Linguistic Applications of Default Inheritance Mechanisms*

Gerald Gazdar

September 1985

Cognitive Science Research Paper

Serial No. CSRP 070

Cognitive Studies Programme,
The University of Sussex,
Brighton BN1 9QN

To appear in: Peter J. Whitelock et al., eds. *Linguistic Theory and Computer Applications*. London: Academic Press, 1987.

*This is a lightly edited transcript of a talk (and the subsequent discussion) given at the UMIST Alvey/ICL Workshop on Linguistic Theory and Computer Applications in Manchester in September 1985. The transcript was made from videotape by Peter J. Whitelock. An antecedent of the talk was given at the Syntax Research Center, University of California, Santa Cruz in May 1985. I am grateful to the locals on that occasion for their comments and to Kelly Booth, Alex Borgida, David Israel, Rich Thomason and Meg Withgott for relevant conversations. The activity from which this paper emerges was supported by a grant (C00242003) from the ESRC (UK) and was greatly facilitated by the tenure of a Fellowship at the Center for Advanced Study in the Behavioral Sciences through 1984-85 and by visitor status at the Center for the Study of Language and Information at Stanford over the same period.

Linguistic applications of default inheritance mechanisms

Gerald Gazdar

University of Sussex

I'm glad I came after Stuart [Shieber], because he says so much more eloquently some things I might otherwise have stumbled through. I will therefore not make my metatheoretical preamble quite as long as it might otherwise be. What I'm going to do is begin with a few remarks about relations between disciplines, in particular the relation between linguistics and computational linguistics. Then I'm going to look at a particular topic, the topic of defaults and more especially default inheritance, to illustrate the rather slight metatheoretical point that I'll begin with. Then depending on time at the very end I might, or might not, talk about a particular approach to default inheritance that I've been thinking about, but I may not get to that. So there are basically three chunks, the third of which is optional, the first of which is fairly brief I trust.

It seems to me that during the 70's, at least in this country and I suspect elsewhere, certainly in the States, touched as it was by the Sloan foundation, there was a notion of interdisciplinarity which was rather a naive one. This was that you just bunged together disciplines (within some sort of domain, like philosophy, linguistics, psychology) and that this was a good thing and that they would have things to say to each other. The idea was that you somehow forced people (with the aid of large cheques) to sit in the same room and eventually these people would end up in reciprocal intellectual relationships. I think with the wisdom of hindsight that was silly in many ways. There's no reason why the relations between disciplines, perfectly sensible disciplines with their own right to exist, should always be reciprocal. So to take an extreme case, mathematics: most sensible disciplines relate to mathematics, in that they pinch stuff from it. Social sciences help themselves to large chunks of statistics, linguistics and computer science invariably help themselves to other chunks of maths, and so on with physics and chemistry. Of course there's some feedback in that some areas of maths are developed because of applications and they then turn out to be intrinsically interesting and so on. But basically the drain of knowledge is from maths into these other disciplines. But nobody thinks the worse of physics, say, because it uses a whole lot of maths.

When we come closer to home and look at linguistics and psychology, for example, there doesn't seem to be any very obvious reason why linguistics should get anything from psychology. Now this would have been a heretical thing to have said 10 or even 5 years ago and I don't really want to pursue it here. Whereas the converse, for example, in some sense has to be true, since language is such an important aspect of being human. Psychologists are concerned with what being human is about - there's necessarily going to have to be a psychology of language. What is language? Well, you ask a linguist what language is, at least you ought to be able to.

I want to actually focus now on linguistics and computational linguistics. Having begun in this rather cynical fashion, you may expect me to say something that I'm actually not going to say. It seems to me that a conventional view of the relation between linguistics and computational linguistics would go something like this. Linguists are doing the science of language and they develop grammar formalisms/theories of language, modulo all the things that Stuart [Shieber] said. They do analyses of language, descriptions of language. Computational linguistics is a branch of engineering and they borrow these descriptions and these grammar formalisms and implement them - they hack away. The transfer of knowledge is entirely from linguistics to computational linguistics; there's no particular point in any feedback the other way round.

I suspect there are large numbers of theoretical linguists, at least, who would regard this as the proper picture of the relationship between linguistics and computational linguistics. Now put the way I've just put it, probably not many computational linguists would subscribe to it, though they might have to think for a while to rebut it. And if I put it in a less pejorative fashion, then maybe I could have got some of you to subscribe to it. But I think it's all wrong, and I think it's all wrong because of a misconception about the way linguistics really is, caused by one of the most effective aspects of linguistics, namely its public relations. Its public relations has it that this is what linguists do, they devise grammatical theories = grammatical formalisms, and they describe languages by grammars and so on. But really they don't do these things, for the most part, as people really know know when they think it through. Look at what the majority of theoretical syntacticians are doing at the moment in the US, the numerical majority, not the schools of syntax represented in this room. They're not working on formal theories of language, they're not developing grammatical formalisms for natural language, they're not doing grammatical description of fragments of natural language, they're not writing grammars. None of those things are being done.

Syntax, the core tradition of theoretical linguistics, has been oriented round "cute" facts for a very long time. Where you score points as an American syntax graduate student, is by first finding "cute" facts, and then developing a "cute" analysis of them. The constraints on a "cute" analysis are very slight. You're not required to write it in a particular formalism. Of course, you may well be required to have some sort of framework allegiance, but that doesn't particularly constrain what apparatus you can use. Indeed the whole nature of your "cute" fact paper may be to argue for some previously unknown bit of apparatus. If one has this (what I take to be the) conventional perception of the relation between linguistics and computational linguistics, then computational linguists are going to find themselves waiting and getting irritated and reading linguistics literature and scratching their heads and saying "Gee where is this stuff, maybe I'm reading the wrong journal but they told me that *Linguistic Inquiry* was the journal to look in, where's this grammar?".

On the contrary it seems to me that, certainly now, the situation is that grammatical formalisms are being developed not by linguists but by computational linguists, or by linguists in collaboration with computational linguists. And for a very good reason. If you're doing computational linguistics you'd better have linguistics formalism. If there's some moral of the last 20 years of computational linguistics it's that you don't just want to mess up your grammar and your phonology in the code. You want to pull it out for good software methodology type reasons. You need some sort of formalism and so you've got to develop one. What Stuart [Shieber] referred to as theoretical grammar formalisms - he mentioned three, one of them I don't believe exists - of the two that I do believe exist, I think it's no coincidence that one of them, LFG, one of the two progenitors of that sitting right here [Ron Kaplan] certainly comes from the computational linguistics side of things. Likewise in GPSG, although historically, and continuingly to some extent, the four authors of the GPSG book (Gazdar et al 1985) are linguists by training, it's again no accident that that book is as formal as it is and that two of those authors have for three or four years now been consultants on a major computational linguistics project run by Hewlett Packard. I doubt somehow that that book would have been the way it is - it would have been much more cute fact oriented - had it not been for that.

