POLYSEMOUS RELATIONS*

Adam Kilgarriff & Gerald Gazdar

Cognitive & Computing Sciences

University of Sussex

January 1993

1. Introduction

In the section of Lyons's *Semantics* that deals with the distinction between homonymy and polysemy, he notes that a major criterion "is unrelatedness vs. relatedness of meaning .. indeed, it is arguable that it is the only synchronically relevant consideration" (1977, 551). However, he goes on to argue that all attempts "to explicate the notion of relatedness of meaning in terms of a componential analysis of the senses of lexemes .. have so far failed" (1977, 552-3). Although Lyons is sympathetic to a treatment of the lexicon that seeks to maximise polysemy at the expense of homonymy, he does not himself go on in that book to reconstruct the notion of relatedness of meaning that such a treatment requires.

Discussing polysemy some nine years earlier, Lyons was a little more explicit about what might be required: "Various .. types of ''extension' or ''transference' of meaning were recognized by the Greek grammarians, and have passed into traditional works on rhetoric, logic, and semantics. Meanings that are more or less closely ''related' in accordance with such principles are not traditionally regarded as being sufficiently diferent to justify the recognition of distinct words" (1968, 406). Metaphorical extension is the only kind of extension explicitly discussed in connection with polysemy in *Introduction to Theoretical Linguistics* although it is clear from the quotation just given that it was not the only one that Lyons had in mind. It is surprising, therefore, to find the following passage in a later work: "it is metaphorical extension .. that is at issue when one refers to the related meanings of polysemous lexemes. There are, of course, other kinds of relatedness of meaning, which are irrelevant in this connection" (1981, 47).

The opening lines of the entry for *silk* in Flexner (1987, 1780) read as follows:

> *silk* (silk), *n.* 1. the soft lustrous fiber obtained as a filament from the cocoon of the silkworm. 2. thread made from this fiber. 3. cloth made from this fiber. 4. a garment of this cloth.

If the notion of polysemy is to earn its keep, then it must surely be applicable to such a set of senses. And yet the relation between a meaning denoting a fibre and a meaning denoting a thread made from that fibre, or that between a meaning denoting a cloth and a meaning denoting a garment made from that cloth, is surely that of metonymy rather than metaphorical extension. Our concern in this paper, however, is not to explore the range of polysemous relations that dictionaries attest to, but rather to attend to their systematic and partial regularity. If a relation holds for *silk* then it may well also hold for *cotton*. And if a relation holds for *silk*, *cotton* and *wool*, then it probably also holds for less familar words like *guanaco*.

Our topic is thus what Apresjan calls "regular polysemy". He distinguishes cases of polysemy where the same relationship holds between the senses for two or more polysemous words from those where the relationship is particular to a single word. He points to a similarity between relations of regular polysemy and those of derivational morphology and proceeds to catalogue the regular polysemous relations of Russian. But he does not explore the formal structure of the regularities, nor does he address their exception-prone character. And he does not consider how such relations might be called into service as part of an account of lexical structure. Apresjan's phrase, "regular polysemy", is potentially misleading in a couple of respects. Firstly, by a standard Gricean inference, use of the phrase implies that there might be a category of "irregular polysemy". However, it seems to us that, once we have a fully developed theory of subregularity, it is unlikely that a distinction between "irregular polysemy" and "homonymy" would serve any purpose in a synchronic description of the lexicon. Secondly, the phrase fails to convey the problematic subregular character of the relations involved. Our thesis is that "regular polysemy", while often less regular than lexical syntax or inflectional morphology, is, like them, subregular and is appropriately described using the same formal machinery. Arguably, polysemy is simply null derivation.

Until very recently, and with odd exceptions, polysemous relations have received little attention in the literature on the lexicon. We suspect that this neglect has a lot to do with the fact that linguists

have not had plausible machinery for dealing with subregular phenomena. That situation has changed over the last few years in the wake of the development of a number of lexical description languages by computational linguists and their application to, inter alia, the representation of polysemy (see Kilgarriff (1992) and Copestake (1993) for full discussion and references to the literature). In this paper we will use the lexical representation language DATR to represent polysemous relations such as those evidenced in the extract from the *Random House Dictionary* cited above. In Section 2, we provide an informal introduction to the DATR language and, in Section 3, we go on to define a lexicon fragment that illustrates how DATR can be used to express the kind of subregular generalizations that are pervasive over lexical senses.

2. Lexical representation

Evans & Gazdar (1989a, 1989b) give formal presentations of a semantics, and a theory of inference, for DATR, a lexical knowledge representation language. The goal of the DATR work was to define (and implement) a simple language that (i) has the necessary expressive power to encode the lexical entries presupposed by contemporary work in the unification grammar tradition, (ii) can express all the evident generalizations about the information implicit in those entries, (iii) embodies an explicit theory of inference, (iv) is computationally tractable, and (v) has an explicit declarative semantics. DATR defines networks allowing multiple default inheritance of information through links typed by attribute paths. This typing provides the basis for a "most specific path wins" default inheritance principle. The language is functional, that is, it defines a mapping which assigns unique values to node attribute-path pairs. Recovery of these values is deterministic -- no search is involved.

