Inheriting Polysemy

Adam Kilgarriff

1 Introduction

This paper builds on the title and theme of Apresjan's 1974 paper, *Regular Polysemy*. Apresjan was concerned merely to define the phenomenon and identify where it occurred. Here, we shall explore how it can be exploited.

Regular polysemy occurs where two or more words each have two senses, and all the words exhibit the same relationship between the two senses. The phenomenon is also called 'sense extension' (Copestake & Briscoe, 1991), 'semantic transfer rules' (Leech, 1981), 'lexical implication rules' (Ostler & Atkins, 1991), or simply 'lexical rules'. An example, taken direct from a dictionary (Longmans Dictionary of Contemporary English (1987 Edition), hereafter LDOCE) is:

gin (a glass of) a colourless strong alcoholic drink ...

martini (a glass of) an alcoholic drink ...[1]

In each case, there are two senses referred to, one with the 'bracketed optional part' included in the definition and the other with it omitted, and the relation between the two is the same in both cases.

Recent work on lexical description has stressed the need for the structure of a lexical knowledge base (LKB) to reflect the structure of the lexicon (Atkins & Levin, 1991) and for the LKB to incorporate productive rules, so the rulebound ways in which words may be used are captured without the lexicon needing to list all options for all words (Boguraev & Levin, 1990). These arguments suggest that generalisations regarding regular polysemy should be expressed in the LKB, and that the formalism in which the LKB is written should be such that, once the generalisation is stated, the specific cases follow as consequences of the inference rules of the formalism.

As 'lexicalism' —the doctrine that the bulk of the information about the behaviour of words should be located in the lexicon— has become popular in theoretical linguistics, so formalisms for expressing lexical information have been developed. The syntax, semantics and morphology of most words is shared with that of many

[1] As the LDOCE entry for *glass* notes, a receptacle need not be made of glass to be a glass.

others, so the first desideratum for any such formalism is to provide a mechanism for stating information just once, in such a way that it is defined for large numbers of words. Inheritance networks serve this purpose. If words are arranged into a taxonomy or some other form of network, then a fact which applies to a class of words can be stated at a nonterminal node in the network and inherited by the words to which it applies. Inheritance can be from a parent or from a parent's parent (and so on). Work in knowledge representation has addressed questions of different kinds of network, and the kinds of machinery needed to retrieve inherited information, in detail (see, e.g., introduction and various papers in Brachman and Levesque (1985)).

The next requirement is that exceptions and subregularities can be expressed. It must be possible to describe concisely the situation where a word or class of words are members of some superclass, and share the regular characteristics of the superclass in most respects, but have different values for some feature or cluster of features. Several lexical representation formalisms addressing these desiderata have been proposed (DATR: Evans & Gazdar, 1989a, 1989b, 1990; Russell *et al.* 1991, Copestake 1991). While the generalisations to be formalised are better understood for morphology and syntax, the theoretical gains, of capturing generalisations and eliminating redundancy, and the practical benefits, in terms of lexicon acquisition and maintenance, apply also to regular polysemy. The work described here will take the DATR formalism and use it to represent a collection of facts and generalisations regarding polysemy.

Patterns of regular polysemy will be called 'alternations' or 'sense alternations', and the sense that results from excluding the bracketed part, or which is listed first in the dictionary, or which is the only one listed, will be deemed the 'primary' sense, with others 'secondary'. Words have various 'senses', of which only the more frequently occurring and less predictable ones (see below) are likely to be specified in dictionaries.

In the fragment presented below, information about both the word and its denotation is accessed through a node in an inheritance network associated with the word. Thus a query regarding the syntax of the word *ash*, and a query asking what type of thing an ash tree is, will both be made at the same node. It might be argued that this is to confuse two different kinds of information. However, there are many kinds of information about denotations which have consequences for words, for example that the kind of thing a word denotes determines (at least in the default case) the alternations it participates in, so there is much to be gained from holding the two types of information together. The matter receives a fuller discussion in Kilgarriff (1992). Here, we proceed on the basis that linguistic and encyclopedic information should inhabit the same representation scheme.