So the rather slight metatheoretical, sociological, moral I want to draw from this is that actually computational linguistics should see itself, since linguistics is not doing the job, as having a theoretical branch, what we might call theoretical computational linguistics, by analogy with theoretical computer science. Now one's familiar with the notion of theoretical computer science, people who do theoretical computer science are a bunch of mathematicians, who have no particular need to touch computers though they may do so. Likewise it seems to me we can have a discipline, or sub-discipline, of computational linguistics which we might choose to call "theoretical computational linguistics" which has much the same character. And at the present time, of course, people doing theoretical computational linguistics are also doing practical, engineering type, computational linguistics, and that's quite reasonable, but some time in the future you might see the sort of specialisation of activity that I believe you see in computer science. I think this would be healthy for linguistics though linguists may not like it.

One of the reasons is that linguists are not trained to develop formalisms, to think about formalisms having semantics, to know about a range of alternative formalisms, formal languages and the like. Whereas people with some computational training are oriented to thinking in that way. The consequence of this is that there are a lot of formal techniques available in the computational linguistics literature, which it seems to me make good, or potentially good, sense from the perspective of theoretical linguistics. Theoretical linguistics has been extremely conservative about the form of mechanisms it's used. It's been promiscuous and profligate, but it's also been conservative. A lot of it's been tied to thinking about strings and operations that map strings to strings because historically that's where generative grammar came from. But of course formal language theory has moved way beyond the early days of grammars thought of in terms of strings. Most linguists don't know that, and are unfamiliar with, for example, tree grammars and the like.

I want to illustrate this point by talking about defaults. The illustrative side of this is largely to do with default inheritance. But I'll begin by just saying a few words about default mechanisms in general. Now these have shown up in linguistics. (1) shows something from Chomsky & Halle (1968).

(1) Default values of phonological features

```
[u high]   →   [+ high]
[u nasal]  →   [- nasal]
[u low]    →   [- low]
[u ant]    →   [+ ant]
[u cor]    →   [+ cor]
[u cont]   →   [+ cont]
```

The *u* notation says that the unmarked value for the high feature is "+high" and so on down the list. Now these are one type of Chomsky & Halle markedness conventions. What they are telling you is the default values, the unmarked values, of these phonological features. There are one or two asides that I could perhaps make - notice this arrow, familiar arrow, familiar bit of notation, but of course really quite different semantics from, say, the rewrite arrow. Although you could presumably construe all that in a rewriting system, but it hardly seems to be an appropriate way to think about it.

Now these notions of markedness, like the rest of phonology feature theory, have found their way into syntax. You quite often see in the syntax literature (all syntax literature, I don't just mean GPSG) remarks of the form "the default values of the feature PRO is minus", or "PRO defaults to minus", or whatever. What you don't see is actually a theory about what that might mean or any kind of maths to go with such claims. However in doing the GPSG book (Gazdar et al 1985) we were forced, or at least we felt obliged to make extensive use of features in that book. Perhaps it would be more accurate to say we made explicit our extensive use of features in that book. In doing so we constantly wanted to have default values of features to save endless redundant specifications of things that were obviously going to have such and such a value. And we had to develop a theory of defaults. Now this actually turned out to be very difficult to do, and I will say a word or two about why.

It's not too difficult to develop a theory of defaults if you restrict yourself to defaults that look roughly like FSDs 1. 2, 3 and 4, in (2).

(2) Feature specification defaults

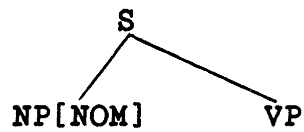
```
PSD l:  [-INV]
PSD 2:  ~[CONJ]
PSD 3:  -[NULL]
FSD 4:  ~[NOM]
FSD 5:  [PFORM]  -»  [BAR 0]
FSD 6:  [+ADV]  -»  [BAR 0]
FSD 7:  [BAR 0]  -»  ~[PAS]
FSD 8:  [NFORM]  -»  [NFORM NORM]
FSD 9:  [INF.+SUBJ]  -•  [COMP for]
FSD10:  [+N.-V.BAR 2]  s  [ACC]
FSDll:  [+V.BAR 0]  -»  [AGR NP[NFORM NORM]]
```

The first one says something like the default value for the feature INV (for inverted sentences) is the value "minus". The second one says the default state for the CONJ feature is for there to be no specification at all for the feature. The details of this don't matter. Even with this relatively simple kind of default there is one problem you get into, which is if you believe that grammatical formalisms should be declarative then that of course commits you to not having any ordering; yet a very natural way of thinking about defaults is to say "you do everything else and then you just paste the default values on at the end". But it you've got a declarative formalism you can't really say that, at least not just as I said it. You have to come up with some declarative version.

However, the real difficulty we got into was the fact that when you actually articulate a large feature system, the defaults you want are in general not straightforward ones like FSDs 1,2,3 and 4 in (2). You want to say things about the default state of particular configurations of features. So you want to say "when this feature's got this kind of value then this other feature defaults to something ot other". So it's not just simple cases of a single feature defaulting to some value or another. You've got configurations or combinations of features that have, as it were, default states. Getting a theory of this together drove us, or me, mad, for a while. It was a painful process. Of course, once one's done it one can look back and think "well, that wasn't so hard", but actually getting away from thinking about the single case to thinking about these multiple cases was very difficult. What you actually have to move to is a view where defaults are not crudely properties of particular features but they're properties of categories. So categories try and satisfy the Boolean things in (2). I don't want to got through the GPSG theory of defaults. I just want to give you a sense of why defaults are a kind of non-trivial issue.

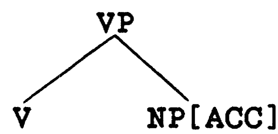
Just to elaborate that point slightly further. Take something like the default [CASE ACC], which probably anybody would want for a treatment of English. You want to say the default case in English is accusative, assuming that there is an accusative/nominative case distinction in the English pronoun system. You're likely to want to say that whatever your grammatical framework is, assuming you have something to say about case. Look what's going to happen. If we take a phrase structure grammar, then in the case of the tensed sentence rule (3), we just want to stipulate that the subject is nominative.