In addition to punctuation, DATR contains two kinds of primitive object: nodes, which, by convention, are always marked with an initial capital letter, and atoms, which appear in lower case. When atoms appear between angle brackets they are referred to as attributes, and sequences of attributes are known as paths. Atoms that appear elsewhere are called values. DATR theories (in the logician's sense of theory -- a set of axioms from which theorems may be derived) consist of a set of

equations.[1] Every DATR equation has a pair of a node and a path as its left-hand side (LHS). The simplest kind of DATR equation simply has zero or more values on the RHS.

Node:Path == Values.

Here are some examples:

ENTITY:<collocates> == .

FIBRE:<genus> == fibre.

Jersey:<alt garment collocates> == football.

Felt:<made-of> == matted fibre.

The first example says, unsurprisingly, that the set of collocates associated with the **ENTITY** node is empty; the second that the **genus** attribute for the **FIBRE** node has the value **fibre**; the third that the **collocates** list for the **alt**ernant **garment** sense of the **Jersey** lexeme consists of **football**; and the fourth that the **made-of** attribute for the lexeme **Felt** equates to the two element value sequence **matted fibre**.[2] (Sequences of) values are the principal ‘‘results’ of a DATR theory: the typical operation involves proving a theorem that will provide the value sequence associated with a given node/path pair.

More generally, the RHS of equations can be values, inheritance descriptors (quoted or unquoted), or (possibly empty) sequences of values and/or descriptors. Inheritance descriptors specify where the required values can be inherited from, and sequences allow arbitrary lists of atoms to be built as values. Inheritance descriptors come in several forms with two dimensions of variation. The unquoted/quoted distinction specifies whether the inheritance context is local (the most recent context employed) or global (in simple cases, the initial context employed). Once the context is established, the descriptor specifies a new node, a new path, or both to be used to determine the inherited value. We will give examples of most of the syntactic possibilities below.

A second type of simple DATR equation uses a node/path pair as its inheritance descriptor and thus has the following form,

Node1:Path1 == Node2:Path2.

which says that the value of **Path1** at **Node1** is to be found by getting the value of **Path2** at **Node2**. A variant of this second DATR statement type uses a **Node** as an inheritance descriptor,

Node1:Path == Node2.

but this can be seen as simply an abbreviation for

Node1:Path == Node2:Path.

And such a statement tells us that the value of **Path** at **Node1** is to be sought at **Node2**. Here are some examples of this frequently used statement type:

Flax:<> == CROP.

CROP:<> == PLANT.

PLANT:<> == ENTITY.

Canvas:<alternants> == ARTEFACT.

Here, the first three statements tell us that the lexeme **Flax** inherits values from corresponding paths at the **CROP** node; that **CROP** inherits values from **PLANT** and that the latter inherits from **ENTITY**. The final example says that the value for the **<alternants>** path of the lexeme **Canvas** is to be sought from the corresponding path at the **ARTEFACT** node.

Another type of simple DATR equation uses a path as an inheritance descriptor and has the form

Node:Path1 == Path2.

which can be seen as no more than an abbreviation for

Node:Path1 == Node:Path2.

and this says that the value of **Path1** at **Node** is to be sought by finding the value of **Path2** at **Node.**

ENTITY:<alt genus word> == <word>.

This example can be glossed as saying that the value for the **<alt genus word>** path at the **ENTITY** node is to be obtained by finding the value for the attribute **word** at the same node.

A third type of DATR equation turns out to be invaluable in DATR analyses, but is conceptually rather different from the equation types that we have just introduced. It uses a quoted path as the inheritance descriptor and looks like this

Node:Path1 == "Path2".

Its interpretation involves a global reference to the node from which one's query originated -- at the risk of oversimplification, we can paraphrase it as saying that the value of **Path1** at **Node** is whatever the value of **Path2** is at the original query node. The following is a characteristic example of the use of this quoted path form,

YARN:<made-of> == "<source>".

this says, in effect, that yarns are made of whatever substance is identified as the value of the **source** attribute at the node from which the query originated. If we had said,

YARN:<made-of> == <source>.

then we would most likely get no value at all, since this is just equivalent to

YARN:<made-of> == YARN:<source>.

and **<source>** is unlikely to be defined at the (abstract) **YARN** node, nor, in the absence of a global reference mechanism, could it be usefully defined so as to return exactly the **source** of whatever particular lexeme we happened to be interested in.

There are two other basic DATR equation types, which use quoted nodes and quoted node/path pairs as their descriptors:

Node1:Path == "Node2".

Node1:Path1 == "Node2:Path2".

Their semantics is subtle and, although our treatment of the flax/linen relation makes use of an instance of the latter, they will not be discussed further here.

One further aspect of DATR is illustrated in the fragment that we present in Section 3 of this paper. This is the possibility of having sequences of values and/or inheritance descriptors on the right-hand side of equations. Thus all of the following are perfectly legal,

Node1:Path1 == Value1.

