Lexical Operations in a Unification-based Framework

Ann Copestake, Ted Briscoe

Computer Laboratory, University of Cambridge, Pembroke Street,
Cambridge, CB2 3QG, UK
aac@cl.cam.ac.uk ejb@cl.cam.ac.uk

Abstract

We consider lexical operations and their representation in a unification based lexicon and the role of lexical semantic information. We describe a unified treatment of the linguistic aspects of sense extension and derivational morphological processes which delimit the range of possible coercions between lexemes and give a preliminary account of how default interpretations may arise.

1 Introduction

In this paper we consider the nature and extent of lexical operations, arguing for a declarative and computationally tractable definition of a lexical rule as a component of a unification-based lexicon employing (default) inheritance and typed feature structures. We claim that this notion of lexical rule is capable of capturing the linguistic element of derivational morphological processes as well as metonymic and metaphoric sense extensions, but is not adequate for the statement of certain types of logical metonymy (e.g. Pustejovsky, 1989a). We argue that such operations must be treated as, in part, 'linguistic' because they have morphological and syntactic consequences, undergo 'blocking', are triggered by grammatically-defined type mismatches and involve default interpretations based on lexical organisation. However, we also argue that such default interpretations can be overridden by contextual information on the basis of more general and open-ended 'pragmatic' inference. Our account contrasts with that of, say, Hobbs et al. (1987) who posit an underspecified and impoverished lexical semantic representation which is enriched through an open-ended process of abductive or deductive reasoning with world (or domain) knowledge. Our lexical knowledge representation language does not support general inference, although our lexical semantic representations are often 'richer' than those standardly assumed.

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 – we follow Pustejovsky (1989a,b,c) in assuming that the lexical semantic representation of nouns includes information about typical relations which the objects they denote enter into; in particular, the "qualia structure" for such nouns will contain (telic) information which allows direct access to the information that they are inhabited. A well-known example of metaphoric sense extension is that involving use of a word denoting an animal to refer to humans ("John is a pig", "John is a wombat" etc). Although the sense extension from animals into metaphorical senses denoting humans with some particular characteristic is apparently productive, the actual characteristics involved, and even whether the word can be applied to men or to women or both, cannot be predicted from knowledge of the animal sense. Thus, the properties ascribed to a person by "pig" are arguably no more than stereotypical associations with the animal, rather than central aspects of its meaning / qualia structure. In the case of "wombat" we would argue that the association of foolishness derives from the phonological form of the word, rather than beliefs about the animal. 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, rather than in terms of general purpose reasoning. As with the metonymic cases (Pustejovsky, 1989c; Briscoe et al., 1990), we believe that the notion of coercion during syntactic and semantic interpretation provides an account of when a metaphorical interpretation will be adopted, and we would like to characterise coercion in terms of possible mappings defined by lexical rules.

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 and metaphoric sense extensions considered above. 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, or indeed metaphorical, usage. Nevertheless, there are clearly derivational processes which do not affect syntactic class (e.g. "re-program", "un-reprogrammable") and of sense extensions which do; for example, countability of nouns changes depending on whether they are interpreted as types, substances or portions (e.g. "There was beer all over the table", "John drank a beer"). Not all derivational processes affect argument structure (e.g. "unkind"), whilst metonymic sense extensions (e.g. "John enjoyed the film", John finished the beer") can, at least given the analyses of Pustejovsky (1989c) and Briscoe et al. (1990). Finally, processes of conversion and derivation can be identical; for example, both "purchase" and "replace" have deverbal nominal forms "purchase" and "replacement", both nouns can denote the action involved and take appropriate complements ("Bill's purchase of his new car", "Bill's replacement of John with Mary"), and both can denote the result of the action ("Bill's purchases were many and varied" "Bill's replacement was young"). Traditionally, this latter resultative meaning would be described as metonymic and probably specialised and non-productive. We think that our definition of lexical rule will allow an account of both conversion and derivation as productive syntactic and semantic operations mapping between lexical entries. The difference between metaphorical and metonymic operations is a matter of the degree to which the interaction of the lexical rule with the basic entry determines or circumscribes the eventual interpretation.