(3) S → NP[NOM] VP



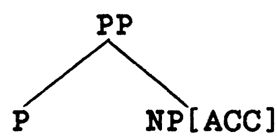
Our theory of defaults is going to have leave this nominative, not overwrite accusative on it. That shouldn't be too difficult, or at least if you can't do that you can't do anything. (4) is the verb phrase rule

(4) VP → V NP



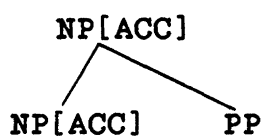
We don't want to say anything about this object NP, it ought to default to accusative. This is precisely why we want a default. We want to get from that rule to a bit of structure like that. Likewise in the case of (5).

(5) PP → P NP



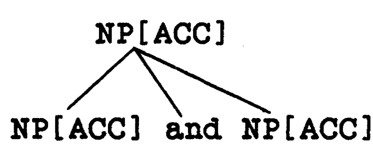
What's going to happen though in cases like (6)? We don't want this thing to default to accusative, because it it does, then we'll never get this construction in sentence subject position.

(6) NP → NP PP



There is, of course, just a bizarre claim about English - this construction occurs in subject position just as easily as it occurs in any other position. Likewise in the case of coordinate NP's in (7) - again the details don't matter, I don't believe that's the right constituent structure.

(7) NP → NP and NP



We certainly don't want this thing defaulting to accusative. So in our theory of defaults, at least in the GPSG version of it, we have to have some notion of things co-varying. We've actually got three cases here - case (3), where the default is just straightforwardly violated or contradicted; cases (4) and (5), with the default operating as we want; and cases (6) and (7), in which we've got a situation where bits of the structure co-vary and we don't want the default to apply.

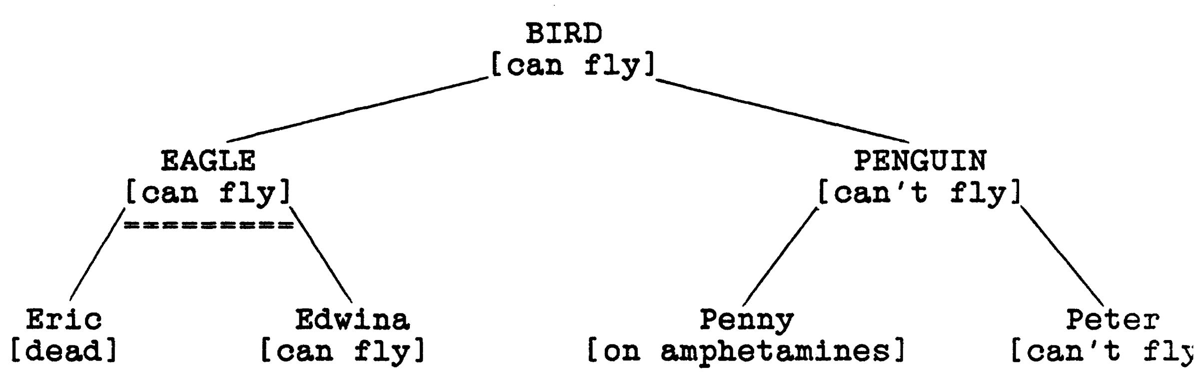
Working out a theory that has those properties is tricky and here's the kind of problem you get into. I said that when things co-vary the default doesn't apply. But here's a case where you do want it to apply. You want to say that the default state of a passive feature is its minus value - things are not passive unless you say they are. The verb-phrase rule (4) gives us a transitive verb and an NP. The VP and the V will co-vary in respect of a feature like passive, so it seems that they should be free of the default and we should be allowed to make them passive. If we're allowed to then we're going to get junk like (8).

(8) *Kim is eaten the chicken.

Those hurdles are jumped in the book. I mention them here only to convey the fact that there is a genuinely difficult problem.

Now, having talked about ordinary defaults in what I hope are fairly familiar kinds of cases, I want to talk about default inheritance. Unlike the notion of marked feature values, unmarked feature values, syntactic features defaulting to accusative and so on, this is a much less familiar notion within linguistics proper, although it's a very familiar notion in the AI literature. Indeed one AI professor said to me about a week ago that more than half of AI was predicated on default inheritance. I don't know enough about the AI literature to know whether that's a true appreciation of the situation. But it is certainly the case that inheritance, and default inheritance, are important in AI. So what do I mean by this, for those of you who are not absolutely au fait with the AI literature? (9) will be familiar to many, though it's nothing to do with linguistics.

(9) Network of properties



Fact 1. If you're dead you can't fly
Fact 2. If you take amphetamines, you can fly

We have this network here rooted in BIRD. BIRDs have a property, namely they can fly. EAGLE and PENGUIN are both types of BIRD. PENGUIN has a property, namely it can't fly. However, nothing's said about whether EAGLE can fly or not. The things that are underlined are not really there. However, since nothing's said about it, then it will just inherit from being a BIRD the property "can fly". Now down at the bottom here we've got Eric who's an EAGLE, only Eric's dead and we know a fact about the world, which is if you're dead you can't fly. So even though these are not superficially inconsistent, given other things that we know about the world, on a particular account of this which somebody might implement, hopefully Eric wouldn't inherit the "can fly" property. On the other hand, nothing's said about Edwina, so she can fly, and we have the kind of opposite case under PENGUIN. This is the sort of thing you will see if you dip into the AI knowledge representation literature. These things are known as "ISA links". Eric *isa* EAGLE, EAGLE *isa* BIRD. Actually, as people have pointed out, in that literature, there's a fudge going on, because there's two types of *isa* relation - EAGLE is a subset of a class of BIRD, whereas Eric is a member of the class of EAGLE. But that is not our concern at the moment.

What kind of applicability might that sort of default inheritance have within linguistics? My sense is that it actually has a good deal of applicability in ways that are interesting far beyond implementation issues. And as my first exemplar, I'll look at a treatment of Dutch verb morphology due to de Smedt (1984).

(10) is a net of essentially the kind we've just been looking at. The top node says VERB, which of course means Dutch verb. Dutch verbs have a bunch of properties - their past tense stem is equal to their present tense stem, their past tense suffix equals *+t or +d* and so on.

```
(10)            VERB
[past tense stem - present tense stem   ]
[past tense suffix - +te or +de/voice   ]
[past participle prefix - ge+           ]
[past participle stem - past tense stem ]
[past participle suffix - +t or +d/voice]

    /                       \
  verk                   MIXED VERB
 (work)      [past participle suffix - +en or +n/syll]
                  /                   \
                 bak               STRONG VERB
               (bake)        [past tense suffix - 0]
                                   /              \
                                zwem              ...
                               (swim)
                      [past tense stem - zvom]
```

Then there's a sub-class, MIXED VERBS, and also just simple members of the class VERB, like *werk.* This is just a verb - nothing else is said about it, so it's going to inherit all those properties. By looking at those properties and applying the rules implicit in those equations, you can infer what the past participle, past tense and so on of *werk* is. In turn, the sub-class MIXED VERBS has a property associated with it, that the past participle suffix is *en.* Of course, this contradicts the information associated with VERB and so overrides the default. The past participle of *bak* will be *gebaken.* On the same basis, we can work out the past participle of *zwern* is *gezwornrnen.*

