriptstyle 3}
\end{array}\right]\\
\text{ARG2} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{binary-formula}}\\
\text{IND} = & \boxed{\scriptstyle 3}\\
\text{PRED} = & \textbf{and}\\
\text{ARG1} = & \boxed{\scriptstyle 5}\left[\begin{array}{l}
\textbf{p-agt-formula}\\
\text{IND} = \boxed{\scriptstyle 3}\\
\text{PRED} = \textbf{p-agt}\\
\text{ARG1} = \boxed{\scriptstyle 3}\\
\text{ARG2} = \boxed{\scriptstyle 1}
\end{array}\right]\\
\text{ARG2} = & \boxed{\scriptstyle 2}
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

$$[e][\textbf{enjoy\_1}(e) \wedge \textbf{p-agt}(e,x) \wedge \text{``enjoyed''}(e,e')]$$

Figure 4.11: Lexical entry for *enjoy* (as mentioned in the previous section, the logical form is incomplete).

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{unary-rule}} \\
0 = \boxed{0} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{sign}} \\
\mathrm{ORTH} = & \boxed{1}\ \textbf{orth} \\
\mathrm{CAT} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{strict-intrans-cat}} \\
\mathrm{RESULT} = & \boxed{\textbf{sent-sign}} \\
\mathrm{DIRECTION} = & \textbf{forward} \\
\mathrm{ACTIVE} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{np-sign}} \\
\mathrm{ORTH} = & \textbf{orth} \\
\mathrm{CAT} = & \boxed{\textbf{np-cat}} \\
\mathrm{SEM} = & \boxed{2}\ \boxed{\textbf{theta-formula}} \\
\mathrm{QUALIA} = & \boxed{\textbf{nomqualia}}
\end{array}\right]
\end{array}\right] \\
\mathrm{SEM} = & \boxed{3}\ \textbf{sem}
\end{array}\right] \\
1 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{nominal-sign}} \\
\mathrm{ORTH} = & \boxed{1} \\
\mathrm{CAT} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{raised-np-cat}} \\
\mathrm{RESULT} = & \boxed{0} \\
\mathrm{DIRECTION} = & \boxed{4}\ \textbf{direction} \\
\mathrm{ACTIVE} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{sign}} \\
\mathrm{ORTH} = & \textbf{orth} \\
\mathrm{CAT} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{complex-cat}} \\
\mathrm{RESULT} = & \boxed{0} \\
\mathrm{DIRECTION} = & \boxed{4} \\
\mathrm{ACTIVE} = & \boxed{\textbf{sign}}
\end{array}\right] \\
\mathrm{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{binary-formula}} \\
\mathrm{IND} = & \boxed{5}\ \textbf{eve} \\
\mathrm{PRED} = & \boxed{6}\ \textbf{and} \\
\mathrm{ARG1} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{formula}} \\
\mathrm{IND} = & \boxed{5} \\
\mathrm{PRED} = & \mathbf{M} \\
\mathrm{ARG1} = & \boxed{7}\ \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{verb-formula}} \\
\mathrm{IND} = & \boxed{5} \\
\mathrm{PRED} = & \textbf{log.} \\
\mathrm{ARG1} = & \boxed{5}
\end{array}\right]
\end{array}\right] \\
\mathrm{ARG2} = & \boxed{8}\ \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{formula}} \\
\mathrm{IND} = & \boxed{5} \\
\mathrm{PRED} = & \textbf{log.} \\
\mathrm{ARG1} = & \boxed{2}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\mathrm{SEM} = & \boxed{3} \\
\mathrm{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{nomqualia}} \\
\mathrm{AGENTIVE} = & \textbf{nomagent} \\
\mathrm{TELIC} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{verb-sem}} \\
\mathrm{IND} = & \boxed{5} \\
\mathrm{PRED} = & \boxed{6} \\
\mathrm{ARG1} = & \boxed{7} \\
\mathrm{ARG2} = & \boxed{8}
\end{array}\right] \\
\mathrm{FORM} = & \boxed{\textbf{nomform}} \\
\mathrm{CONSTITUENCY} = & \boxed{\textbf{nomconst}} \\
\mathrm{OBJECT\text{-}INDEX} = & \textbf{entity}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

Figure 4.12: Unary rule for object to event coercion (**log.** is an abbreviation of **logical-pred**).

$$\left[\begin{array}{l}
\textbf{p-agt-subj-intrans-xcomp/comp-sem}\\
\text{IND} = \boxed{0}\ \textbf{eve}\\
\text{PRED} = \textbf{and}\\
\text{ARG1} = \left[\begin{array}{l}\textbf{verb-formula}\\ \text{IND} = \boxed{0}\\ \text{PRED} = \textbf{enjoy\_1}\\ \text{ARG1} = \boxed{0}\end{array}\right]\\
\text{ARG2} = \left[\begin{array}{l}
\textbf{binary-formula}\\
\text{IND} = \boxed{0}\\
\text{PRED} = \textbf{and}\\
\text{ARG1} = \boxed{1}\left[\begin{array}{l}\textbf{p-agt-formula}\\ \text{IND} = \boxed{0}\\ \text{PRED} = \textbf{p-agt}\\ \text{ARG1} = \boxed{0}\\ \text{ARG2} = \boxed{2}\ \boxed{\textbf{obj}}\end{array}\right]\\
\text{ARG2} = \left[\begin{array}{l}
\textbf{binary-formula}\\
\text{IND} = \boxed{3}\ \textbf{eve}\\
\text{PRED} = \textbf{the\_1}\\
\text{ARG1} = \left[\begin{array}{l}\textbf{obj-noun-formula}\\ \text{IND} = \boxed{4}\ \boxed{\textbf{obj}}\\ \text{PRED} = \textbf{book\_L\_1\_1}\\ \text{ARG1} = \boxed{4}\\ \text{PLMOD} = \textbf{false}\\ \text{QUANT} = \textbf{true}\end{array}\right]\\
\text{ARG2} = \left[\begin{array}{l}
\textbf{binary-formula}\\
\text{IND} = \boxed{3}\\
\text{PRED} = \textbf{and}\\
\text{ARG1} = \left[\begin{array}{l}\textbf{formula}\\ \text{IND} = \boxed{3}\\ \text{PRED} = \textbf{M}\\ \text{ARG1} = \left[\begin{array}{l}\textbf{verb-formula}\\ \text{IND} = \boxed{3}\\ \text{PRED} = \textbf{read\_1}\\ \text{ARG1} = \boxed{3}\end{array}\right]\end{array}\right]\\
\text{ARG2} = \left[\begin{array}{l}
\textbf{binary-formula}\\
\text{IND} = \boxed{3}\\
\text{PRED} = \textbf{and}\\
\text{ARG1} = \left[\begin{array}{l}\textbf{p-agt-formula}\\ \text{IND} = \boxed{3}\\ \text{PRED} = \textbf{p-agt}\\ \text{ARG1} = \boxed{3}\\ \text{ARG2} = \boxed{2}\end{array}\right]\\
\text{ARG2} = \left[\begin{array}{l}\textbf{p-pat-formula}\\ \text{IND} = \boxed{3}\\ \text{PRED} = \textbf{p-pat}\\ \text{ARG1} = \boxed{3}\\ \text{ARG2} = \boxed{4}\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

$$[e][\mathbf{enjoy\_1}(e) \wedge \mathbf{p\text{-}agt}(e,x) \wedge \text{“enjoyed”}(e,e') \wedge$$
$$[e']\mathbf{the}[(\mathbf{book\_L\_1\_1}(y), M\mathbf{read\_1}(e') \wedge \mathbf{p\text{-}agt}(e',x) \wedge \mathbf{p\text{-}pat}(e',y))]]$$

Figure 4.13: Semantics for *enjoy the book*

Pustejovsky also proposes that qualia structure allows variations in the interpretation of adjectival modification to be explained. For example *long book* can mean a book which is physically long (modification of the formal role) or one which takes a long time to read (modification of the telic role). In Briscoe *et al.* (1990) we observed that it was necessary to copy the telic role of the qualia structure (or to be more precise, the variables indicating the event and the agent within the telic role) rather than to simply unify it. Unless this is done, the interpretation of (13a) will be (13b) where the enjoyed event is stated to be long, whereas it should be something like (13c).[5]

(13) a John enjoyed the long book.

b $\exists e, e', x, y[\text{enjoy}(e, j, e') \wedge \text{read}(e', j, y) \wedge \text{book}(y) \wedge \text{long}(e')]$

c $\exists e, e', e'', x, y[\text{enjoy}(e, j, e') \wedge \text{read}(e', j, y) \wedge \text{book}(y) \wedge \text{long}(e'') \wedge \text{read}(e'', x, y)]$

Under the current framework, we have most of the formal machinery to specify that the telic role is inherited (and thus copied) when extracted into the logical form, although a modification to the description language would be required and the specification would have to be made on a case by case basis. What is rather more interesting is to examine why such a step is needed. There is an essential level of inheritance in the usual paradigmatic/syntagmatic interface, in that syntagmatic combination involves instances of lexical entries, not the lexical entries themselves. However in the case of the telic role, an individual entity has a class of (potential) events associated with it, as described by the role, and we only start referring to an individual event when particular processes, such as logical metonymy, result in the telic role participating in the construction of the logical form. For example, the telic role associated with a particular instance of *book* can be assumed to refer to a set of events of reading it. Not all of these events should be assumed to be long or enjoyable, if the sentence interpretation happens to involve the telic role, any more than all books should be taken to be green if a particular sentence refers to a green book. I will not try to treat this problem here, since it seems to me that it would be most naturally resolved in conjunction with a more motivated approach to syntagmatic inheritance of qualia structure.

A problem with the analysis of the type shift as mediated by a unary rule is that the same NP can also act as a complement to a verb where no type shift is involved, as for example in:

(14) John ate and enjoyed the salmon.

The unary rule account predicts that the NP should be systematically ambiguous, whereas it appears in fact to be vague, since (14) seems perfectly acceptable. This problem is discussed in more detail in Copestake and Briscoe (1991). However, as mentioned earlier, an account where a distinct entry for *enjoy* is generated by lexical rule, also seems implausible. There is another option within the current formalism, which is perhaps closer to Pustejovsky's original proposal that the lexical entry for *enjoy* specifies the coercion. The basic idea here is to underspecify the lexical entry for enjoy, so that the value to be instantiated by the complement can be specified to be either a VPing or to an NP. In the latter case the semantics are arranged so that the same coercion happens as with the unary rule. If the type system is set up so that the particular complement to *enjoy* forces specification to one or other of these possibilities (e.g. because the specification of the feature QUALIA on the complement forces classification) then we have the situation envisaged by Pustejovsky where the type shift is directly forced by complementation. A sketch of this account is shown in Figure 4.14, where we assume that the category type of *enjoy* is set to **complex-or-coerc-cat**. The trick here is to specify an extra feature on the category, COERC, to allow for the possibility of coercion. Note that the relationship between the input NP, which will instantiate the value of ACTIVE, and the structure actually unified with the verb semantics, which will be part of the value of COERC, can be made exactly equivalent to the corresponding relationship in the unary rule.

One reason for preferring an account where the type-shifting is not tied to the lexical entry for *enjoy* and other verbs, is the general possibility of the interpretation of the NP as an event, in

[5] Actually even (13c) is incorrect, because it specifies the existence of a long event, but a long book will not necessarily have been read. This is not particularly important here, however.

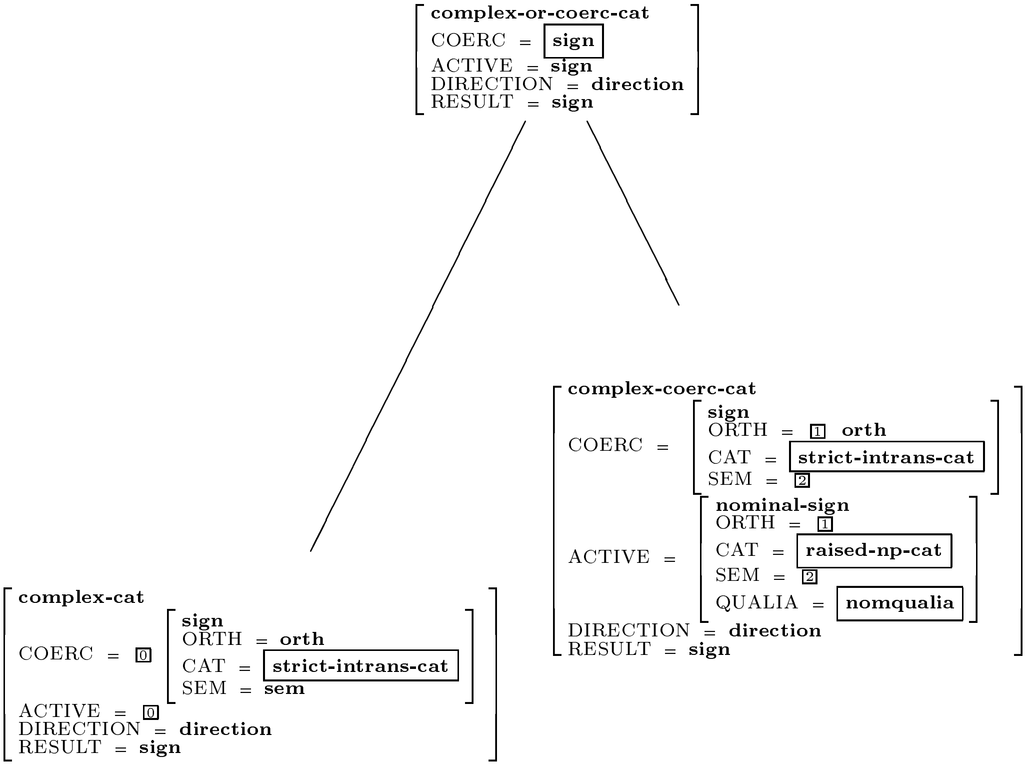


Figure 4.14: Object to event coercion within the verb entry.

examples such as (15a) which is naturally interpreted as (15b).

(15) a After eight glasses of champagne John became slightly tipsy.
b After drinking eight glasses of champagne John became slightly tipsy.

I will assume that the important point here is to characterise the type shift, and from this point of view whether it is internalised in the way outlined above or not is a secondary consideration.

There is much that is tentative about the above account, in particular the syntagmatic inheritance of qualia structure and the interaction with pragmatics. Furthermore it does not show that a purely pragmatic account of interpretation of the metonymic event is wrong. However, our formalisation of sense extensions, considered in the next section, does provide some independent support for the general applicability of qualia structure and type shifting in metonymic interpretation.

4.3 Sense extensions

Much of this section describes work which I carried out in collaboration with Ted Briscoe and is a revised and expanded version of material presented in two papers; Copestake and Briscoe (1991) and Briscoe and Copestake (1991).[6] There we argued that the LAUREL concept of a (lexical) rule is expressive enough to cover both derivational morphological processes and metonymic and metaphoric sense extensions, and that the application of such rules is productive within finely specified subsets of the lexicon, identifiable via the type system. Formulation of these processes in a uniform framework uncovers similarities between them; such as the blocking of metonymic sense extension processes in a fashion similar to the blocking of derivational morphological ones.

Here I will mainly concentrate on the lexical representation issues involved, although I will also discuss the arguments for taking our approach over one where it is assumed that metaphorical and metonymic sense extensions are treated purely pragmatically. I will also discuss the relationship between the relatively conventionalised processes considered in our earlier papers and the very general phenomenon of *deferred reference* described by Nunberg (1978). It will be seen that the formal framework emerging from this is compatible with the use of type-shifting rules discussed in the previous section, and that there are good reasons for generalising from the concept of lexical rule to a concept of a sense extension rule which may apply to non-lexical signs.

4.3.1 Sense extensions and derivational morphology

Linguistic research has emphasised the phonological or graphological consequences of derivational processes to the extent that conversion or zero-derivation has often been treated in terms of 'zero-morphemes' and 'zero-affixation'. However, here I will concentrate on the syntactic and semantic consequences of such processes and argue that all of them can be treated as mappings between lexical entries expressed as typed feature structures. Affixation requires changes to the phonological or orthographic part of the lexical entry, but otherwise there are considerable similarities between the processes of sense extension and more clearly derivational ones.

A standard example of metonymic sense extension is the use of a word denoting a place to refer to (some of) the people inhabiting that place (e.g. *village*, *palace*). This process seems to involve the foregrounding of one component of the meaning of the place denoting word. In the framework assumed here, the qualia structure for such nouns will contain information to the effect that they are inhabited. An example of a derivational morphological process is the addition of the *er* suffix to verbs, typically creating a noun denoting the agent of the action denoted by the verb (e.g. *teach*, *teacher*). There are several apparent differences between this type of process and the metonymic sense extension. The derivational rule involves a change of syntactic class, it affects the argument structure of the derived predicate, it involves affixation, and although there is a foregrounding of one aspect of the verb meaning, the result would not traditionally be described as a metonymic usage. Nevertheless, there are clearly derivational processes which do not affect syntactic class (e.g.

[6] Sections 4.3.2,4.3.4,4.3.5 and 4.3.6 are largely new.

re-program, *un-reprogrammable*) and sense extensions which do; for example, countability of nouns changes depending on whether they are interpreted as kinds, substances or portions (e.g. *There was beer all over the table*, *John drank a beer*). Both syntactic class and argument structure are affected in sense extensions which characterise the relationship between artifacts and verbs denoting events characteristic of their use (e.g. *sugar*, *hammer*). Finally, processes of conversion and derivation can be identical; for example, both *purchase* and *build* have deverbal nominal forms *purchase* and *building*, both nouns can denote the action involved and take appropriate complements (*Bill's purchase of his new car*, *Bill's building of the house*), and both can denote the result of the action (*Bill's purchases were many and varied*, *Bill's buildings were admired*).

There are other similarities between sense extension and derivational morphology; clearly, productivity is an issue in both, and in particular, sense extension processes may apparently be blocked (preempted by synonymy), in a way comparable to the situation in derivational morphology (see e.g. Bauer 1983:87f). For example, the regular form *stealer* does not generally occur, apparently because of the availability of *thief.* Another productive metonymic sense extension is that of animal denoting (count) nouns to (mass) nouns denoting their meat (e.g. *lamb*), but this process too is blocked by the presence of a synonymous lexeme with different form (*pig*, *pork*). Nevertheless in many of these cases the sense extension may still apply in a more marked fashion. For example, using *pig* instead of *pork* to mean the meat is possible but not established, and its use is marked, perhaps suggesting that the meat is of very inferior quality or otherwise objectionable to the speaker. Similarly morphologically blocked forms are not completely ruled out; "Stealer of Souls" is the title of a fantasy novel; here the use of *stealer* seems more appropriate than *thief* because of the alliteration and the slightly archaic effect which is desirable in the genre. In general, we will specify both sense extension processes and derivational ones as fully productive, and formalise them both in terms of rules applying to finely specified subsets of the lexicon, which are defined in terms of both syntactic and semantic properties expressed in the type system. We can account for blocking systematically by checking for syntactic and semantic identity between the result of rule application and an existing lexical entry, rather than explicitly specifying that the rule is not applicable to a particular lexical item.

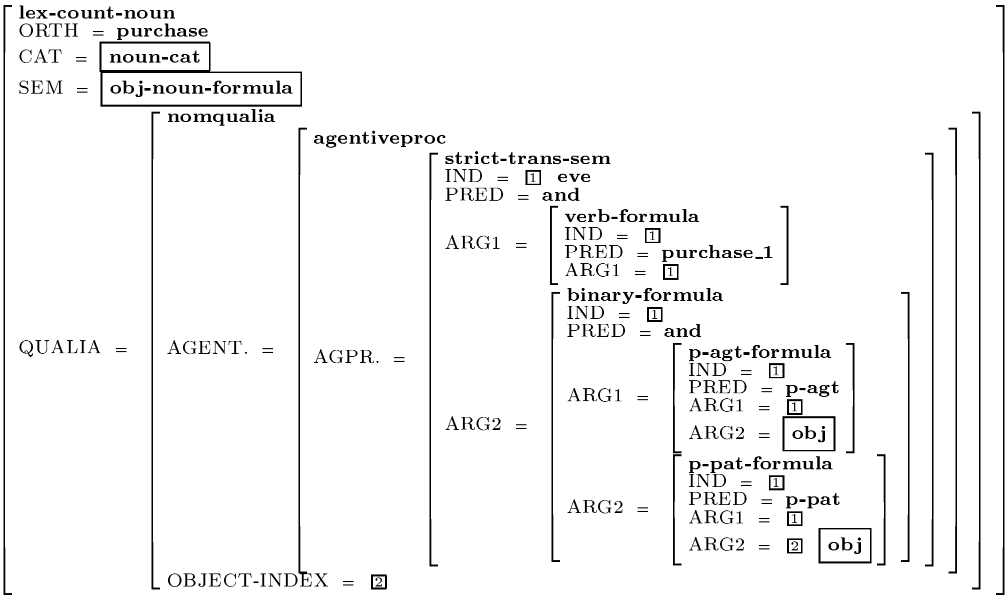
We also treat metaphoric sense extension as lexical rules. Although, for example, the sense extension from animals into metaphorical senses denoting humans with some particular characteristic is apparently productive (e.g. *John is a wombat*), the actual characteristics involved cannot be predicted from knowledge of the animal sense. We would argue, for example, that the properties ascribed to a person by *pig* are stereotypical associations with the animal, which would not be encoded in the qualia structure. In the case of *wombat* the association of foolishness is as likely to derive from the phonological form of the word as from beliefs about the animal[7]. Despite the more associative or analogical nature of metaphorical sense extension, we would argue that there is a core component to such processes which should be expressed in terms of a lexical rule. In general, we assume that the possible mappings defined by lexical rules define the limits to the possible shifts in meaning, but more general reasoning may be involved in determining the meaning more exactly in a particular context.

Some lexicalised items can be naturally represented as equivalent to the result of a productive process, with some specialisation or modification to the orthography, syntax or semantics. For example, rather than regarding the noun *beggar* as completely distinct from the verb *beg* in the lexicon, it is preferable to treat it as equivalent to the result of applying the *er* suffix to the verb, but with a distinct value for orthography. The LAUREL default inheritance mechanism provides a natural means for accomplishing this.

4.3.2 Nominalisation

In this section I will briefly sketch how morphological processes can be formalised in LAUREL, using resultative nominals as an example. I will concentrate on the instantiation of the qualia structure in particular, and ignore most of the other issues.

[7] Compare *womble* (a fat furry creature from a series of televised children's stories) *wobble* and *wamble* (a dialect word meaning to move unsteadily).



QUALIA : AGENTIVE : AGPROCESS $[e][\mathbf{purchase\_1}(e) \wedge \mathbf{p\text{-}agt}(e,x) \wedge \mathbf{p\text{-}pat}(e,y)]$
QUALIA : OBJECT-INDEX: y

Figure 4.15: Resultative for *purchase*

Resultative deverbal nominals denote entities which can be considered as the result of some event of the kind denoted by the base verb. Resultatives can be subdivided according to their relationship with the verb. In the examples I will consider here, the resultative relates to the object of the verb. For example, an entity which is the object of a purchasing event can be referred to as a *purchase* (at least after the event has occurred). The resultative *purchase*, shown in Figure 4.15, can be derived from the verb by the following unary rule:

```
objresultative
lexical-rule
   < 1 > = strict-trans-sign
   < 0 > = lex-noun-sign
   < 0 : QUALIA > = < 1 : ACTIVE : QUALIA >
   < 0 : QUALIA : AGENTIVE : AGPROCESS > = < 1 : SEM >
   < 0 : QUALIA : AGENTIVE : AGPROCESS : ARG2 : ARG2 : ARG2 >
         = < 0 : QUALIA : OBJECT-INDEX >.
```

The morphological process which produces the resultative involves the agentive role of the noun being instantiated with the semantics of the corresponding verb, with the resultative noun in object position.

Although I have formalised this in terms of a unary rule, the rule can be internalised to produce an account of derivation by categorial combination of the verb with a bound morpheme. For example, nouns formed by combination of a verb with the morpheme *ing*, often denote the result of the action described by the verb, which in many cases can be identified with the object (e.g. *painting*, *washing*, *building*). This use of the suffix could be specified as shown below, where the relationship between the feature structures introduced by `< CAT : ACTIVE >` and `< CAT : RESULT >` is precisely the same as the unary rule.

```
ing 1
   < CAT : ACTIVE > = strict-trans-sign
   < CAT : RESULT > = lex-noun-sign
< CAT : DIRECTION > = backward
< CAT : RESULT : QUALIA : AGENTIVE : AGPROCESS >
     =  < CAT : ACTIVE : SEM>
< CAT : RESULT : QUALIA : AGENTIVE : AGPROCESS : ARG2 : ARG2 : ARG2 >
      = < CAT : RESULT : QUALIA : OBJECT-INDEX >.
```

Indeed, if we used the features 1 and 0 instead of ACTIVE and RESULT in categorial specifications, we could simply specify *ing* as inheriting its CAT value from the unary rule. So, even if we mix internalised and externalised lexical rules, we could use the same type system to specify them both. To keep things relatively simple, I will assume the unary rule treatment, which allows a uniform account without having to postulate 'zero-morphemes' for sense extension.

The instantiation of the qualia structure in this way may be used to provide a partial account for the restricted possibilities for the interpretation of the possessive in resultatives. In the case of event denoting nominals, there seem to be good reasons for assuming that the noun has argument structure and that the possessive can be interpreted with respect to this (see Grimshaw, 1990). For example, (16a) refers to an event in which the city was destroyed, (16b) to an event where the soldiers destroyed the city:

(16) a The city's destruction
b The soldiers' destruction of the city

However, resultative nominals which denote physical objects rather than events may also give rise to very restricted interpretations of possessives — in this case as arguments to predicates denoting the events of which they are the result. For example, in (17), John must have been the purchaser of the objects, but is not necessarily their current owner.

(17) John's purchases were on the table.

Suppose John bought some books and gave them to Mary who put them on the bookcase. In this situation (18a) and (18b) are true but (18c) is not.

(18) a John's purchases are on the bookcase.
b Mary's books are on the bookcase.
c Mary's purchases are on the bookcase.

This sort of example poses problems for any attempt at treating the interpretation of the predicate as uniformly underspecified and instantiating it by reasoning with real world knowledge. Clearly, the interpretation of the possessive depends not on the objects denoted but on the predicate with which they were described. However, qualia structure can be used to account for the interpretation since the possessor may be interpreted as filling the agent role in the agentive process. Such interpretation of the possessive is possible for non-derived nouns; for example *John's book* may denote a book owned by John or one written by him. In general, we assume that there are a range of possible interpretations for the possessive, but that the agentive role interpretation blocks the more general ownership interpretation for resultatives like *purchase.* Pragmatic factors are obviously going to be involved in determining the particular interpretation in context, but the lexical semantic structure provides a framework for such interpretation.

This account can be extended to other processes of nominalisation. Nominals formed with *er* are assumed to have their telic roles instantiated rather than their agentive roles, and may either denote the agent or the instrument of some event. The feature structure associated with the result of applying agentive *er* nominalisation to *teach* is shown in Figure 4.16. Just as with resultatives, the interpretation of the possessive may be determined by the qualia structure — with agent denoting nominals, the possessor is normally interpreted as the object of the associated event.

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-noun-sign}}\\
\text{ORTH} = & \left[\begin{array}{l}\textbf{complex-orth}\\ \text{ORTH1} = \textbf{teach}\\ \text{ORTH2} = \textbf{er}\end{array}\right]\\
\text{CAT} = & \boxed{\textbf{noun-cat}}\\
\text{SEM} = & \boxed{\textbf{obj-noun-formula}}\\
\text{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{human}}\\
\text{TELIC} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{strict-trans-sem}}\\
\multicolumn{2}{l}{\text{IND} = \boxed{1}\ \textbf{eve}}\\
\multicolumn{2}{l}{\text{PRED} = \textbf{and}}\\
\text{ARG1} = & \left[\begin{array}{l}\textbf{verb-formula}\\ \text{IND} = \boxed{1}\\ \text{PRED} = \textbf{teach\_1}\\ \text{ARG1} = \boxed{1}\end{array}\right]\\
\text{ARG2} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{binary-formula}}\\
\multicolumn{2}{l}{\text{IND} = \boxed{1}}\\
\multicolumn{2}{l}{\text{PRED} = \textbf{and}}\\
\text{ARG1} = & \left[\begin{array}{l}\textbf{p-agt-formula}\\ \text{IND} = \boxed{1}\\ \text{PRED} = \textbf{p-agt}\\ \text{ARG1} = \boxed{1}\\ \text{ARG2} = \boxed{2}\ \boxed{\textbf{obj}}\end{array}\right]\\
\text{ARG2} = & \left[\begin{array}{l}\textbf{p-pat-formula}\\ \text{IND} = \boxed{1}\\ \text{PRED} = \textbf{p-pat}\\ \text{ARG1} = \boxed{1}\\ \text{ARG2} = \boxed{\textbf{obj}}\end{array}\right]
\end{array}\right]
\end{array}\right]\\
\text{OBJECT-INDEX} = & \boxed{2}
\end{array}\right]
\end{array}\right]$$

QUALIA : TELIC $[e][\textbf{teach\_1}(e) \wedge \textbf{p-agt}(e,x) \wedge \textbf{p-pat}(e,y)]$

QUALIA : OBJECT-INDEX: x

Figure 4.16: Result of application of agentive rule to *teach*

(19) John's teacher
John's carer
John's lover
John's killer
The product's distributor
The product's seller

Again, the same effects can arise with non-derived nominals, e.g. *the book's author*.

The examples of derivation and of the possessive which I have sketched above only cover a few of the possible interpretations. Clearly, this account has to be expanded to allow for the others. However, this does not lead to a multiplicity of unrelated rules and entries in the system, because the inheritance mechanisms in LAUREL can be used to relate the different possibilities. For example, all the possible interpretations of the possessive can be regarded as inheriting from a general entry which states that some unspecified predicate connects the entities involved. I will consider such hierarchical relationships in subsequent sections in the context of sense extension rules.

4.3.3 Grinding

Having sketched an approach to morphology we can now see how similar mechanisms apply in the case of sense extensions, by examining one particular class of examples. Most nouns denoting animals can also be conventionally used to mean the meat/flesh of that animal (*lamb*, *rabbit*, *haddock*, *chicken*) or (in somewhat more restricted cases) its fur/skin (e.g. *lamb*, *rabbit*, *mink*, *beaver*). These can be seen as a special case of a more general sense extension rule, `grinding`. It is well known that any count noun denoting a physical object can be used in a mass sense to denote a substance derived from that object, when it occurs in a sufficiently marked context. I refer to this as grinding because the context normally suggested is the 'Universal Grinder' (see, e.g. Pelletier and Schubert, 1986). So if *a table* is ground up the result can be referred to as *table* (*there was table* *all over the floor*). Several regular sense extensions can be regarded as special cases of grinding, where the extension may have become lexicalised. Thus, besides the animal/meat examples, trees used for wood have a sense denoting the wood (e.g. *beech*), and so forth.

This extension appears to be productive, at least in a sufficiently marked context; for example, in the LOB corpus we find the use of *mole* as a mass term:

> Badger hams are a delicacy in China while mole is eaten in many parts of Africa.

We therefore cannot assume that the 'extended' senses are necessarily lexicalised, even in the relatively conventionalised uses to mean meat, fur etc. Any approach which involved the use of distinct lexical entries would not account for this productivity. As an alternative all nouns can be taken to be initially underspecified with respect to the count/mass distinction. Thus it is possible to produce a grammar where nouns are initially undefined with respect to a syntactic count feature and where **lamb′** is taken as denoting the animal and the meat and fur and so on (see the 'p-theory' in Pelletier and Schubert, 1986, and also Copestake, 1990a). In contexts where one interpretation is forced (*a piece of lamb*, *two lambs*) the predicate can be restricted to denote either count or the mass senses (in this case, either the animal or the meat senses). However, this predicts that NPs such as *the lamb* are vague rather than ambiguous between count and mass readings. But the results of the usual tests for ambiguity (Zwicky and Sadock, 1975) suggest that this is incorrect; for example, sentences such as:

> ? John fed and carved the lamb.

are peculiar. It also seems that there should be some constraint on what an individual predicate connected with a lexical entry can denote, which would exclude a disjunction of the animal and ground senses of *lamb*. Formalising such a constraint is currently impossible, of course, nevertheless the intuitive idea that lexical denotation should be conceptually coherent is in conflict with the underspecification proposal in this case. There are other examples where it seems more reasonable to claim that a nominal should be underspecified with respect to the count/mass distinction; it is frequently unclear whether individuation is occurring with nouns like *data*. But we will regard a noun like *lamb* as ambiguous between mass and count senses rather than vague, and the mass (meat) sense as an extension of the count (animal) sense, specified by lexical rule.

The most general grinding process can be specified in LAUREL as shown in Figure 4.17. The effect of the lexical rule is to transform a count noun with the qualia properties appropriate to an individuated physical object into a mass noun with properties appropriate for an unindividuated substance.

We specialise the grinding rule to allow for cases such as the animal/meat extension explicitly. The typed framework provides us with a natural method of characterising the subparts of the lexicon to which such rules should apply. The lexical rules can, in effect, be parametrised by inheritance in the type system. For example, we can give rules which inherit information from `grinding` such as `animal-grinding`:

```
animal-grinding
   < > < grinding < >
   < 1 : QUALIA > = animal
   < 0 : QUALIA > = c_natural.
```

Here **c_natural** is a type which stands for comestible naturally derived things. Thus given the lexical entry for *rabbit* we can apply the lexical rule to generate a sense corresponding to 'edible stuff derived from rabbits' partially represented as shown in Figure 4.18. Here the specification of the value for the telic role arises from the constraint on the type **c_substance**.