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"). By representing such processes in terms of lexical rules mapping between entries, we hope to account for blocking in terms of syntactic and semantic identity with an entry defined without recourse to the relevant lexical rule. In addition, we hope to express sense extension processes, and indeed derivational ones, as fully productive processes which apply to finely specified subsets of the lexicon, defined in terms of both syntactic and semantic properties expressed in the type system underlying the organisation of the lexicon.

In Briscoe et al. (1990) we offered an account of logical metonymies such as that involved in the interpretation of "Bill enjoyed / regretted that paper", drawing on Pustejovsky (1989a), in which a unary syntactic rule is used to coerce the entity-denoting NP "that paper" into an event-denoting NP with an underspecified predicate. We argued that a default specification of this predicate is supplied by the qualia structure of the noun and that this is determined by the organisation of the lexicon as a default inheritance network. Specifically, "paper" in the relevant sense will inherit a telic role read′ and agentive role write′ and "enjoy" will by default select the telic role of an NP object, whilst "regret" will select the agentive role. We presented evidence based on corpus data that such default interpretations are appropriate in unmarked informationally-weak contexts, but that they are overridden in marked contexts in which such an interpretation would be clearly inappropriate. The corpus evidence suggests, firstly, that the default interpretation is appropriate with most such logical metonymies and, secondly, that where it is not, the context is sufficient to block it. By contrast, an account such as that of Hobbs et al. (1987) has difficulty explaining why the default interpretation is adopted in the *absence* of contextual information, unless the effects of lexical organisation are reconstructed in terms of weightings encoding preferences amongst inferences. Whilst this account of the division of labour between default but circumscribed linguistic processes and more open-ended inference remains attractive, the treatment of coercion in this case as a unary syntactic rule, or in Pustejovsky's (1989c) alternative account as a lexical rule, seems inadequate. The interpretation of an individual-denoting nominal or noun phrase as an event-denoting one may be more a matter of systematic vagueness than ambiguity between determinate senses. Pustejovsky (1989a) argues that in examples such as a), b) or c) "long" will have an interpretation in which it modifies the telic or agentive role of "book" because it is an event modifier.

a) John bought and read the long book.
b) John enjoyed the long book.
c) John bought, read and enjoyed the long book.

However, this coercion of "book" into, say, "long book to read" does not preclude either an event-denoting or individual-denoting interpretation of the complete NP, as a) and b) demonstrate (where presumably "read" and "bought" select the straightforward referential interpretation, whilst "enjoy" forces another round of coercion). Furthermore, an example like c) which involves a "crossed" interpretation in which "the long book" is simultaneously interpreted as individual- and event-denoting does not seem odd. However, in cases of genuine ambiguity, as opposed to vagueness, such readings are usually blocked (Zwicky & Sadock, 1975):

a) John likes landing planes and so does Bill.
b) ? Peter's purchase of hi fi took hours and was expensive.
c) ? John played with and then ate his lamb.
d) John ate and enjoyed the salmon.
e) Bill picked up and finished his beer.
f) John wrote but later regretted that paper.

Thus a) has two, not four, readings in which both John and Bill like watching planes landing or like landing them themselves, but not readings in which the interpretation of "landing planes" varies between conjuncts. Similarly, b) is odd because the first conjunct forces a deverbal event-denoting interpretation of "purchase" whilst the second strongly prefers a resultative reading and c) is odd because the first conjunct prefers the animal-denoting interpretation of "lamb" whilst the second selects the food-denoting one (although the overall preferred interpretation will probably involve treating "played with" to mean something like 'fiddle with or pick at (food)'). By contrast, d-f) all involve moving between individual- and event-denoting readings of the final NPs, but do not seem problematic.

These observations suggest to us that an adequate account of the coercion process in these cases will involve positing systematic vaguenesses in interpretation of NPs, perhaps along the lines of type ladder polymorphism (e.g. Partee & Rooth 1983) or in terms of lexical operations which apply to the predicates, like "enjoy" which introduce such logical metonymies (rather than to the NP objects of these predicates). In this paper, though we concentrate on ambiguities in interpretation which can be treated in terms of lexical rules which apply to noun entries. We illustrate our approach with reference mainly to the process of '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. We refer to this as 'grinding' because the context normally suggested is the "Universal Grinder" (see 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 established. Thus besides the animal/meat examples, trees used for wood ("beech") have a sense denoting the wood, and so forth. Before we describe this process in detail, we present the framework in which our account will be couched.