Node1:Path1 == Value1 Path2.

Node1:Path1 == Path2 Value1.

Node1:Path1 == Value1 Node2.

Node1:Path1 == Node2 Value1.

Node1:Path1 == Value1 Value2.

Node1:Path1 == Value1 "Path2".

Node1:Path1 == Node2:Path2 Value1.

Node1:Path1 == "Path3" Value1 Node2:Path2.

Node1:Path1 == Node2:Path2 Value1 "Path3" Value2.

along with an infinity of others. Here are some concrete examples taken from the fragment with which the present paper deals.

PLANT:<collocates> == grow ENTITY.

Cotton:<alt fibre collocates> == wool FIBRE:<collocates>.

YARN:<made-of> == "<source>" fibre.

CROP:<alternants> == fibre seed PLANT.

These behave as the notation would lead you to expect -- instead of getting atomic values from them, one gets value sequences.

Now that we have presented the syntax of the DATR language, we will turn briefly to a couple of key rules of inference that apply in the language. The first rule implements local inheritance, and uses the following additional meta-notational device: the expression *E0{E2/E1}* denotes the result of substituting *E2* for all occurrences of *E*1 in *E0*.

(LOC) Node2:Path2 == A.

Node1:Path1 == B.

Node1:Path1 == B{A/Node2:Path2}.

Rule LOC says that if we have a theorem **Node1:Path1 == B.** where **B** contains **Node2:Path2** as a subexpression, and we also have a theorem **Node2:Path2 == A.**, then we can derive a theorem in which all occurrences of **Node2:Path2** in **B** are replaced by **A**. In the simplest case, this means that we can interpret a statement of the form

Node1:Path1 == Node2:Path2.

as an inheritance specification meaning "the value of **Path1** at **Node1** is inherited from **Path2** at **Node2**". So for example, from:

Flax:<word> == flax.

Linen:<source> == Flax:<word>.

one can infer:

Linen:<source> == flax.

LOC can also handle inheritance for node and path descriptors, in view of the following, already noted, equivalences:

Node1:Path1 == Node2. is equivalent to

Node1:Path1 == Node2:Path1.

Node1:Path1 == Path2. is equivalent to

Node1:Path1 == Node1:Path2.

Rule LOC implements a local notion of inheritance in the sense that the new node or path specifications are interpreted in the current local context. The second inference rule considered here implements a non-local notion of inheritance: quoted paths specify paths which are to be interpreted in the context of the node in which the original query was made (the global context), rather than the current context.[3]

(GLO) Node1:Path2 == Value.

Node1:Path1 == A.

Node1:Path1 == A{Value/"Path2"}.

To see how the operation of the GLO rule differs from LOC, consider the following theory:

YARN:

<source> == undefined

<made-of> == <source>.

Linen:

<source> == flax

<made-of> == YARN.

The intention here is that the **YARN** node expresses the generalisation that yarns are made of whatever the source material is for the instance of yarn involved. But the theory as stated fails to express this intention since we can derive the following unwanted theorem:

Linen:

<made-of> == undefined.

To achieve the desired result, we must modify the theory as follows:

YARN:

<source> == undefined

<made-of> == "<source>".

Linen:

<source> == flax

<made-of> == YARN.

With this change, the GLO inference rule allows us to derive the theorem we want:

Linen:

<made-of> == flax.

The proof is as follows:

(1) Linen:<made-of> == YARN. **(given)**

(2) YARN:<made-of> == "<source>". **(given)**

(3) Linen:<made-of> == "<source>". **(LOC on 1 and 2)**

(4) Linen:<source> == flax. **(given)**

(5) Linen:<made-of> == flax. **(GLO on 3 and 4)**

For completeness, we state below the only other inference rule that is relevant to the fragment in this paper, but we will not discuss it.

(QNP) Node2:Path2 == Value.

Node1:Path1 == A.

Node1:Path1 == A{Value/"Node2:Path2"}.

In addition to the conventional inference described above, DATR has a nonmonotonic notion of inference by default: each equation about some node/path combination implicitly determines additional equations about all the extensions to the path at that node for which no more specific equation exists in the theory.

To characterise this notion of default inference, we need some auxiliary definitions. The expression **Path1^Path2** denotes the path formed by concatenating the two paths. And we say that **Path3** is an extension of **Path1** if and only if there is a **Path2** such that **Path3** = **Path1^Path2**, and that **Path3** is a strict extension of **Path1** if and only if **Path2** is non-empty. We also use the ^ operator to denote extension of all the paths in a DATR equation, as in the following examples:

If S is N:<a> == v.

then S^<b c> is N:<a b c> == v.

If S is N1:<a> == N2:<b c>.

then S^<b c>is N1:<a b c> == N2:<b c b c>.

If S is N1:<a> == "<>".

then S^<b c>is N1:<a b c> == "<b c>".

