Dependency Structure and Cognition

Invited talk

Richard Hudson
emeritus professor in the Department of Phonetics and Linguistics
University College London
Great Britain
r.hudson@ucl.ac.uk

1 Language and cognition

We probably all share an interest in syntax, so we would dearly love a clear and certain answer to the question: what is syntactic structure like? Is it based on dependencies between words, or on phrases? What kinds of relation are there? And so on. But before we can answer relatively specific questions like these, we must first answer a much more general question: What kind of thing do we think language is? Or maybe: Where do we think language is – nowhere, in society, in our minds? Our answer will decide what basic assumptions we make, and how our discipline, linguistics, relates to other disciplines.

Is language a set of abstract patterns like those of mathematics, without any particular location? This is a popular answer, and makes a good deal of sense. After all, what is language if not abstract patterning? The patterns made by words in a sentence, or by segments in a syllable, are certainly abstract and regular, and can be studied as a branch of mathematics – as indeed they have been studied and still are studied in linguistics. For some researchers who take this approach, the aim is elegance and consistency; so in a competition between alternative analyses, the prize goes to the simplest one. For others, though, the goal is a working computational system, so the criterion is some kind of efficiency. One problem for this approach is that the material in which these patterns are embedded is inescapably human activity; in contrast with mathematical patterns, linguistic patterns only exist because humans create them. And another problem with the mathematical approach is that it provides few explanations for why language is as it is. If language patterns always turned out to be the most elegant possible patterns, the mathematical approach would indeed explain why; but they don't, and as we all know, language can be frustratingly messy.

Another possible answer is that language is a set of conventions that exist in society. For some linguists the social character of language is fundamental (Halliday and Matthiessen 2006), and they like to focus on the role of language in 'construing' experience. Language exists 'out there'in the community, as well as being shared by all its members; so the methods of sociology and cultural anthropology should apply. Similarly, some sociolinguists see the social patterning of variation as belonging to the community, though not to any of its members (Labov 1972). The trouble with this approach is that communities are much harder to define, and much less homogeneous, than we might expect; and once again, the basic data are irreducibly individual products – individuals speaking and listening to each other.

The third answer – and this is my preferred option – is that language is an example of individual knowledge. As in the first answer, the knowledge involves mathematically expressible patterning; and as in the second, it has a strong social dimension – after all, we learn the knowledge from others in our community, and we reveal our knowledge through our own social behaviour as speakers and listeners. But ultimately language is a matter of individual psychology. We learn it as individuals, we use it as individuals, and others know us, as individuals, through it. Who could deny this? And yet the other views of language have been very influential, and still are.

As an important example of its influence, take the criterion of elegance or simplicity. This is very widely accepted in linguistics, and those of us who support dependency structure might argue that one of the attractions of our approach, in contrast with phrase structure, is its simplicity. Just count the nodes! We have precisely one node per word, whereas a phrase-structure analysis contains all these word nodes, plus extra nodes for the phrases. But is this criterion really relevant? If we were physicists, it certainly

Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), pages 1–11,
Prague, August 27–30, 2013.

would be; but we aren't. We're studying a part of the human mind, and any human mind is the product of a long and complicated experience; so why should we believe that any mind is simple? As cognitive linguists argue, we learn our language from 'usage' (Barlow and Kemmer 2000) – from the millions of examples of language that we hear, each embedded in a very specific social context. And we interpret each example in terms of the examples that went before, using a growing system of concepts. Nothing there is simple: for any given language, thousands or millions of speakers all follow different routes to a slightly different adult grammar, with numerous false starts and detours on the way. It's easy to understand why linguists welcome the idea of a simple, perfect and uniform language as a way to escape from this buzz of confusion and complexity. But, like the drunk looking for the keys that he has dropped, we have a choice: we can look under the street lamp, where the light is good; or we can look over in the dark corner, where we know that we actually dropped the keys – a choice between esthetics and truth.

In short, I believe we have to accept that language is part of cognition. And with that acceptance comes the principle that our theories of language structure should be compatible with cognitive science – in fact, our theories are part of cognitive science, and arguably a particularly important part of cognitive science, given the relative clarity and detail of the data found in language. The reality that we are trying to capture in our theories is what is often called 'psychological reality'.

But, you may object, how can we know what is psychologically real? It's true that I can't even look inside my own mind, let alone inside someone else's mind; but then, psychology has moved a long way from the bad old days of introspection, and has findings which are supported by very robust experimental methods. The rest of this paper is an attempt to develop some of the consequences of taking these findings seriously when building models of language. I shall pay special attention to their consequences for my own theory, Word Grammar (**WG**, Hudson 1984, Hudson 1990, Hudson 2007, Hudson 2010, Gisborne 2010, Eppler 2010).

But before I go on to consider some of these findings, I must admit that there is a way to avoid my arguments. This is to claim that although language is part of cognition, it is actually different from everything else – a unique 'module' of the mind (Chomsky 1986, Fodor 1983). Our generative colleagues are free to invent principles, parameters and structures at will, unconstrained by anything but their basic formal assumptions and the purely 'linguistic' facts. As you can guess, I don't think this is a good way to study language because I believe that language is, in fact, just like the rest of cognition in spite of all the attempts to show the contrary.