2 The Lexical Representation Language

Our lexical representation language is unification-based, allowing complex interconnections between syntactic and semantic information to be defined, and making a tight interface possible between the lexicon and a parser/interpreter. It supports a restricted range of operations; (default) unification, (default) inheritance and lexical rule application. It does not support arbitrary inference. The language is based on the use of typed feature structures similar to those described in Carpenter (1990). Feature structures must be well-formed with respect to types and particular features will only be appropriate to specified types and their subtypes. Types are hierarchically ordered; the association of *constraints* with types allows non-default inheritance. We augment this with a restricted concept of default inheritance (allowing only 'orthogonal' multiple inheritance (Touretzky 1986)); default inheritance is formalised in terms of default unification of feature structures ordered by an inheritance hierarchy. The type system constrains both default inheritance and lexical rule application. This representation language is described in detail in Copestake et al (1991); the following sections are an informal description illustrated with relevant examples.

2.1 The type system

The type hierarchy defines a partial ordering (notated $\sqsubseteq$) on the types and specifies which types are *consistent*. Only feature structures with mutually consistent types can be unified

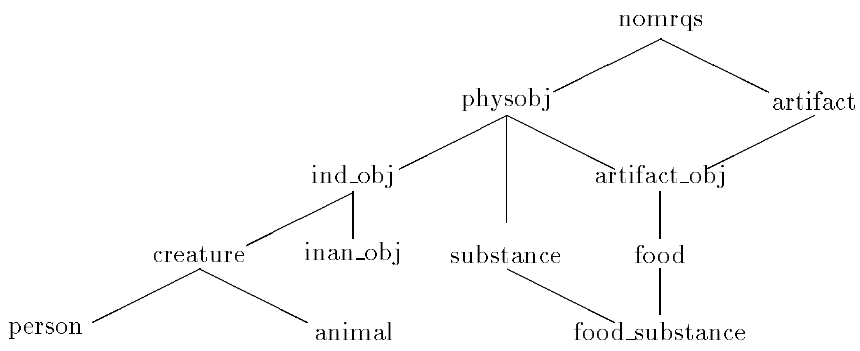

Figure 1: A fragment of a type hierarchy

— two types which are unordered in the hierarchy are assumed to be inconsistent unless the user explicitly specifies a common subtype. Every *consistent* set of types $S \subseteq \mathsf{TYPE}$ has a unique greatest lower bound or meet (notation $\sqcap S$). This condition allows feature structures to be typed deterministically — if two feature structures of types **a** and **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 1 **artifact** and **physobj** are consistent; $\mathbf{artifact} \sqcap \mathbf{physobj} = \mathbf{artifact\_obj}$.

Our system differs somewhat from that described by Carpenter (1990) in that we adopt a different notion of well-formedness of typed feature structures. In our system 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; a well formed feature structure may only contain appropriate features. 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. Constraints can be seen as extending the PATR-II notion of templates (eg. Shieber, 1986) in that the inheritance of constraints allows concise definitions of all feature structures, not just lexical entries; but in an untyped system, such as PATR-II, there is no restriction on the features that can occur in a feature structure.

For example the constraints associated with the types **artifact** and **physobj** might be:

$$\begin{bmatrix} \mathbf{artifact} \\ \textsc{telic} = \mathbf{formula} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{physobj} \\ \textsc{form} = \mathbf{physform} \\ \textsc{physical-state} = \mathbf{solid} \end{bmatrix}$$

Bold case indicates types; thus, for instance **formula** is a type and any feature structure of type **artifact** must have a feature structure of type **formula** as the value for its

TELIC (purpose) feature. **formula** is intended to represent a formula in predicate logic, it therefore has a complex constraint itself:

$$\begin{bmatrix}\textbf{formula}\\ \textsc{ind} = \textbf{entity}\\ \textsc{pred} = \textbf{logical-pred}\\ \textsc{args} = \textbf{arg-list}\end{bmatrix}$$

In contrast **solid** is an atomic type, it has no appropriate features and its constraint is simply the atomic feature structure [**solid**].