Given an equation **S**, we define the root of **S** to be the node/path expression appearing to the left of the equality in S (for example the root of **Node:Path == Value.** is **Node:Path)**. Given a set of equations **T**, a **Node** and a **Path**, we say **Node:Path** is specified in **T** if and only if **T** contains an equation **S** whose root is **Node:Path**.

Let **Node1:Path1** and **Node1:Path2** be such that **Node1:Path1** is specified in **T**. We say **Node1:Path2** is connected to **Node1:Path1** (relative to **T**) if and only if:

(i) **Path2** is an extension of **Path1**, and

(ii) there is no strict extension **Path3** of **Path1** of which **Path2** is an extension such that **Node1:Path3** is specified in **T**.

So **Node1:Path2** is connected to **Node1:Path1** if and only if **Path1** is the maximal subpath of **Path2** that is specified (with **Node1**) in **T**.

Given a set of equations **T**, we define the path closure of **T** to be:

{**S^Q** | **S** is an equation in **T**, with root **Node:Path**, and **Node:Path^Q** is connected to **Node:Path**}

It is clear from these definitions that any **Node:Path** is connected to itself and thus that **T** is always a subset of the path closure of **T**. The path closure contains all those theorems which can be inferred by

default from **T**. The operation of path closure is non-monotonic: if we add more equations to our original theory, some of our derived equations may cease to be true. The two forms of inference in DATR are combined by taking the path closure of a theory first, and then applying the inference rules to the result.

To illustrate path closure, consider the following example theory:

CROP:

<> == PLANT

<syntax> == MASS-NOUN:<>

<alt fibre> == FIBRE:<>.

Cotton:

<> == CROP

<word> == cotton.

We can infer by default the following theorems, among others:

CROP:

<genus> == PLANT:<genus>

<made-of> == PLANT:<made-of>

<artefact> == PLANT:<artefact>

<collocates> == PLANT:<collocates>

<syntax cat> == MASS-NOUN:<cat>

<syntax count> == MASS-NOUN:<count>

<syntax concrete> == MASS-NOUN:<concrete>

<alt fibre alt yarn> == FIBRE:<alt yarn>

<alt fibre alt yarn alt fabric> == FIBRE:<alt yarn alt fabric>.

Cotton:

<word form> == cotton

<genus> == CROP:<genus>

<word root form> == cotton

<made-of> == CROP:<made-of>

<artefact> == CROP:<artefact>

<collocates> == CROP:<collocates>

<syntax cat> == CROP:<syntax cat>

<syntax count> == CROP:<syntax count>

<syntax concrete> == CROP:<syntax concrete>

<alt fibre alt yarn> == CROP:<alt fibre alt yarn>

<alt fibre alt yarn alt fabric> == CROP:<alt fibre alt yarn alt fabric>.

Note the way in which equations that have paths on their RHS are also extended by the subpath used to extend the LHS. This characteristic of the DATR language plays a crucial role in our treatment of alternant meanings in the rest of this paper.

3. Representing polysemy

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. We refer to these eponymous nodes as *lexemes*. Thus a query regarding the syntax of the word *silk*, and a query asking what type of thing silk is, will both be made at the same node. There are many kinds of information about denotations which have consequences for words. For example, the kind of thing a word denotes determines, at least by default, the alternations that it will participate in. And, in many languages, the type of denotation determines the default noun class or gender for a word. There is thus much to be gained from holding the two types of information together. The matter receives a fuller discussion in Kilgarriff (1992). Below, we proceed on the basis that linguistic and encyclopaedic information should inhabit

the same representation scheme.

Our example fragment of lexical semantics embodies the taxonomy shown in Figure 1.

<Figure 1 about here>

This taxonomy gets encoded in DATR as follows:

ARTEFACT:<> == ENTITY.

PLANT:<> == ENTITY.

FIBRE:<> == ENTITY.

CROP:<> == PLANT.

YARN:<> == ARTEFACT.

FABRIC:<> == ARTEFACT.

GARMENT:<> == ARTEFACT.

Cotton:<> == CROP.

Flax:<> == CROP.

Silk:<> == FIBRE.

Wool:<> == FIBRE.

Nylon:<> == FIBRE.

Linen:<> == YARN.

Felt:<> == FABRIC.

Canvas:<> == FABRIC.

Jersey:<> == FABRIC.

Suit:<> == GARMENT.

To that basic structure, we wish to add generalisations about fibre, yarn, fabric and garment senses. Once we have established that *silk*, say, is being used in its fabric sense, we wish to treat the word as we would *canvas* or *felt*. We need to distinguish secondary senses from primary ones in such a way

that the paths for accessing information about them are different. We do this by prefixing the path with **alt** (for alternation). There might be several alternations, so we identify the alternation by the path element following **alt**, for which we shall use the genus terms of the alternate senses. Let us also now add some flesh to the bare bones of the taxonomy, and state some **genus** terms, **word** values (i.e., the word associated with the node), and **collocates** at various low-level nodes, and other information whose role will become clearer as we proceed.