Now I think that this is linguistically interesting. One of the reasons for this, apart from the fact that it provides a non-redundant and therefore generalisation-capturing representation of the Dutch inflexions, is that it accounts automatically for the phenomenon known to linguists as "blocking" - the existence of an irregular form in general stops the simultaneous production of a regular form. If you look at linguistic treatments of morphology, they normally acknowledge the existence of blocking, but normally it has to be stipulated. I'm not familiar enough with the literature to assert that it always has to be stipulated. But in general, the effect of blocking is stipulated as an extra constraint, which says that these regular rules may not apply when there is some more particular rule. There's a whole terminology - "proper inclusion precedence" and so on - which goes with that. Well, in the de Smedt case it's quite automatic - there's no mystery about why the past participle of *zwem* isn't whatever it would be if you followed the rules at the top there. It can't be, because of the way these equations get inherited down the tree. So the regular derivations are not an issue with this kind of knowledge representation.

I think this is an interesting consequence. It's not one, incidentally, that de Smedt pulls out of this, at least I don't think he does, which is a bit of a pity. De Smedt is primarily concerned with the implementation that he presents.

Essentially the same idea is used in Flickinger, Pollard and Wasow (1985), henceforth FPW, which is also about the lexicon. In that paper, they are not primarily concerned with inflectional morphology, but rather with subcategorisation frames. They propose a treatment of lexical organisation which inherits subcategorisation frames in just this fashion. They also incidentally do inflectional morphology the same way de Smedt does.

Both these two things I've referred to are implemented in object-oriented programming languages, and I'll make another remark about that in a moment. Just let me give you another couple of empirical cases in the morphology area, where this sort of default inheritance could provide a theory of things that have been suggested in the linguistics literature.

Looking at Marantz (1984), he proposes that features on affixes override those on the root, but that features on the root get through if there's no conflict. So again here, you have a claim about the way features on affixes and roots combine. I'm not concerned with the veracity of the claim, but you have a claim that can be naturally treated with a default inheritance mechanism. The kind of example he gives is that the passive affix is [-transitive], but it goes on [+transitive] roots. You don't want the thing that's formed to be [+transitive], you want [-transitive] to be inherited from the affix.

There's a paper by Halvorsen & Withgott (1984) on tone in Norwegian. Apparently in the Norwegian tone system, inflectional markers, for example the definite marker, can give a particular tone to an unaccented word. So you have a stem that's not accented, and you put on a definite marker, and then the whole resulting word has some tone, which it has inherited from the definite marker. The tone is not just on the definite marker, it's a global property of the word. However, this doesn't happen when the word already has a tone if its own. So if you have some stem that has an inherent tone, and you bung a definite marker on it, then the tone of the stem wins through. So this is actually the oppositive case to the Marantz case. In Marantz's case, the affix was winning; in the Norwegian tone case, the stem wins.

Turning from morphology and the lexicon to syntax - in the book version of GPSG, though not in earlier (published and unpublished) versions, the Head Feature Convention (HFC), which Stuart [Shieber] alluded to in passing, is treated as a default inheritance mechanism of a rather particular kind. The idea of the HFC is extremely simple, (and this is implicit in an awful lot of (non-GPSG) linguistics work) and is simply that phrases and their heads share various syntactic properties. One way of formalising that is to have features pass from lexical heads to phrasal nodes, carrying information up or down, or merely requiring identity of certain sorts of information. Previous formulations just required identity of relevant classes of features, but what the book version does is not require absolute identity, but rather says "pass up whatever features you can, but if passing something up would lead to an inconsistency on the mother node, then forget it, don't bother". Likewise, "pass things down if you can, but if you're going to get an inconsistency then forget it". So this is bidirectional default inheritance of features. There's an attempt by both mother and daughter, to put it anthropomorphically, to pass information to each other, and only the information that's consistent with what they already think gets carried across. How does this pan out?

What will happen in (11) is that everything associated with the mother category will get passed down into that H position, because there's no reason for it not to. So if the mother category is a finite S, then that H category will be a finite S as well.

(11) S - H, ADV

Likewise in (12), again nothing's going to stop that head having all the properties of its mother. So the property of being a VP. the property of being a bare infinitive (BSE), just get equated.

(12) VPtBSE] -+ H, ADV

Now look at (13)

(13) VP -• HtBAR 0] . . .

Here we have a VP with a lexical head, i.e. BAR 0, a lexical category immediately dominating a word. What happens if we have a bare infinitive mother category? There's nothing to stop this BSE feature going down, but this VP has some BAR level, e.g. 2. This is a feature that we passed down in the previous case, but we can't pass it down here, because it would be inconsistent with the head being a V[BAR O]. Given the way the mechanism works in the GPSG book, that's no problem. This is a default inheritance mechanism - since the bar level is already specified, the mother's bar level does not get handed down - to do so would give rise to inconsistency.

Now the nice feature of that is that you can then treat bar level as the sort of information that is propagated by the HFC. Previous treatments have not permitted that. Why is that a nice thing to do? You get a rather curious consequence, namely that recursion is the default situation with rules like this. So rules that introduce an adverb, like (11-12), by default get recursion.

Now of course that's in flat contradiction to what people like Jackendoff (1977) have said. But then constructions like this were a severe embarrassment to Jackendoff, so this turns Jackendoff on his head. Instead of saying that these are the bizarre, "out-in-left-field" constructions, this says that these are the primary ones. So that's one kind of nice consequence, at least it's nice if you think that's the right way to go.

The other nice consequence of treating information about whether something is a phrase or lexical as a head feature is that you then get its propagation automatically in coordination. The point here is that when you coordinate two noun phrases, what you get is a noun phrase; when you coordinate two verb phrases, what you get is a verb phrase; when you coordinate two sentences, what you get is a sentence. It is normally, possibly always, the case, that the bar level of the two or more daughter conjuncts is the same as that of the mother. If the properties of the daughter conjuncts are theirs simply in virtue of coordination being a multi-headed construction, then if the phrasal status of something is the sort of thing that is manipulated by the HFC, then that just falls out for free. That's another consequence.

I'm actually doing something I didn't intend to do at the moment. I'm trying to motivate this analysis, which I don't really want to do. Its main purpose here is illustrative of the use of a default mechanism within syntax. That analysis may be wrong, but it at least has some arguments going for it, and it is a default inheritance mechanism and its being used in a superficially very different kind of domain to the lexical domains I was just sketching.