The constraint on **artifact_obj** will contain information inherited from both parents, thus:

$$\begin{bmatrix}\textbf{artifact\_obj}\\ \textsc{form} = \textbf{physform}\\ \textsc{physical-state} = \textbf{solid}\\ \textsc{telic} = \textbf{formula}\end{bmatrix}$$

Further examples of constraints and features which we will use in examples in this paper are:

$$\begin{bmatrix}\textbf{ind\_obj}\\ \textsc{form} = \begin{bmatrix}\textbf{physform}\\ \textsc{shape} = \textbf{individuated}\end{bmatrix}\end{bmatrix}$$

$$\begin{bmatrix}\textbf{creature}\\ \textsc{age} = \textbf{scalar}\\ \textsc{sex} = \textbf{gender}\end{bmatrix}$$

$$\begin{bmatrix}\textbf{animal}\\ \textsc{edible} = \textbf{boolean}\end{bmatrix}$$

$$\begin{bmatrix}\textbf{substance}\\ \textsc{form} = \begin{bmatrix}\textbf{physform}\\ \textsc{shape} = \textbf{unindividuated}\end{bmatrix}\end{bmatrix}$$

$$\begin{bmatrix}\textbf{food}\\ \textsc{telic} = \begin{bmatrix}\textbf{formula}\\ \textsc{pred} = \textbf{eat}\end{bmatrix}\end{bmatrix}$$

We shall also make use of the following types to define syntactic properties etc:

$$\textbf{lex-sign} \sqsubseteq \textbf{top} \quad \begin{bmatrix}\textbf{lex-sign}\\ \textsc{orth} = \textbf{string}\end{bmatrix}$$

$$\textbf{noun} \sqsubseteq \textbf{lex-sign} \quad \begin{bmatrix}\textbf{noun}\\ \textsc{syntax} = \begin{bmatrix}\textsc{count} = \textbf{boolean}\end{bmatrix}\\ \textsc{rqs} = \textbf{nomrqs}\end{bmatrix}$$

$$\textbf{count-noun} \sqsubseteq \textbf{lex-sign} \quad \begin{bmatrix}\textbf{noun}\\ \textsc{syntax} = \begin{bmatrix}\textsc{count} = +\end{bmatrix}\end{bmatrix}$$

$$\textbf{mass-noun} \sqsubseteq \textbf{lex-sign} \quad \begin{bmatrix} \textbf{noun} \\ \textsc{syntax} = \begin{bmatrix} \textsc{count} = - \end{bmatrix} \end{bmatrix}$$

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.

$$\begin{bmatrix} \textbf{count-noun} \\ \textsc{orth} = \text{“\textbf{haddock}”} \\ \textsc{syntax} = \begin{bmatrix} \textsc{count} = + \end{bmatrix} \\ \textsc{rqs} = \begin{bmatrix} \textbf{animal} \\ \textsc{sex} = \textbf{gender} \\ \textsc{age} = \textbf{scalar} \\ \textsc{edible} = \textbf{boolean} \\ \textsc{physical-state} = \textbf{solid} \\ \textsc{form} = \begin{bmatrix} \textbf{physform} \\ \textsc{shape} = \textbf{individuated} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Given the type system introduced above, a lexical entry, such as:

```
haddock 1 count-noun
       <rqs> = animal.
```

would be expanded out into such a feature structure[1].

2.2 Default inheritance

To allow default inheritance we introduce the concept of *psort*; a feature structure from which another feature structure inherits information, by default. The hierarchical ordering on psorts (which must be consistent with the type hierarchy) provides an order on defaults. Default inheritance is implemented by a version of default unification. Only orthogonal multiple inheritance (Touretzky 1986) is allowed; information inherited from multiple parents must not be contradictory. (A default inheritance hierarchy which connects semantic parts of lexical entries can be derived semi-automatically from taxonomies extracted from conventional dictionaries, see Copestake, 1990a). We refer to this particular case of the psort hierarchy as an IS_A hierarchy. Values of features can be associated either manually or semi-automatically with psorts in the IS_A hierarchy; the more specific word senses then inherit them, by default. (Defaults may also be useful in the representation of syntactic information in the lexicon (e.g. Flickinger, 1987).)

Since the type system constrains the psort system it also constrains multiple default inheritance. If the value of the FOOD-TEMPERATURE feature for “drink 2 (1)” is **low** then this information would be inherited by the entry for “beer” which is below “drink 2 (1)” in the IS_A hierarchy. However inherited information may be overridden by associating other values with psorts lower in the hierarchy; for example although “tea” is under “drink 2 (1)” in the hierarchy, its FOOD-TEMPERATURE can be specified to be **high** rather than **low**.