It should be noted that the `animal-grinding` rule does not result in the full specification of what might usually be taken as the meaning of the meat/flesh sense. The substance is stated to be edible, and to be derived from the animal, but there is no attempt at defining the meaning to exclude stuff derived from bones, for example. Particular cultural assumptions will affect exactly what is taken to be edible, so *rabbit* will usually be taken to exclude the bones but *whitebait* will not, for example. Thus not all the characteristics are captured by the lexical rule.

$$\left[\begin{array}{l}\textbf{grinding}\\ 0 = \left[\begin{array}{l}\textbf{lex-uncount-noun}\\ \text{ORTH} = \boxed{0}\ \textbf{orth}\\ \text{CAT} = \boxed{\textbf{noun-cat}}\\ \text{SEM} = \left[\begin{array}{l}\textbf{obj-noun-formula}\\ \text{IND} = \boxed{1}\ \boxed{\textbf{obj}}\\ \text{PRED} = \boxed{2}\left[\begin{array}{l}\textbf{modified-pred}\\ \text{MODIFIER} = \textbf{grinding}\\ \text{MODIFIED} = \boxed{3}\ \textbf{string}\end{array}\right]\\ \text{ARG1} = \boxed{1}\\ \text{PLMOD} = \textbf{false}\\ \text{QUANT} = \textbf{false}\end{array}\right]\\ \text{QUALIA} = \left[\begin{array}{l}\textbf{physical}\\ \text{AGENTIVE} = \left[\begin{array}{l}\textbf{agentive}\\ \text{ORIGIN} = \boxed{3}\end{array}\right]\\ \text{FORM} = \left[\begin{array}{l}\textbf{nomform}\\ \text{RELATIVE} = \textbf{mass}\end{array}\right]\end{array}\right]\end{array}\right]\\ 1 = \left[\begin{array}{l}\textbf{lex-count-noun}\\ \text{ORTH} = \boxed{0}\\ \text{CAT} = \boxed{\textbf{noun-cat}}\\ \text{SEM} = \left[\begin{array}{l}\textbf{obj-noun-formula}\\ \text{IND} = \boxed{4}\ \boxed{\textbf{obj}}\\ \text{PRED} = \boxed{3}\\ \text{ARG1} = \boxed{4}\\ \text{PLMOD} = \textbf{false}\\ \text{QUANT} = \textbf{true}\end{array}\right]\\ \text{QUALIA} = \left[\begin{array}{l}\textbf{physical}\\ \text{FORM} = \left[\begin{array}{l}\textbf{nomform}\\ \text{RELATIVE} = \textbf{individual}\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]$$

Figure 4.17: Grinding

$$\left[\begin{array}{l}\textbf{lex-uncount-noun}\\ \text{ORTH} = \textbf{rabbit}\\ \text{CAT} = \boxed{\textbf{noun-cat}}\\ \text{SEM} = \left[\begin{array}{l}\textbf{obj-noun-formula}\\ \text{IND} = \boxed{0}\ \boxed{\textbf{obj}}\\ \text{PRED} = \left[\begin{array}{l}\textbf{modified-pred}\\ \text{MODIFIER} = \textbf{grinding}\\ \text{MODIFIED} = \boxed{2}\ \textbf{rabbit\_L\_1\_1}\end{array}\right]\\ \text{ARG1} = \boxed{0}\\ \text{PLMOD} = \textbf{false}\\ \text{QUANT} = \textbf{false}\end{array}\right]\\ \text{QUALIA} = \left[\begin{array}{l}\textbf{c\_natural}\\ \text{AGENTIVE} = \left[\begin{array}{l}\textbf{agentivestuff}\\ \text{ORIGIN} = \boxed{2}\end{array}\right]\\ \text{TELIC} = \boxed{\textbf{verb-sem}}\\ \text{FORM} = \left[\begin{array}{l}\textbf{nomform}\\ \text{RELATIVE} = \textbf{mass}\end{array}\right]\\ \text{CONSTITUENCY} = \boxed{\textbf{nomconst}}\\ \text{OBJECT-INDEX} = \boxed{0}\\ \text{PROPERTIES} = \boxed{\textbf{phys-properties}}\end{array}\right]\end{array}\right]$$

Figure 4.18: Meat/flesh sense of *rabbit*

Although the denotation of the count sense and the mass sense are distinct there clearly is some relationship between them. A full account of sense extension must be able to represent relationships between the senses' denotations. For grinding in general, the most specific claim that can apparently be made is that the ground sense denotes some "stuff" which was, at some past time, part of one or more individuals denoted by the count sense. I will illustrate how this might be formalised in the next chapter where I will discuss the semantics of mass and count terms in some detail. The ORIGIN feature in the qualia structure encodes the relationship with the unground sense.

Our current treatment thus has similarities to the "s-theory" of Pelletier and Schubert (1986) where *lexical extension rules* are used to produce mass nouns from count nouns. However these rules merely change the value of the syntactic count feature, apply a predicate operator which is the same for all cases of grinding, and mark the mass sense resulting as "+EXT" which is supposed to suggest that it is in some way abnormal. This is clearly inadequate:

> John carved the lamb.

would be marked "+EXT" for the reading where lamb was used in a mass sense and not for the count sense reading. In our approach, lexical rules are hierarchically ordered; the more specific rules, which inherit from the general grinding rule, express the conventionalised processes that apply to semantically specified parts of the lexicon. In this way we account for the possibility of multiple distinct mass senses being possible. For example, *rabbit* is given distinct senses in LDOCE for the meat and the fur, and (in context) an underspecified sense is available:

> After several lorries had run over the body, there was rabbit splattered all over the road.

Multiple lexical rules may be applied in sequence. For example, another lexical rule which we specify is `portioning` which converts food (or drink) denoting mass nouns into count nouns denoting a portion of that substance (e.g. *three beers*, see Figure 4.19). This is clearly productive, it can be used with names of particular types of beer, for instance, such as *three IPAs*. It can also apply to extended senses such as *three lambs*, at least in the context of a restaurant.

The sense extension involved in portioning is relatively tightly specified. The entity is simply some particular quantity of the substance, and while we may not know the size of the portion, the rest of its properties are regularly predictable from knowledge of the substance. I will discuss this in more detail in the next chapter. In contrast, metaphorical uses are very underdetermined, although there are some regularities involved in the metaphorical use of animal words to denote humans. For example, a word denoting a group of animals will denote a group of humans, rather than a single individual, when used metaphorically. The `animal-metaphor` rule is given in Figure 4.20.

4.3.4 Lexical rule application

We think of sense extension rules as defining the limits of coercion amongst lexemes. It is clear that their application has to be controlled. For example, with the rules described in the previous section, (20a) has the interpretations (20b) and (20c), apart from the more likely possibilities:

(20) a John saw some pigs.
b John saw some portions of pig meat.
c John saw some portions of substance derived from humans with some pig-like properties.

However, such control does not fit simply into a constraint-based approach, since it essentially involves preferring particular readings rather than preventing rule application altogether. So in this section, I will describe some of the problems and consider how they might be tackled, rather than putting forward a particular treatment.

The first case I will consider is blocking, or preemption by synonymy. As I mentioned earlier, this occurs with sense extensions in a way which parallels the morphological examples. Because

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lexical-rule}} \\
0 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-count-noun}} \\
\text{ORTH} = & \boxed{0}\ \textbf{orth} \\
\text{CAT} = & \boxed{\textbf{noun-cat}} \\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj-noun-formula}} \\
\text{IND} = & \boxed{1}\ \boxed{\textbf{obj}} \\
\text{PRED} = & \left[\begin{array}{l}
\textbf{modified-pred} \\
\text{MODIFIER} = \textbf{portion} \\
\text{MODIFIED} = \boxed{3}\ \textbf{logical-pred}
\end{array}\right] \\
\text{ARG1} = & \boxed{1} \\
\text{PLMOD} = & \textbf{boolean} \\
\text{QUANT} = & \textbf{boolean}
\end{array}\right] \\
\text{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{comestible}} \\
\text{AGENTIVE} = & \boxed{4}\ \textbf{nomagent} \\
\text{TELIC} = & \boxed{5}\ \boxed{\textbf{verb-sem}} \\
\text{FORM} = & \left[\begin{array}{l}
\textbf{nomform} \\
\text{RELATIVE} = \textbf{portion}
\end{array}\right] \\
\text{CONSTITUENCY} = & \boxed{6}\ \boxed{\textbf{nomconst}} \\
\text{OBJECT-INDEX} = & \boxed{1} \\
\text{PROPERTIES} = & \boxed{7}\ \boxed{\textbf{phys-properties}}
\end{array}\right]
\end{array}\right] \\
1 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-uncount-noun}} \\
\text{ORTH} = & \boxed{0} \\
\text{CAT} = & \boxed{\textbf{noun-cat}} \\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj-noun-formula}} \\
\text{IND} = & \boxed{8}\ \boxed{\textbf{obj}} \\
\text{PRED} = & \boxed{3} \\
\text{ARG1} = & \boxed{8} \\
\text{PLMOD} = & \textbf{boolean} \\
\text{QUANT} = & \textbf{false}
\end{array}\right] \\
\text{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{comestible}} \\
\text{AGENTIVE} = & \boxed{4} \\
\text{TELIC} = & \boxed{5} \\
\text{FORM} = & \left[\begin{array}{l}
\textbf{nomform} \\
\text{RELATIVE} = \textbf{mass}
\end{array}\right] \\
\text{CONSTITUENCY} = & \boxed{6} \\
\text{OBJECT-INDEX} = & \boxed{8} \\
\text{PROPERTIES} = & \boxed{7}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

Figure 4.19: Lexical rule for portioning

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lexical-rule}} \\
0 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-count-noun}} \\
\text{ORTH} = & \boxed{0}\ \textbf{orth} \\
\text{CAT} = & \boxed{\textbf{noun-cat}} \\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj-noun-formula}} \\
\text{IND} = & \boxed{1}\ \boxed{\textbf{obj}} \\
\text{PRED} = & \left[\begin{array}{l}
\textbf{modified-pred} \\
\text{MODIFIER} = \textbf{animal-metaphor} \\
\text{MODIFIED} = \boxed{3}\ \textbf{logical-pred}
\end{array}\right] \\
\text{ARG1} = & \boxed{1} \\
\text{PLMOD} = & \textbf{false} \\
\text{QUANT} = & \textbf{true}
\end{array}\right] \\
\text{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{human}} \\
\text{TELIC} = & \boxed{\textbf{verb-sem}} \\
\text{FORM} = & \boxed{4}\left[\begin{array}{l}
\textbf{nomform} \\
\text{RELATIVE} = \textbf{(individual plural group)}
\end{array}\right] \\
\text{PROPERTIES} = & \left[\begin{array}{l}
\textbf{creature-properties} \\
\text{STATE} = \textbf{solid\_a} \\
\text{SEX} = \boxed{5}\ \textbf{gender}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
1 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-count-noun}} \\
\text{ORTH} = & \boxed{0} \\
\text{CAT} = & \boxed{\textbf{noun-cat}} \\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj-noun-formula}} \\
\text{IND} = & \boxed{6}\ \boxed{\textbf{obj}} \\
\text{PRED} = & \boxed{3} \\
\text{ARG1} = & \boxed{6} \\
\text{PLMOD} = & \textbf{false} \\
\text{QUANT} = & \textbf{true}
\end{array}\right] \\
\text{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{animal}} \\
\text{TELIC} = & \boxed{\textbf{verb-sem}} \\
\text{FORM} = & \boxed{4} \\
\text{PROPERTIES} = & \left[\begin{array}{l}
\textbf{creature-properties} \\
\text{STATE} = \textbf{solid\_a} \\
\text{SEX} = \boxed{5}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

Figure 4.20: Lexical rule for the metaphorical use of animal denoting words.

words like *pig* can be used in the extended sense, in marked contexts, the correct way to treat blocking is not to prevent application of the lexical rule completely, but to mark generated occurrences as being in some way peculiar when they are synonyms of an existing word sense, such as *pork*. Because we are making use of a relatively rich representation which indicates information such as ORIGIN, we can recognise potential synonyms of generated senses by comparison of the qualia structures. This much is possible within LAUREL, at least in principle, although the practical difficulties of ensuring that equivalent senses have equivalent qualia structure are considerable, and it is not clear how detailed the representation would have to be to achieve this effect. However, the idea that the lexical rule should not be applied to generate senses corresponding to existing entries in an unmarked context cannot be formalised within LAUREL, because it involves a condition on lexical rule application which can only be formalised in terms of a comparison operation on the entire lexicon.

The second point is that, in general, if an interpretation can be found without application of a lexical rule, this is to be preferred. This statement needs considerable qualification, however. Many extended uses can be regarded as established, and in these cases this principle would not be expected to apply. I would not claim, for example, that in (21) *chicken* should be interpreted as referring to the live bird by default.

(21) John saw the chicken.

Furthermore, although the extended and the non-extended senses may be distinguishable by definite syntactic criteria in a particular sentence, often the context will favour, rather than force, the extended interpretation. In this case we would have to weigh the contextual effect against the likelihood of rule application, in that context. This does not rule out the possibility that the distinction between base and extended senses could be treated in exactly the same way as normal lexical ambiguity, in which case there would be no independent measure of the likelihood of rule application out of context. This can really only be investigated as part of a general theory of sense distinction and disambiguation.

The third aspect of control of rule application applies to hierarchically organised lexical rules, where the feature structure given by application of the more general rule will subsume that resulting from application of the more specific. We assume that in these cases the more specific rule will be applied in preference. A sentence where the use of the general grinding rule is forced will be odd, without some context, because the sense generated will be very underspecified, and will tend to refer to some substance which is not conventionally used.

(22) ?? John likes table.

The presence of a more specific rule such as `animal-grinding` indicates that a particular, relatively specific, sense will normally be generated if the general process can apply. It thus follows that the more specific rules should be applied in preference to the more general ones. Again, we can state the condition that a subsumption relationship holds in LAUREL, but not the particular effect of preferring the more specific rule.

One option for control of rule application is to develop a language in which to state principles such as those suggested above. If the aim is to incorporate lexical rules in a practical system, then this is probably the best approach. However, from a theoretical point of view, it is more interesting to investigate why these principles might arise, and to try and motivate them in terms of a more general processing model. The second principle suggests that there is some 'cost' to rule application. If this is assumed it could also account for blocking, since presumably if a suitable lexical item exists which can be used without incurring the cost of a lexical rule, then it would normally be used, assuming Grice's Maxim of Manner applies. Ostensibly avoiding the established word carries the implication that the speaker sees it as inappropriate, which in the case of examples like *pig* might lead to the interpretation that the meat was seen as substandard or unpleasant. The cost of rule application could be an indirect effect associated with having to disambiguate between the base and extended senses, but this would not account for blocking in morphology.

4.3.5 Hierarchical rule organisation

The concept of hierarchically organising sense extensions so that the conventional ones are treated as special cases applies in cases other than grinding. For example, Ostler and Atkins (1991) give as examples of conventional sense extensions (lexical implication rules in their terminology):

> LIR **Vehicle Verb**
> NC: singular:vehicle -> VTI(verb transitive/intransitive):to travel/transport using that vehicle
> e.g. cycle
> LIR **Cutter - Cut**
> NC: tool for cutting piercing -> VTI(verb transitive/intransitive):to use that tool on
> e.g. knife

However, we would regard these as special cases of a more general rule relating objects to some act where the object is used as an instrument. Clearly this will normally only apply to objects with a conventional use in some action (which will be represented by their telic role), but examples such as (23), where an unconventional use is forced by context, are also possible.

(23) The mugger tried to cosh John but Mary handbagged him.

Nunberg and Zaenen (1991) take a somewhat different position; they argue that the general processes such as grinding should be explicitly represented but that the specialisation of grinding is predictable on pragmatic grounds and thus should not be expressed by rule; the meat and fur readings are provided by contextual processes. Our account of the specialisation of grinding relies on the representation of detailed semantic information in the lexicon, but under Nunberg and Zaenen's account this is, at first sight, unnecessary.

Nunberg and Zaenen distinguish between lexicology and lexicography. They take the goals of lexicology to provide a description of the language which is as economical as possible, which will account for acceptability when taken along with assumptions about rational agency and world knowledge. Under this account, given the possibility of the grinding process, an agent can predict that a mass use of *rabbit* is likely to refer to the meat or the skin, even in an underspecified context, because world knowledge tells us that meat and skin are the most likely stuff to be derived from rabbits. Thus specialisation of the grinding rule is unnecessary. Under this account, lexical knowledge and encyclopaedic knowledge are distinguished, and the fact that meat can be derived from rabbits is part of encyclopaedic, rather than lexical knowledge.

An important part of Nunberg and Zaenen's case for the pragmatic account of the specialisation of grinding is that the meat and skin/fur senses are not distinct by the usual ambiguity tests; to show this they give example (24).

(24) He calls himself an environmentalist, but he both wears and eats rabbit.

However, this is inconclusive, given the availability of the underspecified ground sense in our account, especially since other examples, such as (25), are less acceptable:

(25) ? John's rabbit was inedible and Bill's was unwearable.

We would argue that the underspecified ground sense is applicable in (24) because of the use of *environmentalist*; it seems quite likely that the fact that the stuff is somehow derived from living animals is what is important in the sentence, rather than its particular function. But in (25) there is nothing to motivate the underspecification, and thus the sentence seems odd.

Furthermore, as Ostler and Atkins (1991) have pointed out, there are a considerable number of apparently quite idiosyncratic restrictions on the use of conventional sense extensions, for example, that although the names of plants can usually be used to denote substances derived from them, in the case of oils this applies to those used in perfume (*lavender*, *jasmine*) but not those used in cooking (?*olive*, ?*sunflower*). As Nunberg and Zaenen point out there is no apparently plausible direct pragmatic explanation for this, though they claim that in context the cooking oil use might be possible. [8] In order to account for restrictions on the conventional or idiomatic use of sense extensions (or *transfer functions* in their terminology), Nunberg and Zaenen have to postulate *lexical licences* for application of the transfer functions, and they suggest that this might also apply to uses to provide the names of meat and hides. Lexical licences cannot operate at a purely pragmatic level, not only because they are conventionalised processes but also because some of the restrictions described by Ostler and Atkins appear to have a phonological, morphological or syntactic basis. Any implementation of lexical licences requires that detailed semantic information be available, such as that associated with the telic role in Pustejovsky's theory.

It is thus rather difficult to see how the two positions can be clearly distinguished, except in the context of a much more complete account of interaction with pragmatics. Our specialisations of lexical rules do differ in behaviour from the most generic ones, in that they are preferentially applied, and I do not think they could be necessarily distinguished from a computational approach to lexical licences. Furthermore, any account of the conventionalised processes should relate them to the general processes, such as grinding. The lexical rule account at least provides the beginnings of such a framework. I think it entirely reasonable that lexicology (in Nunberg and Zaenen's use of the term) should not be concerned with cataloguing the conventional sense extensions, but it is necessary to have a framework in which they can be described, and classified, just as an adequate lexical theory has to provide for the description of lexical items, even though actually enumerating them is the job of the lexicographer. I do not believe that this leads to the situation where the linguistic/lexicological description cannot be divorced from an account of the real world, with all the methodological problems that this would entail, because what is required is a treatment of the *interaction* of linguistic and world knowledge, not a detailed description of the latter.[9]

4.3.6 The pragmatics of reference

We have approached the problem of sense extension from a lexical perspective, and made explicit comparison with morphology, but it does, of course, also have a pragmatic component and a full account should eventually incorporate both. From the pragmatic viewpoint, sense extension as described here looks like a special case of the more general possibility of transfer of reference, considered by Nunberg (1978). In the context of waiters or waitresses talking in a restaurant, Nunberg pointed out that examples such as those in (27) seem quite natural ways of referring to a customer who has ordered a ham sandwich:

(27) The ham sandwich is sitting at table 5.
He's at table 5. (said while pointing to a ham sandwich)

It is clear that the factors governing these extended uses must be mainly pragmatic; it is clearly impossible to have an account where a lexical entry for *ham sandwich* denoting a customer is explicitly represented and there is very little about the extended use which it is possible to predict semantically. However, in the context of the preceeding discussion, I would argue that there is some non-pragmatic element, and that the basic transfer functions which Nunberg assumes should be treated as underspecified sense extension rules in our framework.

[8] Nunberg and Zaenen suggest that one can imagine professional cooks might say things like "I usually fry it in safflower". Even as an ex-professional food technologist, I find "??I usually fry it in olive/ground nut/sunflower/rape seed" very odd (ignoring *safflower* because I don't associate it with frying). Sentences such as (26) are more acceptable, but refer to the macerated substance, not to the oil.

(26) There's olive/ground nut/sunflower/rape seed in the food processor.

This suggests that the conventional use is not possible but that the general grinding rule can be applied.

[9] Within the current framework, we could perhaps distinguish between the general and conventionalised rules in terms of description as well as application, by making the general rules depend on lexical semantic type alone, while the conventionalised rules may refer to the values of the roles of the qualia structure. On the assumption that the types are set up according to linguistic criteria, but that the values of the roles are regarded as reflecting foregrounded real world knowledge, this would give us a principled distinction.

In the account above we treated sense extension as governed by lexical rule, and clearly Nunberg's examples can apply to phrasal units. But our description of metonymic sense extension as lexical was an oversimplification, given examples such as (28) where the metonymy involves the whole noun phrase.

(28) The south side of Cambridge voted Conservative.

Since there is no real distinction between lexical rules and non-lexical unary rules in the current framework, this causes no particular problems. Nunberg assumes that the basic transfer functions can be combined in an indefinite number of ways. Although not all the possibilities for combination which he envisages apply to lexical rules, many of them do, given that sequences of sense extension rules may be applied, and that inheritance hierarchies of rules are possible. I believe that the assumption that the basic transfer functions be represented by something like the lexical rule mechanism is justified, since I think that there are some restrictions on them which are not accounted for purely by the pragmatics of reference. For example, (29) is not a plausible way of referring to a ham sandwich, even though in context it could meet Nunberg's conditions for reference.

(29) *The man in the brown suit has too much mustard on it.

In general it is much easier to transfer reference to humans than away from them.

Given that both sense extension and general transferred reference involve possible shifts in grammatical behaviour, the alternative to the use of rules for representation involves very underspecified lexical entries. Some of the arguments against treating grinding in this way were made above; but in general almost nothing would remain which could be lexically specified about many nouns. Countability and number can both be affected by transfer functions and sense extensions. Grinding and portioning affect countability, the 'place to people' metonymy involves the production of a group noun which can take singular or plural agreement. Pollard and Sag (in press) claim that the transferred reference also changes agreement, giving example (30).

(30) The hash browns at table nine is/*are getting angry.

This is not completely convincing since *hash browns* is a conventionalised plural. The equivalent sentence with *ham sandwiches* is worse, however it is quite clear that if transfer of reference does occur plural agreement is not possible.

(31) The ham sandwiches at table nine ?is/*are getting angry.

Transfer of reference also clearly affects pronoun agreement.

(32) The ham sandwich at table nine just made a fool of himself/*itself. (Pollard and Sag, in press)

Even syntactic category might have to be left underspecified if this approach is taken to its logical conclusion, since sense extensions from noun to verb denoting characteristic action, such as *sugar* and *nail* appear to be productive. Most if not all nouns denoting substances used for covering can be used to denote the action; *paint*, *varnish*, *lacquer*, *Ronseal*, *Sadolin*.[10] The possibility of sentences such as (33), repeated from above, have led to the well-known conclusion that one can verb anything.

(33) The mugger tried to cosh John but Mary handbagged him.

Such examples could perhaps be excluded as metalinguistic but this does not seem reasonable for the more semantically restricted groups, and it is much more attractive to attempt to find a

[10] The latter two are tradenames for varnish or lacquer products.

common account.

We have to exclude an account where production of an underspecified meaning representation is followed by pragmatic processing to determine the specific predicate, both because it incorrectly predicts that predicates are vague between the possible interpretations, and because it does not explain how grammatical behaviour is constrained by the interpretation. It is quite clear that a representation is needed where the shifts in grammatical behaviour are kept in correlation with the shifts in meaning. Pollard and Sag suggest that this could be done for number agreement purely by letting the referent determine plurality. Thus the lexical entry is underspecified, but the anchoring conditions on an instance of a sign will result in its instantiation. The details of this proposal are not clear. However, it does not seem plausible that all lexical entries can be underspecified in this way, since number and countability cannot be determined, in general, purely from the real world referent. We might assume that they are defeasibly determined in this way, but can be lexically overridden, and that in the case of the extended senses no overriding will occur, simply because there is no lexicalised information. I think such an account might be possible, but I do not think that it adequately accounts for Nunberg's examples, since it still requires the denotation of the predicate to be vague, between the basic reading and any pragmatically possible extended reading, in any underdetermined context. This does not accord with the results of ambiguity tests, nor does it explain why the basic readings are assumed by default.

An account where lexical rules are applied prior to a pragmatic processing phase is also highly implausible, because it would lead to generation of an indefinite number of readings. Rule application should be controlled by the pragmatic conditions on reference. For example, we might postulate a very basic transfer function, allowing designators of objects to be used to refer to a person contextually associated with them. The rule might specify the output to be a **lex-count-noun** with lexical semantic type **human** and give a vague predicate connecting the output with the input. The pragmatic constraint would have to ensure that this only operated if its output designated a particular individual, etc., in the current context. Note that this does require something like Pollard and Sag's anchoring conditions. There would also have to be a more global constraint that rules were not applied unless forced by context. If we assume that conventionalised sense extensions are more specific forms of the basic transfer functions, we also need a constraint that a conventionalised extension is assumed, if possible, in preference to a less specific one. Under this assumption, modularity of description can still be retained and pragmatic processing is still seen as selecting between interpretations licensed by the grammar, but this selection operates in parallel with syntagmatic processes.

To summarise this account: I assume that conventionalised sense extensions are specialisations of the basic transfer functions, and that these may also underlie derivation and the type shifts postulated to account for logical metonymy. In sufficiently rich contexts we get the unconventional, ham sandwich type use of the basic transfer functions. In LAUREL, the basic transfer functions are represented by a limited number of types and the semantic content of all sense extension rules, type shifting unary rules and (possibly) derivational rules, must be consistent with one of these types. The unconventional transfers of reference correspond to rules which contain no specialisation of the basic functions, relying on context to specify their interpretation. Conventionalised sense extensions are represented by specialisations of these rules, therefore they specify the interpretation of the extended sense more finely, and are natural in less marked contexts. Particular extended senses may also become conventionalised or established, and these may have associated idiosyncratic information, as may morphologically derived words. There is no fundamental distinction between lexical and non-lexical or morphological and zero-derivational operations with respect to the basic transfer types, since their differences are with respect to the non-semantic component of the rules.

There is clearly much that is highly speculative and schematic about this account, in particular the interface to pragmatics. The treatment as it stands can be only really be falsified in the rather weak sense that it may turn out that it can only be made to 'work' by adding lexical semantic types and features and basic transfer functions to account for each new example. However, taken in the context of a general approach to lexical semantics, I believe that it would become clear whether ad-hoc additions were being made, because they would lack independent linguistic motivation. For example, the descriptions of grinding developed here, depend on the representation of information such as the telic role, which, as we have seen earlier, has independent justification. Further investigations along these lines can only really be carried out within a formal theory of pragmatics and its interrelationship with lexical semantics, which is outside the scope of the current work.

Chapter 5

Individuation

In this chapter I will discuss in more detail the representation of individuation in nominals, by which I mean the way in which (real world) entities can be treated linguistically as single individuals, plural individuals, groups or uncountable masses. I will not attempt a full treatment of these issues, in particular I will only discuss non-generically referring concrete nominals, and I will not cover many of the problems that these raise, but I will show how the lexical semantic qualia structure can complement a formal semantic treatment. In the next chapter some of the representations described here will be used in the analysis of dictionary definitions.

From the linguistic viewpoint we can distinguish between predicates which have some 'built-in' notion of an individual (count nouns), and those which do not (mass nouns). Individuation is clearly a semantic phenomenon, but equally clearly cannot be completely predicted from real world knowledge of the entities involved. There are some generalisations which can be made, for example, in English, nouns denoting humans individuate, furthermore, even nouns denoting groups of humans have uses where the group is treated as a plural entity. Conversely, predicates denoting substances like *gold*, which has no obvious particles, will not individuate. However there are many areas of the lexicon where the count/mass distinction is not straightforwardly predicatable. There is no particle size above which all nouns are count and below which they are mass. Some mass nouns such as *furniture*, *clothing* and *cutlery* refer to collections of objects, rather than mass-like substances. Many of these collection mass nouns are used where the individual objects involved are related by function, but where there is no common lexicalised predicate which describes the individual objects, but again this cannot be predicted from the denotations.

The unpredictability of the mass/count distinction is demonstrated by the instances of pairs of lexical items which are distinguished (mainly) by whether and how they individuate, for example, *garments/clothes/clothing*, *job/work*, *weapons/arms*, *turds/shit*. There are also cross-linguistic differences, even between closely related words and languages: for example, *information* and *fennel* are mass in English, but *information* and *fenouil* are count in French. Such differences may even apply over subparts of the lexicon, for example pasta denoting words, *spaghetti*, *tagliatelli* and so on, are count in Italian and mass in English[1]. However *noodle* is count in English.

Wierzbicka (1988) claims that the mass-count distinction can be accounted for on the basis of what I refer to as real world knowledge. She takes into account not only physical characteristics, but also factors such as whether something is typically seen in large quantities and how a food is characteristically eaten in a particular country. However plausible her accounts may be, they are essentially untestable and more importantly, apply historically rather than synchronically. Wierzbicka does not consider the *spaghetti* example, but, even supposing that the differences between English and Italian were originally due to British cooking techniques resulting in unindividuated pasta, synchronically countability would not be predicted. Even though I invariably cook pasta to be *al dente*, and assume that this is the norm, I cannot, in English, use *spaghetti* as an ordinary count noun.[2] Thus the mass/count distinction sometimes has to be lexically specified, but

[1] This was pointed out to me by Antonio Sanfilippo

[2] Actually I suspect that the historical reasons for pasta denoting words being mass in English have more to do

there is no reason why this should preclude capturing the regularities that I mentioned initially.

These issues are complicated by the extended uses that arise from grinding, portioning and so on. Thus *spaghetti* can be used as a count noun, in the kind or portion readings, and even in ham sandwich sentences, such as:

(1) The two spaghettis left without paying their bills.

(which might be said by a waiter in a restaurant, where *spaghetti* is understood to refer to a customer who ordered spaghetti). Although it is clear that the mass sense of *spaghetti* is the established one, there are other examples, such as *cake*, where this is less obvious. Clearly the mass/count behaviour of true extended senses cannot be specified by lexical entry, since extended senses do not have lexical entries as such.

Apart from the mass/count distinction, there are various complications that arise when treating plurality. At the formal semantic level a treatment of plural denoting predicates is required which relates them to the singular predicates. Such a theory also has to account for distributive behaviour, and to describe how entities can be counted. Normally, plural denotation correlates with plural agreement, but there are obvious exceptions in English, for example the pair nouns, *scissors*, *trousers* and *binoculars*. Although there is some sense of plurality which arises from the denotation of such nouns, and thus we do not want to claim that agreement here is purely syntactic, this does not correspond with a predicate which is plural in terms of the formal semantics. For example, if *the scissors* denotes a single pair of scissors, it cannot be used with partitives such as *one of*. Furthermore the entity denoted by *scissors* cannot be inherently plural with respect to agreement (although it may inherently have a potential for plurality) because the use of a different predicate, for example, *instrument*, changes the agreement.

The treatment of individuation proposed here does not rely on any assumption that the world is divided up into plural, individual and mass-like entities, but treats most aspects of individuation as being relative to some predicate. For example, a particular physical entity may be referred to by one predicate as a single individual, by another as a plural, and by another as a mass (for example, *a report*, *65 sheets of paper*, *garbage*). I will use a formal semantic treatment, largely based on that developed by Krifka (1987), in which there is no fundamental distinction between objects and stuff, but differences in the way that particular predicates individuate may be captured in the model in terms of properties of the predicate. The basic distinction is between quantised and cumulative predicates: ordinary singular count nouns have quantised reference, mass terms and plurals have cumulative reference. Plural predicates are formed from singulars by the application of a closure operator, following Link (1983). In the qualia structure, I make finer distinctions between the individuation possibilities relative to the particular predicate. The classes are specified as types within the FORM role of the qualia structure; these include **individual**, **plural**, **group**, **mass** and **pair**. The overall qualia types are cross-classified by these relative form distinctions, but some types have a restricted range of possible forms — **human**, for example, can only be **individual**, **plural** or **group**.

It is the lexical semantic form which is correlated with agreement information. This allows some of the peculiarities in English number agreement to be described. For example, pair nouns like *trousers* can be formally distinct from normal plurals, such as *dogs*, but nevertheless regularly take plural agreement. Agreement information is treated as part of the internal structure of referential indices, following Pollard and Sag (in press). The only agreement feature I will be concerned with here is NUM, number, which will take values **sg** or **pl**. Extended uses of nouns will have the agreement determined by the lexical rules. For example, the rule specifying the metaphorical use of animal terms to denote humans, given in Figure 4.20 in the previous chapter, applies to both individual and group denoting senses (for example *flock*, *herd*), and specifies that the resulting extended sense will individuate in the same way.

with morphology than cooking techniques.

5.1 Formal semantic of mass terms and plurals

This section is intended as an overview of the formal semantics to introduce the aspects of the theory which are most relevant from the viewpoint of the integration with the qualia structure. There are some issues, such as distributivity, which I will barely mention, since although they are important for a full theory, they are less critical for the lexical aspects. Krifka's treatment is quite different from that of Link, despite their common use of lattice theory, but the distinctions are relevant to the discussion of individuation, and I will make use of a closure operator equivalent to that of Link. I will outline Link's treatment here first.[3]

5.1.1 Link's logic of plurals and mass terms

Link's (1983, 1988) logic of plurals, LP, is a first order logic which introduces a sum operation for its individual terms. This *isum* operation, $\sqcup_i$, has $\oplus$ as its syntactic counterpart. For example, $a \oplus b$ denotes an entity in the domain of individuals, made up of the individuals a and b, but itself an individual of the same type as a and b. Thus a and b would correspond to single individuals, for example, individual students, but $a \oplus b$ is the plural individual comprising both a and b. In Link's theory it is such a plural individual which is involved in the collective reading of sentences like (2) and (3).

(2) Two students lifted the piano.

