Inheritance in Natural Language Processing

Walter Daelemans
Tilburg University

Koenraad De Smedt*
University of Nijmegen

Gerald Gazdar
University of Sussex

In this introduction to the special issues, we begin by outlining a concrete example that indicates some of the motivations leading to the widespread use of inheritance networks in computational linguistics. This example allows us to illustrate some of the formal choices that have to be made by those who seek network solutions to natural language processing (NLP) problems. We provide some pointers into the extensive body of AI knowledge representation publications that have been concerned with the theory of inheritance over the last dozen years or so. We go on to identify the three rather separate traditions that have led to the current work in NLP. We then provide a fairly comprehensive literature survey of the use that computational linguists have made of inheritance networks over the last two decades, organized by reference to levels of linguistic description. In the course of this survey, we draw the reader's attention to each of the papers in these issues of Computational Linguistics and set them in the context of related work.

1. Introduction

Imagine that you are a linguistic innocent setting out on the job of building a computer lexicon for English. You begin by encoding everything you know about the verb *love* and then turn your attention to the verb *hate*. Although they are antonyms, the majority of properties that you have listed for *love* will show up again in your list for *hate*. Your first thought is to put this list of common properties into an editor macro to save you the laborious task of typing them all in each time that you add another verb. But it soon becomes clear to you that adopting this strategy is going to lead to a huge representation for your lexicon, and one that keeps saying the same thing again and again. Your second thought is to put the common property list in just one place and call it, say, TRANSITIVE VERB. Then you amend what you have entered for *love* and *hate* so that all the common material is replaced by a notation that indicates that each is a transitive verb. This works well and you add a couple of thousand more English verbs without difficulty. It is only when you reach *elapse* and *expire* that you find yourself landed with the tedious task of again typing full lists of properties, since these two verbs cannot be accurately represented by including a reference to the TRANSITIVE VERB property list. Looking at the entries for these two anomalous verbs induces a feeling of déjà vu. They too have many properties in common, but just not exactly the same set of common properties as *hate* and *love* and their siblings. Following the strategy that worked well before, you gather their common properties together and give them the name INTRANSITIVE VERB, then you strip the duplicated material

* University of Leiden, Psychology Department, P.O. Box 9555, 2300 RB Leiden, The Netherlands.

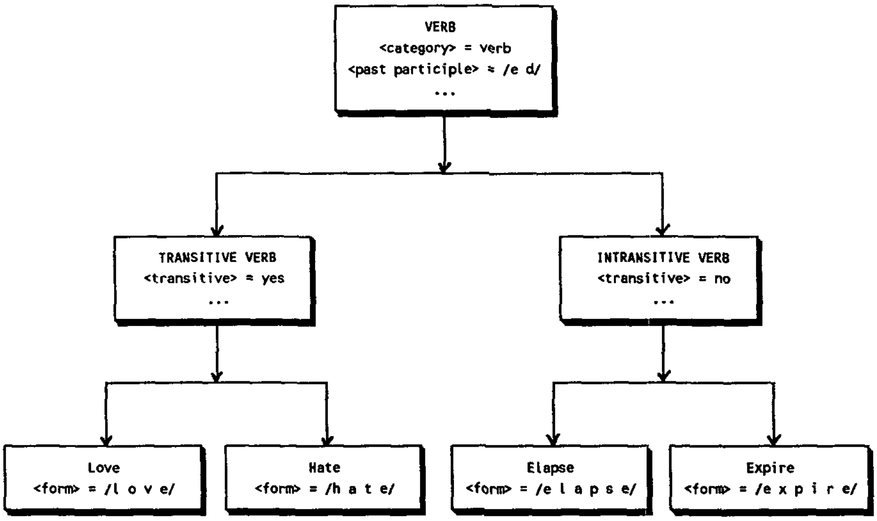

Figure 1
Monotonic single inheritance.

from the entries for *elapse* and *expire* and replace it with a notation that points to your list of intransitive verb properties. As you inspect your handiwork, you notice that the lists of properties associated with TRANSITIVE VERB and INTRANSITIVE VERB now exhibit exactly the kind of duplication that you first saw when you wrote down your entries for *love* and *hate*. Indeed, the number of their commonalities exceeds the number of their differences. Once again you decide to invoke the style of solution that you have used before: you collect the common properties together, give the collection the name VERB and then rework your formulation of TRANSITIVE VERB and INTRANSITIVE VERB so as to strip the shared material and replace it with a notation indicating that each is an instance of VERB.