2 Some things we know about cognition

We start with four very elementary findings which can be found in introductory textbooks on cognitive psychology such as Reisberg (2007), concerning networks, mental relations, complexity and classification.

Knowledge is a **network** of concepts in which each concept is associated with a number of other concepts. These 'associations' explain why experiences evoke neighbouring memories – memories that share links (in the network) to the same concepts; why we make mistakes (including speech errors) when we choose a neighbouring concept in place of the intended target; and why an object in a psychological laboratory 'primes' objects that are its neighbours (as when hearing the word *doctor* makes the word *nurse* easier to retrieve than it would otherwise be). The notion of networks explains all these familiar facts about cognition. But if knowledge in general is a network, and if language is part of knowledge, then language itself must be a network. And that includes not only the whole of language – the grammar and phonology as well as the lexicon – but also the utterances that we interpret in terms of this network of knowledge.

But even though the notion of 'association' is important, we can be sure that the links in our mental network are not merely associations, but **relations** of many different kinds. Just think of all the words you know for kinship relations – words such as *father*, *aunt* and *ancestor*, each of which names a relationship. Then think of all the other person-to-person relationships you can name, including 'father-in-law', 'neighbour' and 'boss'? And then think of the prepositions and nouns you know for non-human relationships, such as *beneath*, *opposite* and *consequence*. The point is that we seem to be able to freely create and learn relational concepts, just as we do non-relational concepts such as 'bird' and 'Londoner'. This conclusion takes us a long way from theories in which our minds recognise only a small, innate set of inbuilt relations called 'syntactic functions' or 'semantic roles'.

However, alongside these learnable relations there is at least one fundamental relation which may well be innate: what AI researchers often call 'is-a', as in 'penguin is-a bird', relating a subcategory to its 'supercategory'. This is the relation that allows all generalisations, so it is bound to play an important part in any theory of cognition. Mental networks seem to be built round taxonomies of concepts related in this way, but with multiple other inter-relations as well. Since this is such an important relation, it has its own special notation in WG: a small triangle whose (large) base rests on the (large) supercategory and whose apex points at the subcategory. This notation is illustrated in the taxonomy of my family members in Figure 1.

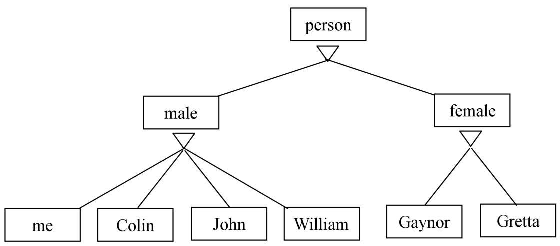

Figure 1: A taxonomy of people

The third relevant claim of elementary psychology is that these knowledge networks can be very **complex**. This is clearly true of language, but other areas of knowledge also turn out to be astonishingly complex. Take once again the example of kinship, as illustrated in Figure 2 by the male members of my immediate family.

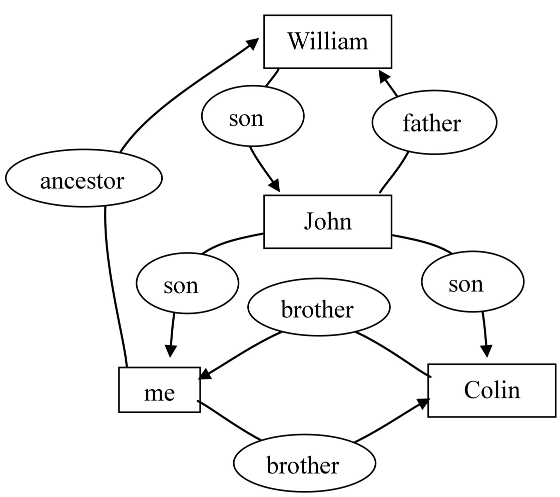

Figure 2: My family

The structure in Figure 2 is part of the same network as that in Figure 1, and like this, it must be part of my cognition because every bit of it is something I know. I know all the people named in the square boxes, and I know how they are (or were) related to each other. Even this tiny fragment of my total knowledge illustrates some important formal properties of the human cognitive network:

- Relations aren't merely 'associations', but are classified (as 'father', 'son' and so on).
- Relations are asymmetrical, in the sense that each one consists of an 'argument' and a 'value' (so the 'son' relation near the top of the diagram has William as its argument and John as its value, showing that John is William's son). In the notation that I use in this diagram (and indeed in later ones), the arrow points towards the value.
- Mutual relations are possible: if John is William's son, then William is John's father; and two individuals may even each have the same relation to the other, as where Colin and I are each other's brother.
- Relations may be recursive. The relevant example here is 'ancestor', which has a recursive definition (A is the ancestor of B either if A is a parent of B, or if A is a parent of an ancestor of B).

These formal properties can be described mathematically, but the one thing they don't do is to limit the space of possibilities: almost anything seems to be possible. This is a very different approach to formal structures compared with the familiar aim of explaining grammars by limiting their formal properties.