(3) John and Mary lifted the piano.

The domain of discourse, E is closed under the isum operation and forms a complete join semilattice. [4]

For example if the atomic individuals in the domain are:

$$A = \{j, m, b\}$$

then we have:

$$E = \{j, m, b, j \sqcup_i m, j \sqcup_i b, m \sqcup_i b, j \sqcup_i m \sqcup_i b\}$$

The intrinsic ordering relationship on E induced by $\sqcup_i$ is $\sqsubseteq_i$, *ipart* (syntactic counterpart Π).

$$a \Pi b \iff a \oplus b = b$$

An entity is said to be atomic if it has no iparts other than itself.

$$\forall x[\mathrm{At}(x) \iff \forall y[y \Pi x \Rightarrow y = x]]$$

There is also a *closure* operator, *, such that $^\star P$ forms all possible isums from the members of the extension of P. So if

$$[\![P]\!] = \{j, m, m \sqcup_i b\}$$

then

$$[\![^\star P]\!] = \{j, m, m \sqcup_i b, j \sqcup_i m, j \sqcup_i m \sqcup_i b\}$$

Thus if the singular noun, $\mathbf{student}'$, denotes the set of individual students, $^\star\mathbf{student}'$ will denote all the individuals, and all the possible plural individuals that they can form.

Landman (1989) points out that there is an isomorphic set-theoretic model. If we start with a basic set A of individuals then the denotations of individual constants are not the elements of A, but sets of elements of A, the power set of A, pow(A). We want to exclude the empty set from

[3] Much of this section is taken from Copestake (1990a).

[4] In Link (1983) a 0-element is assumed but is omitted in Link (1988). I will ignore the 0-element because nothing here depends on its inclusion.

the domain of individuals (although the empty set is relevant to the denotation of predicates). For example,

$$A = \{j, m, b\}$$
$$\mathrm{pow}(A) - \{\} = \{\{j\}, \{m\}, \{b\}, \{j, m\}, \{j, b\}, \{b, m\}, \{j, b, m\}\}$$

So $\mathrm{pow}(A) - \{\}$ has the structure of a complete atomic join semilattice, without the zero element, with union corresponding to Link's isum operator and subset to ipart.

In the theory LPM (the logic of plurals and mass terms) Link (1983) proposed a distinction between an entity and the 'stuff' which constitutes it (constituency is represented by $\triangleright$). Material constituency of 'stuff' is represented using a semilattice but this is distinct from the semilattice of ordinary individuals. So there is an operator, $+$, *material fusion*, such that $a + b$ constitutes but is not identical to, $a \oplus b$. The *material part* relation between the stuff constituting entities is denoted by $\top$:

$$a \Pi b \Rightarrow a \top b$$

So $+$ and $\top$ are defined on the whole domain of individuals E, but their semantics depend on D, the set of portions of matter, which is a subset of A, the atoms of E. All entities which constitute another entity are members of D; members of D have themselves as constituents. D itself forms a semilattice under the $+$ operation, but it is not necessarily atomic with respect to $\top$. Although a set-theoretic model assumes the existence of atomic entities, a lattice structure need not, and thus can be adopted without specifying that mass terms have atomic parts.

Link claims that the distinction between objects and their constituent stuff provides a solution to the puzzle of how a sentence like (4) can be true.

(4) This ring is new but the gold which it is made out of is old.

Link would give this the following sort of representation:

(5) $\mathrm{old}(\iota x[\mathrm{gold}(x) \wedge x \triangleright a]) \wedge \neg \mathrm{old}(a)$(where a is the ring)

New can be true of the object and old can be true of the stuff out of which the object is made; the object and the stuff out of which it is made are not identical but are related by the constitution relation.

One problem with LPM is that it does not give any indication of how to treat mass terms such as *furniture*, which are not so obviously true of stuff as mass nouns like *gold*. It does not seem appropriate to claim that *this furniture* constitutes *these chairs* but even if this were assumed we could not then specify that *this wood* constituted *this furniture* (without making the entities identical). Bach (1986) came to the conclusion that even considering more normal mass nouns than *furniture*, an indefinite number of levels of constituency would be needed. His example is of a snowman, which is made out of snow, which is made out of water molecules. The H_2O molecules may be old, yet the snow is new.

I do not think that iterating constituency would salvage the theory however, since the interpretation of the problem sentence is apparently more complex than suggested by Link's analysis. On my interpretation it corresponds to something like (6):

(6) This ring was made recently out of gold that was mined a long time ago.

The interpretation cannot be that the gold considered purely as stuff is old, since considered in terms of its atoms all gold is of roughly the same age. So we are not really applying *old* to the stuff at all; really *old* is being applied to some previous object (the nugget that was mined, some previous artifact, or whatever). This suggests that we could currently equate the ring with the gold from which it is made, although we will have to be careful when introducing predicates such as *old* and *new* which may refer back to a time when the object did not exist, but the stuff which constitutes it did.

Link's puzzle seems to belong in the larger class of examples which have to be treated by allowing predicates to take context-dependent aspects of an entity as arguments. Obviously some utterances make distinctions between an artifact and its constituents, but there are other aspects of an object that we can distinguish besides constituency. Landman (1989) considers the example of a judge who is also the hangman, but where it is possible to state (7a) without implying (7b).

(7) a The judge is on strike.
  b The hangman is on strike.

Taking Link's approach to its logical conclusion we would have to assume that each of these aspects is realised as a distinct entity in the model, and such multiplication of entities is highly unsatisfactory. Landman argues that the judge/hangman example would still apply even if all judges were necessarily also hangmen, and thus considering intensions rather than extensions of predicates is not sufficient to avoid the problem.

Within a model-theoretic framework solutions to such problems have been suggested which involve property theory (Chierchia and Turner, 1988) or intentional properties (Landman, 1989). Applying Landman's suggestions to the *old gold, new ring* puzzle we would get an interpretation where the entity considered in its aspect as a ring was new, but considered in its aspect as gold was old. Note that this is not the same as interpreting the adjective relative to the predicate; it would distinguish the two aspects even if the criteria for judging oldness and newness were the same for rings and gold. I will not discuss this issue further here, since the only point that is relevant is that Link's distinction between individual objects and the stuff that constitutes them is not adequately motivated, since it does not account for the problem which it was intended to solve. For my current purposes I will assume a purely extensional model-theoretic treatment.

5.1.2 Krifka's theory of nominal reference

Krifka (1987) assumes that objects, including quantities of matter, can be characterised by a predicate O, the extension of which has the structure of a complete, complementary, join semilattice without the 0-element. I will adopt his notation, $\sqcup_o$ for join (*osum*), $\sqsubseteq_o$ for part and $\sqsubset_o$ for proper part. Krifka introduces various higher order predicates and relations to characterise different reference types. The most important of these for my purposes are given here.

Cumulative reference A predicate P has cumulative reference if when P is true of two entities it is true of the join of the two entities.

$$\forall P[\mathrm{CUM_o}(P) \iff \forall x, y[P(x) \wedge P(y) \Rightarrow P(x \sqcup_o y)]]$$

Mass nouns like **gold′** and plurals like **horses′** have cumulative reference.

Quantised reference A predicate P has quantised reference if when P is true of an entity it is not true of any proper part of that entity.

$$\forall P[\mathrm{QUA_o}(P) \iff \forall x, y[P(x) \wedge P(y) \Rightarrow \neg(y \sqsubset_o x)]]$$

Singulars like **horse′** and measured objects like **(five ounces of gold)′** have quantised reference.

Singular reference A predicate P has singular reference if it is true only of a single entity.

$$\forall P[\mathrm{SNG_o}(P) \iff \exists x[P(x) \wedge \forall y[P(y) \Rightarrow x = y]]]$$

Except in the case where a predicate P has singular reference it cannot be both cumulative and quantised.

P-atom x is a P-atom iff x is P and there is no proper part of x which is P.

$$\forall P, x[\mathrm{ATOM_o}(x, P) \iff P(x) \wedge \neg\exists y[y \sqsubset_o x \wedge P(y)]]$$

Atomic reference If P has atomic reference then every x which is P is either a P-atom itself or has a P-atom as a proper part.

$$\forall P[\mathrm{ATM_o}(P) \Rightarrow \forall x[P(x) \Rightarrow \exists y[y \sqsubseteq_o x \wedge \mathrm{ATOM_o}(y, P)]]]$$

It is important to note that the definition of atomicity is with respect to a particular predicate. This an essential difference in the formal apparatus between this theory and Link's LP. Link defines atomicity with respect to Π but Krifka remains neutral on whether the semilattice is atomic with respect to $\sqsubseteq_o$. Krifka's join and part operators are used in such a way as to make them operators on material rather than individuals; this means that Krifka's theory has closer parallels with mereological theories than Link's does. However there is still no question of making the denotation of predicates anything other than a set. [5]

Krifka gives the following representations for phrases with specified quantities:

five ounces of gold
$\lambda x[\mathbf{gold'}(x) \wedge \mathbf{ounce'}(x) = 5]$
five head of cattle
$\lambda x[\mathbf{cattle'}(x) \wedge \mathrm{NU}(\mathbf{cattle'})(x) = 5]$
five cows
$\lambda x[\mathbf{COW}(x) \wedge \mathrm{NU}(\mathbf{COW})(x) = 5]$

These are similar in that they are all treated as though a measure phrase is applied to the head noun. The measure function may be specified directly (**ounce′** for example) but in the case of classifier constructions and ordinary count noun constructions it will depend on the head noun. In this case, the function symbol NU is applied to the nominal predicate to yield a measure function. Krifka discusses the properties that are necessary for measure functions and the conditions under which a measure phrase is well formed. The representation of *five cows* uses **COW** which is assumed to be a nominal predicate similar to **cattle′**, which underlies the count noun *cow*, but with no direct representation in English.

Krifka makes use of similar lattice operations to Link. However, the underlying ontology of his theory is very different. In Krifka's theory, the lattice operations do represent material constituency. The domain of matter is structured by the join and part-of relationships. No concept of atomic individual is built into the model. Certain predicates, normally corresponding to singular count nouns, will only be true of entities that correspond to what we normally think of as singular individuals, but there is nothing to prevent an entity from being singular with respect to one predicate and plural with respect to another. Other predicates, corresponding to mass nouns, will not carry any connotations of individuality at all. A particular entity in the semilattice might equally well be referred to by a mass noun, or a singular or plural count noun.

Krifka's theory increases the complexity of some inferences since it is necessary to consider individuation relative to a particular predicate, but this is reasonable, it provides some indication of why it is difficult to count the 'things' on my desk, for example. The examples of inconsistency in the mass/count distinction, given at the beginning of this chapter, show that we cannot assume that there is some consistent notion of individuality. A theory such as Krifka's, which makes individuation dependent on particular predicates, accounts for this much more naturally than a theory in which atomicity is built into the model.

I will adopt the basic treatment described by Krifka here, but instead of decomposing count nouns so that the individuating natural units are distinct I will treat the predicates which are denoted by lexicalised singular count nouns as atomic, and form plurals by a closure operator

[5] Bunt (1985) gives an alternative approach to the treatment of mass terms which also avoids the assumption of atomic individuals. He introduces a theory of *ensembles* which takes the concept of 'part-of' as fundamental. Ensembles need not have atomic parts — sets are, in effect, the special case of ensembles which do have atomic parts. This approach is elegant in that many of the set operations are special cases of the more general ensemble operations (subset is a special case of part-of, for example). But, for current purposes at least, this is not necessarily an advantage, since the structure of the domain of entities (given here by the lattice operations) is quite distinct from the denotation of predicates (which are sets of entities in the standard way).

equivalent to Link's. The reasons behind this modification are discussed in Copestake (1990a); essentially the problem is that if the entire burden of individuation is carried by a measure function, it is apparently not possible to adequately reflect the integrity of individuals.

5.2 Individuation and qualia structure

In the current type system information which is relevant to individuation appears in three places in the structure of the sign. The presence or absence of the closure operator and the specification of predicates as cumulative or quantised are encoded in the semantic structure. Agreement is specified as a property of the indices in the semantic structure, and fine-grained individuation distinctions are made in the qualia structure.

The type for the semantics of object denoting nominals is **obj-noun-formula**, given in Figure 5.1. This type introduces two features PLMOD and QUANT which both take boolean values.

$$\left[\begin{array}{l} \textbf{obj-noun-formula} \\ \text{IND} = \boxed{0} \left[\begin{array}{l} \textbf{obj} \\ \text{AGR} = \left[\begin{array}{l} \textbf{agr} \\ \text{NUM} = \textbf{number} \end{array}\right] \end{array}\right] \\ \text{PRED} = \textbf{logical-pred} \\ \text{ARG1} = \boxed{0} \\ \text{PLMOD} = \textbf{boolean} \\ \text{QUANT} = \textbf{boolean} \end{array}\right]$$

Figure 5.1: Noun semantic type

Plural modification is specified by giving a value **true** to the feature PLMOD, which will otherwise be **false** and a predicate is specified to have quantised or cumulative reference by specifying QUANT to be **true** or **false**. I have ignored the possibility of singular reference in the current type system.

Agreement is specified on indices, following the HPSG approach (Pollard and Sag, in press). The specification of information on an index is not to be taken as indicating that the real-world entities themselves have properties such as gender and number. In the system described here, agreement does not correspond directly with plurality in the formal semantics, that is with the use of the plural modifier on predicates. Agreement is more closely linked to the semantic classification expressed in the qualia structure, which in turn may be partially determined by real world properties of the entity denoted. The `< FORM : RELATIVE >` specification in the qualia structure encodes a finer-grained distinction. The general type is **form** which has two immediate subtypes: **countable** and **uncountable**. The leaf types **individual**, **pair**, **complex**, **group**, **portion** and **plural** are subtypes of **countable**; **mass** and **collection** are subtypes of **uncountable**.

The type for lexically specified nouns **lex-noun-sign** has subtypes, **lex-count-noun** and **lex-uncount-noun** shown in Figures 5.2 and 5.3. (These do not apply to relational nouns which are treated separately, see Section 5.4.) Finer distinctions between lexical classes are encoded in LAUREL as psorts with one psort corresponding to each of the possible relative form types in the qualia structure. Lexical entries for nouns are specified as inheriting information from one of the eight possible psorts. Psorts are used rather than types here because some of the information associated with them is defeasible. The psorts are described below.

5.2.1 Lexical specification of individuation classes

Individual Most count nouns fall into this class, which is defined in LAUREL as follows:

```
lex-individual
    < > = lex-count-noun
    < CAT : M-FEATS : REG-MORPH > = true
    < QUALIA : FORM : RELATIVE > = individual
    < SEM : IND : AGR : NUM > = sg
    < SEM : PLMOD > = false
    < SEM : QUANT > = true .
```

$$\left[\begin{array}{l}\textbf{lex-count-noun}\\ \text{ORTH} = \textbf{orth}\\ \text{CAT} = \boxed{\textbf{noun-cat}}\\ \text{SEM} = \left[\begin{array}{l}\textbf{obj-noun-formula}\\ \text{IND} = \boxed{0}\left[\begin{array}{l}\textbf{obj}\\ \text{AGR} = \left[\begin{array}{l}\textbf{agr}\\ \text{NUM} = \textbf{number}\end{array}\right]\end{array}\right]\\ \text{PRED} = \boxed{1}\ \textbf{logical-pred}\\ \text{ARG1} = \boxed{0}\\ \text{PLMOD} = \textbf{boolean}\\ \text{QUANT} = \textbf{boolean}\end{array}\right]\\ \text{QUALIA} = \left[\begin{array}{l}\textbf{nomqualia}\\ \text{AGENTIVE} = \textbf{nomagent}\\ \text{TELIC} = \boxed{\textbf{verb-sem}}\\ \text{FORM} = \left[\begin{array}{l}\textbf{nomform}\\ \text{ABSOLUTE} = \textbf{real-form}\\ \text{RELATIVE} = \textbf{countable}\end{array}\right]\\ \text{CONSTITUENCY} = \boxed{\textbf{nomconst}}\\ \text{OBJECT-INDEX} = \boxed{0}\end{array}\right]\end{array}\right]$$

Figure 5.2: Lexical type for count nouns

$$\left[\begin{array}{l}\textbf{lex-uncount-noun}\\ \text{ORTH} = \textbf{orth}\\ \text{CAT} = \boxed{\textbf{noun-cat}}\\ \text{SEM} = \left[\begin{array}{l}\textbf{obj-noun-formula}\\ \text{IND} = \boxed{0}\left[\begin{array}{l}\textbf{obj}\\ \text{AGR} = \left[\begin{array}{l}\textbf{agr}\\ \text{NUM} = \textbf{number}\end{array}\right]\end{array}\right]\\ \text{PRED} = \textbf{logical-pred}\\ \text{ARG1} = \boxed{0}\\ \text{PLMOD} = \textbf{boolean}\\ \text{QUANT} = \textbf{false}\end{array}\right]\\ \text{QUALIA} = \left[\begin{array}{l}\textbf{nomqualia}\\ \text{AGENTIVE} = \textbf{nomagent}\\ \text{TELIC} = \boxed{\textbf{verb-sem}}\\ \text{FORM} = \left[\begin{array}{l}\textbf{nomform}\\ \text{ABSOLUTE} = \textbf{real-form}\\ \text{RELATIVE} = \textbf{uncountable}\end{array}\right]\\ \text{CONSTITUENCY} = \boxed{\textbf{nomconst}}\\ \text{OBJECT-INDEX} = \boxed{0}\end{array}\right]\end{array}\right]$$

Figure 5.3: Lexical type for uncount nouns

$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{pluralise}} \\
0 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-noun-sign}} \\
\mathrm{ORTH} = & \textbf{orth} \\
\mathrm{CAT} = & \boxed{\textbf{noun-cat}} \\
\mathrm{SEM} = & \left[\begin{array}{l}
\textbf{obj-noun-formula} \\
\mathrm{IND} = \boxed{0} \left[\begin{array}{l} \textbf{obj} \\ \mathrm{AGR} = \left[\begin{array}{l} \textbf{agr} \\ \mathrm{NUM} = \textbf{pl} \end{array}\right] \end{array}\right] \\
\mathrm{PRED} = \boxed{1}\ \textbf{logical-pred} \\
\mathrm{ARG1} = \boxed{0} \\
\mathrm{PLMOD} = \textbf{true} \\
\mathrm{QUANT} = \textbf{false}
\end{array}\right] \\
\mathrm{QUALIA} = & \left[\begin{array}{l}
\textbf{nomqualia} \\
\mathrm{AGENTIVE} = \boxed{2}\ \textbf{nomagent} \\
\mathrm{TELIC} = \boxed{3}\ \boxed{\textbf{verb-sem}} \\
\mathrm{FORM} = \left[\begin{array}{l} \textbf{nomform} \\ \mathrm{RELATIVE} = \textbf{plural} \end{array}\right] \\
\mathrm{CONSTITUENCY} = \boxed{5}\ \boxed{\textbf{nomconst}} \\
\mathrm{OBJECT\text{-}INDEX} = \boxed{0}
\end{array}\right]
\end{array}\right] \\
1 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-noun-sign}} \\
\mathrm{ORTH} = & \textbf{orth} \\
\mathrm{CAT} = & \boxed{\textbf{noun-cat}} \\
\mathrm{SEM} = & \left[\begin{array}{l}
\textbf{obj-noun-formula} \\
\mathrm{IND} = \boxed{6} \left[\begin{array}{l} \textbf{obj} \\ \mathrm{AGR} = \left[\begin{array}{l} \textbf{agr} \\ \mathrm{NUM} = \textbf{number} \end{array}\right] \end{array}\right] \\
\mathrm{PRED} = \boxed{1} \\
\mathrm{ARG1} = \boxed{6} \\
\mathrm{PLMOD} = \textbf{false} \\
\mathrm{QUANT} = \textbf{true}
\end{array}\right] \\
\mathrm{QUALIA} = & \left[\begin{array}{l}
\textbf{nomqualia} \\
\mathrm{AGENTIVE} = \boxed{2} \\
\mathrm{TELIC} = \boxed{3} \\
\mathrm{FORM} = \boxed{\textbf{nomform}} \\
\mathrm{CONSTITUENCY} = \boxed{5} \\
\mathrm{OBJECT\text{-}INDEX} = \boxed{6}
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Figure 5.4: Type for plural formation rules

Thus the lexical entry for *rabbit* might be as follows:

rabbit L_1_1

```
   < > < lex-individual < >
   < QUALIA > = animal.
```

The first line specifies that the entire entry is to be regarded as inheriting by default from the psort, but the type of the expanded entry is **lex-count-noun** because of the constraint on specification of inheritance that the type of the psort should be equal to the type of the resulting entry, or a supertype of it.

Plural This class of nouns includes all semantically regular plurals, for example, *rabbits*, *tables* and *children*. The general type for plural formation rules is shown in Figure 5.4, actual rules such as that corresponding to suffixation with *+s*, can be specified in terms of this type. The feature structure corresponding to the plural *rabbits* is partially shown in Figure 5.5. The predicate **rabbit_L_1_1** is read as being modified by the plural operator to give ***rabbit_L_1_1**. The predicate is stated to be cumulative and the agreement to be plural.[6]

Morphologically irregular plurals, such as *children*, can be given lexical entries of the following form:

children 1

```
   < > < ( child_1 + plural) < >.
```

[6] Unfortunately, this feature structure reveals a problem with the current system, in that the type of the qualia structure for the plural is underspecified, although the components of the structure (not shown) will be identical to those in the input entry, with the exception of the relative form specification. This is discussed further in Chapter 8.

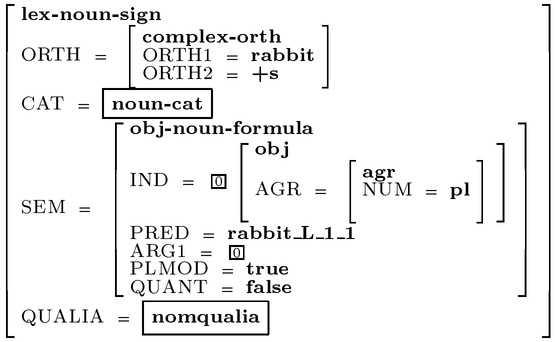


Figure 5.5: Feature structure for *rabbits*

The value for the orthography specified by the lexical rule will be overridden by that implicitly specified in the entry. For lexicalised plural nouns such as *people* which cannot be regarded as related to any singular forms, the psort `lex-plural` is used.

```
lex-plural
   < > = lex-count-noun
   < QUALIA : FORM : RELATIVE > = plural
   < SEM : QUANT > = false
   < SEM : IND : AGR : NUM > = pl.
```

Group The psort `lex-group` is only intended to cover the use of group nouns in examples such as (8a) where the group is treated as a single entity, as opposed to (8b), where the group noun behaves in a way which is very similar to a plural.

(8) a That team was promoted to the First Division last season.
b One of the team was given a medal.

Group nouns are considered in detail in section 5.3.

```
lex-group
   < > < lex-individual < >
   < QUALIA : FORM : RELATIVE > = group
   < QUALIA : CONSTITUENCY > = groupconst.
```

Pair The nouns which fall into this category all denote entities which have a physical structure which could be described as bipartite. This includes garments, such as *trousers*, instruments such as *binoculars* and *scales*, and tools such as *scissors*. They are morphologically plural and usually occur in compounds in a singular form, e.g. *trouser leg*. They all take plural agreement and in some contexts have to occur with the classifier *pair*, for example *a pair of scissors* rather than **a scissors*. However the unmodified predicates (e.g. **scissors**′ as opposed to ⋆**scissors**′) are taken as having quantised reference, which will correctly predict that *one of the scissors* is unacceptable if *the scissors* denotes a single pair of scissors.

```
lex-pair
   < > < lex-individual < >
   < QUALIA : FORM : RELATIVE > = pair
   < SEM : IND : AGR : NUM > = pl .
```

Complex This is a somewhat ad-hoc category consisting of nouns which denote complex entities which may take singular or plural agreement, for example *gallows*, *barracks* and *works*. In other respects they do not differ greatly from ordinary individual denoting nouns.

```
lex-complex
   <> < lex-individual <>
   < QUALIA : FORM : RELATIVE > = complex
   < SEM : IND : AGR : NUM > == num.
```

Mass This class includes the 'normal' mass nouns such as *gold*, *sand* and *fennel.* They have singular agreement and are cumulative (and thus cannot be pluralised).

```
lex-mass
   <> = lex-uncount-noun
   < QUALIA : FORM : RELATIVE > = mass
   < SEM : IND : AGR : NUM > = sg.
```

There are a few examples of nouns which would traditionally be described as *pluralia tantum*, which take plural agreement but otherwise behave more or less identically to mass nouns. One example is *oats*, the singular form *oat* only occurs in compounds and *oats* itself is not countable (e.g. **three oats*). The lexical entry for *oats* is therefore similar to that of an ordinary mass noun, but with agreement specified as **pl**. Since `lex-mass` is a psort rather than a type the value for agreement can be overridden.

```
oats L_0_1
   < > = lex-mass < >
   < SEM : IND : AGR : NUM > = pl.
```

Collection Mass nouns such as *furniture*, *clothing* and *cutlery* refer to possibly heterogeneous collections of objects with a common function.

```
lex-collection
   < > < lex-mass < >
   < QUALIA : FORM : RELATIVE > = collection.
```

Portion I distinguish between ordinary individuals and portions which are conventionally measured amounts of a substance, but which have no inherent individual properties. Examples include the extended senses of *whisky*, *beer* and so on, which denote an amount of the drink. In such cases the individuating unit is given by convention or context and may be relatively underspecified. The portioning rule is discussed in more detail below.

To summarise we have the following possibilities for lexical entries:

psort	agr	CUM/QUA	form (relative)
`lex-individual`	**sg**	QUA	**individual**
`lex-plural`	**pl**	CUM	**plural**
`lex-pair`	**pl**	QUA	**pair**
`lex-group`	**sg** (**num**)	QUA	**group**
`lex-complex`	**num**	QUA	**complex**
`lex-portion`	**sg**	QUA	**portion**
`lex-mass`	**sg** (**pl**)	CUM	**mass**
`lex-collection`	**sg**	CUM	**collection**

The agreement possibilities specified in brackets may apply to individual lexical entries in that class, overriding the normal agreement. The description is incomplete in that it omits some classes of noun, in particular those which refer generically or to unique individuals (e.g. *mankind* and *cosmos*).

Various constructions can be regarded as causing a shift between classes. The rule for plural formation which was given above, takes nouns in any of the quantised classes and converts them to plurals. Portioning converts **mass** to **portion**. All the cumulatively denoting classes can be quantised by the specification of a quantity or form. For example, *three apples*, *three pounds of apples*, *a piece of metal* and *a wedge of cheese* are all quantised. Examples such as the last three will be described in section 5.4.

Some qualia types restrict the possible values of `< FORM : RELATIVE >`. The qualia type **human** specifies that `< FORM : RELATIVE >` is **individual**, **plural** or **group**. The effect of this is to block lexicalised predicates which refer to humans as unindividuated masses, for example. This prediction can be checked to some extent using LDOCE, since the semantic coding scheme specifies human denoting nouns. If we do this we find that there are some human denoting nouns which are specified as uncount, such as *mankind* and *humanity.* A separate category is needed for examples such as these which refer generically, but they are distinct from ordinary mass terms (contrast *some furniture* with **some mankind*). [7] In fact, the same generalisation seems to apply to animals, with the only genuine exception found by querying LDOCE being *game*^{1} *4*, which, although it can denote live animals, clearly refers to them as potential food.

Other generalisations can be made, but these are somewhat more tentative, because they are more difficult to test on LDOCE. The qualia type **substance** is restricted to values of **mass** or **portion** for `< FORM : RELATIVE >`, which prevents lexical entries for liquids, gases and homogeneous solids from individuating except as portions.

For some purposes, a description of 'absolute' form is also needed in the qualia structure. This should have a specification which is dependent only on the nature of the particular entity involved and independent of the predicate used to describe it. One starting point is to allow entities to be described as individuated, unindividuated or indeterminate. Entities such as humans, animals (at least vertebrates) and instruments, which depend on their physical integrity to function, would be described as individuated, and substances such as gold, water and air, which are not in (obvious) physically distinct units would be unindividuated. Many entities, including many plants, foodstuffs and so on, would be classified as indeterminate, because they have no well-defined mode of individuation, but nevertheless are not homogeneous. As an alternative to directly restricting the relative form of types such as **human**, the generalisations could be expressed in terms of absolute form.

An absolute notion of form would be involved in selectional restrictions, on the assumption that these depend on the denotation alone. For example, it has sometimes been suggested that shape-denoting adjectives such as *round*, *square*, *cuboid*, *linear* should be classified as count, and blocked from modifying mass nouns, because of the oddity of phrases such as ?*round gold.* But this would incorrectly predict *square furniture* to be odd while *a square whisky* would be accepted. Within the treatment sketched here, such a classification would be made on the basis on absolute physical form. However I will not pursue this issue further here, since to come up with a scheme that was not ad-hoc would require examination of a considerable amount of data.

5.2.2 Grinding and portioning

Given this framework, the rules for grinding and portioning introduced in the previous chapter can be reexamined. The lexical rules are repeated here as Figures 5.6 and 5.7, with the formal semantic structure emphasised rather than the qualia structure.

The portioning rule can be characterised as turning a **mass** noun into a **portion** noun, by modifying the predicate so that it is quantised in a minimal way. Apart from the specification of relative form, the qualia structure of the noun which results from the application of the rule should be identical to that of the input. The predicate modifier **portionof** can be formalised as specifying that a conventional measure function, CM, applies as well as the modified predicate (compare Krifka's natural units). Thus, according to this characterisation, all that is implied by portioning is that some conventionally (or even contextually) specified quantity of the mass is

[7] Possible counter-examples to this generalisation are the extended uses of words such as *trash* and *scum* to refer to humans. The *Collins COBUILD English Language Dictionary* (Sinclair, 1987) gives *trash* as uncount, but LDOCE describes it as plural, which is confirmed by COBUILD's example ... *the assorted trendy trash who fill the wine bars and bistros.* This does not completely preclude the possibility that it is uncount, but my feeling is that individuation is possible at least in phrases such as *a few of the trash* (compare **a few of the furniture*).

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lexical-rule}} \\
0 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-count-noun}} \\
\text{ORTH} = & \boxed{0}\ \textbf{orth} \\
\text{CAT} = & \boxed{\textbf{noun-cat}} \\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj-noun-formula}} \\
\text{IND} = & \boxed{1}\ \textbf{obj} \\
\text{PRED} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{modified-pred}} \\
\text{MODIFIER} = & \textbf{portionof} \\
\text{MODIFIED} = & \boxed{4}\ \textbf{logical-pred}
\end{array}\right] \\
\text{ARG1} = & \boxed{1} \\
\text{PLMOD} = & \textbf{false} \\
\text{QUANT} = & \textbf{true}
\end{array}\right] \\
\text{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{comestible}} \\
\text{FORM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{nomform}} \\
\text{RELATIVE} = & \textbf{portion}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
1 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-uncount-noun}} \\
\text{ORTH} = & \boxed{0} \\
\text{CAT} = & \boxed{\textbf{noun-cat}} \\
\text{SEM} = & \boxed{\textbf{obj-noun-formula}} \\
\text{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{comestible}} \\
\text{FORM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{nomform}} \\
\text{RELATIVE} = & \textbf{mass}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

Figure 5.6: Portioning rule

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{grinding}} \\
0 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-uncount-noun}} \\
\text{ORTH} = & \boxed{0}\ \textbf{orth} \\
\text{CAT} = & \boxed{\textbf{noun-cat}} \\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj-noun-formula}} \\
\text{IND} = & \boxed{1} \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj}} \\
\text{AGR} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{agr}} \\
\text{NUM} = & \textbf{sg}
\end{array}\right]
\end{array}\right] \\
\text{PRED} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{modified-pred}} \\
\text{MODIFIER} = & \textbf{ground} \\
\text{MODIFIED} = & \boxed{3}\ \textbf{string}
\end{array}\right] \\
\text{ARG1} = & \boxed{1} \\
\text{PLMOD} = & \textbf{false} \\
\text{QUANT} = & \textbf{false}
\end{array}\right] \\
\text{QUALIA} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{physical}} \\
\text{AGENTIVE} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{agentivestuff}} \\
\text{ORIGIN} = & \boxed{3}
\end{array}\right] \\
\text{FORM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{nomform}} \\
\text{RELATIVE} = & \textbf{mass}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
1 = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{lex-count-noun}} \\
\text{ORTH} = & \boxed{0} \\
\text{CAT} = & \boxed{\textbf{noun-cat}} \\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj-noun-formula}} \\
\text{IND} = & \boxed{4} \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj}} \\
\text{AGR} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{agr}} \\
\text{NUM} = & \textbf{sg}
\end{array}\right]
\end{array}\right] \\
\text{PRED} = & \boxed{3} \\
\text{ARG1} = & \boxed{4} \\
\text{PLMOD} = & \textbf{false} \\
\text{QUANT} = & \textbf{true}
\end{array}\right] \\
\text{QUALIA} = & \boxed{\textbf{physical}}
\end{array}\right]
\end{array}\right]$$

Figure 5.7: Individuation effects of the grinding rule

involved.

$$\forall x[\mathbf{portionof}(P)(x) \Rightarrow P(x) \wedge CM(P)(x) = 1]$$

It is possible to pluralise portions (e.g. *three beers*) by applying the closure operator to the predicate in the normal way. Krifka (1987) discusses various conditions which measure functions have to meet to behave appropriately, for example, that they map to positive real numbers and that additivity holds:

Additivity A measure function, M, is said to be *additive* iff:

$$\forall x, y[\neg x o_o y \Rightarrow M(x) + M(y) = M(x \sqcup_o y)]]$$

(where overlap, o_o, can be defined in terms of $\sqsubseteq_o$, $x o_o y$ iff $\exists z[z \sqsubseteq_o x \wedge z \sqsubseteq_o y]$)

I assume that these conditions apply to $CM(P)$ and therefore we get the desirable result that, for example, two beers plus three beers equals five beers. However the precise quantity corresponding to a portion of something is assumed to be contextually rather than lexically specified.