Although you may not realize it, what you have done is build an inheritance network to represent the information that you are including in your lexicon—see Figure 1. The root node of this network is VERB and it has two daughters, TRANSITIVE VERB and INTRANSITIVE VERB, which inherit all the properties associated with the root. Each of these two nodes has further daughters (**Love**, **Elapse**, etc.). The latter inherit all the properties of VERB together with all the properties of their immediate parent. These inherited properties are added to the properties listed as idiosyncratic to the lexical item itself (e.g., the property of being orthographically represented as /l o v e/). This very simple lexical network has a couple of characteristics that it is worth drawing attention to. Firstly, each node has a single parent, and there is thus only one path through which properties may be inherited. A network of this kind either consists of a single tree of nodes, or of a set of (unconnected) trees of nodes, and we will call such a network a single inheritance network.[1] Secondly, in describing our example, we have been assuming that each node inherits *all* the properties associated

1 Two trees are unconnected if and only if they have no nodes in common. For present purposes, a set of unconnected trees can always be trivially converted into an equivalent single tree by adding a new root for all the trees, but one that has no properties associated with it.

with its parent node which, in logician's parlance, means that property inheritance is monotonic.

Neither single inheritance nor monotonicity is a necessary characteristic of inheritance networks. Suppose you try to add **Beat** to the network we have been describing. The obvious thing to do is to insert it as a daughter of TRANSITIVE VERB. But this is likely to entail that your network will claim that the past participle is **beated*. One potential solution to this problem would be to define a node called EN TRANSITIVE VERB and attach **Beat** as a daughter to this. However, this strategy simply pushes the problem further up the inheritance tree: EN TRANSITIVE VERB cannot be a daughter of the TRANSITIVE VERB node since it contains a property (past participle = /e n/) that is inconsistent with a property associated with the latter (past participle = /e d/). Nor can our new node be attached as a daughter of VERB, for exactly the same reason. It seems, therefore, as if the new node may have to be defined wholly from scratch, duplicating all but one of the properties of TRANSITIVE VERB. To avoid this disagreeable conclusion, we might consider another potential solution in which we remove any reference to the past participle suffix at the level of the VERB node, and specify it instead at the level of that node's daughters. At first sight, this appears to be a most attractive option. In fact, by adopting it, we have embarked on a slippery slope that will result in our stripping VERB of almost all the properties canonically associated with verbs. For each property you might expect it to have, if there is a single verb in English that is exceptional with respect to that property, then the property cannot appear at the VERB node. In the case of morphological properties, this is likely to mean that "present participle = /i n g/" is the *only* property that can be associated with the VERB node. And, in the case of syntactic properties, it is likely to mean that banalities such as "category = verb" will be all we are able to list.

How are we to avoid these rather dismal alternatives? There are (at least) two possibilities. One is to abandon single inheritance. Suppose we reorganize our network so that TRANSITIVE VERB and INTRANSITIVE VERB only encode syntactic properties of verbs. We then introduce two further nodes, ED VERB and EN VERB, which only encode morphological properties. Then we allow **Beat** to have both TRANSITIVE VERB and EN VERB as its parents. A network of this kind can no longer be represented as a tree (or set of unconnected trees) and is said to employ *multiple inheritance*—see Figure 2. Another possibility is to abandon monotonicity. We leave **Beat** where we first attached it, under TRANSITIVE VERB in our original network, and we associate the property "past participle = /e n/" with it. If inheritance continues to be construed monotonically, then the network will make contradictory claims about the past participle of **Beat**. But if we adopt a nonmonotonic interpretation of inheritance, in which properties that are attached to a node take precedence over those that are inherited from a parent, then no contradiction will arise. Such nonmonotonic inheritance is known as "default inheritance"—see Figure 3.