[1] The actual type system being employed is considerably more complex, since only the relevant features are being shown in these examples.

Types and features thus provide an organisation on the information which is necessary for interaction with lexical and syntactic rules. The IS_A hierarchy is motivated by defining its semantics in terms of the real world entities corresponding to the word senses and demonstrating that default inheritance of attributes in the lexicon correlates with default reasoning about properties of the entities. Copestake (1990a) outlines a preliminary attempt to formalise the relationship between this aspect of lexical semantics and world knowledge.

2.3 Lexical rules

A lexical rule is a feature structure of type **lexical-rule**. The expanded constraint for the type is:

$$\begin{bmatrix} \textbf{lexical\_rule} \\ 0 = \textbf{lex\_sign} \\ 1 = \textbf{lex\_sign} \end{bmatrix}$$

thus all lexical rules have to have the features 0 and 1 which must both have values which are of type **lex_sign**.

New lexical signs may be generated by unifying a copy of the lexical entry with the feature structure at the end of the path `<1>` in a copy of the lexical rule — the feature structure at the end of the path `<0>` is then the new lexical sign. Lexical rules are indexed by the type of their "input" and "output" feature structures, so they will only be applied to entries of the appropriate type and will only create well-typed entries.

A number of productive or quasi-productive phenomena, such as deverbal nominalisation, 'grinding', and so forth, can be represented as lexical rules which generate further lexical entries. A general type for grinding lexical rules could be specified in our system as follows:

$$\textbf{grinding} \sqsubseteq \textbf{lexical\_rule} \quad \begin{bmatrix} \textbf{grinding} \\ 1 = \begin{bmatrix} \textbf{count-noun} \\ \text{ORTH} = \boxed{1} \\ \text{RQS} = \textbf{ind\_obj} \end{bmatrix} \\ 0 = \begin{bmatrix} \textbf{mass-noun} \\ \text{ORTH} = \boxed{1} \\ \text{RQS} = \textbf{substance} \end{bmatrix} \end{bmatrix}$$

The effect of the lexical rule is to transform a count noun with the 'relativised qualia structure' (RQS, Calzolari, 1991) properties appropriate to an individuated physical object **ind_obj** into a mass noun with properties appropriate for a substance **substance**. Thus the core component of grinding is a linguistic, syntactic operation which affects syntactic realisation, such as the ability to appear without a determiner, correlated with an abstract and underspecified semantic operation. We would claim that specific predicational and syntactic contexts will result in coercion (application of the lexical rule) and that this much, at least, of the 'grinding' family of sense extensions must be seen as a non-default and essentially linguistic process.

We specialise the grinding rule to allow for cases such as the animal/meat regular sense 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 parameterised by inheritance in the type system. As our theory of lexical organisation allows us to make fine distinctions between classes of lexemes, in terms of both syntactic and semantic properties encoded in the type system, we expect that many processes which have been characterised as partially productive or semantic specialisations of productive processes will be characterisable as fully productive rules of sense extension applying to smaller semantically coherent subsets of the lexicon.

For example given the type hierarchy shown in Figure 1 we can give rules which inherit information from **grinding** such as animal_grinding:

$$\text{animal\_grinding} \quad \left[\begin{array}{l} \textbf{grinding} \\ 1 \;=\; \left[\begin{array}{l} \textsc{rqs} \;=\; \left[\begin{array}{l} \textbf{animal} \\ \textsc{edible} \;=\; + \end{array}\right] \end{array}\right] \\ 0 \;=\; \left[\textsc{rqs} \;=\; \textbf{food\_substance}\right] \end{array}\right]$$

Thus given the lexical entry for "haddock" shown above we can apply the lexical rule to generate a sense meaning "haddock-flesh" (partially represented as):

$$\left[\begin{array}{l} \textbf{mass-noun} \\ \textsc{orth} \;=\; \textbf{haddock} \\ \textsc{syntax} \;=\; \left[\textsc{count} \;=\; -\right] \\ \textsc{rqs} \;=\; \left[\begin{array}{l} \textbf{food\_substance} \\ \textsc{telic} \;=\; \left[\begin{array}{l} \textbf{formula} \\ \textsc{pred} \;=\; \textbf{eat} \end{array}\right] \end{array}\right] \end{array}\right]$$

