

# The logic of default inheritance in Word Grammar

Richard Hudson

In principle, default inheritance has always been the underlying logic of grammar, and of morphology in particular. Without inheritance there can be no generalisation; and without default inheritance, there can be no exceptions. Arguably, the study of morphology is the intellectual endeavour where this logic has been tried and tested most successfully in dealing with a myriad of detail which is hard to match in any other intellectual endeavour. Not surprisingly, some version of default inheritance (aka normal inheritance) has been widely used not only in constraint-based theories of grammar, but also in AI more generally, where it has been used in modeling human inference.

The basic idea behind default inheritance is very simple and obvious: more specific patterns override or block more general ones; or where there is a conflict between a rule and an exception, the exception always wins. However, this simple principle raises such profound challenges for any formalised theory that some people reject the whole idea. Indeed, default inheritance challenges elementary traditional logic in just the same way that the modern view of categories as prototypes challenges the classical view of categories as fixed classes defined by necessary and sufficient conditions. (So far as I know, I'm the only person to point out the link between prototypes and default inheritance; but any theory of categorisation that allows default inheritance automatically allows categories to generate 'prototype effects'.) In short, integrating default inheritance into any theoretical package is not a trivial task.

The potential problems with default inheritance are as follows:

- generality: how does the logic generalise beyond the domain of (say) morphology?
- reliability: how can any inference be considered reliable in a non-monotonic logic?
- certainty: how can we recognise which inheritable properties clash, and which of them should take priority?
- economy: how can we prevent inherited properties from being stored, thereby losing the benefits of generalisation?
- relevance: how can we prevent irrelevant properties from being inherited?

Any theory that incorporates default inheritance must confront these questions, and the more explicitly formalised the theory, the more urgent they become. The discussion will present the solutions offered by Word Grammar, and will explain how these solutions are related to the other components of Word Grammar theory. It will not try to explore how, or whether, the Word-Grammar solutions can be applied to other theoretical packages.

## The WG theory of morphology (and other things)

First, though, I will outline the theory of morphology contained in Word Grammar (WG). Like all the other theories represented in this workshop, and in Stump's terminology, WG probably qualifies as an 'inferential-realizational' theory in the European Word-and-Paradigm tradition, with 'realization' linking lexical and syntactic categories to the forms of the words concerned. For example, the word defined by combining the lexical item DOG with the inflectional category 'plural' (DOG, plural) is realized by the form *dogs*. Moreover, generalisations apply by

default inheritance (rather than by unification). However, WG morphology also has some distinctive characteristics:

- Syntactic categories and lexical items are word-classes in a taxonomy rather than features consisting of an attribute and its value; e.g. ‘plural’ and DOG are both word-classes, with the same theoretical status. Features are allowed in WG, but are only used for expressing agreement.
- Morphology is recognised as a distinct level between syntax and phonology; e.g. {dog} and {s} are classified as root and suffix, and the {s} of *dogs* is the same morph as the one in *barks*, and is a different concept from the phoneme /s/.
- Inflectional and derivational morphology share the same realisational apparatus, as do clitics, and differ only in whether the ‘output’ is a different lexical item.
- Words are related not only to their stem, but also to ‘derived’ versions of the stem called ‘x-variants’ (where ‘x’ varies); e.g. {{dog} {s}} is the s-variant of DOG, plural. In cases of syncretism, two distinct words share the same variant of their base; e.g. the past participle and passive participle in English are both realized by their ‘en-variants’.
- The ‘attributes’ of other theories are expressed as network relations which are themselves interrelated in a taxonomy; for example, the relation ‘en-variant’ ‘isa’ (is a particular case of) the more general relation ‘ed-variant’, which in turn isa realization.

These distinctive characteristics of the WG theory of morphology all derive from one very general characteristic of WG: its cognitive orientation. WG isn’t just a theory of morphology, or even of language; it is a theory of language located in a general framework of ideas about cognition. According to WG, language is an ordinary part of cognition rather than a mental module (as in Chomskyan theory). The modularity debate is largely irrelevant for most of the other theories in the workshop, which are cognitively agnostic, but it is crucial for WG because the goal is a model of cognitive structure. This cognitive orientation is one of the (many) reasons for choosing a logic based on default inheritance, given that this is so clearly the logic that we use in everyday thinking about other things.

Morphology is a particularly good area of language to face the challenge of describing language in domain-general terms because it is the area where language is most remote from the rest of cognition – from both meaning and phonetics. If even morphology can be modeled as an example of general cognition, without any special apparatus unique to morphology, then surely language is not a special module of the mind.

## **Default inheritance in morphology**

As mentioned earlier, morphology is also a particularly good area for testing any theory of default inheritance because we have so much relevant data – indeed, our ancestors have probably been assembling such data for at least 4,000 years (since Babylonian scribes first wrote down verb paradigms). Every generalisation applies by inheritance, and every exception overrides the default. Moreover, the data of morphology are (in general) both clear and highly structured. For instance, there is no uncertainty about *dogs* being the plural of DOG, and it is clear that the relation of *dogs* to DOG is exactly the same as that of *cats* to CAT. Moreover, these facts are relatively context-independent, so morphology is a good area for building self-

contained models for an area of human knowledge, as witness the proliferation of computer models for morphology.