The fourth important fact about cognition is that classification ('categorization') is based on **prototypes** – typical cases where a bundle of properties (such as beaks, two legs, flying, feathers and laying eggs which define the typical bird) coincide – with other cases (such as non-flying birds) arranged round these typical ones as more or less exceptional or unclear examples. This way of organising knowledge requires a special kind of logic, called '**default inheritance**', in which generalisations apply 'by default', but can be overridden.

It seems reasonable to assume, therefore, that our minds are capable of handling complex networks in which there are at least two kinds of relations between nodes: the basic 'is-a' relation of categorization and default inheritance, and an open-ended list of relational concepts which are created as needed and learned in the same way as other concepts. This is the mental machinery that we can, and indeed must, assume when building our theories of how language is organised – again, a very different starting point from the rather simple and sparse assumptions behind most of the familiar theories in either the PS or DS families.

3 Dependencies and phrases

These assumption are directly relevant to the debate between PS and DS. The question is how we represent the words in a sentence to ourselves: do we represent them as parts to larger wholes (phrases), or do our mental representations link them directly to one another? For example, is *cows* in (1) related only to the phrase *cows moo*, or is it related directly to *moo*?

(1) Cows moo.

The PS answer evolved out of Wundt's rather impoverished theory of cognition which concentrated on the relation between a whole 'idea' and its immediate parts – the origins of Bloomfield's Immediate-constituent analysis, which in turn led to Chomsky's Phrase structure (Percival 1976). PS analysis rests crucially on the assumption that the whole-part relation between a phrase and its parts is the only possible relation (although of course even PS users talk informally about dependencies such as 'long-distance dependencies').

But the evidence from section 2 shows that the human mind, which creates sentence structure, can handle much, much more complicated structures than whole-part relations. Just think of my family. If we assume, as surely we must, that the full power of the human mind is available for language, and if we can handle direct relations between people then surely we can also handle direct DS relations between words. Moreover, this conclusion confirms what grammarians have been saying for two thousand years about valency links between words. In the fourth century BC, Panini showed the need for semantic relations between verbs and their arguments, and in the second century AD Apollonius pointed out how verbs and prepositions required their objects to have different case inflections (Robins 1967:37). Since then, and through the Arabic grammarians and our Middle Ages up to recent times, these semantic and syntactic links between words have been a regular part of a grammarian's work. It seems very clear, therefore, that our minds are not only capable of recognising word-word dependencies, but actually do recognise them. And in our example, we can be sure that *cows* and *moo* are held together by a direct bond which explains why *moo* has no {s} (in contrast with *the cow moos*).

But where does that leave the notion of phrase? Evidence in favour of word-word relations is not in itself evidence against whole-part relations. By recognising a dependency between *cows* and *moo*, are we also recognising a larger unit, *cows moo*? Here the answer is much less clear, at least to me even after nearly forty years of thinking about it. But I am sure of three things.

- The larger unit, if it exists, is no more than the sum of its parts, because all of its properties – its meaning, its syntactic classification and so on – are the properties of its head word. (I explain in section 8 how the head word carries the meaning of the whole phrase.)
- The larger unit does have boundaries, which certainly are relevant at least for punctuation which marks phrase boundaries: *Cows moo.* No doubt the same is true of intonation. And in phonology and morphology, it is widely accepted that some phenomena are limited to the 'edges' of constituents (Hyman 2008). But maybe that's all there is to a phrase: just its boundaries.
- Unary branching – where a phrase has just one constituent – is where PS is most vulnerable. If we say that *cows* is a noun phrase consisting of a single word, then we are stretching the notion of 'part' beyond its limit. The fact is, or seems to me to be, that we don't normally allow objects to have just one part. For instance, if a box normally has a lid, but we lose the lid, we don't think of the box as having one part. What would that part be, other than the box itself? But if we forbid unary branching, we lose one of the main supposed benefits of phrases, which is to allow generalisations across complex phrases and single words (so that *cows*, *brown cows* and even *they* count as 'noun phrases').

In short, we can be much more sure about the mental existence of word-word dependencies than about that of phrases; but we're certainly capable of recognising whole-part relations, so we can't rule them out altogether. The result is that we certainly need an analysis like the righthand one in Figure 3, but we may also need to include the lefthand structure. (I discuss the unorthodox DS notation below.)

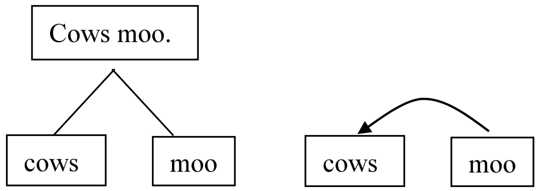

Figure 3: PS or DS?

4 Bundles or levels?

Dependencies do many different jobs, from carrying 'deep' information such as semantic roles to carrying more 'surface' information such as word order and inflectional categories. Moreover, dependency relations can be classified in terms of familiar syntactic functions such as 'subject', whose definitions typically span a range of different kinds of information from deep to surface (Keenan 1976). One major theoretical question for DS analysis is what to do with this diversity of information 'carried' by dependencies. As so often in theoretical questions, we find 'splitters' and 'lumpers' – those who split dependencies into different types according to the information they carry, and those who lump the different relations together. Once again, our cognitive assumptions throw important light on the question.