(where the specification of the value **eat** for the telic role arises from the constraint on the type **food_substance**, inherited from **food**, and the type mass-noun arises from **grinding**.) It would not be possible to apply this lexical rule to "book" and get a sense denoting "book-flesh" because "book" has the type **inanimate_obj** which is incompatible with **animal**. It would still be possible to apply the general lexical rule for grinding and to get a mass use of "book" but the denotation of the mass sense would be underspecified. We would expect the context to provide a more specific interpretation of the mass sense in such a (less-conventionalised) case.

Furthermore, our approach provides a natural mechanism for dealing with semantic specialisation or restriction. We can represent lexicalised items as inheriting information by default from a psort which is the result of applying an appropriate lexical rule to the base form. Such items may have specific information associated with them. In cases where the lexical rule predicts the extended sense exactly, the specific information will duplicate information already present. If the rule is correct, but incomplete, the specific information will augment the inherited information. If it is partially incorrect, the more specific information will override that inherited from the result of lexical rule application. In an untyped system this representation would not constrain the structures associated with lexicalised derived forms, since the entire feature structure output by the lexical rule might be overridden. However default inheritance is constrained by the type system so that information may only be inherited from a structure of the same or higher type, and thus this treatment predicts that a derived form can never have a type which is incompatible with that determined by the lexical rule.

We can illustrate the manner in which this type of semi-productivity might be dealt with if we assume that we are attempting to construct a lexicon semi-automatically from a conventional dictionary (e.g. Copestake, 1990a). If the result of applying a lexical rule to a sense is notated as sense+rule-name (eg lamb_1+animal_grinding) then the representation of the sense lamb (2) ("the meat", from the *Longman Dictionary of Contemporary English* LDOCE) might be:

```
lamb 2 < lamb_1+animal_grinding.
```

In this case no extra information need be added. In contrast the entry for lamb (3) ("a young gentle person" LDOCE) might augment the information inherited from the lexical rule:

```
lamb 3 < lamb_1+animal_metaphor
< rqs : age > = low.
```

In the case of "haddock", where no LDOCE entry is found, the structure derived from the lexical rule alone would be used.

We can regard morphological rules as a particular type of lexical rule where the orthography of the output is not equal to that of the input (we assume that the regular spelling changes involved in affixation will be dealt with by a separate system, e.g. Cahill, 1990). In this case we could represent irregular forms as having an orthographic form which overrides that produced by rule application. In principle, multiple lexical rules may be applied in sequence. For example the resultative senses of "replacement" and "purchase" mentioned in the introduction would be the result of applying a metonymic sense extension rule to the result of the nominalisation process. All outputs of lexical rules must be potentially valid lexical entries. In the case of conversion or zero-derivational processes we wish to restrict the set of lexical rules so that application may not be circular — that is if there is a lexical rule which could generate the set of feature structures F2 from the set F1, no other lexical rule or sequence of lexical rules may be specified which could generate any member of F1, or a feature structure subsuming any member of F1, starting from any member of the set F2, since lexical rule application would not then terminate. (We can check for such potential circularities relatively efficiently by looking at the type of the feature structure that a lexical rule generates rather than the entire feature structure.) However, this condition is overrestrictive in general because some types of derivational rule can apply to their own output iteratively ("meta-meta-theory", "anti-anti-missile", "great-great-grandmother", "re-re-program"). This observation suggests that we need to distinguish types of lexical rule, such as at least derivational rules and processes of conversion, and associate slightly different constraints with them.

3 Grinding

In this section we justify our treatment of grinding as a lexical rule and show why relatively complex semantic information is needed to adequately account for this sense extension.

The first point to consider is that grinding processes appear to be genuinely productive. Thus we find:

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

in the Lancaster-Bergen/Oslo (LOB) corpus. We therefore cannot assume that the ground senses are necessarily lexicalised, even in the relatively conventionalised uses to mean meat, fur etc.