Monotonic single inheritance networks are easy to build and easy to understand. If one designs a notation for defining them, then it is straightforward to say what the semantics of that notation is: translation into first order logic, for example, is quite trivial. Unfortunately, for the reasons hinted at in the example considered above, monotonic single inheritance networks are not really very well suited to the description of natural languages. As a result, as we shall see below, most researchers who have employed inheritance techniques in NLP have chosen to use either default inheritance or multiple inheritance or, very commonly, both. Networks that employ default and/or multiple inheritance are also quite easy to build, but they are much less easy to understand. The combination of default and multiple inheritance is especially problematic: "despite a decade of study, with increasingly subtle examples and counterexamples being

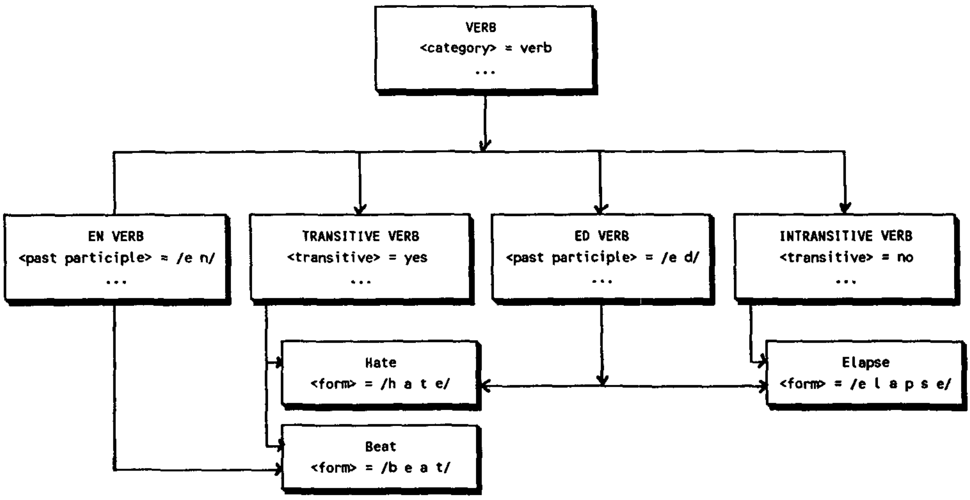

Figure 2
Monotonic multiple inheritance.

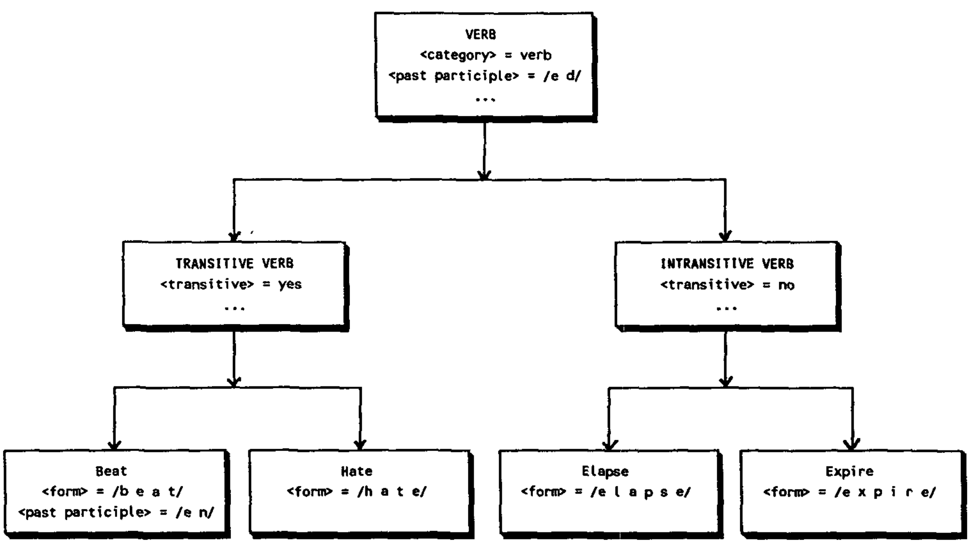

Figure 3
Nonmonotonic single inheritance.

considered, consensus has yet to emerge regarding the proper treatment of multiple inheritance with cancellations" (Selman and Levesque 1989, pp. 1140). Unsurprisingly, the problem has given rise to a large, and growing, list of publications in the knowledge representation literature (see, e.g., Horty, Thomason, and Touretzky 1990, and references therein). Almost all of this theoretical work has concerned itself with very simple networks that are only able to say whether or not a monadic property holds of a node in the network. Recently, however, Thomason and Touretzky (1991) have turned their attention to the properties of more expressive networks, potentially capable of encoding what would need to be encoded in any real NLP application. Nonmonotonic inference more generally (i.e. not just in networks) has been, arguably, the dominant theoretical concern in the AI literature of the late 1980s (as measured, for example, by the proportion of papers that have appeared on the topic in *Artificial Intelligence* over the period).