Remember that relational concepts (such as dependencies) are concepts, so like other concepts, their main function is to bring together properties that tend to combine. For instance, the relation 'father' brings together biological properties (procreation) with social properties (parental rights and responsibilities), just as the closely related concept 'male' does. Splitters might argue that the biological and social are importantly different, so they should be separated to give 'b-father', carrying the biological properties, and 's-father' with the social ones. But lumpers would argue – rightly, in my opinion – that this misses the point. After all, the two property-sets tend to coincide, so even if you distinguish two kinds of father, you also need some mechanism to show the special connection between them. So why not simply call them both 'father', and allow the 'father' prototype to have both sets of properties? The existence of exceptional cases (men who father children without looking after them, or vice versa) is easily accommodated thanks to the logic of default inheritance.

Exactly the same argument supports the lumpers in syntax against those who favour separating 'deep' dependents from more 'surface' ones, as in the separation of semantic, syntactic and morphological dependencies in Meaning-Text Theory (Mel'cuk 2003). So for instance between *cows* and *moo*, we can recognise a single dependency which is classified as 'subject' and carries a wide assortment of information about the two words and their relationship. Of course this is not to deny that a word is different both from its meaning and from its realization in morphology; even lumpers should not be tempted to blur these distinctions. But these other levels of structure are among the typical properties that can be predicted from the syntactic dependency: one dependency, many properties. The kind of analysis I have in mind can be seen in Figure 4, where once again I use a non-standard notation for DS which I justify below. The main point about this diagram is that the relation labelled 'subject' allows the prediction ('inheritance') of at least three very different properties:

- the semantic relation labelled 'actor'
- the word order
- the number-agreement.

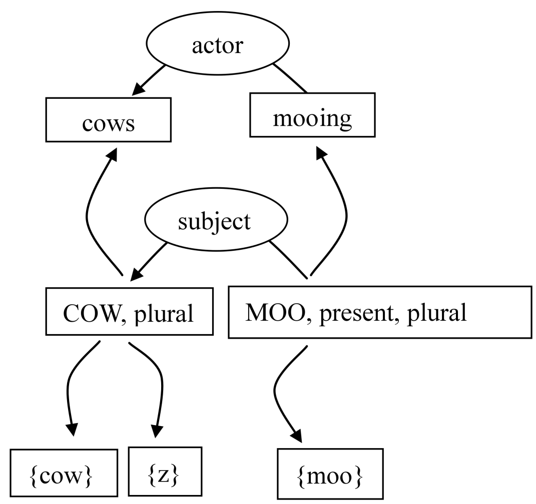

Figure 4: Syntax, semantics and morphology

5 Mutual dependency

Another question for DS theory is how rich DS is; and the answer that I shall suggest will also explain why I use non-standard notation. The standard answer is that DS is about as rich as very elementary PS – in short, very poor. This is the assumption behind the early arguments that DS and PS are equivalent (Gaifman 1965), but of course there is no reason to accept the assumption; indeed, what we know about cognition suggests just the opposite. If our minds are capable of representing complex social structures, then why should the same not be true of syntactic structures?

Take the case of mutual relations such as the relations between me and my father (whereby he is my father and I am his son). All the standard assumptions about syntax rule out the possibility of mutual dependency, but as Mel'cuk comments, mutual government clearly does exist in some languages (Mel'cuk 2003). For example, a Russian numeral such as *dva,* 'two', requires a singular genitive noun, but its gender is determined by the noun; so in *dva stola*, 'two tables', *stola* is genitive singular because of *dva*, but *dva* is masculine because of *stola*. More familiar data confirms this conclusion. Consider (2).

(2) Who came?

Who clearly depends, as subject, on *came*, in just the same way that *cows* depends on *moo* in (1). But the reverse is also true: *came* depends on *who* by virtue of the latter being an interrogative pronoun. This is what allows *who came* to depend on a verb such as *wonder*:

(3) I wonder who came. (compare: *I wonder cows moo)

Moreover, English interrogative pronouns allow ellipsis of the rest of the clause, as in

(4) Apparently someone came; I wonder who.

Facts such as these show strongly that interrogative pronouns (such as *who*) take a finite verb (such as *came*) as an optional complement. But we also know that *who* depends on *came*, so we have a very clear case of mutual dependency.

Mutual dependency cannot be shown in any standard notation, whether for PS or for DS, because these notations all use the vertical dimension for dominance. The problem is that mutual dependency means mutual dominance, and verticality does not allow two objects each to be higher than the other. This is why I prefer in WG to use arrows, where the direction of dominance is shown by the arrow-head (which more generally distinguishes values from arguments). In this notation, then, the structure of (2) is shown in Figure 5.