Now I want to give a final case that is even further afield from what you might think these mechanisms would apply to. This is the theory of implicature and presupposition, that Scott Soames and I developed independently in the 1970's (Gazdar 1979, Soames 1979). I don't want to say anything about the merits of this theory. Indeed it's known to be inadequate in various respects. However, it can be seen, with the wisdom of hindsight, as a default inheritance theory.

As you're all probably aware, many words and construction types in natural language give rise to certain potential presuppositions. A verb like *stop* gives rise to the presupposition that something was going on, something that subsequently stopped. A verb like *regret*, as in (14) presupposes that I missed the workshop. Of course I haven't, so that would be an anomalous thing to say.

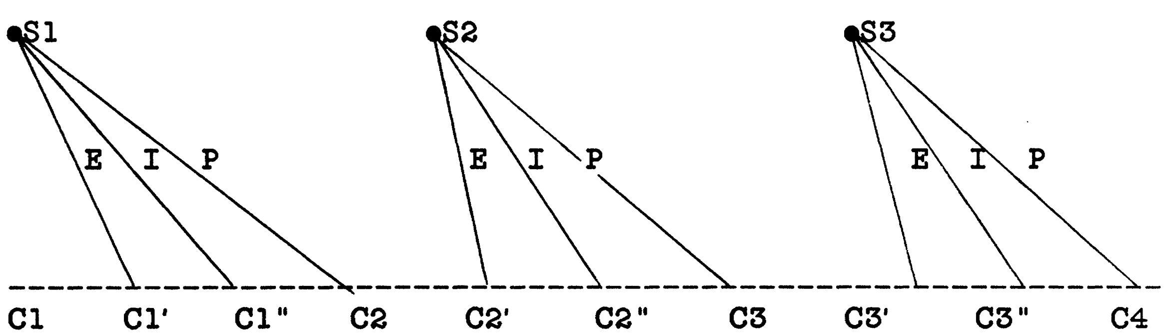
(14) I regret having missed the workshop.

Now these presuppositions get inherited. When you put a verb like *regret* deep inside some large and complicated sentence, the presupposition associated with *regret* may or may not be inherited by the whole sentence. This happens, not in any kind of random way, but in some completely systematic and apparently predictable way. There was a literature in the 1970s on how this worked. The same problem arises with implicatures (this bit of terminology coming from Grice (1975)). When you say (15), you're typically heard as implying that not all of them were.

(15) Some of the post-graduates were at the party.

If we embed that in some large and complicated sentence, does that still imply that not all of the post-graduates were at the party. Well, sometimes it does, and sometimes it doesn't. The Gazdar and Soames account of these phenomena was a theory about when it does and when it doesn't. The way they worked is shown in (16).

(16) Gazdar-Soames Pragmatic Theory



The line represents the passage of time - well it's really pseudo-time, bits of it are time, and other bits aren't. We're in some context C1 which we can take to be mutually accepted beliefs, or whatever your theory of context is, and we utter sentence S1. That moves us to a new context C1', in which all the entailments E of S1 are added to context C1.

This is a very idealised world in which there are no disagreements. Everything that is said is believed by everybody and so on. Our first step is to move to context C1' augmented with all the entailments. The next step (and this is what I mean by pseudo-time, since this is just a formal ordering, not a genuine temporal ordering) is to augment context C1' with all the implicatures that you can assemble on sentence S1. And by "assemble", I just literally mean pulling them all out and adding them together, not doing anything fancier than that. You add all those that are consistent with C1' in order to form C1". Then finally, (although of course there's no notion of process here, it's just static definition.) you add all the presuppositions P of S1 (again assembled just by lumping them all together) which are consistent with the context C1", which you've got by augmenting it with the implicatures. So then you arrive at C2 which is the context in which the next sentence S2 is uttered. Then you go through the routine again.

So it's a theory about how entailments, implicatures and presuppositions augment contexts, in a very idealised model - one speaker giving a monologue of utterly uncontroversial material. My point here is not to defend this or to elaborate it or anything, just to show it to you as an instance that can be seen in retrospect as a kind of default inheritance mechanism. And indeed there's a paper by Mercer & Reiter (1982) that does construe the presuppositional part of this theory in terms of a default logic. They make a point which I think is a point worth pulling out here. They comment on my work and say that what I did was basically ad hoc. They don't mean ad hoc in the linguist's sense, namely that there was no linguistic motivation for it. Their comment was addressed to the formal side of the theory. I developed a bit of formal apparatus that said what this diagram says. But the formal apparatus came out of nowhere. I just spent two years of my doctoral research time staring into space until I developed this thing. It doesn't relate to any other formal apparatus, it doesn't have any other use except this. So from the point of view of a logician it's ad hoc.

What's interesting about the Mercer & Reiter paper is that they show that you can reconstrue all this in terms of their default logic which can also be used to reconstrue the networks you've been looking at. There's another paper coauthored by Reiter, this time with Etherington (Etherington & Reiter, 1983), in which they discuss using Reiter's default logic to reconstruct the eagle/penguin/bird net I exhibited earlier (9).

So I think there is an advantage to be gained - namely avoiding the logician's sense of ad hoc - by taking this kind of perspective on various things that you might want to do with default inheritance mechanisms in linguistics - standing back and saying "maybe we can have a general theory of such and such a kind". But of course it's possible that we may not. It may be that the sort of default inheritance theory we need for this case is actually going to turn out to be formally different, interestingly formally logically different, from the sort you would need in the lexical case. It wouldn't be a priori surprising because this is such a very different domain. However, at least by standing back and seeing that they're similar allows us to ask the question, which we can't if one of them's buried in the pragmatics literature and the other is somewhere else.

Well I'm coming to the end and I'm not going to do my section three. However, I would like to say one or two things to take some of the positive edge off this. I think this is an illustration where linguists who are interested in these matters, but not at all interested in computational linguistics or engineering, could usefully look at the AI literature, for example, particularly the recent AI literature, on networks - stuff like Etherington & Reiter, work by Touretsky (1986) and Brachman (1985) and so on. There's also more exotic, but closely related, work on default logic - by Mercer and Reiter (1982), Bob Moore (1983) (who's here), McDermott & Doyle (1980), and so on, which ties up with this.

When you go and look at that literature, as I've been doing recently, you realise somewhat to your distress, in this case my distress, that actually this stuff is not anywhere near as fully worked out as one would like. What one would really like to do is go to that literature, pick out ready made formal theories of this stuff and import it into linguistics. But you can't really do that. It appears to have been the case, and there are people here probably in a position to correct me if I'm wrong, that until at least 1980 people who were using the network notations were really deeply confused in various fundamental ways about what they were doing and what the status of it was. As far as I can tell, much of the recent literature has been attempting to sort out the confusion and to make sense of what had been done.