One of the key issues in the knowledge representation literature has been how to deal with the default inheritance of mutually contradictory information from two or more parent nodes. Most NLP researchers who have embraced multiple inheritance techniques have chosen to avoid this issue by adopting one of two strategies. On one strategy, information is partitioned between parental nodes. You can, for example, inherit morphological properties from node **A** and syntactic properties from node **B**, but no single property can be inherited from more than one parent node. This is known as "orthogonal inheritance." One way of thinking of it is in terms of a set of disjoint single inheritance networks layered on top of each other. On another strategy, a given property, or set of properties, may potentially be inherited from more than one parent node, but the parents are ordered: the first parent in the ordering that is able to supply the property wins, and contradiction is thus avoided. We will refer to this strategy as "prioritized inheritance."

The use of inheritance networks in current NLP comes from three rather separate traditions. The first is that of "semantic nets" in AI, which goes back to Quillian (1968) through Fahlman's (1979) NETL to the late 1980s monographs by Touretzky (1986) and Etherington (1988). The second is that of data abstraction in programming languages, which has led to (a) object-orientation in computer science with its notions of classes and inheritance as embodied in such languages as Smalltalk, Simula, Flavors, CLOS and C++, and (b) the use of type hierarchies, which have become widely seen in unification-oriented NLP since the appearance of Aït-Kaci (1984) and Cardelli (1984). Of necessity, the type hierarchy work in NLP has remained strictly monotonic. The third is the notion of "markedness" in linguistics, which originates in the Prague School phonology of the 1930s, reappears in the "generative phonology" of Chomsky and Halle (1968) and Hetzron's (1975) and Jackendoff's (1975) models of the lexicon, and shows up in syntax in the "feature specification defaults" of Gazdar, Klein, Pullum, and Sag (1985).[2] Unlike the other three traditions, the linguistic tradition does not embody a notion of inheritance per se. But the issue of how to decide which operations take precedence over others has been a continuing concern in the literature (see, e.g., Pullum 1979, especially Section 1.4.1, and references therein).

The consensus view, though largely unspoken, among computational linguists currently working with default inheritance networks appears to be that nodes that are close (or identical) to the root(s) of the network should be used to encode that which is regular, "unmarked," and productive, and that distance from the root(s) should correlate with increasing irregularity, "markedness," and lack of productivity. At the

2 See Evans (1987), Gazdar (1987), and Shieber (1986a) on the various defaulty characteristics of GPSG.

very least, this is what emerges from their practice. The differences between the current strands of NLP work in this area are partly philosophical (e.g., as to whether psycholinguistic data could or should be relevant to the structure of the network), partly methodological (e.g., as to whether networks should be built in a formal language designed for the purpose or implemented in an existing computer language), partly technical (e.g., whether a negation operator is useful, or whether orthogonal networks are to be preferred to those using prioritized inheritance), and partly theoretical (e.g., the trade-off between the semantic perspicuity of monotonic networks versus the expressiveness and concision of their nonmonotonic competitors).

In the subsequent sections of this paper we will survey the use computational linguists have made of inheritance networks over the last dozen years. To organize this chronologically (e.g. by date of publication) would be to impose a wholly spurious sense of historical continuity on what has, in fact, been a fairly haphazard set of parallel developments. It is tempting to try to organize the discussion that follows by reference to technical and formal parameters, but the area is just too young for that to be possible without a great deal of rather arbitrary taxonomy. So we have chosen to play safe and organize the material by reference to levels of linguistic description. This is not wholly satisfactory, since a significant number of the approaches we discuss have been applied to several different levels of description, which means that we have to refer to them in more than one section. But we hope that readers will bear with us.