6 More about cognition: logic and tokens

We now return to consider another feature of general cognition: **node-creation**. This is the idea that we create mental nodes to represent the tokens of ongoing experience (which psycholo-

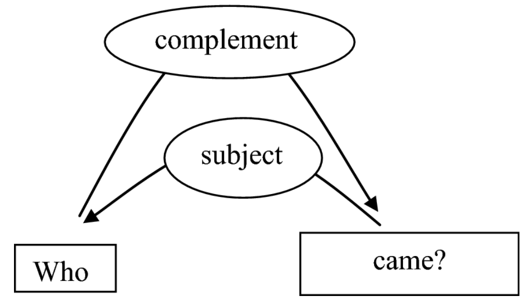

Figure 5: Mutual dependencies

gists call 'exemplars'). For example, when I see an object in the sky, I first create a token node for that object and then try to enrich it by linking it to some stored node (what linguists call a 'type'), such as the 'bird' node from which it can inherit further properties. The token needs a node to itself, most obviously at the point in processing where it hasn't yet been assigned to a type. Moreover the token has properties of its own, such as its unique position in space and time. Because no single node can carry two different sets of properties, we must create a token node which will eventually be classified by an is-a link to some type which effectively integrates the token temporarily into the total network.

This system for handling tokens by creating temporary nodes may seem rather obvious and trivial, but it has important ramifications for my argument below so it is worth pursuing a little.

- The main consequence is that one token may act as supercategory for another; for instance, suppose I see a small yellow bird, for which I create node A1, and then I see another one, and create node A2. The very act of recognising A2 as 'another one' means that I register its similarities to A1, with A1 as the supercategory for A2, and I can recognise this link even if I don't know how to classify A1. The same is true whenever we create one token as a copy of another (as in games such as 'Follow my leader', where everyone does the same as the leader). Thus two distinct objects or events may be linked by is-a even though they are both only temporary tokens.
- But multiple tokens are possible even for single objects or events. For example, suppose I create node B for a rather nondescript brown bird which I can't classify, and then, minutes later, I see another bird of similar size hopping around near the first bird, for which I create node C. From its colour I know that C is a blackbird, so I assume that B is its mate, and is

also a blackbird; but I can also remember my original failure to classify B, so I need a separate node for the newly classified B, which we may call B*. We might say that blackbird C has 'modified' B into B* – an example of one concept's properties being affected by those of a related concept.

- Another possibility is where we predict one token as part of the enrichment for another token. For example, suppose I see a duck swimming in a pond, and wonder where its nest is. This mental operation presupposes two nodes, D1 for the duck and N1 for its nest. Now suppose I think the typical relation between a duck and its nest is for the duck to be sitting on the nest; thanks to default inheritance, I expect D1 to be sitting on N1. But of course this is wrong, because D1 is actually swimming in the pond. I then spot a nest N2 with another duck D2 sitting on it, and, putting two and two together, I work out that D1 is D2's mate, and N2 is their shared nest. In other words, the expected N1 (the nest I expect D1 to be sitting on) is actually D2, which is in the expected relation to D2 but not to D1. Once again, default inheritance provides precisely the right analysis if we recognise N2 as a 'subcategory' of N1 – the actual nest that N1 was meant to anticipate.

All these examples are brought together in Figure 6, where the greyed boxes indicate permanent types and the others are temporary tokens. The main point of this figure is to show that an is-a relation is possible between one token and another, as in A1-A2, B-B* and N1-N2.

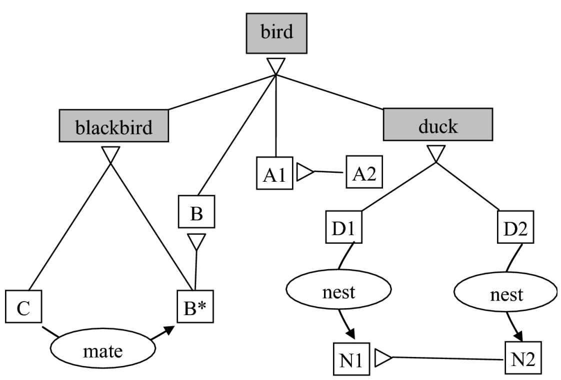

Figure 6: Tokens as supercategories

7 Structure sharing, raising and lowering

Returning to syntax, let's assume that the mental resources we can apply to birds are also available for words. Let's also assume, with Tree Adjoining Grammar, that a dependency grammar consists of 'elementary dependency trees anchored on lexical items' (Joshi and Rambow 2003). For example, by default inheritance the word token *moo* has a subject, in just the same way that a duck has a nest, and in processing this bit of experience we have to identify the expected subject or nest with some other token. And of course in both cases the expected token has a 'valency' of its own: the nest needs an owner, and the noun needs a 'parent' word to depend on. In fact, just the same process lies behind the classification of the tokens: so each token starts with an unknown supercategory which has to be identified with a known type. The little grammar in Figure 7 shows these identifications by the '=' linking the expected but unknown '?' to its target.