ENTITY:

<word> == "<word>"

<collocates> ==

<artefact> == no

<made-of> == itself.

ARTEFACT:

<> == ENTITY

<artefact> == yes.

FIBRE:

<> == ENTITY

<genus> == fibre

<collocates> == spin ENTITY

<alt yarn> == YARN:<>.

Silk:

<> == FIBRE

<word> == silk

<source> == insect

<collocates> == worm FIBRE

<alt yarn alt fabric collocates> == fine FABRIC:<collocates>.

YARN:

<> == ARTEFACT

<genus> == yarn

<collocates> == stitch weave ARTEFACT

<alt fabric> == FABRIC:<>.

FABRIC:

<> == ARTEFACT

<genus> == fabric

<collocates> == cut sew ARTEFACT.

Now, if we query the fragment to obtain a value for

Silk:<genus>

the value is **fibre**, whereas for

Silk:<alt yarn genus>

the **<alt yarn>** path prefix diverts the inheritance (via **FIBRE**) to **YARN**. The effect of the empty path on the right hand side of the equation

FIBRE:<alt yarn> == YARN:<>.

is to direct the inheritance to the **YARN** node with the path prefix replaced by the null path. In this case, that leaves the path **<genus>**, which is evaluated at **YARN** to give **yarn**. This prefix stripping yields the desired results in that, once we have specified that we have the yarn sense of a fibre word, the analysis behaves as if the primary sense were a yarn sense.

The axioms involving **collocates** exploit DATR sequences. The evaluation of sequences is such that each sequence member is treated as if it were alone on the RHS of the equation, is evaluated, and the value is placed back in the sequence. If we wish to find the collocates of the primary fibre sense of *silk*, we need to evaluate

Silk:<collocates>

At the **Silk** node, we find an equation for which the LHS matches, and the RHS. The first element of the sequence is an atom, **worm**, so that becomes the first element of the sequence that is returned. The second element is not an atom, but a node, so for the remainder of the sequence we need to evaluate

FIBRE:<collocates>

which is again specified as a sequence. The first element, **spin** is an atom and thus returned unchanged. The second element is a node, **ENTITY**. For the latter, **<collocates>** is given as the null sequence. We now have all the components of the sequence that forms the value for the original query. The empty sequence disappears and the value sequence returned is **worm spin**.

Alternation is rarely wholly regular and it is often necessary to overrule inherited values, or to add specifications that are not inherited to an inherited sense. This is easily done in DATR. The garment alternant sense of the fabric word *jersey* typically denotes not just any garment made of jersey, but rather one that covers the trunk of the wearer, hence:

Jersey:<alt garment differentia 1> == covers trunk.

Likewise, the collocates of particular alternants may be very specific and not restricted to those one might inherit from the more general meaning category to which the alternant belongs:

Jersey:<alt garment collocates> == football GARMENT:<collocates>.

In general, any number of further specifications may be added to an inherited sense in this way.

Alternation relations display transitivity. Just as words that denote fibres may have alternant senses that denote yarns, so words that denote yarns may have alternant senses that denote fabrics, as with the word *linen*:

YARN:

<> == ARTEFACT

<genus> == yarn

<made-of> == "<source>" fibre

<collocates> == stitch weave ARTEFACT

<alt fabric> == FABRIC:<>.

Linen:

<> == YARN

<word> == linen

<source> == flax

<alt fabric alt garment collocates> == crumpled GARMENT:<collocates>.

FABRIC:

<> == ARTEFACT

<genus> == fabric

<made-of> == "<word>" yarn

<collocates> == cut sew ARTEFACT

<alt garment> == GARMENT:<>.

From the fragment presented so far, we can prove DATR theorems such as the following:

Linen:

<word> == linen

<genus> == yarn

<artefact> == yes

<made-of> == flax fibre

<collocates> == stitch weave

<alt fabric word> == linen

<alt fabric genus> == fabric

<alt fabric artefact> == yes

<alt fabric made-of> == linen yarn

<alt fabric collocates> == cut sew.

Transitivity becomes apparent when we reconsider a word like *silk*, which can denote a fibre, a yarn made from that fibre, or a fabric made from the yarn that is made of the fibre. In the approach developed here, the basic mechanism for transitive alternations is to use as many **<alt x >** prefixes (where **x** is the identifier for the alternation) as required. Thus, from the fragment presented so far, we can prove DATR theorems such as the following:

Silk:

<word> == silk

<genus> == fibre

<artefact> == no

<made-of> == itself

<collocates> == worm spin

<alt yarn word> == silk

<alt yarn genus> == yarn

<alt yarn artefact> == yes

<alt yarn made-of> == insect fibre

<alt yarn collocates> == stitch weave

<alt yarn alt fabric word> == silk

<alt yarn alt fabric genus> == fabric

<alt yarn alt fabric artefact> == yes

<alt yarn alt fabric made-of> == silk yarn

<alt yarn alt fabric collocates> == fine cut sew.