2. Syntax and Morphology

One of the earliest applications of inheritance to syntax was Bobrow and Webber's (1980a,b) use of PSI-KLONE (a variant of KL-ONE) to encode ATNs. In the context of RUS, a system for natural language parsing and interpretation, inheritance was used to organize linguistic knowledge efficiently in terms of grammatical categories. This frame-based representation was used by a process called incremental description refinement, which first determined which descriptions were compatible with an object known to have a set of properties, and then refined this set of descriptions as more properties become known. Subsequent work by Brachman and Schmolze (1985) used PSI-KLONE to translate the ouput of the RUS parser into KL-ONE representations of literal meaning. An inheritance network that the authors refer to as a "syntaxonomy" is used to encode information about syntactic categories.

A rather similar view of language processing is to be found in the Conceptual Grammar of Steels (1978) and Steels and De Smedt (1983). This approach adopted a single frame-based grammar representation for a variety of language processing tasks and for all types of linguistic knowledge. General inference mechanisms based on constraint propagation used the frames, organized in inheritance hierarchies, in generation and parsing. De Smedt (1984) went on to use generic function application to provide one of the earliest illustrations of the descriptive power of default inheritance networks for morphology in a treatment of Dutch verbs, an analysis that is extended to adjectival and nominal forms in De Smedt and de Graaf (1990). In the same paper, the authors indicate how inheritance techniques can be applied to a unification-based formalism called Segment Grammar (Kempen 1987), which is intended to facilitate incremental syntactic processing.

Attempts to reconcile inheritance with unification grammars began in the mid-1980s. Shieber (1986b, p. 57ff) noted that the provision of lexical "templates" in PATR amounted to a language for defining monotonic multiple inheritance networks. He drew attention to the possibility of adding a nonmonotonic "overwriting" operation to PATR and commented that "the cost of such a move is great, however, because the use of overwriting eliminates the order independence that is so advantageous a property in a formalism" (1986, p. 60). In a subsequent implementation of PATR, Karttunen (1986) makes all D-PATR templates subject to overwriting. The very similar notion of "priority union" is introduced in the context of LFG by Kaplan (1987, p. 180). These ideas are developed by Bouma (this issue) who gives a definition of default unification on the basis of a logic for features.

Kameyama (1988) uses PATR-style templates to build a multiple inheritance multilingual lexicon to support Categorial Unification Grammar descriptions of Arabic, English, French, German, and Japanese nominals. Although the system described is monotonic, there is a footnote suggesting a move toward a default inheritance system to deal with marked constituent orders (p. 202, n10).

Of all the unification-based grammar formalisms, it is HPSG which has thus far led to the greatest use of inheritance networks, both default and monotonic. Flickinger, Pollard, and Wasow (1985) proposed a treatment of lexical organization in which English subcategorization frames and inflectional morphology were handled within a default multiple inheritance network implemented in HPRL. They pointed out that such an approach took care of morphological "blocking" phenomena "largely for free" (1985, p. 267).[3] In his 1987 Ph.D. dissertation, Flickinger goes on to provide a monograph length inheritance treatment of the syntactic and morphological information embodied in the English lexicon. His analysis crucially presupposes machinery for multiple default inheritance. Like Shieber (1986b, pp. 60–61), he notes the problem that contradictory attribute values pose for such machinery and entertains the hypothesis that the relevant links "should be disjoint in the set of attributes for which they support inheritance" (1987, p. 61). Flickinger's thesis is probably the most detailed discursive application of a default inheritance framework to the lexicon. In their paper in the present issue, Flickinger and Nerbonne (in press) extend the analysis further still so as to encompass some of the trickiest and most-debated data in the syntax of English.

Pollard and Sag (1987), in the first book-length presentation of HPSG, treat the lexicon as a monotonic multiple-inheritance type hierarchy. They implicitly reject the use of an "overriding mechanism" (p. 194, n4) in favor of a variety of restrictions designed to prevent overgeneration, together with a nonmonotonic formulation of lexical rules (pp. 212–213). A concern to preserve monotonic inheritance in HPSG is likewise evident in more recent work, such as Carpenter and Pollard (1991) and Zajac (this issue).