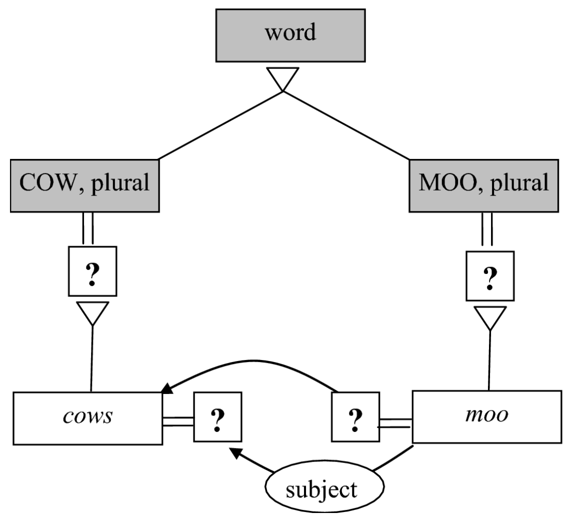

Figure 7: A grammar for Cows moo

This much is probably common ground among DS grammarians. But an important question arises for DS theory: how many parents can one word have? Once again, the standard answer is very simple: one – just the same answer as in PS theory, where the 'single mother condition' is widely accepted (Anderson 1979, Zwicky 1985). But syntactic research over the last few decades has produced very clear evidence that a word may in fact depend on more than one other word. For example, 'raising' structures such as (5) contain a word which is the subject of two verbs at the same time.

(5) It keeps raining.

In this example, *it* must be the subject of *keeps* – for example, this is the word that *keeps* agrees with. But equally clearly, *it* is the subject of *raining*, as required by the restriction that the subject of the verb RAIN must be *it*. Some PS theories (such as HPSG) allow 'structure sharing', which is equivalent to recognising two 'mothers' (Pollard and Sag 1994:4); and this has always been possible in WG. Once again, the arrow notation helps, as in Figure 8 (which I am about to revise):

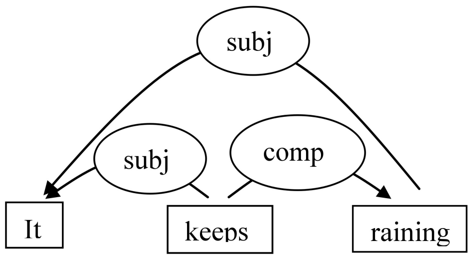

Figure 8: Raising

DS provides a very good framework for discussing structure sharing because it reveals a very general 'triangle' of dependencies which recurs in every example of structure sharing: three words connected by dependencies so that one of the sharing parents depends on the other. In this example, *it* depends both on *keeps* and on *raining,* but *raining* also depends on *is*. We might call these words the 'shared' (*it*), the 'higher parent' (*keeps*) and the 'lower parent' (*raining*).

But the existence of two parents in structure sharing raises a problem. What happens if their parental 'rights' conflict? For example, since *it* depends on *raining*, these two words ought to be next to each other, or at least not separated by a word such as *keeps* which does not depend on either of them; but putting *it* next to *raining* would produce **Keeps it raining*, which is ungrammatical. The general principle that governs almost every case of structure sharing is that the higher parent wins; we might call this the 'raising' principle. But how can we build this principle into the grammar so that raising is automatic?

The answer I shall offer now is new to WG, and builds on the earlier discussion of tokens in cognition, where I argued that one token may take another token as its supercategory. It also develops the idea that each token inherits a 'typical' underlying structure such as the 'tectogrammatical' representations of Functional Generative Description (Sgall and others 1986). Suppose that both the verbs in *It keeps raining* inherit a normal subject, which by default should be next to them: *it keeps* and *it raining*. But suppose also that the two *it*'s are distinct tokens linked by is-a, so that *it,* the subject of *keeps*, is-a *it*,* the subject of *raining*. Formally, this would be just like the relation between nest N2 and nest N1 in Figure 6, and the logic of default inheritance would explain why *it1* wins in the conflict over position in just the same way that it explains why N2 is under a duck but N1 isn't.

This answer requires a change in the analysis of Figure 8, which follows the tradition of WG (and also of other theories such as HPSG). In fact, if anything it is more similar to a Chomskyan analysis with 'traces', where the trace shows the expected position. But unlike the Chomskyan analysis, this does not involve any notion of 'movement'; all it involves is the ordinary logic of default inheritance. The structure I am now suggesting is shown in Figure 9.

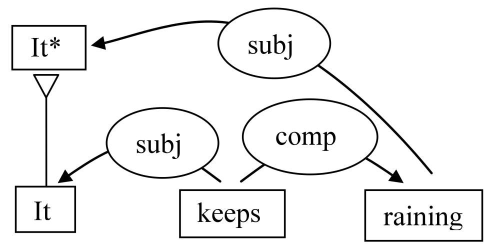

Figure 9: Raising and the raising principle

Why do languages and their speakers prefer raising to its opposite, lowering? I believe there is an easy functional explanation if we think of grammatical dependencies as tools for providing each word with an 'anchor' in the sentence which is in some sense already more integrated than the dependent. Clearly, from this point of view the higher parent must be more integrated than the lower parent, so it provides a better anchor for the shared. I think we can rely on this functional explanation to explain why our linguistic ancestors developed the raising principle and why we so easily learned it; so there is no need to assume that it is innate.

Which is just as well, because there are clear exceptions to the raising principle. In some languages, there are constructions where it is the lower parent that wins – in other words, cases of 'lowering'. For example, German allows 'partial VP fronting' as in (6) (Uszkoreit 1987, Haider 1990).

(6) Eine Concorde gelandet ist hier noch nie.
'A Concorde hasn't yet landed here.'