Morphology faces all the general challenges for default inheritance listed above:

- generality: how does the logic generalise beyond the domain of morphology? If the logic is part of general cognition, it must not depend on analyses or notations peculiar to morphosyntax (such as attribute-value matrices).
- reliability: how can any inference be considered reliable in a non-monotonic logic? If exceptions are always possible, how can we be sure that the general rule for some morphological class applies in any particular case?
- certainty: how can we recognise which inheritable properties clash, and which of them should take priority? This problem becomes especially critical if attribute-value matrices are replaced by some more general format. But in particular,
  - how are conflicts in multiple inheritance handled? In morphology, words normally inherit different properties from a lexical item and from various morphosyntactic categories, and normally the system is organised to avoid conflicts, but what if they do conflict?
  - if an exceptional form overrides a form defined indirectly, are the indirect relations inherited? In morphology, the problem is common with ‘reentrancy’ (Bouma 2006): what if the default realization of some property *p* is the same as the base, but some exceptional lexeme realises *p* as (say) *bac* – does that mean that the base also changes to *bac*?
- economy: how can we prevent inherited properties from being stored, thereby losing the benefits of generalisation? In morphology, how do we stop inherited forms from being stored so that every form is simply retrieved rather than created anew? (The problem is that we know from psycholinguistic experiments that some forms are in fact created anew.)
- relevance: how can we prevent irrelevant properties from being inherited? Any word or form has a host of properties that could be inherited, but which are currently irrelevant – e.g. we know that a typical English word is spoken in the UK, the USA, Australia and so on, and for some words we know their etymology; but both of these properties are irrelevant in most contexts (though they are both potentially crucial in certain imaginable situations). It seems reasonable to assume that we don’t, in fact, inherit such properties, but how does the inheritance mechanism filter them out?

## The Word-Grammar solution

The WG solution to these problems rests on three assumptions fundamental to WG theory:

1. Types and tokens are different. Every bit of ongoing experience is a ‘token’ (though this term tends to be used by linguists rather than psychologists), which we understand by categorizing it as an example of some stored ‘type’. And every token is represented by a separate concept (i.e. by a separate node in the cognitive structure). In theorizing about cognitive structures we can distinguish newly-created tokens from stored types (though learning turns some tokens into permanent types). Rather obviously, but importantly for the WG theory, if a token is an example of a type, then it will always be lower in

- the inheritance hierarchy than that type – and, indeed, lower than any other concept from which it might inherit.
2. Every conceptual node (i.e. every concept) has an activation level. The level depends partly on frequency and recency of previous activation ('entrenchment'), but partly too on current mental activity, which is determined by attention and the ensuing spreading activation. In general, the currently available activation is concentrated on 'relevant' nodes and their neighbours.
  3. The overall conceptual structure is a 'taxonomic network' in which every link is directed. Thanks to 'isa' links, every node (including token nodes) is part of a taxonomy, but any node may be linked freely to any other node (in contrast with a 'directed acyclic graph', in which loops are not possible). The overall taxonomy includes relational concepts as well as entity concepts, so every labelled link (e.g. 'base', 'x-variant') is part of a taxonomy of relations.
  4. Some relational concepts are marked as mutually incompatible (e.g. 'before', 'after').

Given these assumptions, then, the WG algorithm for default inheritance (DI) is as follows:

1. Every token inherits, but it's only tokens that inherit; so inheritance exists just to enrich newly created nodes. Assume a newly created token node Token, which isa one or more type nodes, each labelled Type<sub>i</sub>.
2. Search each Type<sub>i</sub> for inheritable properties – i.e. for links from Type<sub>i</sub> to other concepts, consisting of a relation R and an entity E. These are inheritable if
  - a. they are sufficiently relevant to be active (beyond a certain threshold level).
  - b. the relation R does not already apply to Token, where R already applies to Token if Token already has a link via a concept which either isa R, or which is incompatible with R.
3. For each inheritable property, make a copy 'Token – R' – E' (with R' isa R and E' isa E, and the same directionality as in the original). If the properties of Type<sub>i</sub> form a loop back to Type<sub>i</sub>, make a copy of the entire loop so that each type node only has one copy token. Notice that R' and E' are tokens, so DI immediately applies to them as well, subject to available activation.
4. Repeat 2-3 for every node above each Type<sub>i</sub> in the taxonomy.

This theory solves the problems for DI as follows:

- generality: how does the logic generalise beyond the domain of morphology? The DI algorithm is completely domain-general, and could be applied as easily to knowledge of birds as to our knowledge of morphology.
- reliability: how can any inference be considered reliable in a non-monotonic logic? Every inference is reliable, so the system is actually monotonic, because DI always applies 'bottom-up', applying only to the most specific possible concepts, newly-created tokens.
- certainty: how can we recognise which inheritable properties clash, and which of them should take priority?
  - how are conflicts in multiple inheritance handled? This problem is deliberately left unsolved in WG theory, because it is unsolved in real cognition. The 'Nixon Diamond' was a genuine conflict, and was only 'solved' by stipulating a winner. There are cases in morphology where

unresolved conflicts have no resolution and leave us without any outcome; arguably, the gap where we expect *amn't* or *aren't* is an example.

- if an exceptional form overrides a form defined indirectly, are the indirect relations inherited? Again this conflict has no general solution in cognition, so it needs no general solution in theory. Instead, WG allows solutions to be stipulated by isa links between exceptional and default values to show that indirect relations are inherited; without such isa links, these relations are not inherited.
- economy: how can we prevent inherited properties from being stored, thereby losing the benefits of generalisation? This storage is prevented by allowing DI only for tokens; so the inherited properties will only be stored if the token itself is stored (i.e. remembered and learned).
- relevance: how can we prevent irrelevant properties from being inherited? Relevance controls DI through activation. E.g. if etymology happens to be the current focus of attention, then tokens may inherit known etymologies, but not otherwise.