Grinding is assumed to operate on true individuals, that is lexical entries with `< FORM : RELATIVE >` of **individual**, **group**, **complex** or **pair** but not **portion**, **collection** or **mass**. [8] In contrast to the portioning rule, which involves limiting the denotation of a mass noun (e.g. *a beer* is *beer*), grinding involves a distinct shift in meaning (e.g. *lamb* is not *a lamb*). The most that we can assume in general about grinding is that the ground substance is 'stuff' that at some past time was part of the original entity. This relationship between the modified predicate **ground**(P) and P can be formalised as follows, if we make the additional assumption that predicates are true of entities at particular times.

$$\forall x, t[\mathbf{ground}(P)(x, t) \Rightarrow \exists y, t'[{}^{*}P(y, t') \wedge t' \leq t \wedge x \sqsubseteq_o y]]$$

5.3 Group terms

Group nouns, such as *band*, *crowd*, *quartet*, *flock*, *management* and *group* itself, are distinctive in English in that, when morphologically singular, they behave in some respects like singular nouns and in others like plurals. This manifests itself in several ways:

1. Singular or plural pronouns can be used:

   (9) The band played well tonight. Its/their tour has sold out.

2. Either singular or plural agreement with the verb is possible (plural agreement with group nouns is, in general, less common in American English):

   (10) That band play/plays well.

   In the case of group nouns which denote groups of humans, a relative clause is introduced by *who* if plural agreement is used, and by *which* if it is not:

   (11) The band who get/*gets top billing at the festival receive/*receives £20,000.
   The band which gets/*get top billing at the festival receives/*receive £20,000.

3. Individual members can be referred to by using *one of* etc.

[8] Another way of looking at this is that grinding can apply to portions, but would be equivalent to the reversal of the portioning rule. Grinding results in a **mass** and destroys any lexical specification of form, but **portion**s by definition are **mass**es which have a minimally specified form.

(12) One of the band smashed her guitar.

The final criterion distinguishes between group nouns, and those such as *barracks* and *gallows* which can take either singular or plural agreement (when referring to the same entity). Note also that unpluralised group nouns always take a singular determiner even if verbal agreement is plural.

(13) This barracks is/*are new.
These barracks are/*is new.
That band has/have been playing well.

Many group nouns can behave relationally, as discussed in section 5.4, below. In such cases a prepositional phrase with *of* specifies the members of the group (*crowd of football supporters*, *flock of sheep* etc.).

However there are other nouns which do not behave in this way, even though they refer to entities which can be regarded as being made up of several discrete individuals. For example consider the LDOCE definition of *dolmen*:

dolmen a group of upright stones supporting a large flat piece of stone, built in ancient times in Britain and France

Despite the fact that a *dolmen* can be regarded as a group of entities, it does not behave as a group noun; the following are all unacceptable:

(14) a The dolmen is on a mountain. *They're very eroded.
b *The dolmen have fallen down.
c *One of the dolmen fell down.

There is clearly a semantic distinction between group and non-group nouns; when a group noun is used the individual components of the entity denoted are sufficiently obvious that it can be referred to as though it were a plural term.

Collectives such as *terrace* and *range*, which denote groups of entities of a particular type and which usually appear with *of* phrases (*terrace of houses*, *range of mountains*), do however share some of the behaviour of group nouns. These nouns always take singular agreement when morphologically singular (at least when the *of* phrase is absent), and thus are not group nouns by the first two tests, but they can meet the third, although possibly only in in contexts where the individual members are explicitly mentioned.

(15) The house was one of a terrace.
? One of the terrace had a green front door.

The first example is taken from the Lancaster-Oslo/Bergen (LOB) corpus: when checking for *one of* followed by a morphologically singular noun phrase, this was the only example where the head of the NP was not a group noun by the agreement tests. Contrast *terrace* with *group*, which is not at all limited in the semantic type of its *of* complement (e.g. *group of houses*, *group of statistics*, *group of actions*), but which refers to people when used without the *of* phrase in an unmarked context.

The singular/plural dual behaviour of true group nouns to some extent corresponds to whether the predicate is seen as applying to the group as a whole or to its individual members.

(16) The band was formed in 1977.
The team were killed in a plane crash.

There is a tendency for singular agreement to be used when the group as an entity is referred to, and for plural agreement to be used when the individuals are concerned. The examples above are odd when the agreement is changed:

(17) ? The band were formed in 1977.
? The team was killed in a plane crash.

Sometimes differences in agreement alone suggest a semantic distinction. In (18a) the implication is that the committee as an entity gets the money, (18b) suggests that it goes to the individual members and there is a possible distributive reading, forced in (18c). Note that (18d) is odd.

(18) a The committee gets £20,000 per annum.
b The committee get £20,000 per annum.
c The committee get £20,000 per annum each.
d ? The committee gets £20,000 per annum each.

However, plural agreement with verb phrases which apparently refer to the group as a single entity is quite normal in some contexts, such as when referring to sports teams or clubs.

(19) Forfar are a good side. (LOB corpus)
But there was to be no bargaining [on players' contracts] as far as the club were concerned. (*The Guardian*)

It is useful to distinguish between ordinary group nouns and those which refer generically, since the latter class involve some different problems which I will not discuss here. Ordinary group nouns form plurals in the normal way, (e.g. *bands*, *crowds*, *quartets*, *flocks*), but others do not normally form plurals because they refer generically (e.g. *aristocracy*, *clergy*), or to an entity usually regarded as unique (e.g. *admiralty*), although plurals are possible in phrases such as *the admiralties of England and France.* The generic group nouns have dual group-entity/plural behaviour, but their plural behaviour parallels that of bare plural noun phrases in 'universal' position, in contrast to normal group nouns. For example (20a) implies (20b) but (20c) does not imply (20d), but only (20e).

(20) a The clergy are badly paid.
b Clergymen are badly paid.
c The committee are badly paid.
d Committeemen are badly paid.
e The committeemen are badly paid.

It appears that group nouns in English always refer to entities whose individual members are seen as capable of independent, agentive action, usually humans or other 'higher' animals. I have tested this assumption in a preliminary way using LDOCE: group nouns can, in theory, be retrieved as a class quite simply, since they are marked in the dictionary by the grammar codes GC (group countable, e.g. *committee*) or GU (group uncountable, e.g. *Admiralty*). Unfortunately, the LDOCE coding is far from comprehensive in this case (*army*, *assembly*, *band*, *coven* have no senses marked as being group nouns, for example) and the GU code has been given to a considerable number of entries which would not be characterised as group nouns by the tests given above, especially plural forms such as *letters* and *tactics.* I thus considered only nouns with grammar code GC, and excluded the morphologically plural forms *games*, *Olympic Games* and *vibes*, which also do not meet all the tests. The remaining senses all refer to collections of humans or human organisations, or (less frequently) animals, with the exceptions *fleet* and *convoy*, where the individual entities are ships.[9] There are some group nouns which can refer to collections of people or organisations which may themselves be groups;

league^2 **3** a group of sports clubs or players ...

Thus grouping is not restricted to a single level.

[9] There may be a connection here with the use of feminine gender personal pronouns when referring to ships, given that group nouns are normally associated with humans and higher animals.

This evidence suggests that, to provide an adequate lexical semantics for group nouns, information about their individual membership must be represented in order to allow appropriate semantic information to be associated with phrases such as *one of the band*, and to block or mark phrases such as *flock of football players*. It seems that lexicalisation of a concept of a collection of entities as a group noun is restricted to a small semantic class of entities, which I will provisionally limit to humans, organisations and animals, ignoring the ship examples for the time being, since further work is necessary to more precisely delimit the class.

5.3.1 Representing group denoting nouns

In what follows I will make the simplest assumption about the nature of the plural reading of group nouns, which is that it is equivalent to a normal plural. Thus in the current framework, the plural reading corresponds to an entity which can be regarded as the osum of the members of the group, and has a qualia structure appropriate for a normal plural entity. This straightforwardly accounts for the verbal and pronominal plural agreement, and for the use of partitives. On this assumption, we should get both distributive and collective plural readings. Distinguishing the group reading and the collective plural reading is not easy but there are examples such as (21a) which should probably be treated as a collective plural, since the predicate refers to individual members rather than the group entity. Similarly (21b) has a cumulative reading (Scha 1983), where the committee members are distributed in some unspecified way between the cars. [10]

(21) a The committee are arriving in a car.
b The committee are arriving in three cars.

I will assume that a group and the plural sum of its members are distinct entities and that the plural reading is produced from the group reading by a process of logical metonymy. This allows for the examples of group nouns such as *club*, *committee* and *company* where the entity denoted seems to have an existence independent of its members, even when a purely extensional viewpoint is taken. One can imagine a club, for example, which currently has no members but nonetheless still exists as a legal entity. This is problematic for theories which make the representation the group entity dependent on its members, but does not pose any problems for the metonymic account. Note that this is therefore quite a different situation from the ring/gold constituency example, discussed in section 5.1.1, since physical objects such as rings cannot exist as objects while having no composition. [11]

The treatment of the group/members logical metonymy is very similar to that of the entity/event coercion discussed in Section 4.2. In this case, type coercion occurs in contexts where a plural entity is required. As before, I will implement this coercion in LAUREL with a unary rule. This raises the question of whether the interpretation of the group noun phrase is ambiguous rather than vague. This is more plausible for group nouns than for the object to event coercion since it is more difficult to find a clearcut example where there is a conjunction of a group denoting and an individual denoting predicate. In examples such as (23), a plural pronoun refers back to a

[10] I am grateful to Dick Crouch for suggesting these examples.

[11] The metonymic treatment is less plausible with group nouns such as *crowd* which cannot exist without members, and thus where the distinction between the group and its members has less justification on a purely extensional treatment. Landman (1989) provides an extensive discussion of groups in the context of the treatment of plurality. For example, Landman treats the collective reading of sentences such as (22) as involving a group, rather than the plural isum.

(22) John and Mary lifted a piano.

Landman makes a type distinction between a group and the plural sum of its members: groups are sets of sets of individuals. However, since group formation can iterate indefinitely, this leads to a proliferation of types. It also does not account for group nouns such as *club*, if clubs do not necessarily have members. A fuller discussion of this is outside my current scope, since Landman's approach is so closely connected with the treatment of distributivity. However the general account developed here is basically compatible with an approach where the relationship between the group and non-group readings is metonymic in some cases, but more direct in others.

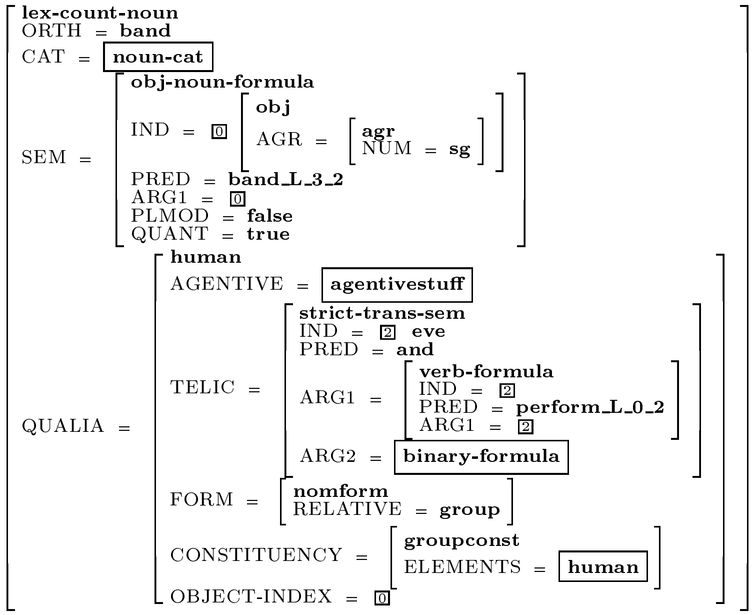


Figure 5.8: Lexical entry for *band*

singular entity, but pronominal reference is not a good test for ambiguity, because pronouns can refer to entities which are implicitly introduced, as in (24).

(23) The team was formed in 1977 but they were killed in a plane crash the next year.

(24) Mary's got engaged. He's very rich.

Examples involving a change in agreement, such as (25), are bad, which would be expected if *the team* is ambiguous, but this would also be accounted for within the unification based account if the noun phrase were initially underspecified with respect to agreement, since unification with the first VP would set it to **sg**.

(25) ? The team was formed in 1977 but were killed in a plane crash the next year.

So we could perhaps postulate a systematic polymorphism here, but, as before, I will leave this question open, and adopt a unary rule account.

An example entry for *band*, in the sense meaning a group of musicians, is shown in Figure 5.8. Here, I have assumed that number agreement is tightly correlated with the group/plural distinction, and thus agreement is specified as **sg**. Alternatively it could be underspecified as **num** to allow for examples, such as those given earlier, where plural agreement is used in sentences which involve an uncoerced group entity. The sign which would result from the application of the `group-to-plural` rule to *the band* is shown in Figure 5.9. This sign has obligatory plural agreement and denotes the plural entity which consists of the members of the band. The operator **membership** applies to the group entity, to give the plural entity. Similarly the qualia structure is that appropriate for a plural entity. The unary rule application is forced in contexts where a formally plural entity is required, for example a partitive construction, such as *one of.* When this is used with a group noun (e.g., *one of the band*), the unary rule is applied to the group to give a plural entity, and the appropriate specification of the individuals involved is then produced in much the same way as for the ordinary plural (e.g. *one of the members of the band*).

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{nominal-sign}} \\
\text{ORTH} = & \left[\begin{array}{l}\textbf{complex-orth}\\ \text{ORTH1} = \textbf{the}\\ \text{ORTH2} = \textbf{band}\end{array}\right] \\
\text{CAT} = & \boxed{\textbf{raised-np-cat}} \\
\text{SEM} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{binary-formula}}\\
\multicolumn{2}{l}{\text{IND} = \boxed{0}\ \textbf{entity}}\\
\multicolumn{2}{l}{\text{PRED} = \textbf{the}}\\
\text{ARG1} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{binary-formula}}\\
\text{IND} = \boxed{1} & \left[\begin{array}{l}\textbf{dummy-or-obj}\\ \text{AGR} = \left[\begin{array}{l}\textbf{agr}\\ \text{NUM} = \textbf{pl}\end{array}\right]\end{array}\right]\\
\multicolumn{2}{l}{\text{PRED} = \textbf{the\_1}}\\
\text{ARG1} = & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textbf{obj-noun-formula}}\\
\text{IND} = \boxed{2} & \left[\begin{array}{l}\textbf{obj}\\ \text{AGR} = \left[\begin{array}{l}\textbf{agr}\\ \text{NUM} = \textbf{sg}\end{array}\right]\end{array}\right]\\
\multicolumn{2}{l}{\text{PRED} = \textbf{band\_L\_3\_2}}\\
\multicolumn{2}{l}{\text{ARG1} = \boxed{2}}\\
\multicolumn{2}{l}{\text{PLMOD} = \textbf{false}}\\
\multicolumn{2}{l}{\text{QUANT} = \textbf{true}}
\end{array}\right]\\
\text{ARG2} = & \left[\begin{array}{l}\textbf{binary-formula}\\ \text{IND} = \boxed{1}\\ \text{PRED} = \textbf{membership}\\ \text{ARG1} = \boxed{1}\\ \text{ARG2} = \boxed{2}\end{array}\right]
\end{array}\right]\\
\text{ARG2} = \boxed{3} & \left[\begin{array}{l}\textbf{formula}\\ \text{IND} = \boxed{0}\\ \text{PRED} = \textbf{logical-pred}\\ \text{ARG1} = \textbf{sem}\end{array}\right]
\end{array}\right]\\
\text{QUALIA} = & \boxed{\textbf{human}}
\end{array}\right]$$

Figure 5.9: Coerced form of *the band*

The representation for the lexical semantics of group denoting nouns has to be compatible with the existing type system. Given the results above, which suggest that there are only restricted semantic classes of group nouns, it clearly would be inappropriate to parallel the entire existing lexical semantic type hierarchy with a group type hierarchy. Furthermore much of the information about group nouns will be comparable with that about their individual members. I therefore allow types such as **creature** to apply to both individuals and groups, and distinguish between the two mainly by specifying that the CONSTITUENCY feature either takes type **nongroupconst** or **groupconst**. Only the latter has ELEMENTS as an appropriate feature. The lexical semantics of the individuals making up a group noun are specified as the value of the ELEMENTS feature. In the case of *band* the individuals involved are *musicians*, thus the ELEMENTS slot is instantiated with the qualia structure corresponding to the pluralised form of `musician_L_0_1`. In Figure 5.8 this is shown unexpanded so only the type, **human**, is apparent.

This representation of semantic information about the individuals which comprise the group allows the plural-like aspects of the behaviour of group nouns to be accounted for. For example the `group-to-plural` rule specifies the qualia structure of the resulting plural entity according to the composition described in the CONSTITUENCY feature and the effect on the lexical semantics thus parallels the effect on the logical representation. The representation of the elements also allows collective phrases to be checked for compatability. For example a reading for *herd of buffalo* would be possible with the usual sense of *herd* because the elements would be specified to be of type **animal**, but *herd of voters* would require the metaphorical extended reading which applies to **human**s.

5.4 Relational nouns and individuation

To conclude this chapter I will briefly consider the following classes of phrases which can be regarded as having an effect on individuation:

- Portionings of mass nouns or plurals into discrete quantities: *an amount of cheese*, *a pound of cheese*, *a kilo of beans*. These can be treated as changing the individuation from **mass** or **plural** to **portion**.

- Portionings of mass nouns which also specify the physical form of the resulting entity to a greater or lesser extent, for example: *a piece of cheese*, *a wedge of cheese.* These change the individuation from **mass** to **individual**.
- Collections of entities: *a collection of records*, *a line of mountains*, *a terrace of houses.* These turn **plural**s into **individual**s.
- Groups of individuals: *a group of men*, *a crowd of football supporters*, *a herd of bison.* These take **plural**s to **group**s.

My main reason for discussing this class of phrases here is their role in dictionary definitions, discussed in the next chapter.

I will use the term *partitive noun* to refer to the nouns involved in these constructions, such as *amount*, *piece*, *terrace* and so on (see Quirk *et al.*, 1985:249f). It appears possible to class all partitive nouns as relational, taking the prepositional phrase as an argument (see Gazdar *et al.*, 1985:127–130). The same entailment relationship holds as with relational nouns like *sister*:

(26) Lee is a sister.
Therefore, Lee is a sister of someone.
This is a piece.
Therefore, this is a piece of something.

Relational nouns can be assumed to be subcategorised for the PP argument because the position of the PP is so restricted. The PP cannot occur in predicative position:

(27) * The sister is of Lee.
* The amount is of cheese.
?? The wedge is of cheese.

Modifiers cannot (normally) be inserted before the PP. This behaviour of relational nouns contrasts with derived nominals where the PP can occur in a variety of positions:

(28) The discovery of penicillin
The discovery in 1923 of penicillin
The discovery of penicillin by Fleming
The discovery by Fleming of penicillin
The piece of cheese with a hole
* The piece with a hole of cheese

In the case of the partitive nouns, the argument PP contrasts with PP modifiers, which can occur with nouns which should not be classed as relational. There is a use of *of* as a modifier to denote composition, for example:

(29) This table of Indian mahogany was sold for £500.
This dress of washed silk was displayed in the shop window.

This contrasts with the purely relational use, even though the PP denotes composition in both cases. Modifiers and arguments are distinguished by the predicative position and interpolation tests:

(30) This dress of washed silk
This piece of washed silk
This dress is of washed silk.
? This piece is of washed silk.
The dress in the shop window of washed silk
? The piece in the shop window of washed silk

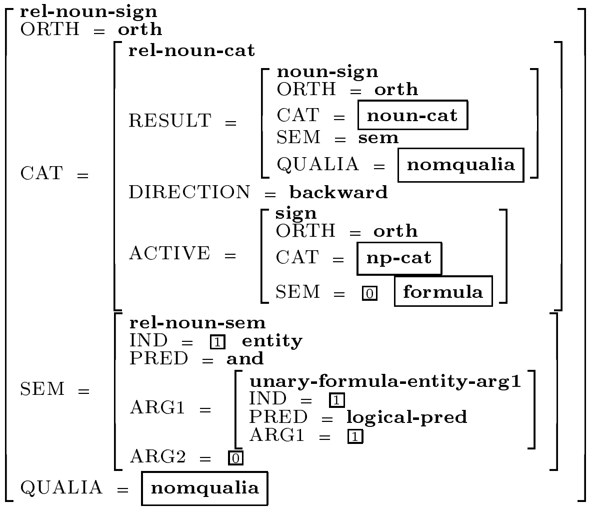


Figure 5.10: The type **rel-noun-sign**

However this position is complicated because many partitive nouns have non-relational senses which do take PP modifiers and do not take arguments. For example *ribbon* behaves relationally in phrases such as *a ribbon of land* but is not necessarily relational in *a ribbon of silk*. In the sense of *ribbon* which does not take an argument, the composition of the entity is understood as some sort of fabric, but the relational use is wider.

The partitive nouns in general have senses where the composition of the entity is understood. For example, *group* can occur without an argument in a context where the composition of the group is not given, but apparently always denote humans or organisations when it does so. There is also the even more specialised 'pop group' sense.

In the next section I illustrate how the representation of the individuation shifting relational nouns interacts with the representation of mass, plural and group terms discussed earlier. I will treat the relational uses as subcategorising for an obligatory argument PP. I will return to the relationship with the non-relational senses at the end of that section.

5.4.1 Representation of relational nouns

All relational nouns have type **rel-noun-sign**, shown in Figure 5.10. Signs of type **rel-noun-sign** will actually act as arguments to the PP which is treated as being almost identical to the type raised NP sign. This type will be instantiated according to the shift in individuation which is involved, and the qualia structure may be further specified according to the type of entity involved. Thus *herd*, for example, is treated as turning plural entities into groups, but as applying only to **animal**s (except in a metaphorical, extended sense). The lexical entry for *herd* is shown expanded in Figure 5.11.

The non-relational uses can be treated as being derived from the relational use by a lexical rule which deletes the argument, shown in Figure 5.12. This rule will leave the composition of the entity essentially unspecified for words such as *piece*, *amount* and *group*. Argument deletion will only be possible when the composition is given either by context or by convention. In the latter case, there will be one or more distinct word senses, which inherit from the result of rule application but further specify it. For example, the lexical entry for *group* in the sense meaning a group of musicians might be:

```
group 2
   < > < ( group_1 + relargdel ) < >
   < QUALIA : CONSTITUENCY : ELEMENTS > < musician_L_0_1 < QUALIA >.
```

If the rule is applied when the composition of the result cannot be instantiated, the result is odd:

$$\left[\begin{array}{l}
\textbf{rel-noun-sign} \\
\text{ORTH} = \textbf{herd} \\
\text{CAT} = \left[\begin{array}{l}
\textbf{rel-noun-cat} \\
\text{RESULT} = \left[\begin{array}{l}
\textbf{noun-sign} \\
\text{ORTH} = \textbf{orth} \\
\text{CAT} = \boxed{\textbf{noun-cat}} \\
\text{SEM} = \textbf{sem} \\
\text{QUALIA} = \left[\begin{array}{l}
\textbf{animal} \\
\text{FORM} = \left[\begin{array}{l} \textbf{nomform} \\ \text{RELATIVE} = \textbf{group} \end{array}\right] \\
\text{CONSTITUENCY} = \left[\begin{array}{l} \textbf{groupconst} \\ \text{ELEMENTS} = \boxed{0}\ \boxed{\textbf{animal}} \end{array}\right]
\end{array}\right]
\end{array}\right] \\
\text{DIRECTION} = \textbf{backward} \\
\text{ACTIVE} = \left[\begin{array}{l}
\textbf{nominal-sign} \\
\text{ORTH} = \textbf{orth} \\
\text{CAT} = \boxed{\textbf{np-cat}} \\
\text{SEM} = \boxed{1}\ \boxed{\textbf{formula}} \\
\text{QUALIA} = \boxed{0}
\end{array}\right]
\end{array}\right] \\
\text{SEM} = \left[\begin{array}{l}
\textbf{rel-noun-sem} \\
\text{IND} = \boxed{2}\ \textbf{entity} \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \left[\begin{array}{l}
\textbf{unary-formula-entity-arg1} \\
\text{IND} = \boxed{2} \\
\text{PRED} = \textbf{herd\_1} \\
\text{ARG1} = \boxed{2}
\end{array}\right] \\
\text{ARG2} = \boxed{1}
\end{array}\right]
\end{array}\right]$$

Figure 5.11: Lexical entry for *herd*

$$\left[\begin{array}{l}
\textbf{lexical-rule} \\
0 = \boxed{0}\ \left[\begin{array}{l}
\textbf{lex-count-noun} \\
\text{ORTH} = \boxed{1}\ \textbf{orth} \\
\text{CAT} = \boxed{\textbf{noun-cat}} \\
\text{SEM} = \boxed{2}\ \boxed{\textbf{obj-noun-formula}} \\
\text{QUALIA} = \boxed{\textbf{nomqualia}}
\end{array}\right] \\
1 = \left[\begin{array}{l}
\textbf{rel-noun-sign} \\
\text{ORTH} = \boxed{1} \\
\text{CAT} = \left[\begin{array}{l}
\textbf{rel-noun-cat} \\
\text{RESULT} = \boxed{0} \\
\text{DIRECTION} = \textbf{backward} \\
\text{ACTIVE} = \boxed{\textbf{sign}}
\end{array}\right] \\
\text{SEM} = \left[\begin{array}{l}
\textbf{rel-noun-sem} \\
\text{IND} = \boxed{4}\ \left[\begin{array}{l}
\textbf{obj} \\
\text{AGR} = \left[\begin{array}{l} \textbf{agr} \\ \text{NUM} = \textbf{sg} \end{array}\right]
\end{array}\right] \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \boxed{2} \\
\text{ARG2} = \boxed{5}\ \boxed{\textbf{formula}}
\end{array}\right] \\
\text{QUALIA} = \boxed{\textbf{nomqualia}}
\end{array}\right]
\end{array}\right]$$

Figure 5.12: Lexical rule for deletion of the argument of relational nouns

(31) ?? A piece was on the table.

This sentence is odd at the beginning of a discourse. The possible interpretations involve the lexicalised non-relational senses of *piece* which happen to be relatively obscure or inappropriate in this sentence. However in context the sentence is fine:

(32) Mary had cut up a cake. A piece was on the table.

Because the context provides an entity for *a piece* to relate to, the sentence is no longer odd. Sentences such as *a piece was on the table* can be regarded as triggering abductive inference, which will fail if no there is no appropriate context. Abduction is not forced if an argument is supplied, however underspecified, and hence (33) is not so odd. [12]

(33) A piece of something was on the table.

In the examples where the relational noun is restricted to arguments which involve more specific types of entities, the out of context non-relational examples are more acceptable.

(34) ? A herd was grazing in the field.
There is one range which extends from the north to south coasts.

This process is analogous to that involved in logical metonymy, where the acceptability of the result of coercion, out of context, depended on the instantiation of the telic role. The process here is similar, but instantiation involves constituency.

Note that, in at least some cases, the direction which I have described for rule application does not coincide with intuitions about which use is derived. For example, *forest* has a relational use *forest of oaks*, *forest of hands* but intuitively this seems to be an extension of the non-relational sense. I will consider the issue of the directionality of lexical rules briefly in Chapter 8. It should also be noted that many partitives have non-relational abstract senses where no physical composition is implied. Familiar examples are *line*, *circle* and so on, but there are also more technical examples such as *group* and *forest*. I leave the issue of interrelating these senses open.

5.5 Summary

I had three aims in this chapter: to show that the use of qualia structure can complement a more conventional formal semantics in the treatment of individuation, to implement this account in LAUREL in order to demonstrate the construction of a reasonably detailed and coherent fragment, and to provide a treatment of some of the more common constructions which occur in dictionary definitions of nouns and affect the extraction of information from them. I argued that a distinct lexical semantic representation of individuation is necessary because individuation has to be stated relative to a predicate. Some nouns are count, others mass, but this is only partially predictable on the basis of non-linguistic knowledge of the entities denoted. Similarly, I claimed that number agreement cannot be treated as purely semantic, because of nouns like *scissors* and *barracks*, which take plural agreement, but do not refer to entities which behave like normal plurals with respect to properties such as distributivity. I have suggested one treatment of this within a formal semantic framework based on work by Krifka, using qualia structure to make fine-grained distinctions between various classes of nouns with respect to their agreement and individuating properties. I have shown how the lexical rules of `grinding` and `portioning` which were introduced in the previous chapter fit into this framework. I proposed a treatment of the plural behaviour of group nouns as a metonymic sense extension and discussed how some of the properties of relational nouns which affect individuation can be encoded in the qualia structure.

[12] Or at least not odd in the same way. The query "a piece of what?" is prompted by (30), in (33) the hearer wonders why the speaker doesn't seem to know what the stuff was. Hence it might be appropriate as the first line in a horror story, whereas (30) would not be.

Chapter 6

Dictionary definitions

In the previous chapters I have discussed some examples of lexical semantic representation in the context of ordinary language. In the remainder of this thesis I will consider the specialised sublanguage of dictionary definitions and the use of MRDs for the extraction of lexical semantic information. [1] I will discuss the linguistic basis for the extraction of semantic information in this chapter, concentrating on noun definitions; the practical issues involved in the construction of lexicons from MRDs are discussed in Chapter 7.

Much of this chapter will be concerned with the way in which definitions can be used as sources of data about hyponymy. In Chapter 3 I gave the example of *book* having hyponyms such as *autobiography*, *novel* and *dictionary*, and *dictionary* having the hyponym *lexicon*. This information is implicit in the dictionary definitions:

autobiography 1 a book written by oneself about one's own life.

dictionary 1 a book that gives a list of words in alphabetical order with their pronunciations and meanings.

lexicon a dictionary, esp. of an ancient language

These definitions are quite straightforwardly hyponymic in character, and conform to the classical description of a definition being composed of a *genus term* (which is *book* in these examples) and discriminating *differentia*. In AI terminology they describe IS_A relationships between the *definiendum* (or term being defined) and the genus. The network of links between words or word senses which can be generated by extracting genus terms is referred to as a taxonomy in computational lexicography. The fact that most noun definitions are IS_A in a more or less straightforward way, follows from the conventions that lexicographers use, and so I will begin this chapter with some comments on defining, then discuss taxonomy and the representation of the classical IS_A definitions in LAUREL. The bulk of the chapter will be concerned with definitions which, although IS_A or hyponymic in character, do not fit into the classical pattern, such as synonyms and coordinated genus terms. I will conclude by discussing non-IS_A definitions.

6.1 Defining

The assumption behind the use of dictionary definitions as sources of linguistic data is that they encapsulate important aspects of a word's meaning, at least in a way which makes dictionaries useful to humans. It is important when treating MRDs as a source of data to have some idea

[1] Much of this chapter describes work which was carried out jointly by Piek Vossen and myself which was originally reported in Vossen and Copestake (in press). However I have revised that material to be consistent with the representations given in the previous chapter and made some other modifications. The examples, statistical data and many of the intuitions about dictionary definitions are Piek's, the more formal discussion of the semantics of the definitions, and the details of representations are my own.

of how lexicographers go about defining; Landau (1984) gives a detailed description, of which I summarise the aspects which are most important from the viewpoint of automatic processing of definitions.

Essential principles of defining

1. Avoid circularity.
   The prototypical (and apocryphal) example of circularity in definition is:

   recursion see recursion

   but the sort of circularity which occurs in practice is usually less direct. Circularity causes the same sort of problems for computerised extraction of information as it does for human users; clearly a well-designed program will not go into an infinite loop, but it is impossible to automatically link a circularly defined group of words with the rest of the lexicon. The irony, which Landau does not point out, is that some group of dictionary definitions must inevitably be circular because every word used in a definition must itself be defined. However if such circularity is confined, as it usually is, to the definitions of very general words such as *thing*, *object* and so on, this causes no particular problems for automatic construction because such entries will be manually specified in any case.

2. Define every word used in a definition.
   Because this principle is generally strictly adhered to, all dictionaries are designed to have a limited defining vocabulary, although obviously this will usually be much larger than that used in LDOCE. As Landau describes, this principle may be knowingly broken in the case of regular derivations, as long as the stem and the affix are themselves entered. This causes problems for automatic extraction since some such combinations will be obvious to a human reader although not strictly compositional. LDOCE lexicographers were also allowed to use derivations which were not in the core vocabulary and some which were used are not straightforward derivations.

3. Define the entry word.
   A definition should define the entry word rather than just talk about it. Landau discusses this using some examples of bad definitions, which contain encyclopaedic information about the concept, without stating the essential properties explicitly. He then goes on to discuss the principles which underlie good definitions.

Good defining practice

- Priority of essence.
  The most essential elements of meaning come first. In the case of noun definitions Landau states:

  > The defining noun may relate to the appearance, purpose or composition of the thing defined, but it should pinpoint that property of the thing that is regarded by most speakers as being essential to it. If that essential property were not present, the thing would not be regularly identified by the definiendum.

- Substitutability
  In general, the definition should be substitutable for the word in context. Exceptions to this rule include closed class words and expletives but dictionary definitions of these are not, on the whole, useful in construction of a lexical knowledge base. More significantly, Landau distinguishes between definitions which are *extracted* from examples of actual usage, for which the substitutability principle can usually be applied, and those which are *imposed*, the latter usually being scientific or technical definitions, where the principle breaks down. Imposed definitions may be given for common words such as *rose* and *feather*. Their non-substitutability, however, seems to me to be more a matter of connotation than denotation.

LDOCE, in common with other learners' dictionaries, avoids technical definitions and so the substitutability principle can be taken to hold, in general, for the definitions which I will discuss.

- Reflection of grammatical function
  Nouns are usually defined in terms of nouns and so on. As I will show below, taxonomies of nouns can be constructed from definitions because of the combination of the three principles, reflection of grammatical function, substitutability and priority of essence, as applied to noun definitions.