A word like *cotton* has alternant senses that stretch all the way from the crop grown in the southern states of the USA through to the garments we are wearing, as the following examples attest:

The children were harvesting the cotton. [CROP]

Every gin was clogged with cotton. [FIBRE]

He threaded the needle with black cotton. [YARN]

I'll buy three metres of the red cotton, please. [FABRIC]

We always wash the cottons separately. [GARMENT]

We can augment our fragment to allow all these senses of *cotton*:

GARMENT:

<> == ARTEFACT

<genus> == garment

<made-of> == "<word>" fabric

<collocates> == wear clean ARTEFACT.

PLANT:

<> == ENTITY

<genus> == plant

<source> == vegetable

<collocates> == grow ENTITY.

CROP:

<> == PLANT

<collocates> == seed sow harvest PLANT

<alt seed> == SEED:<>

<alt fibre> == FIBRE:<>.

Cotton:

<> == CROP

<word> == cotton

<collocates> == field picking gin belt CROP

<alt fibre collocates> == wool FIBRE:<collocates>

<alt fibre alt yarn collocates> == mill YARN:<collocates>

<alt fibre alt yarn alt fabric alt garment collocates> == cool GARMENT:<collocates>.

We can now prove a variety of theorems about the various alternant senses of *cotton*. Note, in particular, how the DATR treatment of the **Cotton** lexeme allows one to fine tune the set of collocations associated with each sense.

Cotton:

<word> == cotton

<genus> == plant

<artefact> == no

<made-of> == itself

<collocates> == field picking gin belt seed sow harvest grow

<alt fibre word> == cotton

<alt fibre genus> == fibre

<alt fibre artefact> == no

<alt fibre made-of> == itself

<alt fibre collocates> == wool spin

<alt fibre alt yarn word> == cotton

<alt fibre alt yarn genus> == yarn

<alt fibre alt yarn artefact> == yes

<alt fibre alt yarn made-of> == cotton fibre

<alt fibre alt yarn collocates> == mill stitch weave

<alt fibre alt yarn alt fabric word> == cotton

<alt fibre alt yarn alt fabric genus> == fabric

<alt fibre alt yarn alt fabric artefact> == yes

<alt fibre alt yarn alt fabric made-of> == cotton yarn

<alt fibre alt yarn alt fabric collocates> == cut sew.

<alt fibre alt yarn alt fabric alt garment word> == cotton

<alt fibre alt yarn alt fabric alt garment genus> == garment

<alt fibre alt yarn alt fabric alt garment artefact> == yes

<alt fibre alt yarn alt fabric alt garment made-of> == cotton fabric

<alt fibre alt yarn alt fabric alt garment collocates> == cool wear clean.

There may be any number of **<alt x >** prefixes, and a query may be redirected any number of times. There can be any number of alternations specified at nodes: **CROP**, for example, has two, one pointing to the **FIBRE** node and the other pointing to a **SEED** node (the latter omitted from the example fragment). Each alternation possibility redirects inheritance to another node and strips off the relevant

<alt x > prefix. Thus, as the number of **<alt x >** prefixes grows, so the number of potential usage-types which the theory is describing for the word increases exponentially (in principle, if not in practice). All the alternations directly available to the primary sense of the word form a set of possibilities at depth 1 (thus **fibre** and **seed** are depth 1 alternants for crops). These, in turn, may lead to nodes at which further alternations are defined (e.g. **yarn** at **FIBRE**) and these form the set of possibilities at depth 2. And so on, recursively.

Suppose we now add another crop to our fragment:

Flax:

<> == CROP

<word> == flax

<collocates> == linseed CROP.

Unsurprisingly, we can immediately prove theorems like these:

Flax:

<word> == flax

<genus> == plant

<collocates> == linseed seed sow harvest grow

<alt fibre word> == flax

<alt fibre genus> == fibre

<alt fibre collocates> == spin.

Those results look plausible enough, but if we push into the depth two alternations then a problem emerges:

Flax:

<alt fibre alt yarn word> == flax

<alt fibre alt yarn genus> == yarn

<alt fibre alt yarn collocates> == stitch weave.

The problem here is that the word **flax** is not used to refer to the yarn derived from flax fibre. The alternation from crop senses to yarn senses has exceptions, even if we restrict the notion of crop to "fibre-producing crop". We have already seen how DATR permits idiosyncratic collocations to be stipulated for remote alternants. It is no harder to use it to stipulate that certain remote alternants are not available to a lexeme. The simplest way to achieve this is illustrated in the revised definition for the **Flax** node shown below (**UNDEFINED** is the name of a node that is not defined).

Flax:

<> == CROP

<word> == flax

<collocates> == linseed CROP

<alt fibre alt yarn> == UNDEFINED.

With the additional axiom, we lose all the problematic theorems that could be inferred in its absence. However, in this particular case, we might want instead to do something a bit more interesting:

Flax:

<> == CROP

<word> == flax

<collocates> == linseed CROP

<alt fibre alt yarn> == "Linen:<>".

Linen:

<> == YARN

<word> == linen

<source> == Flax:<word>

<alt fabric alt garment collocates> == crumpled GARMENT:<collocates>.