There is overwhelming evidence that *eine Concorde* is the subject of both *gelandet* and *ist*, but it is equally clear that it takes its position from the non-finite, and dependent, *gelandet* rather than from the finite *ist*. In this case, then, the expected raising relation is reversed, so that the subject of the lower parent is-a that of the higher parent, and the lower parent (*gelandet*) wins.

Moreover, German isn't the only language with lowering. Sylvain Kahane has drawn my attention to apparently lowered examples in French such as (7), which are easy to find on the internet.

(7) Avez-vous lu la lettre qu'a écrite Gilles à Pauline? 'Have you read the letter which Gilles wrote to Pauline?'

The important thing here is that *Gilles* is the subject of both the auxiliary *a* ('has') and its complement, the verb *écrite* (written), but it takes its position among the dependents of the latter, which is the lower parent.

It would seem, then, that sharing usually involves raising, but can involve lowering; and if raising has functional benefits, then presumably lowering also has benefits, even if the attractions of raising generally win out. And of course the two patterns can coexist in the same language, so we may assume that learners of German and French can induce the generalisation shown in Figure 10, with the general raising pattern shown as the A-B-C-A* configuration, and the exceptional lowering one as D-E-F-D*.

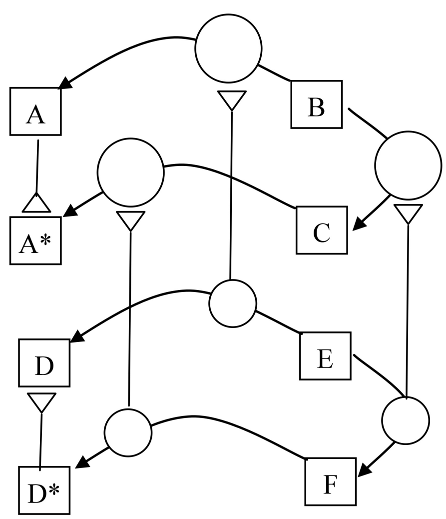

Figure 10: Raising and lowering

Once again, the main point is that we can analyse, and perhaps even explain, the most abstract of syntactic patterns if we assume that the full apparatus of human cognition is available for learning language.

One of the challenges for the very 'flat' structures of DS is to explain examples like (8) (Dahl 1980).

(8) typical French house

The problem here is that a DS analysis treats both *typical* and *French* as dependents of *house*, so there is no syntactic unit which contains *French house* but not *typical*; but the meaning does involve a unit 'French house', because this is needed to determine typicality: the house in question is not just French and typical (i.e. a typical house), but it is a French house which is like most other French houses. This phenomenon is what I have called 'semantic phrasing' (Hudson 1990:146-151), but I can now offer a better analysis which builds, once again, on the possibility of multiple tokens for one word.

This problem is actually a particular case of a more general problem: how to allow dependents to modify the meaning of the 'parent' word (the word on which they depend) – for example, how to show that *French house* doesn't just mean 'house', but means 'French house'. In PS analysis, the answer is easy because the node for the phrase *French house* is different from that for *house*, so it can carry a different meaning. I anticipated the solution to this problem in section 6 when I was discussing the case of the unclassifiable bird turning out to be a blackbird. In that discussion I said that the male bird 'modified' the classification of the other bird, deliberately using the linguistic term for the effect of a dependent on the meaning of its parent.

Suppose we assign the word token *house* not one node but two, just as I suggested we might do with the female blackbird. One node carries the properties inherited directly from the type HOUSE, including the meaning 'house', and the other, which of course is-a the first, shows the modifying effect of the dependent *French*, giving the meaning 'French house'. My suggestion is that modification works cumulatively by creating a new word token for each dependent. If this is right, then we have an explanation for *typical French house*, because 'French house' is the meaning which *typical* modifies. One challenge for this analysis is to find suitable names for the word tokens, but there is a simple solution: to name each token by a combination of the head word and the dependent concerned: *house* – *house+French* – *house+typical*.

This multiplication of word tokens would also explain many other things, such as why the anaphoric ONE can copy either meaning of a phrase such as *French house* as in (9) and (10).

(9) He bought a French house last year and she bought a German <u>one</u> [= house] this year.

(10) He bought a French house last year and she bought <u>one</u> [= French house] this year.

Once again the challenge for DS is how a single word token (*house*) can simultaneously mean either 'house' or 'French house'. But if *house* and *house+French* are actually different tokens, the problem disappears. Moreover, this example also reminds us that anaphora essentially involves the copying of one word token's properties onto another – in other words, an is-a relation between two word tokens, further evidence that one token may act as a supercategory for another. The relations in (9) and (10) are displayed in Figure 11.

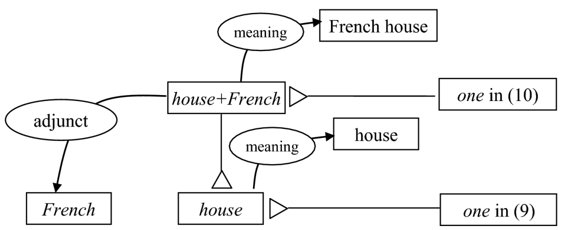

Figure 11: Anaphora between word tokens

This general principle has the interesting consequence of greatly reducing the difference between DS and PS. Both analyses assign an extra node to any word for each dependent that that word has, and assign to that node the modified meaning as affected by the dependent. The similarities are illustrated in Figure 12.