Monotonic multiple inheritance type hierarchies figure in a good deal of recent work in unification-based grammars. Examples include papers by Porter (1987), Emele and Zajac (1990), and Emele et al. (1990), who all use a semantics based on Aït-Kaci (1984); the use of sorts in Unification Categorial Grammar (Moens et al. 1989); the CLE project (Alshawi et al. 1989) and theoretical work by Smolka (1988).

Default multiple inheritance also figures centrally in a couple of grammatical frameworks. One is Hudson's (1984, 1990) Word Grammar, and a detailed exposition is provided by Fraser and Hudson in this issue. Word Grammar is a feature-based variant of dependency grammar, one that makes pervasive use of a (multiple) inheritance relation. The latter is unusual in that stipulated exceptions do not automatically override an inherited default: the latter has to be explicitly negated if the grammar requires its suppression (compare Flickinger, Pollard, and Wasow's (1985) approach to "blocking," noted above).

3 The existence of an irregular form typically means that the corresponding regular form is not a permissible option. This is known as "blocking."

The other is ELU (Russell et al. in press), which extends a PATR-like grammar formalism with a language for defining default multiple inheritance networks for the lexicon. Inspired by CLOS, an object-oriented extension of Common LISP, they adopt prioritized inheritance to escape the problem caused by conflicting inherited information. Russell et al. (in press) illustrate their approach with ELU analyses of English and German verbal morphology.

Evans and Gazdar (1989a) outline the syntax and theory of inference for DATR, a language for lexical knowledge representation, and (1989b) they provide a semantics for the language that is loosely based on the approach taken by Moore (1985) in his semantics for autoepistemic logic. DATR allows multiple default inheritance but enforces orthogonality. Evans et al. (in press) show how DATR can also be used to encode certain kinds of prioritized inheritance. Unlike ELU and the Word Grammar notation, DATR is not intended to be a full grammar formalism. Rather, it is intended to be a lexical formalism that can be used with any grammar that can be encoded in terms of attributes and values. Kilbury et al. (1991) show how a DATR lexicon can be linked to a PATR syntax, while Andry et al. (in press) employ a DATR lexicon in the context of a Unification Categorial Grammar. The use of DATR to describe morphology is illustrated, for Latin, by Gazdar (in press) and in the fragments of Arabic, English, German, and Japanese included in Evans and Gazdar (1990).

All of our discussion thus far has presupposed the use of inheritance networks to store essentially static information. But, following the precedent set by Brachman and Schmolze (1985), a number of researchers have begun to explore their utility in language processing itself. Thus, for example, van der Linden (this issue) exploits the structure of the network in order to avoid premature lexical disambiguation and to identify lexical preferences during incremental parsing with a Lambek categorial grammar. And Vogel and Popowich (1990) add a new twist to the now familiar "parsing as deduction" strategy: instead of construing parsing as, for example, inference in a Horn clause logic, they describe an HPSG parser that operates by means of path-based inference over an inheritance network.

3. Phonology, Orthography, and Morphophonology

Computational phonology is perhaps the youngest and least studied branch of NLP. But notions of default have played such a prominent role in linguistic discussion of the area that it is not surprising that default inheritance networks have found a place in this subfield right from the start.

Thus Gibbon and Reinhard have made extensive use of DATR networks to describe lexical morphophonological phenomena such as German umlaut, Kikuyu tone, and Arabic vowel intercalation (Gibbon 1990b, in press; Reinhard 1990; Reinhard and Gibbon 1991). And Daelemans (1987a,b, 1988) uses the object-oriented knowledge representation language KRS to implement default orthogonal inheritance networks for the lexical representation of phonological, orthographic, and morphological knowledge of Dutch and shows how such a lexicon architecture can be used for both language generation and automatic dictionary construction. The work of Calder (1989) and his associates at Edinburgh and Stuttgart on "paradigmatic morphology" also fits within this tradition in that it invokes a restricted kind of default orthogonal inheritance for morphophonological description. However, the emphasis in this work is on the use of string unification to define morphological operations rather than on the default structure of the lexicon per se. In subsequent work, Calder and Bird (1991) use a general nonmonotonic logic to give a formal reconstruction of "underspecification phonology" (Archangeli 1988).[4]

4. Semantics and Pragmatics

> Given that knowledge representation was principally driven by natural language concerns right up to the beginning of the decade, one would have expected substantial progress to have been made in the 1980s on knowledge representation support for natural language semantics. This seems not to have been the case (Brachman 1990; p. 1088).