This version of the theory leads to almost the same set of theorems as our original theory did:

Flax:

<alt fibre alt yarn word> == linen

<alt fibre alt yarn genus> == yarn

<alt fibre alt yarn collocates> == stitch weave.

The crucial difference lies in the value of the **word** attribute of the depth 2 yarn alternant: in place of **flax**, we now find **linen**. We will not take a stand here on the question of whether one should say of *flax* that its yarn alternant is undefined or whether one should say that the alternant is defined and is exactly what you would expect it to be, except in that this particular alternant is written as *linen* not *flax*.[4] The latter approach is perfectly coherent in the context of a DATR lexicon although it takes us beyond the territory of polysemy as standardly construed.

Our use of an **UNDEFINED** node, above, is a way of eliminating alternants that do not exist. For depth 1 alternants, it may be that descriptive concision is better achieved by means of a positive specification of the applicable alternations. To this end, we introduce an attribute **alternants** whose value will be a list of the available alternants at the node.

ENTITY:

<alternants> == .

CROP:

<alternants> == fibre seed PLANT.

FIBRE:

<alternants> == yarn ENTITY.

YARN:

<alternants> == fabric ARTEFACT.

FABRIC:

<alternants> == garment ARTEFACT.

Canvas:

<alternants> == ARTEFACT.

The mechanism employed for **alternants** is exactly the same as that already employed for **collocates**: the RHS is given as zero or more alternant attributes followed by the name of a higher node from which further alternants may be sought. The ultimately general **ENTITY** node, unsurprisingly, has an empty set of alternants associated with it. The only interesting case in the list just given is that of *canvas*: the **Canvas** node is a daughter of the **FABRIC** node which, by default, leads to alternant garment senses in its daughters. But, by stipulating that **Canvas** alternants are to come from **ARTEFACT** rather than **GARMENT**, the default is bypassed.[5]

Our topic in this paper has been polysemous relations and thus polysemy. But what are we to do when no subregular relation holds? Although we share Lyons's sympathy for maximal polysemy, a DATR lexicon is perfectly well able to represent homonymies.

Jersey1:

<> == FABRIC

<word> == jersey

<made-of> == knitted yarn

<alt garment collocates> == football GARMENT:<collocates>.

Jersey2:

<> == ISLAND

<word> == jersey

<alternants> == cattle ISLAND

<collocates> == tax Bergerac Channel ISLAND:<collocates>.

In a sense, however, our priorities have been the reverse of those of Lyons in his writings on homonymy and polysemy. For us, alternation relations and the formal explication of the notion of relatedness of lexical meaning form the primary object of study. Polysemy is simply the name for the sets of multiple senses that fall out from a theory of such relations: "the only synchronically relevant consideration", as Lyons puts it. And that leaves homonymy as the name for whatever same-form/multiple-sense cases remain. What makes the study of polysemous relations difficult is the pervasive subregularity of the phenomenon. Our claim is that that subregularity is formally no different from the subregularity one finds in the lexical representation of syntax and morphology. DATR was developed on a test-bed of lexical examples involving syntactic subcategorization and inflectional morphology, but, as we hope to have shown, it offers a rather natural way of formalizing polysemous relations also.

*This paper is a sequel to Kilgarriff (1993). We are grateful to Ann Copestake, Roger Evans and Lionel Moser for relevant conversations and to SERC for the grant to Kilgarriff during his doctoral research.

[1]There is actually a formal distinction in DATR between two kinds of equation (notated with "==" and "=", respectively) but the discussion in this paper will simply ignore this technical nicety.

[2]Collocates are words that commonly occur as near neighbours of instances of the lexeme that have the specified sense. Our intended interpretation of collocates is the purely statistical one to be found in Church & Hanks (1989). However, the examples given for the example fragment in this paper are entirely hypothetical -- we have not done the empirical work that would be required to discover what the collocates *really* are.

[3]The correct formulation of the rule of inference depends on the already noted distinction between two kinds of DATR equation, but we continue to ignore that technicality here.

[4]An exactly analogous case occurs with the **garment** alternation for the fabric sense of *wool*: the relevant form is *woollen* or *woolly*, not *wool*. The same analytical alternatives are available.

[5]Nothing in the fragment as presented in this paper formally connects the value of the **alternants** attribute to the **<alt x >** machinery. In effect, **alternants** simply tells the user of the lexicon which **<alt x >** paths have meaningful values. It is not difficult to make a formal link between the two, within the DATR code, such that, for example, **<alt x z>** only has a defined value at node **N** just in case **x** is to be found on the RHS of the evaluation of **N:<alternants>**. And recursively for **<alt x alt y z>**, etc. But the technicalities of how one might do this are of no real relevance to the present paper.

References

Apresjan, Ju. D. (1974) Regular polysemy. *Linguistics* 142, 5-32.

Church, Kenneth & P. Hanks (1989) Word association norms, mutual information and lexicography. *27th Annual Conference of the Association for Computational Linguistics,* 76-83.