- Simplicity

- Brevity

- Avoidance of ambiguity.
  Unfortunately definitions which are quite unambiguous to a human reader can have a very large number of readings when processed with a comprehensive grammar, especially if the lexicon contains the same sense distinctions as the dictionary (Briscoe and Carroll, 1991). One disadvantage of the LDOCE limited defining vocabulary is that the common words which comprise it tend to be the most ambiguous. The claim that words were used in their most central sense in definitions is not found to be substantiated in practice, at least if the most central sense is taken to be a single sense as given in LDOCE. Hence sense-disambiguation heuristics are essential for processing.

There will also be constraints specific to particular dictionaries. I have mentioned the use of a limited defining vocabulary in LDOCE, but editors also produce manuals for lexicographers which will give the recommended style for particular types of definitions. For example, these may list the recommended and proscribed phrases for introducing definitions of adjectives. These conventions and constraints can be exploited for automatic processing, but they occasionally lead to definitions which are badly phrased or inadequate. Lexicographers may therefore bypass the convention.

Lexicographers generally produce definitions on the basis of citations and previously published dictionaries. Commercial dictionaries are produced by teams of lexicographers working under considerable time pressure. Until very recently, the use of database technology to help the process of ensuring the consistency of definitions was limited to relatively simple checks, such as ensuring that a word used in a definition was itself included in the dictionary. Space is all important, so definitions may have to be shortened. Thus dictionaries will contain errors and infelicities in individual entries, and groups of entries may be inconsistent.

Because dictionaries are printed books, there are obvious limitations in the techniques that can be used in representing relationships between words and between senses of words. The most obvious artifact in dictionaries is the listing of distinct senses. Dictionaries adopt different conventions with respect to sense distinctions; some use numbers or letters to distinguish senses within an entry, others just use punctuation. LDOCE, for example, adopts a complex variety of techniques for sense distinction:

1. Separate entries are usually given for different parts of speech, and for words which are known to have distinct etymologies.

2. Within entries senses are numbered. Occasionally letters are used to further subdivide senses.

3. Implicit techniques used to indicate variations in senses include separation by semi-colons and bracketing.

In many cases, sense distinctions which seem directly comparable on a linguistic basis will be represented in different ways in different entries. For example:

carafe (the amount contained in) an ornamental bottle ...

glass 5 a a drinking vessel **b** also **glassful** — the amount which this holds

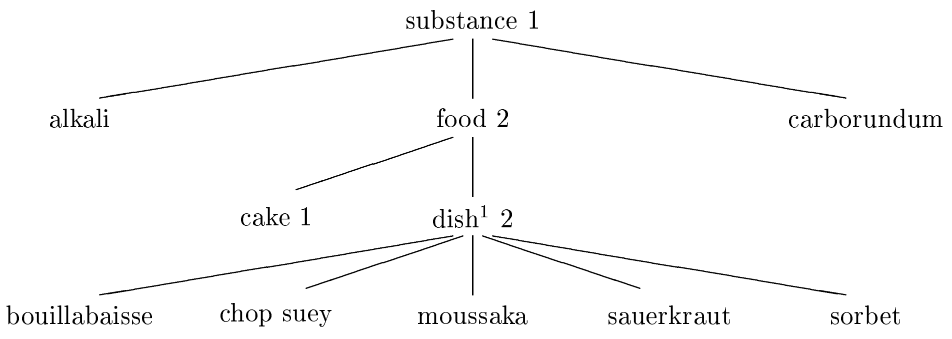


Figure 6.1: Partial taxonomy

bottle 1 a container, typically of glass or plastic . . .
2 also **bottleful** — the quantity held by a bottle

These variations do not just arise from perceived differences in the relative importance of the distinction, but from an interplay between a range of factors. These may include variations between lexicographers, changes in style as the dictionary evolves, relative importance of the entry as a whole (senses are more likely to be distinguished if an entry is seen as being important), and the need to give extra information about a sense (such as a synonym) which can only be done if the sense is made distinct. Furthermore, a group of words might consistently pattern in a way which was distinct from another group, simply because a lexicographer may base a new definition on an existing one for a similar word. Thus, although dictionaries are a potentially useful source of data on sense distinction, there are considerable problems in interpreting the information.

6.2 Taxonomies

Most attempts at representing semantic information derived from dictionary definitions are based on the use of taxonomies (Amsler, 1981). The notion of taxonomy that has been used in work on MRDs by Amsler, and more recently by authors such as Chodorow *et al.* (1985) and Guthrie *et al.* (1990), is essentially an informal and intuitive one: a taxonomy is the network which results from connecting headwords with the *genus terms* in their definitions but the concept of genus term is not formally defined. For noun definitions it is, in general, taken to be the syntactic head of the defining noun phrase. The defined term, or *definiendum*, is normally a hyponym of the genus term. The modifiers to the genus are the *differentia*. For example, the LDOCE definition of *moussaka* is:

moussaka a Greek dish made from meat and AUBERGINEs, often with cheese on top

and here the genus term is taken to be (a particular sense of) *dish*. The network is formed by connecting *moussaka* to *dish* and by connecting *dish* to its genus term and so on. However, if no sense disambiguation is carried out, the network will be a directed graph rather than a hierarchy. To form a coherent hierarchy sense disambiguation is essential, as discussed in section 7.1.

Assuming for the moment that all definitions have this classical structure, with a single sense which can be taken as the genus term, of which the definiendum is a strict hyponym, and assuming that we stop tracing genus terms when we reach a sufficiently general definition, the structure which results is a tree, such as that partially shown in Figure 6.1. Such a tree is not a classical taxonomy, since sister nodes need not be incompatible, and there is no restriction to natural kinds. For example, *stallion*, *pony*, *gee-gee*, *Arab* and *palomino* all come under *horse* in a taxonomy derived from LDOCE. It is nevertheless clear that this sort of structure can form the basis of an inheritance hierarchy — properties which are true of horses (animality, four-leggedness and so on) are also true of the entities corresponding to the daughter nodes. Taxonomies are thus usually seen as the basis for representation of semantic information extracted from MRDs, at least as far as nouns are concerned, with the taxonomic link corresponding to an IS_A link between senses.

I will only consider nouns in detail, and will concentrate on concrete nouns in particular, but it is worth noting that this concept of taxonomy works much less well for verbs and seems to have little utility for adjectives. Lexicographers normally find it relatively easy to describe a genus term for concrete nouns, but, although it is quite straightforward to group some verbs in a hierarchy, those involving hitting or killing for example, others are less easy to group in this way (see Pulman, 1983). Taxonomies of verbs tend to be very flat, which is perhaps not surprising, since verbs generally have more dimensions on which to vary lexically than nouns do, because of their more complex argument structure. Isolating genus terms in verb definitions can be difficult because of the use of phrasal verbs. For example, the genus term in the following definition is *take off* rather than *take*:

undress^1 **1** to take one's clothes off

Adjectives pose even more problems — although they can sometimes be grouped under a common adjectival parent, this is relatively rare. They are more frequently defined in terms of nouns or verbs and so a straightforward hierarchy of adjectives cannot be derived. For example,

foreign 1 to, from, of, in, being, or concerning a country or nation that is not one's own or not the one being talked about

Although most definitions of concrete nouns do fit into the classical structure of genus and differentia, and taxonomies six or seven levels deep can be extracted from LDOCE, the intuitive concept of taxonomy breaks down on many definitions. In examples such as:

swallow^1 a type of small bird with pointed wings ...

it is clear that a more useful hierarchy is formed by taking *bird* and not *type* as the genus term. In general, in many definitions of the form "an X of Y", both X and Y are candidate genus terms and the relationship involved may not be one of hyponymy. Definitions of this form are said to have *complex heads* or *complex kernels* and X, which might be *kind*, *piece*, *amount*, *group*, or *part*, for example, is sometimes referred to as the *relator*. The proper treatment of such definitions is a well-known problem area for computational lexicography (it is discussed in Guthrie *et al.* (1990) and Klavans and Wacholder (1990), for example).

In the definition above, it is reasonable to claim that there is an IS_A or hyponymy relationship between *swallow*^{1} and *bird 1*, but not between *swallow*^{1} and *type*, if *swallow*^{1} is taken as denoting a set of individual swallows rather than as denoting a kind. In a definition such as

nose^1 **1** the part of the face above the mouth ...

the relationship between *nose*^{1} *1* and *face*^{1} *1* is not an IS_A relationship, but there is a (rather uninformative) IS_A relationship with *part*^{1} *1*.

Several authors have proposed that many more link types be distinguished to deal with the varieties of complex kernels. For example Guthrie *et al.* suggest that a HAS_MEMBER link is necessary as well as an IS_A link. However, since a semantics for these proposed links is not provided, it is impossible to evaluate these claims. Furthermore, the problem of non-hyponymy relationships is not restricted to the complex heads. There are, for example, definitions in LDOCE such as

yoghurt milk that has turned thick and slightly acid ...

where a temporal element is involved in the relationship between the genus term (*milk*) and the defined sense; creating a new type of link (a WAS_A link?) without giving it any semantics is clearly no solution to this. Even regarded as a notational device, creating new types of link for each potentially different case should probably be avoided, since it is likely to result in the imposition of a structure on the entries which is not really there, because it complicates the concept of genus term as accepted by lexicographers. It is essential to go beyond intuitive notions and to develop a formal theory. Although networks based on taxonomies extracted from dictionaries have been built, which are useful for tasks such as sense-disambiguation, these are not generally directly utilisable as NLP lexicons. For a (re)usable lexicon, a declarative, formally specified, representation language, such as LAUREL is essential.

In the rest of this chapter, I will discuss various classes of dictionary definitions and how they give rise to slightly different inheritance relationships which may be represented in LAUREL. Most of this discussion will involve considering individual definitions rather than the inheritance hierarchy as a whole, but I will reexamine the notion of taxonomy at the end of the chapter. I will argue that IS_A links can be seen as a special case of default inheritance between the semantic components of lexical entries, and that this inheritance relationship is licensed by the semantics of the definitions. For classical definitions the feature structure corresponding to the lexical semantics of the genus term is treated as a psort in LAUREL from which the lexical semantics of the definiendum can be inherited.

The treatment of constructions such as complex kernels in dictionary definitions is ultimately dependent on their general linguistic treatment. In principle, the treatment of a definition should be to construct an appropriate representation for it, just as we would if parsing the phrase, and then to use this as the psort from which to inherit information (by default). However, since we are primarily interested in characterising the structures which are used relatively frequently in definitions, and their usual semantic effect there, we can regard the definition language as a specialised sublanguage and ignore many of the complexities which would be involved in a full treatment of the various constructions involved, and concentrate on how they affect the lexical semantic qualia structure.

6.3 Semantics of genus phrases in noun definitions

In order to describe how dictionary definitions can be used to construct inheritance links, it is easiest to look at some examples. As I described in Chapter 3, hyponymy relationships can be treated as licensing default inheritance between the qualia structure of lexical entries. By using dictionary definitions to construct an inheritance hierarchy we can automatically create entries for words such as *autobiography*, *novel* and so on, which inherit their lexical semantic properties including the telic role, by default from *book*. The definition of the relevant sense of *autobiography* in LDOCE is:

autobiography a book written by oneself about one's own life.

If an inheritance link is created on the basis of the taxonomic relationship, then the qualia structure will be inherited from *book* by *autobiography*, allowing the lexical entry to be produced without having to specify the telic role directly. However, this information is clearly defeasible; consider the definitions of *dictionary* and *lexicon* in LDOCE:

dictionary 1 a book that gives a list of words in alphabetical order with their pronunciations and meanings.

lexicon a dictionary esp. of an ancient language

Although a dictionary is a book, the purpose of a dictionary is to be referred to rather than read. Entries lower in the taxonomy must thus be allowed to override the inherited information; although *dictionary* is under *book*^{1} *1* in the hierarchy, the inheritance of the telic role must be cancelled by more specific information about the purpose of dictionaries. This modified value should be inherited by *lexicon*.

It is not always appropriate to construct inheritance links of this form between the genus sense and the sense being defined. For example, consider the following definition:

rocket^1 **1** a tube-shaped case packed with gunpowder ...

The appropriate sense of *case* here is presumably:

case^2 **3** a box or container for holding and protecting something

However, to state that *rocket*^{1} *1* inherited from *case*^{2} *3* would probably be a mistake, since its lexical semantic properties are quite different; for example *case*, like other container denoting words, can be used either to mean the physical object, or the stuff contained:

> John smashed open the case.
> Oddbins sells 30,000 cases (of wine) a year.

This is clearly not true of *rocket*. It is intuitively plausible to claim the definition of *rocket*^{1} *1* does not express an IS_A relationship, but intuitions on the question of what should be taken as an IS_A relationship vary widely.

I assume as a working hypothesis that the taxonomic relationships which licence inheritance of lexical semantic properties between word senses correspond to the cases where the definitions can be taken as asserting that a subset relationship exists between the extensions of the defined word sense and the genus word sense. For example:

$$\text{autobiography1 ISA book}^1\ 1 \equiv [\![\textbf{autobiography1}]\!] \subseteq [\![\textbf{book}^1\ \textbf{1}]\!]$$

Thus I assume that the subset relationship is (usually) entailed by the semantics of dictionary definitions such as that for *autobiography*, although I would certainly not claim that this is all that there is to their semantics. As we saw earlier, traditional dictionaries follow the principle of substitutability; that is that the word sense being defined (the definiendum) can be replaced by the definition when it occurs in a sentence with a result which is equivalent in meaning. In the ideal dictionary definition the extension of the definition and the definiendum would be identical. This ideal is clearly rarely, if ever attainable. However, this principle, plus that of priority of essence, lead to a characteristic pattern for definitions where the modifiers are intersective and specialise the meaning of the genus term to approximate to that of the definiendum. The genus term itself will tend to be the most specific term consistent with the other constraints on the lexicographer, such as the need to avoid uncommon words. In the case of definitions such as that for *autobiography*, the modifiers of the genus term are intersective; a *book written by oneself* is a *book*. In contrast, a *case packed with gunpowder* is no longer a *case*, but such definition structures are less frequent.[2] Given the semantics described in the previous chapter, where singular, plural and mass entities have the same model-theoretic status, this assumption about the inheritance relationship can be taken to cover mass terms and plurals as well as singular count nouns.

If we assume that lexical semantic properties are at least consistent with real world properties, the subset relationship supports the inheritance behaviour which I have just outlined. It is clear that properties which necessarily hold of some set of entities also hold for any subset of those entities and thus inheritance of necessary properties is justified. Lexical semantic properties are not usually necessary properties, but the inheritance behaviour described does follow from the 'Penguin Principle', that properties which hold by default of some set of entities also hold by default for any subset of those entities, but that more specific information about the default behaviour of the subset will override the inherited behaviour. The Penguin Principle characterises the behaviour of default inheritance networks (e.g. Touretzky, 1986) and recently has been taken as a desirable property of a non-monotonic logic (by Delgrande (1989) and Asher and Morreau (1991), for example). However, in representing the lexicon there is no need to implement a non-monotonic logic directly, because the full power of such a formalism is not needed. It is sufficient to use a default inheritance mechanism for lexical semantic information which is consistent with these observations. As shown in Chapter 3, LAUREL's default inheritance mechanism is consistent with these assumptions and can therefore be used in this way.

[2] It is clear that *case* does not pinpoint the essential properties of *rocket* and the LDOCE definition seems bad. However the lexicographer probably could not find a better genus term which was in the restricted defining vocabulary, which does not include *projectile*, for example.

6.4 The hyponymic relationship in LAUREL

The following path specifications for *autobiography*, *dictionary* and *lexicon* make their lexical entries inherit qualia structure from `book_L_1_1`:

```
autobiography L_0_1
   < QUALIA > < book_L_1_1 < QUALIA >.

dictionary L_0_1
   < QUALIA > < book_L_1_1 < QUALIA >
   < QUALIA : TELIC > <= refer_to_L_0_2 < SEM > .

lexicon L_0_0
   < QUALIA > < dictionary_L_0_1 < QUALIA > .
```

These path descriptions are similar to those used in Chapter 3, but I am now assuming the use of the full type system. Rather than specifying the predicate **refer_to_L_0_2** directly, the value of the TELIC role for `dictionary_L_0_1` is non-default inherited from the semantic specification of `refer_to_L_0_2`. The complete semantic structure is thus specified correctly, relative to the predicate. (The reentrancy specification which ensures that the patient argument of **refer_to_L_0_2** is instantiated by the variable representing the dictionary is inherited from the lexical semantic type **artifact**.)

The classical hyponymic definition therefore has a straightforward translation into LAUREL. The genus term is interpreted as a psort from which the lexical semantic structure of the entry can be inherited by default. Further information expressed in the differentia or acquired from other sources can modify the inherited information. Such modification is regarded as non-defeasible, because it is assumed to correspond to information which is known directly about that particular entry. This applies to the telic role of `dictionary_L_0_1`, even though it is described as being inherited from another lexical entry, because the semantic structure is regarded as a chunk — the predicate **refer_to_L_0_2** necessarily has this particular argument structure.

Because default inheritance is constrained by the type system in LAUREL so that the daughter must have a type which is either more specific than the mother or the same as it, the default inheritance specification allows the qualia type of the entry to be inferred, non-defeasibly. Thus a taxonomy derived by linking inheritance relationships will be consistent with the type hierarchy.

Multiple taxonomic inheritance will tend not to occur, since sense-disambiguated taxonomies extracted from dictionaries will be tree-structured, unless there are multiple genus terms. These can occur as a result of coordination with *and* (coordination with *or* should not be represented in this way, since multiple inheritance corresponds to conjunction of information). However, genuine conjunctions are relatively rare in definitions (see section 6.7). The other case where multiple inheritance is desirable is for the representation of cross-classification, see below. In this case, multiple inheritance tends be orthogonal because it involves different parts of the qualia structure. Because we do not find significant numbers of genuine multiple inheritance conflicts,[3] the default inheritance mechanism in LAUREL was designed so that any discrepancies were reported as errors, since they are far more likely to arise as the result of problems in the automatic extraction process. This at least gives the user the option of manually editing the lexical entries in order to get the desired behaviour, whereas any approach which did not signal the presence of conflicts would not. Top-down rather than bottom-up inheritance also seems justified for this application, since the taxonomic relationship is best described as a relationship between fully formed lexical entries.

6.4.1 Variations in classification

Although taxonomies derived from genus terms are tree structured when coordination is excluded, this can sometimes be demonstrated to be more of an artifact of the defining conventions than a genuine observation about the nature of hyponymy. Furthermore, it is not always the case that

[3] In fact I have not found any convincing examples in LDOCE so far.

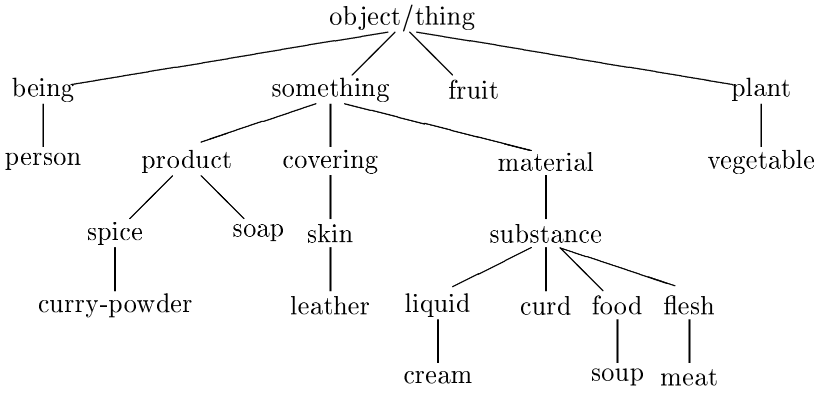


Figure 6.2: Partial taxonomy showing variations in classification.

senses are found at the most specific level possible. These effects, which have been investigated in detail by Piek Vossen, are clearly illustrated by an example of his (taken from Vossen and Copestake, in press) which shows that many senses which could be considered to be substances are not in fact directly related by dictionary definitions to the word *substance*. A partial taxonomy is shown in Figure 6.2, based on the definitions in LDOCE. Vossen points out that words such as *soap* and all *spices* are defined as *product*, which is not related to *substance*. Some edible objects and substances are not found as subordinates of *food*: e.g. *fruit*, *vegetable*, *cream*, *meat*. In the case of *curd*, the level of *food* is completely bypassed and it is directly classified as a *substance*. These different classifications seem to be triggered by the lexicographers either taking the functional classification as primary (*food*, *covering*, *product*) and specifying the properties of the form in the differentiae (*soup* is "liquid food"), or taking the form as primary (*substance*) and specifying the function by modification.

Vossen (1990b, 1991) describes this sort of systematic variation in classification in terms of:

1. A different priming of features: given a set of features that are essential for the meaning of a word, a lexicographer can make a different choice in selecting one feature as the genus and the other features as the differentiae.

2. A different abstraction of features: given a set of features a lexicographer can always choose to classify the sense at a more abstract level and to specify the extra discriminating features as differentiae.

For example Vossen describes two features, "covering" and "material" as competing for the status of genus in the following definitions:

armour 1 strong protective metal covering on fighting vehicles, ships and aircraft

blanket^{1} **1** a thick, esp. woollen covering...

carpet^{1} **1** heavy woven often woollen material for covering floors or stairs

daub^{2} **1** (a) soft sticky material for covering surfaces like walls

These sort of examples illustrate the inadequacies of taxonomies when taken in isolation, since a classification in terms of the genus term alone will lead to one or other of these aspects being omitted. Because the type system is not a strict hierarchy but allows types with multiple parents, it is possible to classify such examples consistently, as **artifact-physical**, for example. To produce a finer grain of specification, multiple inheritance from psorts can be used. For example it is possible to specify that *soup* inherits from two psorts *liquid* and *food*.

```
soup L_0_1
   < QUALIA > < liquid_L_2_1 < QUALIA >
   < QUALIA > < food_L_0_1+2 < QUALIA >.
```

The type system separates features which describe function (the TELIC role in the qualia structure) from those describing form (the FORM role). Although some information about physical form is inherited from psorts such as those for *food*, which primarily denote function, and thus conflicts are potentially possible, no clear cases have yet actually arisen where this would happen. The more serious difficulty with dealing with such definitions is the automatic extraction of the extra information, which we have no way of doing reliably at the moment.

6.4.2 Indeterminacy of sense

In the discussion above, I have assumed that the genus term corresponds to a single sense in the dictionary, and thus that a single entry from which information should be inherited can be found in a lexicon derived from that dictionary. Given the difficulties of making sense distinctions, it is surprising how well this assumption works, at least for LDOCE nouns. The program for constructing disambiguated taxonomies, TAXUS, which I will describe in the next chapter, relies on making sense distinctions according to a series of heuristics and then checking these with the user. Extensive use of this with concrete nouns in LDOCE has shown that decisions as to the most appropriate sense for a genus term are usually easy to make. This suggests that, although the sense distinctions made in LDOCE are arbitrary to some extent, they also have a large amount of internal consistency, which supports the assumption that they offer a reasonable starting point in building a system with more motivated sense distinctions.

However, in some cases, it is not possible to identify a distinct sense for a genus term because several might apply in the definition in which they have been used. The lexicographer sometimes separates senses which are difficult to distinguish manually (and impossible automatically) and which would generate indistinguishable lexical entries, at least with the type system and approach to lexical semantics assumed here. Such examples have also been discussed by Vossen (1990b). The most appropriate treatment for these cases in the construction of a lexicon is to generate a single entry for the merged senses from which other entries will inherit as usual. However there are other cases where this is not straightforwardly possible.

For example *cake* and *stone* have separate senses in which they are countable (grammar code "C") and uncountable (grammar code "U").

Headword	sense	code	definition
cake^1	1	U	a food made by baking ...
	2	C	a piece of this food ...
stone^1	1	C	a piece of rock, esp. ...
	2	U	solid mineral material ...

In the following entries, where either *cake* or *stone* are the genus word, the uncountable and countable uses have been merged into one sense, and both senses of *cake* or *stone* apply.

Headword	sense	code	definition
gâteau		U; C	a specially attractive type of cream cake
seedcake		C; U	(a) sweet cake containing...
tortilla		C; U	(a) thin round flat cake made...
diamond	1	C; U	a very hard, valuable, precious stone, ...
onyx		C; U	a (type of) precious stone ...
opal		C; U	a (type of) precious stone which ...

In other examples, however, only one sense is applicable:

Headword	sense	code	definition
bun	1	C	a small round sweet cake
shortbread		U	a thin hard kind of sweet cake ...
pebble	1	C	a small roundish smooth stone ...
portland stone		U	a type of yellowish-white stone...

Thus it would not be appropriate to merge the senses of *cake* and *stone*.

Where such polysemy is regular, it can be handled by formulating sense extension rules. In the case of *cake* the portioning rule mentioned in Section 4.3.3 and 5.2.2 would be applicable; this captures the fact that mass nouns denoting foodstuffs can also refer to portions of that substance. Thus the count sense of *cake* can be regarded as an established extended sense. Of the senses given above, *seedcake*, *tortilla* and *gâteau* can also be regarded as mass nouns, which have established uses corresponding to a portion reading. It is thus appropriate to represent these entries as inheriting from the mass sense of *cake*, but undergoing portioning to produce an established sense. [4]

```
tortilla L_0_0_U
   < > <= lex-mass < >
   < QUALIA > < cake_L_1_1 < QUALIA >.

tortilla L_0_0_C
   < > <  ( tortilla_L_0_0_U + portioning ) <> .
```

In contrast *shortbread* inherits from cake, but does not have a portion sense represented in LDOCE, so would simply have the entry:

```
shortbread L_0_0
   < > <= lex-mass < >
   < QUALIA > < cake_L_1_1 < QUALIA >.
```

Finally, *bun* is regarded as inheriting from the established portion sense, but as being itself a distinct individual, rather than a portion:

```
bun L_0_0
   < > <= lex-individual < >
   < QUALIA > < cake_L_1_2 < QUALIA >.
```

It seems reasonable to regard the sense extension involved in the *stone* examples as a specialised form of grinding, with the mass sense of *stone* being established. The representations derived would be straightforwardly analogous to those given above.

Assuming that it is possible to detect that the rule is applicable to the lexical entries generated for these definitions, the relationship between the senses can be derived straightforwardly. However recognition of merged senses is complicated in practice by the kind reading which makes it possible for any mass noun to be used as a count noun (e.g. *limestone* can be used as a count noun in *Portland stone is a limestone* which is equivalent to *Portland stone is a type of limestone*); the lexicographers sometimes give count grammar codes as an option in order to mark this usage when it is especially frequent (e.g. *fabric*).

6.5 Synonymy

The first example of non-classical IS_A definition which I will consider is synonymy, where in terms of denotation the genus term and the definiendum are identical. For example, Vossen and Copestake (in press), list the following dictionary definitions which lack distinguishing modifiers:

bobby *infml BrE* a policeman

bull^{1} **3** *sl, esp. AmE* a policeman

cop^{2} *infml* policeman

copper^{4} *infml* a policeman

[4] In the longer term, it would be better to produce an LKB which is less tied to a particular dictionary, and which has a more motivated notion of lexicalisation and sense distinction. However it seems better to attempt to create this indirectly, via an LKB of the form assumed here, rather than directly from definitions.

flatfoot *sl* a policeman

peeler^2 *BrE old sl* a policeman

pig^1 **4** *derog sl* policeman

The relationship between these senses and *policeman* is not one of hyponymy since their referents are the same. The differences from *policeman* are in respect of register indicated by the various labels in their definitions such as “infml” and “sl” (slang). In terms of semantic features, therefore, they cannot be seen as being strictly subsumed by *policeman* and it would be more appropriate to analyse them as ‘cognitive synonyms’ (Cruse 1986) with specific values for the feature REGISTER. This makes them different from the following examples which exhibit hyponymy, since the words do refer to more specific concepts:

constable 1 a policeman of the lowest rank

outrider a policeman riding on a motor cycle …

policewoman 1 a female policeman

ranger 2b a policeman who rides through country areas …

What is needed is a representation which explicitly constrains the semantics of *cop*^{2} to be identical to those of *policeman*. The constraint that the feature structure representing the lexical semantics of *cop*^{2} is identical to that of *policeman* can be straightforwardly captured in LAUREL by the use of the equality constraint introduced in section 2.6.2. The lexical entry for *cop*^{2} thus contrasts with that for *policewoman 1* where the extra information expressed in the differentia is reflected in the entry, but it also contrasts with *ranger 2b*, where the additional information is not expressible within the current type system:

```
cop L_2_0
   < QUALIA > == policeman_L_0_0 < >
   < REGISTER > = informal.

policewoman L_0_1
   < QUALIA : SEX >  = female
   < QUALIA > < policeman_L_0_0 < > .

ranger L_0_2_b
   < QUALIA > < policeman_L_0_0 < >.
```

6.6 Kinds and types

Up to now, I have assumed that dictionary definitions can be taken as referring, at least indirectly, to properties of the extension of the definiendum. However, some dictionary definitions are explicitly phrased in terms of kinds. For example:

cheddar a type of firm smooth usu. yellowish cheese

limestone a type of rock containing material from bones …

In practice it is clear that most examples of such definitions can be treated in the same way as the ordinary hyponymic definitions, allowing inheritance of lexical semantic properties. Thus we can ignore the relator *type* and treat the definition of *cheddar* as being equivalent to:

cheddar firm smooth usu. yellowish cheese

Given that most of the predicates involved in such examples are true primarily of individual instances rather than kinds, and that thus any formal representation of the original definition should involve those properties being asserted of individual instances (at least by default), there is an empirical justification for this approach. Furthermore the parts of the definition which are not directly true of individuals, for example *usually yellowish*, tend to be qualifications of predicates which are true at an individual level.

It could be claimed that all dictionary definitions implicitly involve relationships between kinds. However some hyponymic definitions would be very odd if *type of* or *kind of* were inserted:

stallion a fully-grown male horse kept for breeding

Cruse (1986) gives the following frame as diagnostic of taxonomy:

An X is a kind/type of Y

but, as mentioned earlier, dictionary taxonomies are not taxonomies in the classical sense, and so fail on this test. It is therefore clear that the semantics of dictionary definitions in general cannot be described in terms of kinds in the informal sense of the term.

The semantic effect of *kind* and *type* in dictionary definitions is unclear and difficult to establish, because the lexicographers' use is uneven. For most cases where *kind* and *type* are used, apparently parallel definitions may be found where they are omitted, for example:

limestone a type of rock containing material from bones ...

slate^1 **1** heavy rock formed from mud by pressure ...

It is well-known that all nouns can have either a type or a token reading, but it is possible that *kind* and *type* in a definition indicate that the type usage is especially salient. [5] For examples which derive from place names, or other proper names, such as *cheddar* and *Stilton*, the type usage might be intuitively said to be primary. However Nunberg (1978) suggests that attempting to make a distinction between the primacy of unit/type readings linguistically is unrevealing.

It is, however, clear that a full representation of some dictionary definitions would involve reference to something other than ordinary individuals in the semantics. Dictionary definitions, after all, are generic statements, and generic utterances are usually assumed to refer to kinds (see, for example, Pelletier and Schubert, 1988). There are cases where definitions clearly predicate properties of the kind rather than of the ordinary individual. For example:

rabbit^1 **1** [C] a common type of small long-eared animal of the HARE family that lives in a hole (BURROW) in the ground.

winkle a type of small sea animal that lives in a shell and is eaten as food

An individual rabbit cannot be *common* (in the relevant sense) and it is not the case that the typical individual winkle is eaten. However, in the second example in particular, it is not clear exactly what the predicate can be taken to be true of. To say that it is true of the kind requires a definition of kind such that it makes sense to say that the kind "is eaten" and yet allows the correct entailments to go through about individuals. None of the treatments of kinds of which I am aware seem adequate, because they do not treat cases such as this where the predicate is not true of all or even most individuals but is only true of some of them (see Pelletier and Schubert, 1988, for one review).

However, the aim here is not to represent dictionary definitions for their own sake, but to use them as sources of data for instantiating lexical entries. The lexical representations produced are intended to be appropriate for token usages; type usages should be derivable from them and not stored individually. Thus, since none of the properties which I have suggested should be represented in the qualia structure demand a treatment in terms of kinds rather than individuals, I will assume that continuing to ignore kinds in dictionary definitions is justified, within the context of the current enterprise.

[5] Note though that remarks similar to those at the beginning of this chapter about the apparent arbitrariness of sense distinctions could be made here. In some cases, for example, space considerations might be responsible for the omissions of *kind* and *type*.

6.7 Coordination

Many entries (about 20% of all noun senses) have definitions in which genus terms are coordinated, in some cases in combination with complex kernels:

staging 2 movable boards and frames for standing on

breeder 2 an animal, bird, or fish that produces young

borer a person, tool, or insect that makes round holes

In some previous work (e.g. Chodorow *et al.* 1985) coordinated genus terms are interpreted as each being parents of the word being defined in the taxonomic structure, which is one way in which a tangled hierarchy results. This is clearly inappropriate if the taxonomic structure is viewed as allowing inheritance; in LAUREL it would mean that each coordination is interpreted as the unification of the parents' feature structures. This is only appropriate for strict logical conjunction which is rare in definition kernels. If we look at the distribution of coordination we see that three elements are mainly used (*or*, *and* and *etc.*) and that most cases of coordination are disjunctions.

As far as the interpretation of the coordinations is concerned, two major classes can be distinguished: either the coordinated elements are alternatives between which a choice has to be made (e.g. "**albino 2** an animal or plant…"), or the entry word is a complex whole of which they are components or constituents. There are also examples where *and* is used with a temporal meaning which can also be regarded as describing the elements of a complex event denoting whole:

montage 3 the choosing, cutting and combining together of separate photographic material…

Composites are always coordinated with *and* but coordination with *and* does not necessarily result in a composite, as is illustrated by the following examples, where *and* seems to be used to indicate non-exclusive alternative descriptions:

growing pains 1 aches and pains in the limbs…

glamour 1 charm and beauty with a ROMANTIC power of attraction

wealth (a large amount of) money and possessions

However most alternatives are indicated by the use of *or* or *etc.* Examination of the definitions suggests that a distinction can be usefully made between *close* and *distant* alternatives. Close alternatives share many features and will probably be found in the same section of the taxonomy:

allegory 1 a story, poem, painting, etc., in which…

tribulation trouble, grief, worry, suffering, etc.

albino 2 an animal or plant that lacks the typical colouring

breeder an animal, bird, or fish that produces young

A special case of close alternatives are *level alternatives* in which a superordinate word is coordinated with a subordinate:

acorn the fruit or nut of the oak tree…

bearer 5 a fruit-producing tree or plant

In these examples, the entry word is related to its genus but also directly to the genus of the genus:

nut[1] **1** a dry fruit with a seed…

tree[1] **1** a type of tall plant…

Distant alternatives (coordinator is always "or") share few features and will be found in quite separate parts of the taxonomy.

threat 2 a person, thing or idea regarded as a possible danger

stationer a person or shop that sells stationery

non-starter a person or idea without any chance of success

borer a person, tool, or insect that makes round holes

legend 5 a famous person or act, esp. in a particular area of activity

performer a person (or thing) that performs (2), esp an actor, musician etc.