One approach which allows for this productivity is to treat all nouns as being 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, in effect, taken as denoting both animals and meat and so on (see the "p-theory" in Pelletier and Schubert 1986 and also Copestake 1990b). In contexts where one interpretation is forced ("a piece of lamb" vs "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 seems to predict that NPs such as "the lamb" are vague rather than ambiguous between count and mass readings. Thus the peculiarity of sentences such as:

> ? John fed and carved the lamb.

is not accounted for (see also the introduction). It is perhaps significant that in most dictionaries the mass sense is specified as well as a count sense for the conventionalised grinding examples we have been considering. Lexicographers are sometimes aware of the regularity of the extension (Atkins 1990) but have no way of representing this in a conventional dictionary. We thus 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. (There are cases where it is 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", however in the grinding examples there is a clear change in meaning.)

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. By having an inheritance ordering on lexical rules we can express the conventionalised processes that apply to semantically specified parts of the lexicon and 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.

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. (We can formally specify this relationship between the ground sense G and the base sense B as

$$\forall x, t[G(x,t) \Rightarrow \exists y, t'[{}^{\star}B(y,t') \wedge t' \leq t \wedge x \sqsubseteq_o y]]$$

using the formalisation developed in Copestake (1990a) following Krifka (1987) where nominal predicates are taken as being true of quantities of matter at some time index, where ${}^{\star}B$ denotes a potentially plural entity, and where $\sqsubseteq_o$ represents a relationship of material constituency. In the lexicon we actually use a feature ORIGIN which can be taken as an abbreviation for the relationship specified above.)

A good theory of sense extension should give some treatment of blocking, which appears to occur with some cases of regular sense extension in a way that seems similar to derivational morphology. For example the use of "pig" to denote the meat seems to be blocked by the existence of "pork" — "pig" can be used in the extended sense but such a use is marked, suggesting for example that the meat is distinctly inferior. To account for this we need to be able to recognise that "pork" is equivalent to the sense obtained from "pig" using the lexical rule. Although there are many problems with this (what do we mean by equivalence, why does this apparently not apply to metaphorical sense extension) in order to do it at all we clearly need a rich representation which indicates information such as "origin".

Bauer (1983) distinguishes two types of non-productivity (which he refers to as *established* senses) – lexicalisation and institutionalisation. Lexicalisation is defined as irregular and unpredictable modification of some or all of the semantic, syntactic or phonological properties of a derived form. Institutionalisation, by contrast, involves restriction, rather than modification along one of these dimensions; thus "telephone box" is institutionalised to mean telephone kiosk unambiguously, although the liberal rules of noun compounding predict other possibilities. In our approach, we can treat institutionalisation as a form of blocking in which forms such as "telephone box" would have separate entries equivalent to one productive meaning predicted by a putative lexical rule of compounding. This would predict a strong preference for this interpretation (except in a marked context). Bauer (1983:58) points out that some treatment along these lines will be required since the other meanings are not completely ruled out, and therefore simply listing them as independent entries will be inadequate. The more specific rules of grinding (as opposed to the most general rule) are instances of productive institutionalisation within sub-classes in that the specific interpretations they introduce can be overridden in marked contexts.

Similarly, lexicalisation is usually a partial process which affects one aspect of a derived lexical entry, whilst the rest remains productive. Bauer gives the example of "disbelieve" which can be productively derived through a lexical rule which prefixes "dis+" to the verb "believe" with a predictable change of meaning, except that "disbelieve" does not inherit the syntactic properties of "believe" because it cannot take sentential or infinitival complements. The productive aspects of the relation between the two verbs can be expressed by a lexical rule for "dis+" prefixation, whilst the non-productive aspects can be captured naturally in this framework by positing an independent entry for "disbelieve" which overrides some of the information provided by the lexical rule.

We think of lexical rules as defining the limits of coercion amongst lexemes and argue that lexical rule application, or selection of the derived entry (which is equivalent in many cases), will be forced when the type of the basic entry is incompatible with the syntactic or predicational context in which the lexeme occurs. Consider the following example, taken from the Lancaster-Bergen/Oslo (LOB) corpus:

More than 1,000 union men and their families arrived to play bowls, eat barbecued chicken and row on his fish-infested lake.