After some brief comments on related work and on the approach adopted towards dictionaries, DATR will be described. The paper uses DATR but neither presupposes a knowledge of it, nor gives a formal description. Then evidence regarding regular polysemy will be introduced, in stages, with the proposed DATR account of the evidence worked through at each stage.

1.1 Related work

Polysemy has been the subject of remarkably little research. Linguists have found it insufficiently regular to succumb to their theoretical accounts, lexicographers have been concerned with instances rather than theory, and NLP has, until recently, overlooked the homonymy/polysemy distinction and ignored the relations between polysemous senses. A full literature review is to be found in Kilgarriff (1992).

Recent work which does aim to formalise phenomena closely related to regular polysemy includes Pustejovsky (1991), Briscoe, Copestake, and Boguraev (1990), Copestake and Briscoe (1991) and Kilgarriff and Gazdar (ming). The first three papers are concerned with metonymy rather than polysemy, but the overlap is considerable. Crystal (1987) defines metonymy as "The use of an attribute in place of the whole, e. g. *the stage* (the theatrical profession), *the bench* (the judiciary)" (p 70); from a lexicographic perspective, both examples are straightforward cases of polysemy. The first two papers focus on the mechanisms for parsing and semantic composition rather than inheritance. The more recent work by Copestake concentrates on inheritance, and is also concerned with metonymies more closely related to the alternations described here. The work is a similar enterprise in a different formalism. The last is a companion paper to this, exploring similar regularities in another domain and exploring the implications for linguistics.

1.2 Dictionary and LKB

At the most general level, a monolingual dictionary and an LKB share the same goal: to describe the lexical resources of the language. If a fact is in the dictionary, the lexicographers have considered it worth stating there, and that is a *prima facie* reason for it also needing stating in the LKB. Work which pursues the goal of automatically extracting information from dictionaries for 'populating' LKBs is reported in Byrd et al. (1987) and Boguraev and Briscoe (1989) amongst other places. A further matter of interest is the manner of representation of the facts in the dictionary. It too can be used as a clue to how the facts might fit into the LKB. The alternations studied are all represented in LDOCE using the 'bracketed optional part' mechanism, and this device indicates a regular or predictable relationship between the two types of use of a word (Kilgarriff, 1990). Working from a dictionary provides not only a supply of facts to be formalised, but also some indications, from the way they are presented, of what types of facts they are and what place they have in the overall lexical description (see also Neff and Boguraev (1989)). The dictionary is not, of course, without inconsistencies. Although liquid/glass-of alternations are generally indicated using the 'bracketed optional part', the glass-of sense of whisky is, for no apparent reason, presented as a distinct, numbered sense of the word. While a formalisation may be based on a dictionary, care must be taken not to reproduce its inconsistencies.

More interesting than its inconsistencies are its systematic omissions. LDOCE does not note both options for all drinks. For *bourbon*, for example, no glass-of sense is noted. Types of usage of words are systematically omitted where they are predictable, and where they occur only rarely. For a combination of the two reasons the *bourbon* sense is omitted. Here we see a contrast between LKB and dictionary. A human dictionary-user (with the level of competence in English assumed by the LDOCE lexicographers) knows enough about the regular patterns of English not to need to have the glass-of sense always specified. They can infer its availability from the facts that bourbon is a drink, and that other, more familiar drink-words display both 'liquid' and glass-of senses. They have at their disposal a general principle to the effect that words from the same semantic field that have been found to behave alike to date, will in all likelihood behave alike in novel situations. For an LKB all such generalisations and principles need to be built into the system in such a way that it follows as an inference that the alternation applies to *bourbon* and (possibly) to all drinks words, no matter how rare the glass-of sense may be. In the dictionary the glass-of sense of *bourbon* is not stated because it is predictable (and of low frequency). In the LKB it will not be explicitly stated, again, because it is predictable — but it will be there, by inference, just as it is there amongst the lexical resources of the language.