Now there's a bit of a moral here. It's a moral that's not one that needs to be drawn for people with a computational background, but I think does for linguists. Just because somebody's got an implementation of something doesn't mean that they've got a a theory of it. Unfortunately the cases I've given you, de Smedt and FPW, are in their present state simply implementations. They used a couple of different object-oriented program languages to implement lexicons that did the sort of thing that I said they did. But as far as I know there's no abstract declarative theory of what those lexicons are. That's what one needs if one's going to do theoretical linguistics. One probably needs it if one's going to do what they're doing as well.

So the issue of the implicit semantics of default inheritance networks seems to be the absolutely crucial one. One route to go is the Etherington & Reiter route, using default logics. That doesn't actually seem to me an attractive route, at least not for these cases of lexicon and morphology; my sense is, and I don't have any arguments here, this is really just concluding back of the head remarks, is that there's much more power there than one wants and one can get away with a much simpler semantics for such devices, such formalisms - for those applications. Maybe you need the other for the general knowledge representation issues that are addressed in AI. But for the particular linguistic applications I've talked about, (though not the last one which I suspect gets you into the full horror of default logic) my hunch is that something much simpler and more straightforward will do. Had I allowed myself time for the third section I would have talked about a possible route into that. But I didn't.

Discussion

Karen SPARCK JONES: What is the moral for natural language processing? You carefully said at the beginning that shoving things together doesn't [necessarily] mean that there's any real fertilisation - in particular, you can use some of the formal apparatus and concepts of one field, for your own purpose, without implying that this generates a new synthesis. Now you're saying "let's make the assumption that AI has got all this stuff about default inheritance nicely tidied up".

Gerald GAZDAR: I wasn't.

SPARCK JONES: Let's imagine that. Then you might say "this will serve our purposes - this is very nice for us as theoretical linguists". And we say "how nice it is for us to be useful". But what do we get back from it? Maybe you don't think anything.

GAZDAR: No, I don't think you do, at least not immediately. The problem with what I called "the conventional view of the relation" is that it attributes to linguistics what linguistics really hasn't got to offer. It pretends to have them to offer, but really doesn't. The moral is that one shouldn't waste too much time looking for things that aren't there. Another moral is that it behoves computational linguists to do theoretical computational linguistics and applied computational linguistics in parallel. They're going to have to do that, since linguists aren't doing theoretical computational linguistics for them, for the most part.
In this particular instance, here's a case of something developed over in computer science/AI, which is an illustration of the kind of thing that linguists should be thinking about - novel formal technique that has wide applicability.

SPARCK JONES: So it's entirely a contingent matter, that we might or might not get something back, but we should have no expectations about it.

GAZDAR: Yes

Ron KAPLAN: I want to defend linguists a little bit. But first I want to say that I agree with your assessment that linguists don't do formalisms, rather that is something that comes from computer science and computational linguistics. One of the reasons that Chomsky has been so successful all these years is that he's the one guy in that whole area who can diddle formalisms - he's a hacker. All the others are playing a kind of [inaudible] game.

GAZDAR: But he doesn't really anymore. If you look through the stuff he's written in the last ten years, you'll barely find an alpha. Counting alphas is a crude metric, but it'll do.

KAPLAN: Generally, your assessment of the vast majority of present day linguists and their motivations and activities is probably correct. They're not writing grammars, many of them are not going out and getting the crucial facts and so forth. But if that is true of the majority it's not true of all. There is something that linguists who really do linguistics do bring to the table - they have a vast understanding of the facts of language. I used to think I could do linguistics, for example; I was doing ATN grammars in the early days. Then I met some real linguists and started to work with them. There's such a vast difference in their talent for perceiving and organising facts, even pre-theoretically. I think there really is a difference in the skills of people from these two disciplines. I can fiddle around with formalisms in a way that, say, Joan Bresnan, can't. But she has a grasp of the data - and what matters are her intuitions about the data, where the generalisations are likely to lie - that goes way beyond anything I can ever conceive of. I think a linguist who really does linguistics well can bring back to NLP or CL a real sense of what the descriptive problem is - what kind of data bear on the problem, where to go look for the data, how to look for the data.

GAZDAR: I was concerned to highlight a disparity between the PR and the actuality.

KAPLAN: I think you're right.

GAZDAR: I don't think that linguists are worthless people who do nothing at all.

KAPLAN: Many of them are.

GAZDAR: To make your point, which I forgot to make. The closest analogy I can think of would be the surgeon's knowledge of the human body. I take it that there are all kinds of things that surgeons know about people's insides - say. that fat people have their spleens displaced to the right, that kind of thing. There are lots of things that you don't find out by reading medical text books, but you do by talking to surgeons. It seems to me that what many linguists are good at is that; you can find out an awful lot by talking to them that you don't by reading their papers.

KAPLAN: Or even by going out and pretending to do linguistics yourself, if you're not that well-trained or competent, as several of us are not.

I wanted to make one further point about this interdisciplinary foolishness. Again much of what you say is correct. But there is a legitimate new discipline where the particular talents, skills and interests of linguists, computational linguists, and psycholinguists can relate to each other. That's where we take, as the intellectual problem we're concerned with, the question of how it is people understand, learn and produce sentences in their languages. You can go about that in various ways. One particular way is to make the assumption that there's some common core of knowledge for all those behaviours and that store of knowledge is what linguists should be characterising. The question then remains, how does that relate to computational interpretations and psychological models. But there is an enterprise that does integrate all the disciplines. Small subsets of people have taken that seriously. Some of that emerged from the Sloan [Foundation] effort. So it's slightly misleading if you say there was just a random mixture of people. There really is a serious interdisciplinary area there. But perhaps it doesn't fall within any of the original disciplines.

Bob MOORE: Do you know of any examples from linguistics that suggest a need for interacting defaults, because that's one of the big technical problems of the formal work that's been done on defaults?

GAZDAR: I think at least the FPW implementation fits multiple inheritance. A question not addressed at all in that paper is what happens when you get clashes between information from the multiple sources of inheritance. I know of nothing else in linguistics that has anything to say on the matter at all. It's something I've been thinking about.

KAPLAN: I have a technical question. These unification based theories, they're descriptive theories which have propositions in them in the form of equations of whatever. Even though they're propositional. they're not full first order predicate calculus, because they're really quantification free - all the quantifiers are universal. I wonder, when you go to these default logics, is there a special case of default logics without the quantification, that is what is needed for linguistic cases?

Henry THOMPSON: It goes exactly the other way. I think there is quantification in the system Gerald outlined, in the worst case.

KAPLAN: You get interactions of existentials and universals?