Both close and distant alternatives could, in principle, be represented by use of a disjunctive feature structure. However, LAUREL does not permit disjunction of arbitrary feature structures, for reasons discussed in Section 2.5. The operation which we use instead is that of generalisation, $\sqcup$, the opposite of unification. As described in Chapter 2 this yields a single, non-disjunctive, feature structure, containing only the information common to both feature structures. For example the lexical entry for *albino 2* would be:

```
albino L_0_2
   < > <= lex-individual < >
   < QUALIA > < ( animal_L_1_1+2 < QUALIA >
                 \/  plant_L_2_1 < QUALIA > ) <> .
```

The generalisation of two level alternatives will result in a psort which is equivalent to the higher of the two, except in the case where some inherited values were overridden by the lower alternative, since if $F_1 \sqsubseteq F_2$ then $F_1 \sqcup F_2 = F_2$.

For the close alternatives, the decision not to allow disjunction in LAUREL is not unduly restrictive. A reasonably specific psort can be produced by generalisation, and this allows quite well for the use of *etc* which disjunction does not (as *etc* presumably indicates that some members of a class are being enumerated). On the whole we expect that if the type system is set up in a way which accords with the lexicographer's intuitions about classes, generalisation will tend to give a reasonably specific and appropriate result.

Distant alternatives are more problematic because the results of generalisation will be very uninformative. The feature structure resulting from analysing "person, tool or insect" in the definition of *borer* above, for example, will be of type **physical**, the most general type for concrete nouns. This seems to be a case where the disjunction would be preferable because it would carry so much more information. One alternative would be to split the entries into different senses. However, it is quite noticeable that many distant alternatives arise in definitions of derived nominals. In cases such as *borer*, the LDOCE lexicographers have apparently used the genus terms more loosely than usual to indicate the prototypical agents or instruments of an action. If taken strictly, this definition and a considerable proportion of the other LDOCE definitions of derived nominals are over-restrictive; here worms and molluscs are excluded, which is inappropriate.

As outlined in section 4.3.2, my proposed representation of derived nominals in LAUREL involves the use of default inheritance from the result of the application of a regular morphological rule to the verb, and thus the bulk of the information in the lexical entry would not be derived from the genus term. In those cases such as *borer*, which are formed by subject nominalisation (Bauer, 1983), information about the lexical semantic type of the derived nominal will be obtained from the selectional restriction on the subject of the verb. Hence using the (relatively) uninformative generalisation of the disjuncts as a psort to augment the derived structure is a better option than taking the probably over-restrictive disjunction.

Thus our common strategy for both close and distant alternatives is to take the generalisation of the psorts involved. Although this might lead to an underspecified type for the lexical entry in the case of those distant alternatives where there is no connection with a verb entry, at least it is possible to detect such entries, and to make use of other strategies (possibly even user intervention) in order to further refine them.

6.8 Individuation shifting kernels

The last major class of definitions which I wish to consider are those where a shift in individuation takes place. In the following section these are discussed in some detail, using the representations described in Chapter 5.

It is frequently the case that dictionary definitions involve a shift in individuation between the genus and the definiendum. In the following examples I have underlined the relator where appropriate and italicised the genus:

meal^{1} **1** an amount of *food* eaten at one time

whisky 2 an amount of *this* [whisky 1] drunk in one glass

band^{1} **1** a thin flat narrow piece of *material* . . .

waste^{1} **1** an unused or useless stretch of *land*

haulm [U] the stems of crops like peas, beans, potatoes etc, left after gathering

down^{6} **1** [U] fine soft feathers

policeman a member of a *police force*

band^{3} **2** a group of *musicians* . . .

One difference between *band*^{1} *1* and *material*, for instance, is that the former is a countable individualised object and the latter is an uncountable mass. The effect of *piece* is that it individuates *material*, shifting the type from an uncountable mass to a countable individual. In this particular case the shape of the resultant individual is specified ("thin flat narrow"). The 'portioning' sense extension, exemplified by whisky, is a special case of individuation. We refer to this class of definitions as QUANTITY/MASS.

In contrast, the MEMBER/GROUP class of definitions shift between individual and group denoting nouns. I distinguish between these cases and PART/WHOLE examples, discussed in section 6.9.2, where there is not necessarily any shift of semantic class. For example, even though a dolmen is described as being made up of individual stones, it is not a group noun as defined in section 5.3, in contrast to *band*^{3} *2*.

dolmen a group of upright *stones* supporting a large flat piece of stone, built in ancient times in Britain and France

Particular kernel structures tend to indicate different classes of relationship. Their effects are summarised in Figure 6.3. I will consider the representation of both of these classes in the following sections. This will lead to a refinement of the informal definition of the categories of relationship given above.

6.8.1 Quantity/mass

Definitions of the form exemplified by *meal*^{1} *1*, *whisky 2* and *band*^{1} *1* can be regarded as producing an individual from a mass denoting genus term by describing the way in which the entity is to be individuated; in the case of concrete nouns this may involve specifying its physical shape to some extent (e.g. *strip*), or its quantity (e.g. *glass*), or merely indicating that there is some limit to physical extent without saying what that limit is (e.g. *piece*, *amount*).

Thus, if we assume that the lexical semantics of nouns is defined in terms of qualia structure, we can regard the lexical entry as inheriting all its qualia from the genus term except that which refers to physical form. The FORM feature can be regarded as being specified by the relator; for *amount* all that can be specified is that there is some individuation, (i.e. that the value for relative form in the qualia structure is **portion**) for *piece* we might want to assume physical separation, for *band* the FORM role could be more completely instantiated. Of course the other qualia (TELIC,

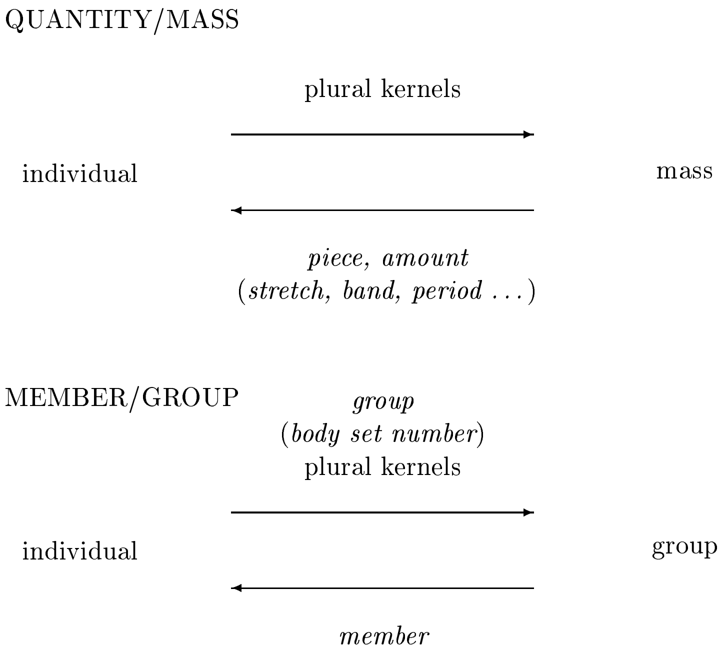


Figure 6.3: The more significant definition structures which cause shifts in individuation.

origin, constituency) may also have to be further instantiated, possibly even overriding the inherited information, but this is no different from the normal cases of hyponymy. We can motivate this because it appears that mass nouns can be taken as referring distributively, in the context of dictionary definitions, thus a "piece of material" will always be "material" and all the properties of "material" should apply. There has been considerable discussion of whether this property can be taken to hold of mass nouns in general, see for example Bunt (1985), Pelletier and Schubert (1986), but the sort of examples which have been claimed as exceptions to this (e.g. "are electrons gold?", "is a chair leg furniture?") do not arise in dictionary definitions in our experience. Figure 6.4 shows an entry for *meal*^{1} *1* and the instantiated feature structure that results.

The example of *whisky 2* is clearly a special case in that the individual and the mass reading are senses of the same headword. This is an example of the 'portioning' sense extension. As discussed in previous chapters (in Sections 4.3.3 and 5.2.2) most food denoting nouns can also be used to denote a portion of that food, and in the case of drinks the extended sense is frequently lexicalised. The portioning rule transforms the syntactic type of the lexical entry from mass to count, but preserves the qualia structure with the exception of the feature form. The effects of the lexical rule on the qualia structure thus correspond to the inheritance behaviour described above. Since a lexicalised example of sense extension may override the information predicted, treating *whisky 2* as formed by sense extension from *whisky 1* is precisely equivalent to the inheritance treatment given above (see Figure 6.5).

The other possibility that I will consider under the QUANTITY/MASS heading is the use of count denoting kernels in the definition of a mass denoting sense. This is usually indicated by a syntactically plural kernel, which may or may not be complex. For example:

haulm [U] the stems of crops like peas, beans, potatoes etc, left after gathering

darning 2 [U] clothes that need to be or are being darned

down^{6} **1** [U] fine soft feathers

gravel 2a [U] small bits of stone-like material in the bladder

Such definitions correspond to cases where something is regarded as individuated with respect to one predicate (e.g. *feather*^{1} *1*) but not with respect to the sense being defined (*down*^{6} *1*). This meal L_1_1

```
   < > <= lex-portion < >
   < QUALIA > < food_L_0_1+2 < QUALIA >.
```

$$\left[\begin{array}{l} \textbf{lex-count-noun} \\ \text{ORTH} = \textbf{meal} \\ \text{QUALIA} = \left[\begin{array}{l} \textbf{comestible} \\ \text{AGENTIVE} = \textbf{nomagent} \\ \text{TELIC} = \left[\begin{array}{l} \textbf{strict-trans-sem} \\ \text{IND} = \boxed{0}\ \textbf{eve} \\ \text{PRED} = \textbf{and} \\ \text{ARG1} = \left[\begin{array}{l} \textbf{verb-formula} \\ \text{IND} = \boxed{0} \\ \text{PRED} = \textbf{eat\_L\_0\_1+2} \\ \text{ARG1} = \boxed{0} \end{array}\right] \\ \text{ARG2} = \boxed{\textbf{binary-formula}} \end{array}\right] \\ \text{FORM} = \left[\begin{array}{l} \textbf{nomform} \\ \text{RELATIVE} = \textbf{portion} \end{array}\right] \\ \text{CONSTITUENCY} = \boxed{\textbf{nomconst}} \\ \text{OBJECT-INDEX} = \boxed{1}\ \textbf{entity} \\ \text{PROPERTIES} = \boxed{\textbf{phys-properties}} \end{array}\right] \end{array}\right]$$

Figure 6.4: Lexical entry and expanded feature structure for *meal*^{1} *1*.

$$\left[\begin{array}{l} \textbf{lex-uncount-noun} \\ \text{ORTH} = \textbf{whisky} \\ \text{QUALIA} = \left[\begin{array}{l} \textbf{comestible} \\ \text{AGENTIVE} = \textbf{nomagent} \\ \text{TELIC} = \boxed{\textbf{verb-sem}} \\ \text{FORM} = \left[\begin{array}{l} \textbf{nomform} \\ \text{RELATIVE} = \textbf{mass} \end{array}\right] \\ \text{CONSTITUENCY} = \boxed{\textbf{nomconst}} \\ \text{OBJECT-INDEX} = \boxed{1}\ \textbf{entity} \\ \text{PROPERTIES} = \left[\begin{array}{l} \textbf{phys-properties} \\ \text{STATE} = \textbf{liquid\_a} \end{array}\right] \end{array}\right] \end{array}\right]$$

whisky 1

$$\left[\begin{array}{l} \textbf{lex-count-noun} \\ \text{ORTH} = \textbf{whisky} \\ \text{QUALIA} = \left[\begin{array}{l} \textbf{comestible} \\ \text{AGENTIVE} = \textbf{nomagent} \\ \text{TELIC} = \boxed{\textbf{verb-sem}} \\ \text{FORM} = \left[\begin{array}{l} \textbf{nomform} \\ \text{RELATIVE} = \textbf{portion} \end{array}\right] \\ \text{CONSTITUENCY} = \boxed{\textbf{nomconst}} \\ \text{OBJECT-INDEX} = \boxed{1}\ \textbf{entity} \\ \text{PROPERTIES} = \left[\begin{array}{l} \textbf{phys-properties} \\ \text{STATE} = \textbf{liquid\_a} \end{array}\right] \end{array}\right] \end{array}\right]$$

whisky 2

Figure 6.5: Lexical entries for whisky.

```
down L_0_1
   < > <= lex-mass < >
   < QUALIA > < ( feather_L_1_1 + plural ) < QUALIA > .
```

$$\left[\begin{array}{lll}
\textbf{lex-uncount-noun} & & \\
\text{ORTH} & = & \textbf{down} \\
\text{QUALIA} & = & \left[\begin{array}{lll}
\textbf{physical} & & \\
\text{AGENTIVE} & = & \textbf{nomagent} \\
\text{TELIC} & = & \boxed{\textbf{verb-sem}} \\
\text{FORM} & = & \left[\begin{array}{lll} \textbf{nomform} & & \\ \text{RELATIVE} & = & \textbf{mass} \end{array}\right] \\
\text{CONSTITUENCY} & = & \left[\begin{array}{lll} \textbf{nomconst} & & \\ \text{PARTICLES} & = & \textbf{feather\_L\_1\_1} \end{array}\right] \\
\text{OBJECT-INDEX} & = & \boxed{0}\ \textbf{entity} \\
\text{PROPERTIES} & = & \boxed{\textbf{phys-properties}}
\end{array}\right]
\end{array}\right]$$

Figure 6.6: Lexical entry for *down*^{6} *1*, showing inheritance of qualia structure from *feathers.*

will only apply to mass nouns where there is some obvious granularity; most are defined in terms of other (potentially) mass nouns such as *material* or *substance.*

Again inheritance from the genus term will involve the TELIC, ORIGIN and CONSTITUENCY qualia, but the FORM will be specified as unindividuated with respect to the sense being defined (although we also represent the potential for individuation with respect to the genus term, in order to make the appropriate inferences about distributivity of reference possible). We can motivate this difference from the normal hyponymy relation, since we are in effect claiming that properties are inherited from the plural entity (i.e.`( feather_L_1_1 + plural )`). See Figure 6.6.

6.8.2 Member/group

Because of the principle of substitutability, definitions of group nouns in dictionaries such as LDOCE will normally be group denoting noun phrases. In some cases the genus term will be a relatively specific group noun, for example:

crew ^{1} **3** a rowing team

Such definitions can be treated in the same way as that described for normal cases of hyponymy; by default the entire qualia structure is inherited from the entry for the genus term *team* 2, but more specific information in the differentia can augment this and may override it.

However, there is another class of definitions where the genus phrase is of the form 'DET *group of* N '. Here the noun is principally being defined in terms of its members, for example:

band^3 **2** a group of musicians ...

These definitions pose more problems since there is very little semantic information that can be inherited from *group.* As illustrated with the example of *dolmen,* given earlier, the use of *group of* does not necessarily indicate a noun which is group-denoting in the technical sense. In other cases, a group noun may be defined using a plural genus term, for example:

audience 1 the people listening to or watching a performance, speech, television show, etc.

In this case, the definition cannot be substituted for the *audience* in contexts where it is used with singular agreement or refers to the group as a whole.

(1) The audience were very noisy tonight.
The people listening to the performance were very noisy tonight.
The audience was very noisy tonight.
*The people listening to the performance was very noisy tonight.
The audience was tiny.
*The people listening to the performance was tiny.

$$\left[\begin{array}{l}
\textbf{lex-count-noun} \\
\text{ORTH} = \textbf{band} \\
\text{CAT} = \boxed{\textbf{noun-cat}} \\
\text{SEM} = \left[\begin{array}{l}
\textbf{obj-noun-formula} \\
\text{IND} = \boxed{0} \left[\begin{array}{l} \textbf{obj} \\ \text{AGR} = \left[\begin{array}{l} \textbf{agr} \\ \text{NUM} = \textbf{sg} \end{array}\right] \end{array}\right] \\
\text{PRED} = \textbf{band\_L\_3\_2} \\
\text{ARG1} = \boxed{0} \\
\text{PLMOD} = \textbf{false} \\
\text{QUANT} = \textbf{true}
\end{array}\right] \\
\text{QUALIA} = \left[\begin{array}{l}
\textbf{human} \\
\text{AGENTIVE} = \boxed{\textbf{agentivestuff}} \\
\text{TELIC} = \left[\begin{array}{l}
\textbf{strict-trans-sem} \\
\text{IND} = \boxed{2}\ \textbf{eve} \\
\text{PRED} = \textbf{and} \\
\text{ARG1} = \left[\begin{array}{l} \textbf{verb-formula} \\ \text{IND} = \boxed{2} \\ \text{PRED} = \textbf{perform\_L\_0\_2} \\ \text{ARG1} = \boxed{2} \end{array}\right] \\
\text{ARG2} = \boxed{\textbf{binary-formula}}
\end{array}\right] \\
\text{FORM} = \left[\begin{array}{l} \textbf{nomform} \\ \text{RELATIVE} = \textbf{group} \end{array}\right] \\
\text{CONSTITUENCY} = \left[\begin{array}{l} \textbf{groupconst} \\ \text{ELEMENTS} = \boxed{\textbf{human}} \end{array}\right] \\
\text{OBJECT-INDEX} = \boxed{0}
\end{array}\right]
\end{array}\right]$$

Figure 6.7: Lexical entry for *band*

(Presumably the lexicographer felt that it was better to use *people* than *group of people* for example, which perhaps suggests a greater cohesion between the individuals than is appropriate here.) For such examples a representation has to be built based on information about the individual members.

Clearly, given the type system outlined in Chapter 5, information about the type of a group's members makes it possible to infer the type of the group noun. If the members are of type **human** then the group as a whole will be of type **human**, and so on. Furthermore the CONSTITUENCY role can be instantiated with the appropriate information. So the following entry could be produced for *band*^{3} *2*:

```
band L_3_2
    < > <= lex-group < >
    < QUALIA > = human
    < QUALIA : CONSTITUENCY : ELEMENTS >
              < ( musician_L_0_1 + plural ) < QUALIA >.
```

However, this leaves some information unspecified, in particular the TELIC role. I assume that this can be inherited from the TELIC role of the members, so the TELIC role of *band*^{3} *2* is inherited from *musician*:

```
    < QUALIA : TELIC >
           < musician_L_0_1 < QUALIA : TELIC >.
```

This gives the lexical entry shown in Figure 6.7. The inheritance of the telic role needs some justification, since in general usage it is not possible to assume that a group as a whole has a property even if all its members have that property, and the property is one which could hold of the group as a whole. For example:

> All the members of the committee are against the poll tax.

does not entail that:

> The committee is against the poll tax.

Inheritance of the telic role might also be problematic. There could be a group of musicians who got together to play football, for example, so:

The King's Road football team is a group of musicians.

could be true, in which case the purpose of the group described would not be equivalent to that lexically specified by *musician*. However such examples are exceptional, and in the special case of dictionary definitions, if a group is defined in terms of its members it can be taken to inherit appropriate properties from them; we have not found any counter-examples to this in LDOCE so far. Intuitively, it seems unlikely that a dictionary would define a group in terms of its members, if they had a purpose or function which was distinct from that of the group. Even if a concept such as the musicians' football team were lexicalised, the lexicographer would have to specify the function of the group explicitly to avoid being misleading, and thus the default inheritance from its members' telic role would be overridden.

The other half of this category of definitions are those in which an individual is defined in terms of the group of which it is a member. For example:

policeman a member of a police force

committeeman a member of a committee

In general, the appropriate type can be obtained from examination of the ELEMENTS feature of the CONSTITUENCY part of the qualia of the group denoting psort, if this has been fully specified. Since the use of complex kernels of the form "a member of X" is, in LDOCE, almost entirely restricted to human denoting definitions, it can normally be assumed that the lexical semantic type is **human**, even if the elements of the group are not known.

In contrast with the mass noun case, group nouns do not refer distributively, so we would not expect to be able to make any logically valid inheritance about the members of a group based on information specified for the group as a whole. Even when predicates which normally refer distributively are applied to a group, it is not possible to conclude that they apply to individual members.

The committee voted for Major.

does not entail that each member of the committee voted for Major. Again dictionary definitions are a special case, since a noun would not be defined in terms of its group membership if it denoted an atypical member of the group. Or, at least, the atypicality would be explicitly marked or indicated. Thus inheriting the TELIC role from the group genus will normally give an appropriate result, since the function of an individual, regarded as the member of a group, will normally be the same as the function of the group as a whole, although possibly more specialised. Inheriting the agentive role is more problematic, since it is likely to refer to the group collectively.

To summarise, this treatment of group denoting nouns defined in terms of their members, and of individuals defined in terms of the group of which they form a part, is based on two assumptions. The first is that there are limitations to the lexicalisation of group formations in English; the second is that in the context of dictionary definitions we can take distributive predicates as being true of members of groups, by default.

6.9 Non-IS_A definitions

Informally all the definitions discussed up till now can be referred to as IS_A definitions, in that appropriate inheritance relationships can be described, which will lead to their instantiation with an appropriate lexical semantic type. In this section I describe three types of definitions which are difficult to treat automatically, because no appropriate inheritance relationship can be derived. Fortunately, the percentage of noun definitions which cause such difficulties is relatively low (around 4-5%) in LDOCE. I will refer to definitions in any of these three classes as non-IS_A definitions.

6.9.1 Non-IS_A adjectival modification

Perhaps the most straightforward case of non-IS_A definition is exemplified by:

teddy bear a toy bear filled with soft material

When this definition is processed automatically *teddy bear* is specified as inheriting its qualia structure from *bear*[1] *1*, which is clearly not appropriate. As is well known the logical form for a phrase such as *toy bear* will not involve *bear* being predicated of the individual denoted, and so the inheritance relationship would not be expected to hold. Other examples of adjectives that behave in this way (sometimes referred to as *privatives*) are *fake*, *former* and *alleged*. If *bear* is regarded as the genus term in the usual way such definitions violate Landau's pinpointing principle. The only structure from which *teddy bear* can be defined to inherit is that appropriate for *toy bear* as a whole. In principle an appropriate psort feature structure could be constructed in LAUREL by parsing *toy bear*, and in practice this might be feasible since a relatively small number of privative adjectives are in the defining vocabulary. However, it is impossible to predict anything very specific about the lexical semantics of combinations involving *fake*, *artificial* and so on, so it is not clear that this is worthwhile. Furthermore, in LDOCE at least, very few definitions seem to involve this type of adjectival modification.

6.9.2 Part/whole definitions

Some dictionary definitions define entities as parts of some larger whole, or, conversely, in terms of their components. However, entities often consist of many different types of component and it is not possible to predict in general how the semantics of the whole relate to the semantics of the components. In definitions such as the following, *part* can be taken as the genus term but is very underspecified.

juice[1] **1** the liquid part of fruit, vegetables, and meat

flesh 3 the soft part of a fruit or vegetable, which can be eaten

organ 1 a part of an animal or plant that has a special purpose

albumen 1 the white or colourless part of an egg

sleeve 1 a part of a garment for covering (part of) an arm

backwater 1 a part of a river, usu. a branch,...

base 9 the main part or substance of a mixture

magazine 3 the part of a gun or weapon of that type...

bastion 1 a part of the wall of a castle or fort...

morning 1 the first part of the day...

cathode the part of an electrical instrument...

collar[1] **1** the part of a shirt, dress, or coat...

root[1] **1** the part of a plant that...

artillery 2 the part of the army trained to use such weapons

proscenium 2 the part of a stage that...

In all these examples the entry word is related to the of-complement via *part*. Only in those cases, however, where the complement is a uniform homogeneous entity can the semantics of the defined word sense be predicted e.g. *part of a river*. A *part* of *fruit*, *vegetable* or *meat* is not necessarily *liquid* and *mass* (*juice* and *flesh*), and *part of an animal or plant* (*organ*) can also be *mass* and *liquid* instead of a countable individualised thing. There are a few generalisations; a part of some physical entity will also be a physical entity, part of some time period will also be a time period etc, but this is only adequate to predict semantic type in extremely broad terms.

The reverse situation also occurs when an entry is described as a complex whole of components:

batter^2 a mixture of flour, eggs and milk...

mortar^2 a mixture of lime, sand, and water...

block and tackle an arrangement of wheels and ropes...

assembly line an arrangement of workers and machines...

waterworks 1 buildings, pipes, and supplies of water...

post chaise a carriage and horses...

goulash meat and vegetables cooked together with paprika...

bust^1 **1** the human head, shoulders, and chest...

Either a relator is used to indicate the whole (*mixture*, *arrangement*) or a coordination is used from which the complex whole has to be inferred. In either case, however, there is no way to infer the lexical semantic type of the whole from the composition, at least without extensive knowledge and inference. Some of the examples above are relatively straightforward, in that the whole refers to a simple grouping of components, e.g. *bust*^{1} *1*, but others involve further modification to their properties. This may be explicitly mentioned, for example in the definition of *goulash* the meat and vegetables are stated to be cooked, or implicit, as in the case of *mortar*^{2}, where the mixing initiates a chemical reaction. Complex wholes are often more covertly and implicitly coded, as in the example of *rocket*, repeated here:

rocket 1 a tube-shaped case packed with gunpowder ...

The same sort of remarks apply here, but it is much more difficult to detect automatically that this should be treated as a non-IS_A definition.

Thus it does not seem possible to produce a uniform semantics for constituency that usefully predicts the lexical semantics of the word sense defined. No 'PART-OF' relationship can be defined in LAUREL since there is no consistent relationship between feature structures to denote. The COMPONENT/WHOLE definitions form a relatively small proportion of the total number of noun definitions (estimated at around 3.5% on the basis of frequency of the relators *part*, *mixture* and so on, and of conjunction with *and*). Thus failure to treat them automatically is not too problematic from the practical viewpoint of extracting information from MRDs. It is clear that for some classes of definitions a predictive relationship can be established; parts of the body, for example, but these remain to be investigated on a case by case basis (and automatic treatment would involve the use of more sophisticated analysers of dictionary definitions).

6.9.3 Grey areas

A further group of definitions involve modifiers which indicate that some change in the properties of the genus term has taken place, but where it is not clear whether an IS_A relationship is still involved. Consider, for example, the definition of *ice*:

ice^1 **1** water which has frozen to a solid ...

It is not self-evident whether this can be taken as an IS_A relationship or not, since usually *frozen* modifies intersectively (for example, frozen parsley is still parsley). In the case of *water*, however, information about physical state might be a necessary component of the meaning, at least in the sense that the LDOCE lexicographers are using:

liquid^2 **1** (a type of) substance not solid or gas, ...

water^1 **1** the most common liquid ...

Under this view both state and chemical composition are specified and the definition is not IS_A.

However, *water* can be used to refer to composition alone, at least technically. The statement *ice is water* seems to be true. It is thus not clear whether to take liquidity as being a default or necessary property of this sense of *water*. Furthermore *liquid*, *gas* and *solid* do seem to be used in LDOCE to mean the state that some substance has under normal conditions:

carbon dioxide the gas produced when animals breathe out or when CARBON (1) is burned in air

dry ice CARBON DIOXIDE in a solid state ...

It is clear that *carbon dioxide* refers to composition rather than to state. If *water* is taken to be a liquid by default, and the definitions for *ice* and *water* are both IS_A, we are driven to the unintuitive conclusion that there is an IS_A relationship between *ice* and *liquid*, and that *a liquid* denote some substance which is normally rather than necessarily liquid.

Such problems would be more serious if we were attempting to derive a full description of word meaning from dictionary definitions. However the aim of instantiating lexical semantic properties in the sort of type system which I have been discussing is much more modest, especially since the relationships set up in the LAUREL description language are interpreted simply as allowing for inheritance of parts of the feature structure. From this viewpoint whether this particular definition is a problem in practice just depends on how the type system is set up. In the fragment described, physical-state is a property which can have values **solid**, **liquid** etc. In effect this means that this information is defeasible, and thus allowing the entry for *ice* to inherit from *water* is possible, since the value for PHYSICAL-STATE will be overridden. However if we had set this up as being a necessary property, by distinguishing subtypes of **physical**, then the inheritance relationship would be invalid.

6.10 Conclusion

In this chapter I have discussed the way in which dictionary definitions may be used as sources of data with which to to instantiate lexical semantic representations. The simple cases of straightforwardly hyponymic definitions give straightforward inheritance relationships and I have discussed the classes of definitions which are the most important exceptions to this simple treatment. I have tried to show how the LAUREL representations suggested are consistent with the semantics of the definitions. Although the semantics of the constructions involved are, in some cases, complex and not well understood, we have found that we can make some simplifying assumptions and treat LDOCE definitions as a specialised sublanguage, which allows us to extend the process of semi-automatically constructing a lexicon containing detailed lexical semantic information to the great majority of noun definitions.

Having gone through these examples in a relatively formal way, it is possible to re-examine the notion of taxonomy. As originally stated in Chapter 3, the notion of an inheritance hierarchy in LAUREL is essentially a derived concept; a network of individual inheritance specifications involving, for example, the qualia part of the lexical signs can be regarded as a hierarchy. It is clear that if we have a taxonomy of word senses derived from definitions which individuate consistently, which do not have coordinated genus terms and which only involve intersective modification, then the taxonomy will be isomorphic to the LAUREL qualia inheritance hierarchy. (Perhaps unsurprisingly taxonomies derived from dictionary definitions of animals tend to fit this structure, at least if group nouns are ignored, but many other taxonomies do not.) However, it is also clear that tracing inheritance relationships in LAUREL will not lead to simple tree-like structures in many cases, but will involve, for instance, inheritance from artificially constructed nodes (disjunction represented as generalisation), partial and multiple inheritance (individuation shifts), cycles (equality assertions in synonym representation) and so on. In the non-intersective definitions there will be no inheritance relationship between the simple genus and the definiendum. Taxonomies are useful as initial approximations to classification (as will be illustrated in the next chapter) and as descriptive devices, but the examples given in this chapter illustrate that attempting to define inheritance over taxonomies by distinguishing between different types of links cannot be productive, because simple hierarchies are inadequate to capture the sort of inheritance relationships discussed, and because in some cases there is no appropriate inheritance relationship between word senses.

Chapter 7

Automatic construction of lexicons

Until now I have omitted any detailed discussion of the practical techniques involved in applying this approach to lexical semantic representation and analysis of dictionary definitions to the construction of large lexicons for NLP and linguistic research. In this chapter I will outline how it is possible to build entries described in LAUREL from MRDs in an effective way. I will describe TAXUS, a program for the identification and disambiguation of genus terms in definitions which allows LAUREL representations to be derived without extensive hand-coding. I will also summarise some other work on extracting information from dictionaries to produce entries in the ACQUILEX LKB system which uses LAUREL as the representation language.

In many ways, an ideal approach to extracting lexical semantic information would be to produce the most probable parses of the definition with a general purpose analyser (possibly parametrised for the specialised sublanguage of dictionary definitions), and to go from this to the instantiated representation. However, this is still impractical with current analysers. All the practical techniques described involve the use of some special purpose heuristic analysis and the representations are constructed semi-automatically. In general, this involves user interaction in cases where the heuristics seem to be failing, or to confirm important choices. The work described has demonstrated the construction of large scale fragments rather than complete lexicons.

Any use of MRDs depends on the previous conversion of the sources available from the publishers into a tractable format. MRDs are typically available as typesetting tapes which use more or less arcane codes to indicate the structure of the definitions; these have to be deciphered before it is possible to convert the data into a form suitable for storage in a database. Most of the techniques for extracting information described here make use of MRD information stored in the Cambridge lexical database system (LDB). This preparatory stage is essential (and time-consuming), and the issues of database representation involved are non-trivial. The process of going from the typesetting tape for LDOCE to a dictionary mounted in the LDB is described in Alshawi *et al.* (1989); the latest version of the LDB software itself is described in Carroll (1992). Boguraev *et al.* (1992) discuss the range of options for representing dictionaries in a database. The LDB itself adopts the *two-level model* in which an arbitrarily deeply tagged version of the MRD entry is stored and a set of arbitrarily complex indices are constructed as a separate level. This allows flexible querying, while still retaining all the information in the original dictionary; thus the entry as it appears in print can be reconstructed from the information in the LDB.

The LDB can be used to store data derived from MRDs in such a way that it can be queried in conjunction with the original. Thus intermediate stages in analysis can be saved and stored in the database as a *derived dictionary.* Briscoe (1991) makes a clear distinction between an LDB and an LKB. Even though a dictionary may undergo considerable processing before it can be stored in an LDB, there is no requirement for representation of the information in a formal language and there is no attempt to fit the description into any particular semantic framework. An LKB, on the other hand, is an instantiation of a theoretical treatment of the lexicon, from which it is possible to make precise predictions about the behaviour of individual words. Although this distinction is extremely important, in practice there is often some blurring of the division. The process of constructing lexical entries described here results in an LKB which is defective in that it retains the LDOCE sense distinctions (except in some limited cases of sense extension). There is little alternative to this currently, given the lack of an adequate theory of sense distinction, although it would clearly be desirable to have a more linguistically motivated treatment. As I will show in section 7.2, some ACQUILEX LKB fragments are close to the original dictionary in other ways. Furthermore, in order to use an LKB for linguistic research some database functionality is needed. For example some form of indexing to allow efficient querying is essential, which is not part of the functionality of an ordinary lexical representation language. The LKB system supports some limited indexing, and since the LDB has no rigid data-model, the two could be combined to allow indices to be created for accessing fully expanded lexical entries represented in LAUREL.