Copestake, Ann (1993) *The representation of lexical semantic information.* DPhil dissertation, University of Sussex.

Evans, Roger & Gerald Gazdar (1989a) Inference in DATR. *Fourth Conference of the European Chapter of the Association for Computational Linguistics,* 66-71.

Evans, Roger & Gerald Gazdar (1989b) The semantics of DATR. In Anthony G. Cohn, ed. *Proceedings of the Seventh Conference of the Society for the Study of Artificial Intelligence and Simulation of Behaviour.* London: Pitman/Morgan Kaufmann, 79-87.

Flexner, Stuart Berg (1987) *The Random House Dictionary of the English Language,* Second Edition. New York: Random House.

Kilgarriff, Adam (1992) *Polysemy.* DPhil dissertation, University of Sussex.

Kilgarriff, Adam (1993) Inheriting polysemy. In Patrick Saint-Dizier & Evelyne Viegas, eds. <*Proceedings of Touluse Workshop on Computational Lexical Semantics.*> Cambridge: Cambridge University, 00-00 [COPY EDITOR: please get title and page numbers for this reference from the relevant CUP colleague.]

Lyons, John (1968) *Introduction to Theoretical Linguistics.* Cambridge: Cambridge University Press.

Lyons, John (1977) *Semantics,* Volume 2. Cambridge: Cambridge University Press.

Lyons, John (1981) *Language, Meaning and Context.* London: Fontana.

APPENDIX A

Example lexicon fragment

ENTITY:

<word> == "<word>"

<made-of> == itself

<alt $alt alt> == <alt>

<alt $alt word> == <word>

<alternants> ==

<collocates> ==

<artefact> == no

<syntax> == NOUN:<>.

NOUN:

<cat> == noun

<count> == yes

<concrete> == yes.

MASS-NOUN:

<> == NOUN

<count> == no.

ARTEFACT:

<> == ENTITY

<artefact> == yes.

PLANT:

<> == ENTITY

<genus> == plant

<source> == vegetable

<collocates> == grow ENTITY.

CROP:

<> == PLANT

<syntax> == MASS-NOUN:<>

<collocates> == seed sow harvest PLANT

<alt fibre> == FIBRE:<>

<alt seed> == SEED:<>

<alternants> == fibre seed PLANT.

Cotton:

<> == CROP

<word> == cotton

<collocates> == field picking gin belt CROP

<alt fibre collocates> == wool FIBRE:<collocates>

<alt fibre alt yarn collocates> == mill YARN:<collocates>

<alt fibre alt yarn alt fabric

alt garment collocates> == cool GARMENT:<collocates>.

Flax:

<> == CROP

<word> == flax

<collocates> == linseed CROP

<alt fibre alt yarn> == "Linen:<>".

FIBRE:

<> == ENTITY

<genus> == fibre

<syntax> == MASS-NOUN:<>

<collocates> == spin ENTITY

<alt yarn> == YARN:<>

<alternants> == yarn ENTITY.

Silk:

<> == FIBRE

<word> == silk

<source> == insect

<collocates> == worm FIBRE

<alt yarn alt fabric collocates> == fine FABRIC:<collocates>.

Wool:

<> == FIBRE

<word> == wool

<source> == animal

<collocates> == sheep shear fleece FIBRE

<alt yarn alt fabric collocates> == warm FABRIC:<collocates>.

Nylon:

<> == FIBRE

<word> == nylon

<artefact> == yes

<source> == synthetic

<alt yarn alt fabric

alt garment collocates> == stocking GARMENT:<collocates>.

YARN:

<> == ARTEFACT

<genus> == yarn

<syntax> == MASS-NOUN:<>

<made-of> == "<source>" fibre

<collocates> == stitch weave ARTEFACT

<alt fabric> == FABRIC:<>

<alternants> == fabric ARTEFACT.

Linen:

<> == YARN

<word> == linen

<source> == Flax:<word>

<alt fabric alt garment collocates> == crumpled GARMENT:<collocates>.

FABRIC:

<> == ARTEFACT

<genus> == fabric

<syntax> == MASS-NOUN:<>

<made-of> == "<word>" yarn

<collocates> == cut sew ARTEFACT

<alt garment> == GARMENT:<>

<alternants> == garment ARTEFACT.

Felt:

<> == FABRIC

<word> == felt

<made-of> == matted fibre

<collocates> == hat FABRIC.

Canvas:

<> == FABRIC

<word> == canvas

<made-of> == woven yarn

<collocates> == sail tent painting boxing FABRIC.

<alternants> == ARTEFACT.

Jersey:

<> == FABRIC

<word> == jersey

<made-of> == knitted yarn

<alt garment collocates> == football GARMENT:<collocates>.

GARMENT:

<> == ARTEFACT

<genus> == garment

<made-of> == "<word>" fabric

<collocates> == wear clean ARTEFACT.

Suit:

<> == GARMENT

<word> == suit

<made-of> == fabric

<collocates> == grey business GARMENT.