THOMPSON: I think that's there. There's a comment in Stu's paper [Shieber, this volume] unfortunately he's out of the room at the moment - about the potential existence of negation in the logic that underlies the HFC in the default system Gerald was talking about. My sense is that it is that complicated, but I don't know exactly how complicated, because I don't know what the questions are. I'm sympathetic to a lot of what Gerald says, but I have to be a little bit more negative than he is.

The inheritance mechanisms of the "ERIC isa EAGLE" variety bear a lot of relevance to the morphological examples. I'm not at all convinced that that goes very far towards a theoretical foundation for feature defaults and inheritance mechanisms which are the other example that Gerald used. There's an obvious distinction in that there is a directionality in the simple cases that isn't present, or anywhere as near obvious, in the syntactic case. There is also a distinction in that the hierarchies in all the other cases are tight hierarchies with a "member" link at the bottom. That's really not the generalisation you want at all for the tree structures you see in the syntactic case. For these reasons, and some others, I think the formal foundation of the feature mechanism that Gerald talked about in the syntactic case is going to turn out very different, and rather more complicated, than the formal mechanisms for the inheritance networks, whether there's multiple inheritance or not.

GAZDAR: I think that what you call the lack of directionality, which I call bidirectionality, may get you into graphs with cycles in, which are much more complicated objects than graphs without cycles in, but they're all in the same space.

KAPLAN: It's not clear they get you into circular graphs. There's a way of thinking about the structures as being described. The structures you come up with are not cyclical. It's only in your description of them that you have what looks like a cycle.

THOMPSON: The problem that you have to take on board is that the dimensionality of the pictures appears to be the same in the two cases - the basic mechanism (not underlined) and the inherited properties (underlined) in (9). The dimensionality of the morphological and syntactic cases appears to be the same. But that's misleading. In the syntactic case you can't actually fold the multiple dimensions down into the page in that way. The relation you end up needing is between sets of trees, not between individuals as you have in the morphological case. The truth of the matter is that the defaults of syntactic features are, as you said, properties of whole categories, (in fact whole trees, but certainly sub-trees). The relationships are between trees, whereas the relationships in the morphological case and the "ERIC isa EAGLE" case are relationships between sets, or individuals, and that's a lot simpler.

MOORE: In the syntactic case, if there's any quantification, isn't it normally bounded quantification - you're dealing with some restricted set of values of features - so in principle you could expand that out into a purely propositional case. You'll be happy to know that the propositional case for all the default logics I'm familiar with is at least decidable. Whereas the general first-order case is normally worse than undecidable - in some cases the sets of "theorems" are not even recursively enumerable.

THOMPSON: That means we're OK for English and we're worried about Swedish. What you just said makes one concerned about the finiteness of the category vocabulary.

KAPLAN: No, but for any particular language, or for any particular grammar of a particular language, the grammar has all the features in it that could possibly be mentioned - finitely specified.

THOMPSON: Not if there are recursive categories.

KAPLAN: So you're not talking about GPSG anymore.

THOMPSON: Sure I am. As it says very carefully in the book (Gazdar et al 1985), we are able to avoid recursive categories because we are focusing on English. But there is at least suggestive evidence that in other languages, we are not going to be able to avoid recursive categories. You [Gazdar] may want to take that back or I may be misrepresenting you.

GAZDAR: Your point, that if you allow recursive categories, then there isn't just a finite set of possible expansions, is right. But that still might keep you in some restricted space.

KAPLAN: That basically puts you in the LFG-type and there are a lot of things that are decidable in that space. I've been looking at an operation in the description language of LFG to represent defaults and interpretation of fragments, which I call "priority union". I don't really understand its mathematical properties. It's a thing I'll talk about in my session tomorrow.

THOMPSON: Its clear that there are a number of people splashing around in the same area.

KAPLAN: Splashing each other.

Mitch MARCUS: Are the cases where you want to use this default reasoning going to interact in a way that's going to blow you out of a finite space?

THOMPSON: I hope not. In the contexts that I'm worrying about, category-valued features and defaults tend to be relatively disjoint phenomena.

GAZDAR: That's my sense as well.

THOMPSON: That's a contingent fact of the grammars that one tends to write, rather than a necessary property of the theory at the moment.

GAZDAR: Since the theory doesn't cover them.

THOMPSON: Even if you restrict yourself to non-recursive categories, it seems to be the case that the kind of defaulting mechanisms that you need to do, tend to steer clear of constructing default categories to be the values of category-valued features.

MARCUS: Fine. Another question. In terms of the AI discussions on default reasoning, there are different kinds of situations that one wants to deal with, but they're fairly impoverished. One is the monotonic reasoning case, Bob [Moore (1983)]'s autoepistemic restriction is another interesting case, that seems weaker. I wonder if there's a kind of hierarchy of these things emerging in any interesting sense.

GAZDAR: My sense is that there is, but Bob would be the one to have anything intelligent to say about that.

MOORE: A lot of the uses that have been made of default inheritance in AI are in some sense incomparable, because they're fundamentally different from a pragmatic, in a formal sense, point of view. I've particularly tried to argue in my paper on the subject (Moore, 1983) that default reasoning as the term is used in AI is default conjectural reasoning. Whereas the defaults here are more in the form of default specification. So there's no sense of getting additional information. The defaults in the syntactic and phonological cases are not cancellable, whereas in the pragmatic case, in some instances, they are.

GAZDAR: In the syntactic case, they are.

MOORE: Not in the same sense. This is very difficult to explain.

KAPLAN: Let me try with this. Do you know any cases where you could dispense with default mechanisms by redundantly elaborating grammatical and lexical specifications?

GAZDAR: In the cases I'm most familiar with you could always do that, but you would end up with a horrendous grammar.

KAPLAN: But if its a substitution for a finite specification then you would expect it to be well-behaved mathematically.

GAZDAR: Yes, and I think in the lexical cases it'll be the same, at least in the cases that have been talked about. What you do with the lexicon that has some recursion in it, or plain iteration, I haven't thought through. One can't assume that lexicons are finite, or rather that the set of words is finite.

KAPLAN: [To Moore] Does that get at the issue?

MOORE: That gets at the issue I was going to deal with next. I won't try to expand on that pragmatic distinction. As far as the formal properties are concerned, one would like to look for special cases. Reiter and his colleagues have identified a special case of what they call "normal defaults", where if all defaults are of the form, "if *p* is possible then *p*", things are computationally tractable. But one of the cases you were discussing amounted to "if it's possible that *p* iff *q* then *p* iff *q*". That might be the same case, I'm not sure whether p has to be atomic. So some of the defaults you had in the syntactic case were certainly more complicated than suggest any simple restriction of at least propositional default logic.