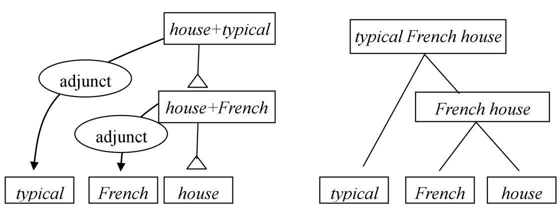

Figure 12: DS (WG style) and PS

Nevertheless, important differences remain: DS allows structures that are impossible in PS, including mutual dependencies, and conversely, PS allows structures that are impossible in DS, including not only unary branching but also exocentric constructions (even if these are excluded by the X-bar principle – Jackendoff 1977). And most importantly, the relevant relations are logically very different: the whole-part relation in PS, and the supercategory-subcategory relation in DS.

References

Anderson, John 1979. 'Syntax and the single mother', *Journal of Linguistics* 15: 267-287.

Barlow, Michael and Kemmer, Suzanne 2000. *Usage based models of language*. Stanford: CSLI.

Chomsky, Noam 1986. Knowledge of Language. Its nature, origin and use. New York: Praeger.

Dahl, Östen 1980. 'Some arguments for higher nodes in syntax: A reply to Hudson's 'Constituency and dependency'.', *Linguistics* 18: 485-488.

Eppler, Eva 2010. *Emigranto: the syntax of German-English code-switching.* Vienna: Braumueller Verlag.

Fodor, Jerry 1983. *The Modularity of the Mind*. Cambridge, MA: MIT Press.

Gaifman, Haim 1965. 'Dependency systems and phrase-structure systems.', *Information and Control* 8: 304-337.

Gisborne, Nikolas 2010. *The event structure of perception verbs.* Oxford: Oxford University Press.

Haider, Hubert 1990. 'Topicalization and other puzzles of German syntax.', in Gunther Grewendorf & Wolfgang Sternefeld (eds.) *Scrambling and Barriers*. Amsterdam: Benjamins, pp. 93-112.

Halliday, Michael and Matthiessen, Christian 2006. *Construing Experience Through Meaning: A Language-based Approach to Cognition.* London: Continuum.

Hudson, Richard 1984. *Word Grammar.* Oxford: Blackwell.

Hudson, Richard 1990. *English Word Grammar.* Oxford: Blackwell.

Hudson, Richard 2007. *Language networks: the new Word Grammar*. Oxford: Oxford University Press

Hudson, Richard 2010. *An Introduction to Word Grammar*. Cambridge: Cambridge University Press.

Hyman, Larry 2008. 'Directional Asymmetries in the Morphology and Phonology of Words, with Special Reference to Bantu.', *Linguistics* 46: 309-350.

Jackendoff, Ray 1977. *X-bar Syntax: A Study of Phrase Structure.* Cambridge, MA: MIT Press.

Joshi, Aravind and Rambow, Owen 2003. 'A Formalism for Dependency Grammar Based on Tree Adjoining Grammar.', in Sylvain Kahane & Alexis Nasr (eds.) *Proceedings of the First International Conference on Meaning-Text Theory*. Paris: Ecole Normale Supérieure.

Keenan, Edward 1976. 'Towards a universal definition of 'subject'.', in Charles Li (ed.) *Subject and Topic*. New York: Academic Press, pp. 303-333.

Labov, William 1972. *Sociolinguistic Patterns*. Oxford: Blackwell.

Mel'cuk, Igor 2003. 'Levels of Dependency in Linguistic Description: Concepts and Problems', in V Agel, Ludwig Eichinger, Hans-Werner Eroms, Peter Hellwig, Hans Jürgen Heringer, & Henning Lobin (eds.) *Dependency and Valency. An International Handbook of Contemporary Research, vol. 1*. Berlin: de Gruyter, pp. 188-229.

Percival, Keith 1976. 'On the historical source of immediate constituent analysis.', in James McCawley (ed.) *Notes from the Linguistic Underground.* London: Academic Press, pp. 229-242.

Pollard, Carl and Sag, Ivan 1994. *Head-Driven Phrase Structure Grammar*. Chicago: Chicago University Press.

Reisberg, Daniel 2007. Cognition. Exploring the Science of the Mind. Third media edition. New York: Norton.

Robins, Robert 1967. *A Short History of Linguistics*. London: Longman.

Sgall, Petr, Hajicová, Eva, and Panevova, Jarmila 1986. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Prague: Academia.

Uszkoreit, Hans 1987. 'Linear precedence in discontinuous constituents: complex fronting in German.', in Geoffrey Huck & Almerindo Ojeda (eds.) *Discontinuous Constituents (Syntax and Semantics 20)*. San Diego: Academic Press, pp. 405-425.

Zwicky, Arnold 1985. 'The case against plain vanilla syntax', *Studies in the Linguistic Sciences* 15: 1-21.