7.1 Constructing LAUREL entries using disambiguated taxonomies

In order to (semi-)automatically produce inheritance relationships in LAUREL from dictionary definitions, it is necessary to identify and disambiguate their genus terms. It is also necessary to distinguish between definitions which licence inheritance (IS_A definitions) and those which do not, or which require a more complex treatment. In this section I will describe the TAXUS program which does this semi-automatically. [1] Although I discussed the construction of inheritance relationships from definitions in a way which de-emphasised taxonomies, TAXUS does construct taxonomies, working top-down from user-specified word senses. The aim in using taxonomies is to achieve a degree of conceptual coherence. It is highly desirable to work with a conceptually related set of definitions because it simplifies the user-interactions and the use of heuristics. Furthermore, given that it is not possible or desirable to construct an entire lexicon at once, it is useful to produce groups of entries with a limited number of parents, in order to be able to efficiently specify information manually and check that is inherited appropriately.

The finer details of constructing the LAUREL entries, such as individuation shift, can be ignored at this stage, because the inclusion of count, mass and group terms in the same taxonomy does not lead to lack of coherence, since most of their properties will be straightforwardly inherited. All except one of the various classes of IS_A definitions discussed in the previous chapter are accepted as forming part of an IS_A taxonomy. The exception is coordination, which is not currently dealt with by TAXUS. However the senses related by the non-IS_A definitions are excluded because these can be very different in properties from their genus terms, and thus do not give a sufficiently coherent taxonomy[2].

The stages involved in the construction of lexical entries are illustrated in Figure 7.1. LDOCE_Inter is a derived dictionary containing information extracted from the LDOCE grammar codes as an intermediate stage in producing the Alvey Tools lexicon (Carroll and Grover, 1989). LDOCE_Sem contains information extracted from partially parsing the definitions, including the genus terms. This contains two sets of data, one version derived using the parser developed by Alshawi (1989), the other extracted by Vossen (1990a). LDOCE_Tax contains sense disambiguated IS_A genus terms identified semi-automatically from information in LDOCE_Sem (and LDOCE) using the TAXUS program. LAUREL entries are constructed by running a series of conversion rules which access information in LDOCE_Tax, LDOCE_Inter and LDOCE itself.

7.1.1 Parsing definitions to find genus terms

As I described in the previous chapter, the genus term for nouns can normally be assumed to be the syntactic head of the defining noun phrase. Quite simple pattern matching techniques can be used to identify the genus term, by utilising heuristics to determine the likely end of the noun phrase

[1] Much of this section is taken from Copestake (1990b), where I described a slightly different (and unnamed) version of this program.

[2] TAXUS can also be used to construct PART_OF taxonomies, but in LDOCE at least, the only coherent taxonomy of significant size that I have found is that involving parts of the body.

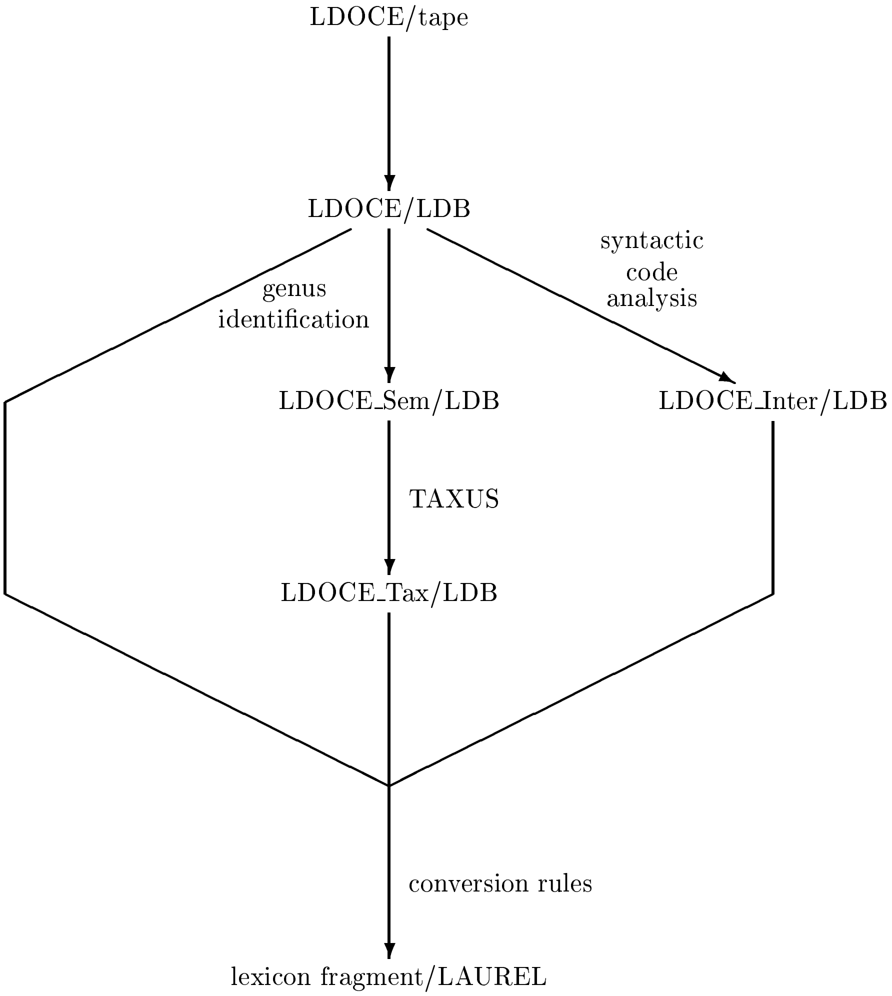


Figure 7.1: Constructing a lexicon fragment

(e.g. Chodorow *et al.*, 1985). However, in order to adequately represent a high proportion of the definitions it is necessary to take account of the less straightforward possibilities, in particular, the complex kernels and coordinated structures. Alshawi's (1989) robust parser used hierarchically organised phrasal patterns to identify genus terms and extract some further information from the differentia. FPar (for flexible parser) is a system based on Alshawi's work, which was incorporated in the LDB system by Carroll (1992), and has been used with a more elaborate pattern system to analyse definitions from the Spanish VOX dictionaries by Ageno *et al.* (1992).

Other researchers have used more conventional parsers with special purpose grammars, including Slator working on LDOCE (Wilks *et al.*, 1989) and Vossen who has also worked on LDOCE (Vossen, 1990a) and subsequently the Van Dale Dutch dictionary. This approach allows a more detailed analysis of the definitions (with an associated cost in development time). There have been attempts to use general purpose grammars to parse definitions. Ahlswede and Evens (1988) used the Linguistic String Parser on definitions from Webster's Seventh, but only managed to parse about two-thirds of the definitions in their sample, and extending the grammar did not improve the success rate sufficiently for further effort to be thought worthwhile. Montemagni and Vanderwende (1992) report the use of the PLNLP system to parse definitions from the Italian Garzanti Dictionary. This system was developed to be very robust, and to accept unrestricted input text, at the cost of producing a representation which may be much less useful for further interpretation than that which would be given by a grammar with more restricted coverage. Thus the analyses produced may be regarded as "preliminary syntactic sketches" (Calzolari *et al.*, 1991) which do not resolve lexical or most structural ambiguity. Thus considerable further processing is required to produce a semantic representation.

Briscoe and Carroll (1991) report the use of a wide-coverage general grammar and lexicon (the Alvey Tools system) on LDOCE definitions. Since this does produce disambiguated representations, large numbers of parses are produced for even quite short definitions. A stochastic technique was used to find the most probable parses, using semi-automatically acquired training data. The training phase also involved the augmentation of the grammar and lexicon. Although the lexicon was discovered to be quite inadequate in some respects, little modification to the grammar was needed. Ultimately, the use of a large coverage general grammar, combined with disambiguation heuristics, is likely to be the best approach to parsing definitions. But this is not yet a feasible way of extracting large-scale information from MRDs.

Initially I used Alshawi's system for work on LDOCE. However the phrasal patterns developed by Alshawi do not handle all the cases of complex kernels which have proved important and the identification of the genus term is rather unreliable. Although these problems could indubitably have been resolved by extending the set of patterns, in practice this would have been very time-consuming. The system developed by Vossen is considerably more reliable and I have therefore used data extracted by him for the work described here. This gives the genus word (or phrase for coordinated genus terms) and the relator, where appropriate. Both the genus word and relator could be lexically ambiguous with respect to both homonyms and senses — I will use the term *class* to refer to the undisambiguated genus term. The data was stored in the LDB as the derived dictionary, LDOCE_Sem.

7.1.2 Sense disambiguation

The TAXUS procedure creates sense-disambiguated taxonomies top-down, using the classes identified by the parser. Starting from a user-supplied word sense, all the entries in which this is used as the genus term are found, and the program then recurses on the word sense corresponding to each of these entries, terminating on word senses which are never used as genus terms. A series of heuristics are used to disambiguate the genus terms, and to distinguish between IS_A definitions and others, the decision being confirmed by the user in the more critical cases. So starting from *liquid*^{2} *1*, *oil*^{1} *1* is one of the (large number of) entries found; this in turn is the genus term in entries such as *paraffin 1*. If *paraffin 1* is not used as a genus term in any entries, the process will bottom out at this point, and *paraffin 1* will be a leaf node in the taxonomy.

Finding the entries with a given genus term is a two stage process; first all the entries with the appropriate word in the genus term position will be found by querying the LDB system and then these are filtered by disambiguating. For example, in order to find the entries where *liquid*^{2} *1* is used as the genus term, the entries which have *liquid* as a class are first retrieved and these are then examined to attempt to determine whether *liquid*^{2} *1* is the sense being used, as opposed to one of the adjectival senses under *liquid*^{1}, or the technical sense *liquid*^{2} *2* “either of the consonants r or l”. A fragment of the taxonomy derived starting from *liquid*^{2} *1* is shown below; indentation is used to indicate depth. All the nodes correspond directly to LDOCE senses.

- liquid^2 1
  - drink^2 1
    - absinth
    - anis
    - aperitif
    - applejack
    - arrack
    - barley water
    - beer 1
      - ale
      - barley wine
      - bitter^3
      - bock
      - home brew
      - lager 1
      - light ale
      - mild^2
      - pale ale
      - porter^3
      - stingo
      - stout^2
      - wallop^1 2
    - beer 2
    - beer^3

Even if the eventual LAUREL representation is not considered, disambiguation is essential in order to produce a coherent taxonomy and to allow termination of the taxonomy building process before it potentially includes all the entries in the dictionary. For example, one of the entries which has *liquid* as a class is *spirit*^{1} *13*. If the program recursed on *spirit* and did not do the sense discrimination (or did not do it correctly) entries such as *devil* and *ghost* would be retrieved. These in turn would give rise to further problems of the same kind.

In the case of *liquid*, disambiguation will be quite straightforward. The adjectival homonym is excluded since the taxonomy is restricted to nouns because of the convention that a sense is defined by a phrase of the same part of speech (so the genus terms in definitions of nouns will always be nouns, for example). A set of (weighted) heuristics is used to find the “best match” among the possibilities with the same syntactic category as the potential genus term — *liquid*^{2} *1* and *liquid*^{2} *2* are thus the only candidates.

LDOCE has two features which can be exploited to facilitate this process. The use of a restricted core vocabulary in the definitions makes parsing easier and restricts the possible non-leaf nodes in the taxonomies, and the entries have a series of associated codes which can be exploited for sense-disambiguation. Thus the set of heuristics used on LDOCE rely partially on a feature particular to it — most entries have a *box code* in the MRD which gives a limited amount of semantic information. In particular, one part of the box code contains codes such as L for liquid, H for human, T for abstract. One heuristic is based on whether the value of this code for the potential genus sense is compatible with the code for the entry. For example, the codes for *oil*^{1} *1* and *liquid*^{2} *1* are both L, but that for *liquid*^{2} *2* is T, which is not compatible with L. This heuristic acts as a rather crude filter: to distinguish between senses with compatible box codes, a more general heuristic is used which depends on the degree of overlap between the uninflected forms of the open-classed words in the definitions. This provides some measure of semantic “closeness”. Another heuristic is simply that lower numbered senses are more likely to be used as genus terms, because the ordering of senses tends to reflect their relative frequency/typicality (although there are many exceptions).

Such sense disambiguation is, of course, not completely reliable. So taxonomy derivation is semi-automatic — the user is asked confirm or override the important choices, those concerning the entries which might be non-leaf nodes in the taxonomy. In tests on deriving taxonomies for concrete nouns from LDOCE this resulted in the user being consulted on about 5% of the entries. This check is enough to prevent spurious paths being followed, whilst an incorrect decision about a leaf node cannot affect the final taxonomy too greatly. The user also has the option to specify that the parser has assigned the undisambiguated genus term incorrectly, and is also consulted as to whether an IS_A link is appropriate. This again ensures that the taxonomy being created remains conceptually coherent. Such yes/no decisions are normally fairly quick and easy to make. The user also has to decide whether to merge closely related senses which might be too difficult to disambiguate.

As discussed in the previous chapter, it is necessary to recognise definitions where it is inappropriate to form a default inheritance relationship in LAUREL. Some such definitions can be automatically recognised. In other cases problematic definitions of leaf nodes are checked with the user. The user is always asked to confirm that potential non-leaf nodes are connected by an IS_A relationship. Vossen's parser returns X as the relator and Y as the class for definitions where the genus phrase is of the form "DET X of Y" and some other cases such as "any of several kinds of Y" are also covered. The taxonomy building program is currently parametrised to automatically assume that an IS_A relationship is appropriate when X is *type* or *kind* and that it is never appropriate if X is *part*. These relators form the majority of cases of complex heads, but in others the user is consulted as part of the taxonomy building process. Recursion is continued if the user decides that the relationship involved is IS_A in that some form of inheritance from the putative genus term is valid, even if the LAUREL representation will have to be complicated as discussed in the previous chapter.

In some cases heuristics based on pattern matching can be used to automatically determine whether a relationship is IS_A or not. For example patterns like "a number of Y" versus "the number of Y" differ systematically in their interpretation in definitions such as:

crowd^2 **1** a large number of people gathered together

population 1 the number of people (or animals) living in a particular area, country etc

The semi-automatic approach allows data gathering in the course of taxonomy building to be utilised in order to determine such heuristics, which allows refinements to be made to the procedure so that the user need be consulted less frequently. Some of the problematic definitions are not, however, currently recognised automatically. In particular, definitions which involve the modification of the genus term by some process, such as that of *rocket*, repeated below, are not automatically detected as non-IS_A at this stage.

rocket 1 a tube-shaped case packed with gunpowder ...

It would only be worth further refining the heuristics to pick up such examples if it turned out that fairly fixed constructions were involved. Examination of the data for the non-leaf nodes where the user has specified that no IS_A relationship is involved allows recognition of any systematic patterns that occur sufficiently frequently. However, TAXUS is intended to be a 'quick and dirty' approach to sense-disambiguation and taxonomy construction, as an alternative to more sophisticated and time-consuming analysis techniques, and there is little point in complicating it to cope with low frequency patterns.

7.1.3 Results

I have used TAXUS mainly on concrete nouns in LDOCE. Several IS_A taxonomies were built, starting from *animal, plant, person, man, woman, substance, instrument* and so on, covering about 7,500 word senses. These taxonomies were built at a rate of 500-1000 word senses per hour (depending on the degree of interaction needed, determined largely by the potential ambiguity in the genus term). Errors are mostly caused by failures of the disambiguation heuristics. These tend to be localised (and thus easy to detect) because they are frequently caused by a particular genus term being hard to disambiguate. The rate of failure of the heuristics depends on the taxonomy being built, but seems acceptable, even in some of the more difficult cases. For example *instrument* is used as a class in 185 entries, and is ambiguous between 3 senses; tool, musical instrument and a very general sense, intended to cover uses such as "an instrument of fate".

instrument 1 an object used to help in work
2 an object which is played to give musical sounds (such as a piano, a horn, etc.)
3 someone or something which seems to be used by an outside force to cause something to happen

The first two senses both have the box code J which denotes a "movable solid", the third is given a box code Z, which covers anything. The current heuristics (which had been developed and tuned on other taxonomies) gave an incorrect assignment in 16 cases (8.5%). This is a relatively difficult case because the box code heuristic will not distinguish between the first two senses. Disambiguation depends mainly on the word overlap heuristic. However, the unreliability of the box code causes problems — 6 of the failures were incorrect assignments to the general sense of instrument, because the lexicographer had used a box code which was incompatible with J. In most cases these were errors, but *organ* is one example for which an incompatible code, N, denoting an immovable solid, was assigned which was probably correct. The rest of the failures were caused by the word overlap heuristic preferring the wrong sense. For example, *metronome* and *record player* were assigned to the second sense of instrument rather than the first because *music* and *sound* occur in their definitions.

The errors caused by the parser finding an incorrect class are relatively infrequent when using the grammar developed by Vossen. However, there are occasional problems with structural ambiguity in definitions such as:

armadillo a small animal native to the warm parts of the Americas

which was assigned the class *native*.

Some further problems are inherent in the dictionary entries. There are cycles in the definitions. For example, *animal*^{1} *1* is defined as "a living creature ..." and *creature*^{1} *1* is defined as "an animal of any kind". Because the taxonomy was produced starting from *animal* this does not cause any serious problems — *creature*^{1} *1* is included in the hierarchy under *animal*^{1} *1*, recursion on *creature*^{1} *1* happens as normal but the cycle is detected by the program, marked in the output, and infinite recursion is avoided. A more serious problem is when the cycles result in words being excluded altogether. For example, the parser assigns *cow* as the genus term of *cattle* and *cattle* as the genus term of *cow*, and about 17 word senses were excluded from the animal taxonomy because of this. I checked for omissions in the animal hierarchy by retrieving all LDOCE entries with box codes appropriate for animals, but this is not possible in general, because most codes do not correspond neatly with taxonomies. In general, omissions can only be detected by creating taxonomies which cover the majority of the entries and then determining which words have not been incorporated. When the final representation is produced, cycles have to be dealt with manually.

The maximum depth of the taxonomies derived was 7. The taxonomies are fairly flat, with about 95% leaf nodes. Taxonomies derived from LDOCE will have a greater percentage of leaf nodes than those from other dictionaries because of the use of a restricted defining vocabulary.

The taxonomy creation program is a general tool which can be customised for use on any MRD where the definitions can be parsed to produce an undisambiguated genus term. The heuristics can be modified for other dictionaries — only the use of box codes is really LDOCE specific. Thus this work is more generally applicable than that reported in Guthrie *et al.* (1990) which relies heavily on LDOCE box codes and also on LDOCE subject codes. The basic program described here has been used with different heuristics for LDOCE verbs for which the box codes are in general unhelpful (although verbal taxonomies tend not to be particularly useful in constructing lexical entries). Ageno *et al.* (1992) incorporated a taxonomy builder based on the TAXUS program into SEISD, a considerably more elaborate system for automatic extraction of information from MRDs which has been used on a Spanish monolingual dictionary, VOX. Disambiguation in both these cases is more difficult; however reasonable results can be obtained even with less reliable heuristics, because some assignments are checked by the user, although far more interaction time is required.

7.1.4 From taxonomies to LAUREL representation

TAXUS can be used in conjunction with the Cambridge lexical database software. The taxonomies created can be stored and queried in the LDB in conjunction with the dictionary for which they were derived; so producing queries to retrieve, for example, all the group nouns that are in the *animal* taxonomy is very easy. Consider, for example, the following entry from the machine readable version of LDOCE:

vixen *n* **1** a female fox

The corresponding entry in LDOCE_Tax created by TAXUS is:

```
vixen
1 isa ((fox) 1 (1)) ((animal) 1 (1 2)) ((creature) 0 (1))
((animal) 1 (1 2))
```

This indicates the IS_A link with *fox* and also specifies the higher levels in the taxonomic chain. This is necessary to allow queries which retrieve all the direct and indirect daughters of a particular taxonomic node, since the LDB, like most database systems, does not support recursive queries. The two-element cycle in this particular taxonomy is indicated by the repetition of the sense of *animal*.

Once the taxonomically linked lexical entries derived from LDOCE have been stored in the LDB, it is quite easy to automatically create LAUREL entries for the definitions which license straightforward inheritance. These entries can also incorporate syntactic information derived from the LDOCE grammar coding scheme and some rather limited semantic information derivable from the LDOCE box codes. The box code for the entry for *vixen* is D, indicating a female animal. The entry in LDOCE_Inter is:

```
vixen
(1 ((Cat N) (Takes NULL) (COMMON +) (COUNT +)
    (GROUP -) (PLU -) (REG +)))
```

In this case the feature specification is the default for nouns which are unmarked by specific grammar codes.

The procedures for converting these entries to a LAUREL description are written directly in Lisp. The example shown here results in the following output:

```
vixen L_0_1
   < > <= lex-individual < >
   < QUALIA : PROPERTIES : SEX > = female
   < SENSE-ID : DICTIONARY > =  "LDOCE"
   < SENSE-ID : LDB-ENTRY-NO > =  "39499"
   < SENSE-ID : HOMONYM-NO > =  "0"
   < SENSE-ID : SENSE-NO > =  "1"
   < QUALIA > < fox_L_1_1 < QUALIA >.
```

The default syntactic properties for nouns correspond with the psort, `lex-individual`. The specification of SEX is determined by the box code. I have not shown the SENSE-ID information in the previous examples, of lexical entries. It is specified to retain the connection with the original dictionary — it allows the LKB system to incorporate menu commands to view the dictionary entry, which is helpful for debugging purposes, for example. The parent psort `fox_L_1_1` has a very similar automatically derived LAUREL entry, without the specification for SEX. However its parent psort, `animal_L_1_1+2`, was manually created, and specified to have qualia type **animal**.

Once such entries have been produced, they can be loaded into the LKB system and the valid descriptions are expanded into well-formed typed feature structures. At this point, any errors in the extraction process which give type mismatches will become apparent. The utility of typing for error checking when representing automatically acquired data can be seen in the following example. In the current type system the feature SEX is introduced at type **creature-properties**, which is an appropriate value for the PROPERTIES feature of **creature** and its subtypes. A few LDOCE entries have incorrect semantic codes; *Irish stew* for example has code K, which should correspond to a male human or animal. Since *Irish stew* is under `food_L_1_1` in the taxonomy, its lexical semantic type is **c_artifact**, and the type of its PROPERTIES feature is not consistent with **creature-properties**. Therefore SEX was detected as an inappropriate feature. When expansion of the automatically generated lexical entry fails to produce a well-formed feature structure, an error message is output, and the mistake can then be manually corrected. If LAUREL were not a typed language, errors such as this would not be detected automatically in this way.

In contrast, automatic classification of lexical entries by type, according to feature information, can be used to force specification of appropriate information. A lexical entry which has not been located in a taxonomy will be given the most general possible type for its QUALIA. However if a value for the feature SEX has been specified this forces a type for PROPERTIES of **creature-properties**. This is of limited utility, however, since the specification of the value of PROPERTIES as **creature-properties** does not force the qualia type to be **creature**, even if it is not appropriate for any other qualia type. For this, full classification would be needed as I discussed in Chapter 2.

In principle, once the entries derived from the individual senses have been produced, they should be reorganised according to a linguistically motivated account of sense distinction. This might involve comparing data from other dictionaries. At the moment the only gesture that can be made towards this is the removal of the senses which are wholly predicatable by sense extension or morphological rule. This can be accomplished by applying all possible rules to a lexical entry (suitably restricting the number of recursive applications to ensure the process terminates) and removing any existing entries which are equivalent to those generated. However, this really requires more extensive data, which must be acquired from the differentia.

The techniques described above only cover the more straightforward taxonomic relationships and cannot produce fully instantiated lexical semantic structures automatically. The current conversion rules do not deal with all the cases of complex kernels, but could be extended. However, this would involve the use of heuristics, to determine what sort of individuation shift was occurring, for example, since some of the patterns indicated in Section 6.8 apply to more than one case. Heuristics would also be needed to augment the LDOCE grammar code information, for group terms and relational nouns, for example.

By manually associating information with lexical entries which corresponded to some of the non-leaf nodes in the taxonomy it is possible to semi-automatically create entries such as those shown in previous chapters, with instantiated TELIC and AGENTIVE roles, on the assumption that inheritance is not overridden. A considerably more detailed parse of the definitions is needed to derive entries at a level of detail sufficient to ensure that inherited information is overridden where appropriate, but work by Vossen (1992), Hagman (1992) and Ageno *et al.* (1992) has shown that this is feasible, at least with respect to a limited domain. Their work is discussed in the next section.

7.2 The ACQUILEX LKB

A variety of approaches for acquisition have been investigated by participants in the ACQUILEX project, aimed at the eventual representation of information in the ACQUILEX LKB using a common type system. (One of the most important aspects of the use of a typed feature structure system on the ACQUILEX project was that it enforced a compatible representation, although there was obviously no guarantee that the interpretation of the features and values would be consistent.) The verb type system is essentially that described in section 4.1. This has an underlying linguistic motivation and the interaction of the entries on the syntagmatic plane is described (in fact is an essential part of the encoding). Despite the theory-specific nature of the representation, the lexicon is reusable in that a mapping between the lexical entries in this system and other formalisms can be defined, since a relatively rich representation is used. Thus it is possible to generate a lexicon using the Alvey Tools representation, for example, and to ignore the thematic role specification. Sanfilippo and Poznanski (1992) discuss how the lexicon can be instantiated semi-automatically using information derived from MRDs. However, it turns out that the paucity of the grammar coding in most dictionaries makes it difficult to derive subcategorisation information for languages other than English; this of course demonstrates a problem in relying entirely on MRDs, not a problem in the type system. Furthermore, Sanfilippo's technique for extraction of the lexical semantic information about psychological predicates and movement verbs depended on the thesaurus-like organisation of the Longman's Lexicon, which was derived from LDOCE and therefore could be automatically correlated with it. It is considerably more difficult to extract such information from conventional dictionaries.

For the noun lexical semantic type system, a different approach was adopted. An attempt was made to extract and represent very detailed semantic information about a relatively small subset of the vocabulary, concentrating on food and drink. Representation and acquisition were considered together and the representation was far more closely determined by the nature of the information

$$
\left[\begin{array}{l}
\textbf{lex-noun-sign} \\
\text{ORTH} = \textbf{ketchup} \\
\text{CAT} = \boxed{\textbf{noun-cat}} \\
\text{SEM} = \left[\begin{array}{l} \textbf{unary-formula-entity-arg1} \\ \text{IND} = \boxed{0}\ \textbf{entity} \\ \text{PRED} = \boxed{1}\ \textbf{ketchup\_g\_0\_0} \\ \text{ARG1} = \boxed{0} \end{array}\right] \\
\text{SENSE-ID} = \boxed{\textbf{sense-id}} \\
\text{RQS} = \left[\begin{array}{l}
\textbf{c\_art\_subst} \\
\text{ORIGIN-AREA} = \textbf{string} \\
\text{TELIC} = \left[\begin{array}{l} \textbf{formula} \\ \text{IND} = \textbf{entity} \\ \text{PRED} = \textbf{logical-pred} \\ \text{ARG1} = \left[\begin{array}{l} \textbf{formula} \\ \text{IND} = \textbf{entity} \\ \text{PRED} = \textbf{eat\_L\_0\_1+2} \\ \text{ARG1} = \textbf{sem} \end{array}\right] \end{array}\right] \\
\text{PHYSICAL} = \textbf{true} \\
\text{OBJECT-INDEX} = \textbf{dummy-or-obj} \\
\text{ANIMACY} = \textbf{false} \\
\text{PHYSICAL\_STATE} = \textbf{state\_a} \\
\text{QUAL} = \left[\begin{array}{l} \textbf{phys-qual} \\ \text{COLOUR\_SPEC} = \boxed{\textbf{colour\_spec}} \\ \text{TRANSPARENCY} = \textbf{transparency} \\ \text{SMELL} = \textbf{smell} \\ \text{TASTE} = \textbf{bitter-sweet} \\ \text{TEMPERATURE} = \textbf{temperature} \\ \text{TEXTURE} = \textbf{semi-liquid} \\ \text{SIZE} = \textbf{size} \end{array}\right] \\
\text{QUANT} = \textbf{quantity} \\
\text{SIMILAR} = \textbf{string} \\
\text{CONST.} = \left[\begin{array}{l} \textbf{constituency} \\ \text{UNIFORM} = \textbf{boolean} \\ \text{CONSTS.} = \left[\begin{array}{l} \textbf{full\_list\_of\_predicates} \\ \text{FIRST.} = \textbf{pomodoro} \\ \text{AMOUNT} = \textbf{degree} \\ \text{REST.} = \left[\begin{array}{l} \textbf{full\_list\_of\_predicates} \\ \text{FIRST.} = \textbf{peperone} \\ \text{AMOUNT} = \textbf{degree} \\ \text{REST.} = \left[\begin{array}{l} \textbf{full\_list\_of\_predicates} \\ \text{FIRST.} = \textbf{aceto} \\ \text{AMOUNT} = \textbf{degree} \\ \text{REST.} = \textbf{empty.} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right] \\
\text{FORM} = \left[\begin{array}{l} \textbf{physform} \\ \text{VOLUME} = \textbf{scalar} \\ \text{WEIGHT} = \textbf{scalar} \\ \text{SHAPE} = \textbf{non-individuated} \end{array}\right] \\
\text{EVALUATIVE} = \textbf{evaluative} \\
\text{INGESTION} = \boxed{\textbf{ingestion}} \\
\text{AGEING} = \textbf{age} \\
\text{AGENTIVE} = \boxed{\textbf{formula}} \\
\text{ALCOHOLIC} = \boxed{\textbf{alcoholic}}
\end{array}\right]
\end{array}\right]
$$

Figure 7.2: Entry for *ketchup* extracted from the Italian Garzanti Dictionary

in the dictionary definitions than was the case with the verb type system.

In order to extract detailed semantic information, the class of food and drink definitions was first isolated, using genus term information, and some of the definitions were then analysed manually in order to determine what properties were most commonly described. For food the list of properties included the following: age, capacity, colour, constituency, effect, evaluative, ingestion, quality, similar_to, storage, transport. On the basis of this, an initial sketch of a lexical semantic type system was augmented with a much more detailed set of attributes and values, designed to reflect the patterns found in the dictionary definitions. The lexical semantic noun type system is referred to as the 'relativised qualia structure' (RQS, Calzolari 1991). The patterns found in the definitions can also be exploited as templates in order to analyse them. Some phrases recur frequently in definitions: for example, *per, usato per, atto a, che serve a, utile a* are some of the phrases which indicate use or purpose of instruments in Italian. Ageno *et al.* (1992) directly related the templates used in the FPar robust pattern matching parser to the type system. Their system, SEISD, uses the parse to instantiate the feature structure skeleton, but allows user interaction to confirm or override the automatic analysis. Vossen (1992) and Hagman (1992) used similar templates to guide the extraction of information from the initial output of their parsers. An example of a noun entry expressed in the RQS system is given in Figure 7.2.

From this example it can be seen that the rather more detailed semantic information necessary to fully instantiate the representations described in the previous chapter can potentially be extracted automatically from MRDs. To date many hundreds of such entries have been extracted and represented in the LKB system, in four languages (English, Italian, Spanish and Dutch). Features such as COLOUR may be unmotivated linguistically, but they are easy to extract and can be used for semi-automatic cross-linguistic linking of lexical entries, disambiguating the translations given in bilinguals by selecting the target language sense most semantically similar to the source language sense (Copestake *et al.*, 1992). In order to allow this, the values for some features are specified in terms of a restricted set of types. For example, **bitter-sweet** is one of the 13 possible values for TASTE. In other cases predicates related to LDOCE senses have been used as common values across the different languages.

The advantages of the RQS approach are that the representation guides the process of extraction of information from the definition and there is some guarantee that the entry can be instantiated from the MRD. However, problems arise because neither the needs of an NLP system, nor the overall linguistic theory, have been considered in constructing the entries. Since the elaboration of the RQS type system was driven by analysis of the information contained in the dictionary entries themselves, rather than any coherent view as to their actual use, the lexical entries produced are closely linked to the original dictionary definitions. In many cases the interpretation of the features outside the context of the definitions was not considered. In terms of Briscoe's (1991) distinction between an LDB and an LKB, discussed at the beginning of this chapter, these entries are more LDB than LKB like, notwithstanding the use of a formalised representation language. Thus it is more appropriate to regard these structures as database entries, representing a partial analysis of the dictionary definitions, from which lexical entries might be derived. However, a considerable amount of further processing would be required to do this. The features would have to be interpreted relative to the particular requirements of the target system, and this might be difficult given the lack of theoretical motivation. The values for some features are essentially unanalysed strings and these would have to be replaced with particular predicates. This further processing in itself requires lexical semantic information pertaining to the words in the definitions. It thus makes more sense to build a broad coverage shallow lexicon first before attempting to build such detailed entries.

Even though representations such as that shown in Figure 7.2 are inadequate as lexical entries, this work has shown that MRDs can be used to extract quite complex lexical semantic information. Although further processing would be necessary to produce lexical semantic representations such as those discussed in previous chapters from the RQS information, this does not seem intractable, if we continue to rely on semi-automatic methods and to make use of the regularities in definitions. Much of the RQS information could, in any case, be ignored, since its inclusion in the lexical entry would be unmotivated according to the criteria I discussed in Chapter 1. Most MRDs are better sources of lexical semantic information than they are of syntactic information (other than part of speech). Indeed, since most dictionaries do not contain detailed grammatical coding, the only way of obtaining information from them about the mass/count distinction, or verb subcategorisation, for example, would have to be via the lexical semantic specification. Although LDOCE is an exception, which makes it possible to extract much more information reasonably reliably, it has deficiencies; it is not possible to determine which nouns are relational, for example, from its grammar coding scheme. In such cases corpora might be used to supplement the information. However work on MRDs such as LDOCE has at least demonstrated that lexicographers can produce data which is useful for linguistic research and NLP, as well as conventional dictionary users.