THOMPSON: A difficult question is the interaction of defaults and well-formedness constraints, which is a characteristic of GPSG. It's a characteristic of the EAGLE network, but that aspect of semantic nets seems to me underdeveloped. It's a question of where "dead eagles can't fly" is represented - it's not in the network, it's somewhere else. There's a way of reconstructing that with multiple inheritance, which says that there are dead things, and they inter alia don't fly, and eric is one of those as well as an eagle. Then the question becomes one of resolution in the case of conflicting inheritance.

MARCUS: I'm just thinking of the consequences of defining a set of dead things. You quickly find yourself in a situation where you take all implications and create categories for the things of which the left hand side is true. There are lots of things that dead animals don't do - they don't play the piano either.

One other comment, which is not about default inheritance per se, more about defaults in general. If one is trying to do a competence theory, where everything is simply specified, you end up with one kind of story about defaults which is the restricted kind you're talking about. If one thinks about the processing case, either for engineering purposes or as a psychological model, one might want to bring the entire non-monotonic apparatus to bear, to allow one to do default reasoning about things that are incrementally true. In my work I've tried to avoid that, but it's an interesting possibility - to consider kinds of things true by default until you discover evidence further on in the sentence that they're false. In some sense it's the difference between the original ATN approach which was chronological backtracking, vs. the kind of dependency-directed backtracking that people in AI are working on.

To my knowledge this has not been applied to natural language understanding. If you set up some state in an ATN, and it turns out to be wrong, you back up entirely and push on again. Whereas the rest of the AI literature has moved away from that to dependency-directed backtracking and even more subtle kinds of things. From my point of view, it's an interesting engineering question. What kind of parsers might result if we try to apply dependency-directed techniques, the kind of things that Johan de Kleer is doing. Has anybody done this, Ron?

KAPLAN: I'm sure this kind of thing has been done.

MARCUS: If people here don't know about it, I guess it hasn't.

KAPLAN: I'm not sure it was interesting.

Mary McGee WOOD: One linguistic theory which is based almost entirely on this kind of thing is Dick Hudson's (1984) *Word Grammar*, which isn't very well known. Since that book was published he's explicitly started using the *isa* relation from AI. He has a network that says "verb *isa* word", "transitive verb *isa* word", etc. One interesting consequence you can get if you push that far enough, which is coming out in some stuff I've done, is that you end up abolishing the lexical component. I'm throwing this out as a marker for tomorrow really. Once you see the whole thing as a flexible default network, with priority to the particular, then you lose any very good justification for putting a sharp cut-off point across that network and saying these guys are general enough to be syntax and these guys are specific enough to be lexicon. If you're implementing you can create a virtual lexicon and virtual syntax by putting a cut-off point at whatever level suits your application. So this kind of mechanism, if you use it thoroughly enough, can have some pretty far-reaching implications for the form of your grammar.

Graeme RITCHIE: Have you got the ordering sorted out? When you gave about a dozen default rules, the non-inheritance ones, you mentioned in passing there was a problem with integration and the notion of ordering if you had a declarative approach.

GAZDAR: There's a solution to that in the book (Gazdar et al, 1985). There's an apparatus which does not give rise to the bad cases and which is not particularly ad hoc in it's own terms. But it's pretty indigestible stuff. It all hangs on the notion of "privilege". It takes me about half an hour to reconstruct what I meant by it whenever I look at it. But it isn't particularly unnatural - it's not hard to reconstruct because it uses weird notions and ad hoc tricks, it's just hard to reconstruct - probably because it's a bad way of doing it. As far as the authors of the book now, it's a solution of sorts.

RITCHIE: It solves all known problems?

GAZDAR: We had a bunch of paradoxes, some of which I showed you, that we were addressing throughout that work. And all the ones we knew about were taken care of by what we developed. There aren't any residual ones that we know of. There may be others - I think it likely that there are, because it feels kind of fragile to me.

References

Brachman, R. 1985. "I lied about the trees", (or defaults and definitions in knowledge representation). *AI Magazine*, 6.3, 80-93.

Chomsky, N. & Halle, M. 1968. *The Sound Pattern of English*. New York: Harper and Row.

de Smedt, J. 1984. Using object-oriented knowledge-representation techniques in morphology and syntax programming. In T. O'Shea (ed.) *ECAI-84: Proceedings of the Sixth European Conference on Artificial Intelligence* Amsterdam: Elsevier, 181-184.

Etherington, D.W. & Reiter, R. 1983. On inheritance hierarchies with exceptions. *Proceedings of the 3rd National Conference on Artificial Intelligence (AAAI-83)*, Washington DC, August 1983. W. Kaufmann, pp. 104-108.

Flickinger, D., Pollard, C.J. & Wasow, T. 1985. Structure sharing in lexical representation. *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics* (Chicago), 262-267.

Gazdar, G. 1979. *Pragmatics: Implicature, Presupposition and Logical Form*. New York: Academic.

Gazdar, G., Klein, E., Pullum, G. & Sag, I. 1985. *Generalised Phrase Structure Grammar*. Oxford: Basil Blackwell.

Grice, H.P. 1975. Logic and conversation. In P. Cole & J. Morgan (eds.) *Syntax and Semantics 3: Speech Acts*, New York: Academic Press, 41-58.

Halvorsen, P-K. & Withgott, M. 1984. Morphological constraints on Scandinavian tone accent. Technical Report CSLI-84-11, Centre for the Study of Language and Information, Stanford, CA.

Hudson, R.A. 1984 *Word Grammar*, Oxford: Basil Blackwell.

Jackendoff, R. 1977. *X' Syntax: A Study of Phrase Structure* Cambridge, Mass.: MIT Press.

Marantz, A.P. 1984. *On the Nature of Grammatical Relations*. Cambridge, Mass.: MIT Press.

McDermott, D.V. & Doyle, J. 1980. Non-monotonic logics. *Artificial Intelligence* 13, 401-72.

Mercer, R.E. & Reiter, R. 1982. The representation of presuppositions using defaults. *Proceedings of the 4th National Conference of the Canadian Society for Computational Studies in Intelligence* (Saskatoon), 103-107.

Moore, R. 1983. Semantical considerations on non-monotonic logics. *Proccedings of the 8th International Joint Conference in Artificial Intelligence (IJCAI-83)*, Karlsruhe, 272-279.

Reiter, R. 1978 On reasoning by default. *Theoretical Issues in Natural Language Processing 2 (TINLAP-2)*, 210-218.

Soames, S. 1979. A projection problem for speaker presupposition, *Linguistic Inquiry* 10, 623-666.

Touretsky, D.F. 1986. *The Mathematics of Inheritance Systems*. London: Pitman.