The application of a grinding lexical rule is triggered by a combination of syntactic and semantic effects arising from the context (for example the predicate "eat" takes an object denoting food in preference to an animal, the bare NP "mole" in the earlier example must have a negative value for the syntactic feature count). By default, the most specific lexical rule applicable will be used, in this case "animal_grinding" as opposed to the general grinding rule and so the interpretation of "chicken" as "chicken-flesh" and "mole" as "mole-flesh" is possible, by default. More open-ended (non-lexical) reasoning might cause the default interpretation of the mass sense to be overridden in some marked informationally-rich contexts. For example:

John bit into the lamb.
It kicked and struggled.

We assume a similar account of the overriding of default interpretations with these types of example as we offered in the case of logical metonymies (Briscoe et al., 1990).

4 Conclusion

We have argued that our notion of lexical rule can capture the productive linguistic element of derivational morphological processes and metonymic and metaphorical sense extensions. In order to do this adequately we need rich lexical semantic information but we do not need to resort to general deductive or abductive inference on unconstrained world knowledge. By using semantic information to structure the lexicon, by means of types and inheritance, we can represent relationships between lexical rules and view them as essentially fully productive over defined subparts of the lexicon, while providing an initial account of blocking and lexicalisation. However the work described here is at a preliminary stage. We need to provide detailed accounts of a range of derivation and conversion processes to see how adequately we can represent them as lexical rules, while structuring the lexicon and type system appropriately to constrain their operation.

Acknowledgements

This work was supported by Esprit BRA-3030, ACQUILEX 'The Acquisition of lexical knowledge for Natural Language Processing systems'.

References

Atkins B(1990) *Lexical Rules: a starter pack,* Ms. OUP

Bauer L(1983) *English Word-formation,* CUP

Briscoe E J, Copestake A A and Boguraev B K(1990) 'Enjoy the paper: Lexical semantics via lexicology', *Proceedings of the 13th Coling,* Helsinki, pp.42–47

Cahill L(1990) 'Syllable based morphology', *Proceedings of the 13th Coling,* Helsinki, pp.48–54

Calzolari N(1991) *Representation of semantic information in a lexical knowledge base,* ACQUILEX WP no ??, 1st ACQUILEX workshop, Cambridge

Carpenter R(1990) ‘Typed feature structures: Inheritance, (In)equality and Extensionality’, *Proceedings of the Inheritance in Natural Language Processing,* Tilburg, pp.9–18

Copestake A A(1990a) *An approach to building the hierarchical element of a lexical knowledge base from a machine readable dictionary,* Workshop on Inheritance in NLP, Tilburg

Copestake, A.A.(1990b) *Some notes on Mass Terms and Plurals,* Technical Report 190, Computer Laboratory, University of Cambridge

Copestake A A, de Paiva V C V, Sanfilippo A and Briscoe E J(1991) *Functionality of the LKB,* Ms. Computer Laboratory, University of Cambridge

Flickinger D(1987) *Lexical rules in the hierarchical lexicon,* PhD dissertation, Stanford University

Hobbs J, Croft W, Davies T, Edwards D and Law K(1987) ‘Commonsense metaphysics and lexical semantics’, *Computational Linguistics, vol.13,* 241–250

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

Partee B and Rooth M(1983) ‘Generalized Conjunction and Type Ambiguity’ in Bauerle R, Schwarze C and von Stechow A (eds.), *Meaning, Use and Interpretation of Language,* de Gruyter, pp.361–368

Pelletier, F.J., and Schubert, L. K.(1986, forthcoming) ‘Mass Expressions.’ in Gabbay and Guenthner (eds.), *Handbook of Philosophical Logic, Vol 4,* Reidel, Dordrecht

Pustejovsky J(1989a) *The Generative Lexicon,* Ms. Brandeis University

Pustejovsky J(1989b) ‘Current issues in computational lexical semantics’, *Proceedings of the 4th European ACL,* Manchester, pp.xvii–xxv

Pustejovsky J(1989c) ‘Type coercion and selection’, *Proceedings of the West Coast Conference on Formal Linguistics,* Vancouver

Shieber S(1986) *An Introduction to Unification-based Approaches to Grammar,* University of Chicago Press, Chicago

Touretzky D F(1986) *The mathematics of inheritance systems,* Morgan Kaufmann, Los Altos

Zwicky A M and Sadock J M(1975) ‘Ambiguity tests and how to fail them’ in Kimball J P (eds.), *Syntax and Semantics IV,* Academic Press, New York, pp.1–36