Chapter 8

Conclusion

This thesis has shown that some interesting problems involving lexical semantics can be tackled within a formally defined, unification-based framework. I have defined a representation language, LAUREL, and implemented a system based on it, which has been used to construct the lexicon/grammar fragment described here, and has also been utilised extensively on the ACQUILEX project. The use of MRDs for the acquisition of data demonstrates that this approach to lexical representation is compatible with the needs of NLP. I have shown that it is possible to build large, highly structured lexicons, containing detailed lexical semantic information.

I have described several specific innovations. The representation language has several new features including the use of default unification within a typed feature structure language and the use of psorts. In some ways psorts are a trivial extension to the language, since they are very similar to templates, but they allow relationships between lexical entries to be encoded directly. This means that in building the lexicon it is possible to take advantage of existing structures. It allows some of the intuitively appealing ideas underlying semantic network representations to be applied in a formally defined unification based language. In particular, one very natural way of describing some aspects of lexical semantics is to state that one word sense inherits from another, but specialises or overrides some features. Encoding this directly allows a succinct representation with the advantage that alterations to the description may be made relatively straightforwardly because information is appropriately localised. Since it is possible to acquire information about inheritance relationships between word senses from MRDs, it also provides a practical route to building a lexicon.

I have demonstrated the utility of the language for constructing representations of lexical semantics which are integrated with syntax by developing a treatment of English nouns and their behaviour with respect to individuation and agreement. This account is significant in itself as there is no comparable fragment implemented in any lexical representation language that I know of. I have also shown how the beginnings of a formalisation of the generative aspects of the lexicon have been developed, including treatments of logical metonymy, derivational morphology and conventionalised sense extension. There are challenging linguistic and representational problems here, but the contribution of this thesis has been to clarify what can be done within a relatively conventional unification-based language augmented with lexical defaults and what cannot. In particular, I have pointed out the need to be able to express operations which apply over the whole of the lexicon for the treatment of blocking and discussed several issues in the description of lexical rules which suggest directions for future research.

From the viewpoint of computational lexicography, this thesis has described significant advances over previous work on the extraction of information from definitions. It is straightforward to map information from dictionary definitions which have the classic genus and differentia structure into statements in the LAUREL language. Even though this only partially captures their semantics, it has made it possible to clarify the status of noun definitions which have a more complex structure. This was possible because LAUREL has a well-defined semantics, unlike the semantic network representations previously used for most work of this type. In particular, the options for the treatment of coordinated genus terms are clear and the cases of complex kernels which involve a shift in individuation have been related to a linguistically motivated treatment. From a practical standpoint, my implementation of the ACQUILEX LKB system has allowed myself and other researchers on the project to develop programs to extract lexical entries from MRDs which give good results for a high proportion of definitions. As I have discussed, the use of a typed language is a significant advantage here, since error-checking happens automatically. Furthermore the type system provides a relatively succinct and explicit statement of the encoding used, which is advantageous both because it makes it easier to collaborate in constructing a consistent LKB and also enables lexicons to be reused more effectively when a change in representation is desirable. I have also described the TAXUS program, which allows dictionary genus terms to be disambiguated, which is a necessary preliminary to an adequate representation of taxonomic relationships.

To conclude, I will consider some representation issues which I have left unresolved and discuss how LAUREL might be modified and extended in the light of experience with the system. As I mentioned in Chapter 2, a notion of ultimate well-formedness can be defined in LAUREL which is much stricter than the well-formedness conditions which are actually imposed. The LKB system should probably be extended to enforce ultimate well-formedness, since without this feature structures are not adequately constrained. The difficulty here is the development of an appropriate algorithm — I believe that classification in LAUREL is decidable, given the current conditions on the constraint system, but I have not proved this and its practical tractability is unclear. A related issue is the adoption of a constraint-resolution based technique for parsing/generation, which might also be desirable, although it is not clear how the use of lexical rules would apply in such a system.

The lexical rule mechanism itself could be improved in a variety of ways. Currently it allows arbitrary transformations of feature structures, and although in theory this can be controlled by the type system, it is far from clear how this should be done. The problem here, however, is not so much the language itself, but the lack of a fully developed theory of sense extension. For example, it seems on an intuitive basis, that there is a considerable difference between the `grinding` and `portioning` rules, since portioning can be described as a monotonic addition of constraints but grinding involves a distinct shift in meaning. Sanfilippo (1990) argues that lexical rules, such as that responsible for passive formation, should be seen as an accumulation of constraints, starting from an underspecified feature structure. Possibly `portioning` and similar rules should be defined to act in this way, but I do not think this can apply to `grinding` or other metonymic or metaphoric sense extensions.

The current portioning rule is defined to change the relative form specification in the qualia structure from **mass** to **portion**, although in other respects it can be seen as increasing semantic information. This has the undesirable effect that the lexical semantic type of the result will be underspecified, because although all the qualia structure features, apart from the relative form specification, are specified as being identical on the input and output structures, the qualia type as a whole cannot be (similar remarks apply to the rule for plural formation). We could get round this by allowing identity of types to be encoded in the description language. An alternative would be to change the encoding of information so that relative form was unspecified for the mass cases and the lexical rule only involved addition of constraints. One reason why I did not originally adopt this approach was that it does not allow the type **lex-uncount-noun** to be constrained by the `< FORM : RELATIVE >` value — if this is stated to be underspecified then it could be incorrectly instantiated by **individual**, for example. However it is possible that the concept of type equality constraints could be extended to the type description language to allow a way around this, although I have not worked out the details of such an analysis.

I discussed the control of lexical rules at the end of Chapter 4. Although I believe that the long term goal is an interface to a pragmatic/contextual component which controls rule application, in the short term a more direct approach will have to be taken, if lexical rules are to be used effectively in parsing and generation. Besides the actual control of application of the rules themselves, the representation of established and lexicalised senses has to be considered. I mentioned at the end of Chapter 5 that the direction of application of the lexical rule which I used to derive the non-relational sense(s) of words like *group*, from the relational sense, did not agree with intuitions about which sense was more basic for words such as *forest*. There are other cases where the directionality of a sense extension is unclear. For example, consider the relationship between nouns denoting some liquid used for covering a surface and verbs describing the application of that liquid. I view this as a productive sense extension from noun to verb, since trade names, such as *Sadolin*, can be used in this way:

(1) I've Sadolined that door.

However, *paint* was used as a verb for centuries before it was used as a noun (Quirk *et al.*, 1985:1529). In the case of this particular sense extension, it may well be that most speakers assume the noun to verb direction despite the historical derivation, but intuitions becomes less clear with other examples (see Cruse, 1986:132–134). From the formal viewpoint, directionality matters if a rule is used non-reversibly in that it deletes lexically specified information. It is also relevant to the control of rule application, on the assumption that the non-extended sense will be preferred. Intuitions about directionality are particularly important with respect to the latter criterion, though historical data may not be directly relevant.

Turning now to the use of defaults, the discussion in Chapter 3 can be seen as establishing a range of possible definitions for lexical inheritance mechanisms based on default unification. The choice of the definition actually adopted in LAUREL was based on practical considerations about the sort of data that we were most interested in representing. In the light of experience with the system, I think that the definition should be expanded to allow priorities to be put on defaults. The intention here is not to attempt to resolves multiple inheritance conflicts in taxonomically derived hierarchies, but to allow different types of data to be inherited at different priorities. For example, in acquiring information from MRDs, I used non-default inheritance from the psorts representing individuation. The reason for this was essentially to ensure that in the case of conflicts with taxonomically inherited information, the individuation specification took precedence. However, ideally information inherited from the individuation psorts themselves should have been defeasible, and so non-default inheritance was being used as a crude approximation to priorities on defaults. The notion of priority that seems to be needed is somewhat similar to that used in HAEL (Konolige, 1988), which allows discrete levels of information to be specified, rather than a more general scheme for ordering information in case of conflicts, since I think that only a limited number of classes need be defined. It is likely that this would remove much of the need for non-default inheritance between psorts, so this could be dispensed with, leaving a rather clearer distinction between psorts and types.

I have mentioned in several places the parallels between default in the lexicon and non-monotonic logics, but I do not think that making the connection more formal is a priority, until various aspects of the role of defaults in the lexicon become clearer. One of the problems that arises in practice is to do with the granularity of defaults; intuitively, when one develops a feature structure representation, some parts are more closely related than others, and should be overridden or inherited as 'chunks'. For example, the entire feature structure which makes up the value for the telic role can be considered as a single unit, and it is not clear whether it is reasonable to override parts of it. The treatments of default unification discussed here essentially assume that parts of the feature structure should be inherited independently, unless conflicts arise directly because of reentrancy. A typing scheme may constrain this, because it restricts the set of well-formed feature structures in such a way that connections can be expressed in ways other than reentrancy, as shown in section 2.4.6, but on the whole the assumption is an essential part of the definition of default unification. On an informal level there is a connection here with default reasoning and the assumption of independence. In many treatments of default reasoning it is assumed that given, for example, the default statements *Ravens fly* and *Ravens are black*, that if we know *Tweety is a white raven*, we will still conclude *Tweety flies* on the assumption that, in the absence of information to the contrary, abnormality of colour is independent of flying ability. However, Morreau (in press) has shown that the principle of independence is formally incompatible with a theory of non-monotonic reasoning based on prototypes (under reasonable assumptions about independence and prototypes). It looks as though producing a single theory of non-monotonic reasoning, which covers all intuitive inference patterns, is not possible. However, it is not clear whether the general assumption of independence is required in the lexicon, or whether we should explicitly state which parts of the feature structure are assumed to be independent. It seems to me that formalisation of the inheritance mechanism in terms of non-monotonic logic would be premature, until the answers to issues such as this become clearer.

The use of defaults in LAUREL operates purely paradigmatically and does not carry over into syntagmatic combination. Although default unification as an operation could apply equally well syntagmatically, defaults in LAUREL are associated with the description language. Given this, a treatment of defaults which operated purely within the description language, as in DATR, and thus does not involve any attempt at defining a variant of unification, is perhaps in some ways more appropriate. However, the use of defaults in the description of lexical semantics does interact with pragmatic processing, and in the long-term default inheritance of lexical semantic information and non-monotonic inference at a pragmatic level should be integrated.

The concept of using defaults in the syntagmatic plane within a unification based framework raises a whole range of complex issues which are largely unexplored (although see Evans (1987) on feature specification defaults in GPSG). The distinction is quite fundamental; none of the notions of defaults which I have described affect the basic notion of a feature structure. The default techniques introduced simply allow non-monotonicity to be involved in the description of feature structures and from the syntagmatic viewpoint this could be seen as an abbreviatory convention. Thus changing the basic definition of a feature structure, so that parts of it could be treated as defeasible, would be one possible approach to syntagmatic defaults, but would require a quite radical redefinition of the language. Alternatively it might be possible to adopt an inheritance based approach throughout — syntagmatic combination could be viewed in terms of default inheritance. The relevance of this to lexical semantics is that it is possible that syntagmatic inheritance of qualia structure should be seen as defeasible, as I mentioned in Section 4.2. But developing such an account requires a more precise specification of the relationship between qualia structure and real world knowledge than I have attempted here.

Bibliography

Ageno, A., I. Castellon, G. Rigau, H. Rodriguez, M.F. Verdejo, M.A. Marti and M. Taule (1992) 'SEISD: an environment for extraction of semantic information from on-Line dictionaries', *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP-92),* Trento, Italy, pp. 253–255.

Ahlswede, T. and M. W. Evens (1988) 'A lexicon for a medical expert system' in M. W. Evens (ed.), *Relational models of the lexicon: representing knowledge in semantic networks,* Cambridge University Press, Cambridge, England, pp. 97–111.

Alshawi, H. (1989) 'Analysing the dictionary definitions' in B. Boguraev and E. J. Briscoe (ed.), *Computational lexicography for natural language processing,* Longman, London, pp. 153–169.

Alshawi, H., B. Boguraev and D. Carter (1989) 'Placing the dictionary on-line' in B. Boguraev and E. J. Briscoe (ed.), *Computational lexicography for natural language processing,* Longman, London, pp. 41–63.

Amsler, R. A. (1981) 'A taxonomy for English nouns and verbs', *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics (ACL-81),* Stanford, pp. 133–138.

Asher, N. and M. Morreau (1991) 'Commonsense entailment — a modal theory of non-monotonic reasoning', *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91),* Sydney, Australia, pp. 387–392.

Bach, E. (1986) 'The algebra of events', *Linguistics and Philosophy,* **9**, 5–16.

Bauer, L. (1983) *English word-formation,* Cambridge University Press, Cambridge, England.

van den Berg, M. and H. Prüst (1991) 'Common denominators and default unification', *Proceedings of the Computational Linguistics in the Netherlands (CLIN), First Meeting,* Utrecht, pp. 1–16.

Bird, S. (1992) 'Finite-state phonology in HPSG', *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92),* Nantes, France, pp. 74–80.

Boguraev, B. and E. J. Briscoe (1988) 'Large lexicons for natural language processing: exploiting the grammar coding system of LDOCE', *Computational Linguistics,* **13 (3-4)**.

Boguraev, B. and E. J. Briscoe (eds) (1989) *Computational lexicography for natural language processing,* Longman, London.

Boguraev, B. and B. Levin (1990) 'Models for lexical knowledge bases', *Proceedings of the 6th Annual Conference of the UW Center for the New OED,* Waterloo, pp. 65–78.

Boguraev, B., E. J. Briscoe, J. Carroll and A. Copestake (1992) 'Database models for computational lexicography', *Proceedings of the Euralex '90,* Málaga, Spain. (Edited and published by Biblograf S.A.), pp. 59–78.

Bouma, G. (1990a) 'Defaults in unification grammar', *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics (ACL-90),* Pittsburgh, USA, pp. 165–173.

Bouma, G. (1990b) 'Non-monotonic inheritance and unification', *Proceedings of the First International Workshop on Inheritance in Natural Language Processing,* Tilburg, The Netherlands, pp. 1–8.

Brachman, R. J. and J. G. Schmolze (1985) 'An overview of the KL-ONE knowledge representation language', *Cognitive Science,* **9**, 171–216.

Briscoe, E. J. (1991) 'Lexical issues in natural language processing' in E. Klein and F. Veltman (ed.), *Natural language and speech,* Springer-Verlag, pp. 39–68.

Briscoe, E. J., and J. Carroll (1991) 'Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars', Technical Report No 224, University of Cambridge, Computer Laboratory.

Briscoe, E. J., A. Copestake and B. Boguraev (1990) 'Enjoy the paper: lexical semantics via lexicology', *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90),* Helsinki, pp. 42–47.

Briscoe, E. J., and A. Copestake (1991) 'Sense extensions as lexical rules', *Proceedings of the IJCAI Workshop on Computational Approaches to Non-Literal Language,* Sydney, Australia, pp. 12–20.

Briscoe, E. J., A. Copestake and V. de Paiva (eds) (1991) 'Proceedings of the ACQUILEX workshop on default inheritance in the lexicon', Technical Report No 234, University of Cambridge, Computer Laboratory.

Briscoe, E. J., A. Copestake and V. de Paiva (eds) (in press) *Inheritance, defaults and the lexicon,* Cambridge University Press, Cambridge, England.

Bunt, H. (1985) *Mass terms and model-theoretic semantics,* (Cambridge Studies in Linguistics 42) Cambridge University Press, Cambridge, England.

Cahill, L. J. and R. Evans (1990) 'An Application of DATR: the TIC Lexicon', *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI-90),* Stockholm, pp. 120–125.

Calder, J. (1987) 'Typed unification for natural language processing' in E. Klein and J. van Benthem (ed.), *Categories, polymorphism and unification,* Centre for Cognitive Science, University of Edinburgh, Edinburgh, pp. 65–72.

Calder, J. (1991) 'Some notes on priority union', Paper presented at the ACQUILEX Workshop on Default Inheritance in the Lexicon, Cambridge.

Calzolari, N. (1991) 'Acquiring and representing semantic information in a lexical knowledge base', *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation,* Berkeley, California, pp. 188–197.

Calzolari, N., T. Marti and P. Vossen (1991) 'Taxonomies and feature structures', ACQUILEX Deliverable No. 9.

Carpenter, R. (1990) 'Typed feature structures: inheritance, (in)equality and extensionality', *Proceedings of the First International Workshop on Inheritance in Natural Language Processing,* Tilburg, The Netherlands, pp. 9–18.

Carpenter, R. (1991) 'Skeptical and credulous default unification with applications to templates and inheritance' in E. J. Briscoe, A. Copestake and V. de Paiva (ed.), *ACQUILEX Workshop on Default Inheritance in the Lexicon,* Technical Report No 234, University of Cambridge, Computer Laboratory.

Carpenter, R. (1992) *The logic of typed feature structures,* (Tracts in Theoretical Computer Science), Cambridge University Press, Cambridge, England.

Carpenter, R. (in press) 'Skeptical and credulous default unification with applications to templates and inheritance' in E. J. Briscoe, A. Copestake and V. de Paiva (ed.), *Defaults, Inheritance and the Lexicon,* Cambridge University Press, Cambridge, England.

Carroll, J. (1992) 'The ACQUILEX lexical database system: system description and user manual' in A. Sanfilippo (ed.), *The (other) Cambridge ACQUILEX papers,* University of Cambridge Computer Laboratory. Technical Report No. 253, pp. 24–48.

Carroll, J. and C. Grover (1989) 'The derivation of a large computational lexicon for English from LDOCE' in B. Boguraev and E. J. Briscoe (ed.), *Computational lexicography for natural language processing,* Longman, London, pp. 117–134.

Chierchia, G. and R. Turner (1988) 'Semantics and property theory', *Linguistics and Philosophy,* **11**, 261–302.

Chodorow, M. S., R. J. Byrd and G. E. Heidorn (1985) 'Extracting semantic hierarchies from a large on-line dictionary', *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics (ACL-85),* University of Chicago, pp. 299–304.

Copestake, A. (1990a) 'Some notes on mass terms and plurals', Technical Report 190, Computer Laboratory, University of Cambridge.

Copestake, A. (1990b) ‘An approach to building the hierarchical element of a lexical knowledge base from a machine readable dictionary’, *Proceedings of the First International Workshop on Inheritance in Natural Language Processing,* Tilburg, The Netherlands, pp. 19–29.

Copestake, A. (1992) ‘The ACQUILEX LKB: representation issues in semi-automatic acquisition of large lexicons’, *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP-92),* Trento, Italy, pp. 88–96.

Copestake, A. and E. J. Briscoe (1991) ‘Lexical operations in a unification based framework’, *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation,* Berkeley, California, pp. 88-101.

Copestake, A., B. Jones, A. Sanfilippo, H. Rodriguez, P. Vossen, S. Montemagni and E. Marinai (1992) ‘Multilingual lexical representation’ in A. Sanfilippo (ed.), *The (other) Cambridge ACQUILEX papers,* University of Cambridge Computer Laboratory. Technical Report No. 253, pp. 117–129.

Cruse, D. A. (1986) *Lexical semantics,* Cambridge University Press, Cambridge, England.

Dahlgren, K. (1988) ‘Using commonsense knowledge to disambiguate word senses’ in V. Dahl and P. Saint-Dizier (ed.), *Natural language understanding and logic programming, II,* Elsevier Science Publishers (North-Holland), pp. 255–275.

Delgrande, J. P. (1988) ‘An approach to default reasoning based on a first-order conditional logic: revised report’, *Artificial Intelligence,* **36**, 63–90.

Dowty, D. (1979) *Word meaning and Montague Grammar,* Reidel, Dordrecht.

Dowty, D. (1989) ‘On the semantic content of the notion ‘thematic role’.’ in G. Chierchia, B. Partee, and R. Turner (ed.), *Property theory, type theory and natural language semantics,* D. Reidel, Dordrecht, pp. 69–129.

Dowty, D. (1991) ‘Thematic proto-roles and argument selection ’, *Language,* **67**, 547–619.

Dowty, D., R. Wall and S. Peters (1981) *Introduction to Montague Grammar,* Reidel, Dordrecht.

Emele, M. and R. Zajac (1990) ‘Typed unification grammars’, *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90),* Helsinki, pp. 293–298.

Evans, R. (1987) ‘Towards a formal specification of defaults in GPSG’ in E. Klein and J. van Benthem (ed.), *Categories, polymorphism and unification,* University of Edinburgh, Edinburgh, pp. 73–93.

Evans, R. and G. Gazdar (1989a) ‘Inference in DATR’, *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics (EACL-1989),* Manchester, England, pp. 66–71.

Evans, R. and G. Gazdar (1989b) ‘The semantics of DATR’ in A. G. Cohn (ed.), *Proceedings of the Seventh Conference of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB-89),* Pitman/Morgan Kaufmann, London, pp. 79–87.

Evans, R. and G. Gazdar (1990) ‘The DATR Papers. Cognitive Science Research Paper CSRP 139’, School of Cognitive and Computing Sciences, University of Sussex.

Flickinger, D. P. (1987) ‘Lexical rules in the hierarchical lexicon’, PhD thesis, Stanford University.

Gazdar, G. (1987) ‘Linguistic applications of default inheritance mechanisms’ in P. Whitelock, H. Somers, P. Bennett, R. Johnson, and M. McGee Wood (ed.), *Linguistic theory and computer applications,* Academic Press, London, pp. 37–68.

Gazdar, G. (1990) ‘An introduction to DATR’ in R. Evans and G. Gazdar (ed.), *The DATR Papers. Cognitive Science Research Paper CSRP 139,* School of Cognitive and Computing Sciences, University of Sussex, pp. 1–14.

Gazdar, G., E. Klein, G. Pullum and I. Sag (1985) *Generalized phrase structure grammar,* Basil Blackwell, Oxford.

Grimshaw, J. (1990) *Argument structure,* MIT Press, Cambridge, Mass..

Guthrie, L., B. M. Slator, Y. Wilks and R. Bruce (1990) ‘Is there content in empty heads?’, *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90),* Helsinki, pp. 138–143.

Hirst, G. (1987) *Semantic interpretation and the resolution of ambiguity,* Cambridge University Press, Cambridge, England.

Hobbs, J. (1987) 'World knowledge and world meaning', *Proceedings of the 3rd Workshop on Theoretical Issues in Natural Language Processing (TINLAP-3),* Las Cruces, New Mexico, pp. 20–25.

Hobbs, J., W. Croft, T. Davies, D. Edwards and K. Laws (1987) 'Commonsense metaphysics and lexical semantics', *Computational Linguistics,* **13**, 241–250.

Jackendoff, R. (1990) *Semantic structures,* MIT Press, Cambridge, Mass..

Kaplan, R. (1987) 'Three seductions of computational psycholinguistics' in P. Whitelock, H. Somers, P. Bennett, R. Johnson, and M. McGee Wood (ed.), *Linguistic Theory and Computer Applications,* Academic Press, London, pp. 149–188.

Kasper, R. T. and W. C. Rounds (1986) 'A logical semantics for feature structures', *Proceedings of the 24th Annual Conference of the Association for Computational Linguistics (ACL-86),* Columbia University, pp. 235–242.

Kasper, R. T. and W. C. Rounds (1990) 'The logic of unification in grammar', *Linguistics and Philosophy,* **13 (1)**, 35–58.

Kay, M. (1979) 'Functional Grammar', *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistic Society,* Berkeley, pp. 142–158.

Kay, M. (1984) 'Functional unification grammar: a formalism for machine translation', *Proceedings of the 10th International Conference on Computational Linguistics (COLING-84),* Stanford, California, pp. 75–78.

Kilbury, J., P. Naerger, and I. Renz (1991) 'DATR as a lexical component for PATR', *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL-91),* Berlin.

Klavans, J. L. and N. Wacholder (1990) 'From Dictionary to Knowledge Base via Taxonomy', *Proceedings of the 6th annual conference of the Waterloo Centre for the New OED and Text Retrieval,* Waterloo.

Konolige, K. (1988) 'Hierarchical autoepistemic theories for non-monotonic reasoning: preliminary report', SRI Technical note 446. SRI International, Menlo Park, California.

Krieger, H-U. and J. Nerbonne (in press) 'Feature-based inheritance networks for computational lexicons' in E. J. Briscoe, A. Copestake and V. de Paiva (ed.), *Inheritance, defaults and the lexicon,* Cambridge University Press, Cambridge, England.

Krifka, M. (1987) 'Nominal reference and temporal constitution: towards a semantics of quantity', *Proceedings of the 6th Amsterdam Colloquium,* University of Amsterdam, pp. 153–173.

Landau, S. I. (1984) *Dictionaries: the art and craft of lexicography,* Scribner, New York.

Landman, F. (1989) 'Groups I + II', *Linguistics and Philosophy,* **12 (5,6)**, 559–606, 723-744.

Levin, B. (1985) 'Lexical semantics in review', MIT Working Papers on the Lexicon.

Levin, B. (1992) *Towards a lexical organization of English verbs,* Chicago University Press, Chicago.

Levinson, S. C. (1983) *Pragmatics,* Cambridge University Press, Cambridge, England.

Link, G. (1983) 'The logical analysis of plurals and mass terms: a lattice-theoretical approach' in Bäuerle, Schwarze and von Stechow (ed.), *Meaning, use and interpretation of language,* de Gruyter, Berlin, pp. 302–323.

Link, G. (1988) 'The logic of plurals, LP: review of the basic ideas.' in Gärdenfors (ed.), *Generalized quantifiers: studies in linguistics and philosophy.,* Reidel, Dordrecht.

McDermott, D. and J. Doyle (1980) 'Non-monotonic logic, I', *Artificial Intelligence,* **13**, 41–72.

Mel'čuk, I. and A. Zholkovsky (1988) 'The explanatory combinatorial dictionary' in M. W. Evens (ed.), *Relational models of the lexicon: representing knowledge in semantic networks,* Cambridge University Press, Cambridge, England, pp. 41–74.

Moens, M., J. Calder, E. Klein, M. Reape, and H. Zeevat (1989) 'Expressing generalizations in unification-based grammar formalisms', *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics (EACL-89),* Manchester, England, pp. 66–71.

Montemagni S. and L. Vanderwende (1992) 'String patterns versus structural patterns for extracting semantic information from dictionaries', *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92),* Nantes, France.

Moore, R. (1985) ‘Semantical considerations on nonmonotonic logic’, *Artificial Intelligence,* **25**, 75–94.

Moore, R. (1989) ‘Unification-based semantic interpretation’, *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL-89),* Vancouver, pp. 33–41.

Morreau, M. (in press) ‘Norms or inference tickets? A frontal collision between intuitions’ in E. J. Briscoe, A. Copestake and V. de Paiva (ed.), *Inheritance, defaults and the lexicon,* Cambridge University Press, Cambridge, England.

Nunberg, G. D. (1978) ‘The pragmatics of reference’, PhD thesis, University of California, reproduced by the Indiana University Linguistics Club.

Nunberg, G. D. and A. Zaenen (1992) ‘Systematic polysemy in lexicology and lexicography’, Paper presented at Euralex ’92.

Ostler, N. and B. T. S. Atkins (1991) ‘Predictable meaning shift: some linguistic properties of lexical implication rules’, *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation,* Berkeley, California, pp. 76–87.

Partee, B. and M. Rooth (1983) ‘Generalized conjunction and type ambiguity’ in R. Bauerle, C. Schwarze and A. von Stechow (ed.), *Meaning, use and interpretation of language,* de Gruyter, pp. 361–368.

de Paiva, V. (in press) ‘Types and constraints in the LKB’ in E. J. Briscoe, A. Copestake and V. de Paiva (ed.), *Inheritance, defaults and the lexicon,* Cambridge University Press, Cambridge, England.

Pelletier, F. J., and L. K. Schubert (1986) ‘Mass expressions’ in D. Gabbay and F. Guenthner (ed.), *Handbook of philosophical logic, Vol 4,* Reidel, Dordrecht.

Pelletier, F.J., and L. K. Schubert (1988) ‘Problems in the representation of the logical form of generics, plurals, and mass nouns’ in LePore (ed.), *New directions in semantics,* Academic Press, London, pp. 385–451.

Pereira, F. C. N. and S. M. Shieber (1984) ‘The semantics of grammar formalisms seen as computer languages’, *Proceedings of the 10th International Conference on Computational Linguistics (COLING-84),* Stanford, California, pp. 123–129.

Pollard, C. and I. Sag (1987) *An information-based approach to syntax and semantics: Volume 1 fundamentals,* CSLI Lecture Notes 13, Stanford CA.

Pollard, C. and I. Sag (in press) *Head-driven phrase structure grammar,* Chicago University Press, Chicago.

Procter, P. (editor) (1978) *Longman Dictionary of Contemporary English,* Longman, England.

Pulman, S. G. (1983) *Word meaning and belief,* Croom Helm, London.

Pustejovsky, J. (1989) ‘Type coercion and selection’, *Proceedings of the Eighth West Coast Conference on Formal Linguistics (WCCFL VIII),* Vancouver.

Pustejovsky, J. (1991a) ‘The generative lexicon’, *Computational Linguistics,* **17(4)**, 409–441.

Pustejovsky, J. (1991b) ‘The syntax of event structure’, *Cognition,* **30**.

Pustejovsky, J. and S. Bergler (eds) (to appear) *Lexical semantics and commonsense reasoning,* Springer Verlag, Berlin.

Quillian, M. R. (1968) ‘Semantic memory’ in M. Minsky (ed.), *Semantic Information Processing,* MIT Press, Cambridge, Massachusetts, pp. 216–70.

Quirk, R., S. Greenbaum, G. Leech and J. Svartvik (1985) *A comprehensive grammar of the English language,* Longman, London.

Russell, G., J. Carroll, and S. Warwick (1990) ‘Multiple inheritance in a unification-based lexicon’, *Proceedings of the First International Workshop on Inheritance in Natural Language Processing,* Tilburg, The Netherlands, pp. 93–103.

Russell, G., J. Carroll and S. Warwick (1991) ‘Multiple default inheritance in a unification-based lexicon’, *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL-91),* Berkeley, California, pp. 215–221.

Saint-Dizier, P. and E. Viegas (1992) *Proceedings of the second seminar on computational lexical semantics,* IRIT, Toulouse, France, forthcoming with Cambridge University Press, England.

Sanfilippo, A. (1990) ‘Grammatical relations, thematic roles and verb semantics’, PhD thesis. Centre for Cognitive Science, University of Edinburgh, Scotland.

Sanfilippo, A. (in press) 'LKB encoding of lexical knowledge from machine-readable dictionaries' in E. J. Briscoe, A. Copestake and V. de Paiva (ed.), *Inheritance, defaults and the lexicon,* Cambridge University Press, Cambridge, England.

Sanfilippo, A. and V. Poznanski (1992) 'The acquisition of lexical knowledge from combined machine-readable sources.', *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP-92),* Trento, Italy, pp. 80–88.

Scha, R.J.H. (1983) 'Logical foundations for question answering', PhD thesis.

Shieber, S. M., H. Uszkoreit, F. C. N. Pereira, J. Robinson and M. Tyson (1983) 'The formalism and implementation of PATR-II', Research on Interactive Acquisition and Use of Knowledge, volume 1894 of SRI Final Report. SRI International, Menlo Park, California.

Shieber, S. M. (1986) *An introduction to unification-based approaches to grammar,* CSLI Lecture Notes 4, Stanford CA.

Sinclair, J. (ed.) (1987) *Collins COBUILD English language dictionary,* Collins, London.

Touretzky, D. S. (1986) *The mathematics of inheritance systems,* London/Los Altos: Pitman/Morgan Kaufmann.

Touretzky, D. S., J. F. Horty and R. M. Thomason (1987) 'A clash of intuitions: the current state of nonmonotonic multiple inheritance systems', *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87),* Milan, Italy, pp. 476–482.

Vossen, P. (1990a) 'A parser-grammar for the meaning descriptions of LDOCE', Links Project Technical Report 300-169-007, Amsterdam University.

Vossen, P. (1990b) 'The end of the chain: where does stepwise lexical decomposition lead us eventually?', *Proceedings of the 4th Functional Grammar Conference,* Copenhagen, Denmark.

Vossen, P. (1991) 'Comparing noun-taxonomies cross-linguistically', ACQUILEX Working Paper no 014, University of Amsterdam..

Vossen, P. (1992) 'An empirical approach to automatically construct a knowledge base from dictionaries', *Proceedings of the 5th EURALEX,* Tampere, Finland.

Vossen, P. and A. Copestake (in press) 'Untangling definition structure into knowledge representation' in E. J. Briscoe, A. Copestake and V. de Paiva (ed.), *Inheritance, defaults and the lexicon,* Cambridge University Press, Cambridge, England.

Wierzbicka, A. (1988) *The semantics of grammar,* John Benjamins, Amsterdam.

Wilks, Y. (1975) 'A preferential pattern-seeking semantics for natural language inference', *Artificial Intelligence,* **6**, 53–74.

Wilks, Y. (1977) 'Good and bad arguments for semantic primitives', *Communication and Cognition,* **10**, 181–221.

Wilks, Y., D. Fass, C-M. Guo, J. McDonald, T. Plate and B. Slator (1989) 'A tractable machine dictionary as a resource for computational semantics' in B. Boguraev and E. J. Briscoe (ed.), *Computational lexicography for natural language processing,* Longman, London, pp. 193–231.

Zajac, R. (in press) 'Issues in the design of a language for representing linguistic information based on inheritance and feature structures' in E. J. Briscoe, A. Copestake and V. de Paiva (ed.), *Inheritance, defaults and the lexicon,* Cambridge University Press, Cambridge, England.

Zeevat, H., E. Klein and J. Calder (1987) 'An introduction to unification categorial grammar' in N. Haddock, E. Klein and G. Morrill (ed.), *Categorial grammar, unification grammar, and parsing: working papers in cognitive science, Volume 1,* Centre for Cognitive Science, University of Edinburgh, pp. 195–222.

Zwicky, A. M. and J. M. Sadock (1975) 'Ambiguity tests and how to fail them' in J. P. Kimball (ed.), *Syntax and Semantics IV,* Academic Press, New York, pp. 1–36.