2 Overview of DATR[2]

Evans and Gazdar (1989a,b) presented the basic features of DATR. Here we briefly review those features: more detailed discussion accompanies the formalisation developed below. DATR is a declarative network representation language with two principal mechanisms: orthogonal multiple inheritance and nonmonotonic definition by default. The primary unit of a DATR network description is called a **node** and consists of a set of *path/definition* pairs where *path* is an ordered sequence of arbitrary atoms (enclosed in angle brackets), and *definition* is either a value, an inheritance specification or a sequence of definitions. **Nodes** are syntactically distinct from other atoms: they start with a capital letter. The primary operation on a DATR description is the evaluation of a query, namely the determination of a value associated with a given *path* at a given **node**. Such a value is either (a) defined directly for *path* at **node** or (b) obtained via an inheritance specification for *path* at **node** or (c) determined from the definition for the longest leading subpath of *path* defined at **node**, when *path* itself is not defined at **node**.

[2]This section borrows heavily from Cahill (1990)

Inheritance specifications provide a new node, new path or both to seek a value from. The simplest form of inheritance, called 'local' inheritance, just changes the node and/or path specification in the current context. To specify that <path1> at Node1 inherits locally, we use one of the following.

```
Node1:<path1>   == Node2.
```

specifies that we inherit the value from <path1> at Node2.

```
Node1:<path1>   == <path2>.
```

specifies that we inherit the value from <path2> at Node1.

```
Node1:<path1>   == Node2:<path2>.
```

specifies that we inherit the value from <path2> at Node2.

When a requested path is not defined at a node, the longest subpath (starting from the left) is used to provide a definition, with all the paths (if any) in the definition specification extended by the extra requested atoms. Thus if paths <a b c> and <a b c d> are not defined at Node1, a definition such as:

```
Node1:<a b>     == Node2: <x>.
```

implicitly defines both the following:

```
Node1:<a b c>   == Node2:<x c>.
Node1:<a b c d> == Node2:<x c d>.
```

This 'definition by default' (in the absence of any more specific path definition) gives DATR its nonmonotonic character: add a definition to a node and some of the theorems which were previously valid, but derived by this default mechanism, may cease to hold. The most common form of definition in the DATR below can now be explained:

```
Node1:<>        == Node2.
```

specifies that Node1 inherits from Node2 for all paths where a leading subpath of the query is not matched at Node1, and thus equates to the only kind of inheritance there is in a simple inheritance network.

DATR has certain desirable formal and computational properties. It is a formal language with a declarative semantics. Retrieving values for queries involves no search. The problems of clashing inheritance often associated with nonmonotonic multiple inheritance are avoided, yet the kinds of generalisation most often associated with the lexicon can be simply stated. For fuller details, see references.

DATR has to date been used as a formalism for expressing syntactic, morphological, phonological and a limited amount of semantic lexical information (Evans & Gazdar, 1990; Cahill, 1990; Cahill & Evans, 1990; Gibbon, 1990). Polysemy has been addressed only briefly, in Cahill and Evans (1990), and that account makes no mention of the generalisations to be made regarding polysemy.

3 Trees, wood, fruit: a DATR fragment

The data we shall consider will concern trees, wood and fruit. Firstly, consider the following definitions, from LDOCE.

> **ash** (the hard wood of) a forest tree ...
>
> **beech** (the wood of) a large forest tree ...

The bracketed optional part mechanism, combined with the near-identical form of words within the brackets, suggests an alternation, and indeed the tree/wood alternation applies to most if not all trees. In the basic taxonomy of the domain `Ash` and `Beech` inherit from `TREE`, which in turn inherits from `Plant` which, in the fragment offered here, inherits directly from `Entity`.

```
PLANT: <>        == ENTITY.
TREE: <>         == PLANT.
Ash: <>          == TREE.
Beech: <>        == TREE.
```

This is shown as a taxonomy in Fig. 1. The motivation for the higher levels of the taxonomy is theory-internal: what structure permits the most elegant treatment of the generalisations in the domain? At the lower levels, this consideration applies alongside dictionary genus terms and folk taxonomy (see papers by Rosch and Berlin in Rosch and Lloyd (1978).