If one compares the progress made in morphology and syntax in NLP in the 1980s, then Brachman's judgment is surely correct. And yet there has been a steady tradition of using semantic networks in the service of natural language understanding that goes back at least as far as Simmons (1973). Much of the work in this tradition has concerned itself with domain and world knowledge relevant to disambiguation and to drawing inferences from what is said, but not to the semantic representations of words, phrases and utterances per se. Exceptions to this generalization are not hard to find, however.

For example, Barnett et al. (1990) use the same language (CycL) for linguistic semantic representation as is used in the encyclopedic inheritance network for which they are providing a natural language interface. And Jacobs (1986, 1987) proposes a uniform hierarchical encoding of both linguistic and conceptual knowledge in a frame-based formalism called ACE. Jacobs then uses the resulting inheritance network to give an account of metaphor, inter alia. By contrast, Allgayer et al. (1989) employ two separate inheritance networks, one for linguistic semantic knowledge and the other for conceptual knowledge, both being implemented in a KL-ONE derivative called SB-ONE.

Several of the inheritance-based linguistic knowledge representation formalisms that we have introduced in earlier sections are being used for semantic purposes. Thus Fraser and Hudson (this issue) make crucial use of the Word Grammar inheritance network to reconstruct the meanings of various types of constituent (e.g. verb phrases) that cannot be syntactically reconstructed in a dependency grammar. Weischedel (1989) uses the taxonomic language NKL (based on KL-ONE) to express selectional restrictions, while Andry et al. (in press) use DATR for the same purpose. Cahill and Evans (1990) use DATR to build up complex lambda calculus representations in the lexicon of a message understanding system. Briscoe et al. (1990) use a version of PATR augmented with defeasible templates to implement a default orthogonal inheritance network for a Pustejovskian analysis of metonymic sense extension in lexical semantics (e.g. interpreting the film in *Enjoy the film!* as *watching the film*).[5] Their approach is further elaborated in Briscoe and Copestake (1991) and Copestake and Briscoe (1991).

A semantic analog of the monotonic type hierarchies discussed above in connection with syntax is manifested in the situation theoretic "infon lattices" introduced by Kameyama et al. (1991) to deal with meaning mismatches in machine translation.

The use of inheritance networks for specifically linguistic pragmatic purposes (as opposed to general reasoning) is notable largely for its absence. The only example we know of is Etherington et al.'s (1989) proposal to represent the consequences of Gricean

4 Compare Gibbon's (1990a) use of DATR to the same end.
5 See Pustejovsky (1989, 1991).

maxims in a default inheritance network designed for fast (though not necessarily correct) reasoning.

Most recent computational linguistic work on pragmatics has tended to turn to general nonstandard logics as tools for the job, rather than their less expressive network relations. Thus Joshi et al. (1984) and Lascarides and Asher (1991) have made the case for nonmonotonic logics in formalizing Gricean maxims, while Schubert and Hwang (1989) show how a probabilistic logic might be used in story understanding. Mercer and Reiter (1982) and Mercer (1988) have employed Reiter's default logic to capture the behavior of natural language presuppositions. Perrault (1990) uses default logic to express a theory of speech acts, while Appelt and Konolige (1988) have deployed an extension to Moore's (1985) autoepistemic logic for the same purpose.

5. Concluding Remarks

Within computational linguistics, it is possible to see three distinct trends emerging. The first is the increasing employment of monotonic type lattices in unification-based grammars to constrain the space of permissible descriptions. The second is the use of a variety of general nonmonotonic logics for formalizing pragmatic components of NLP systems. And the third is the development of a variety of restricted default inheritance languages designed for the representation of phonological, morphological, syntactic, and compositional semantic properties of lexemes.

This last trend is partly driven by descriptive linguistic considerations (e.g. capturing linguistically significant generalizations) and partly by considerations of software engineering. The latter are somewhat analogous to the considerations that have encouraged the spread of object-orientation in computer science and include (i) parsimony—inheritance lexicons can be made one or two orders of magnitude smaller than their full-entry counterparts; (ii) ease of maintenance—changes or corrections will typically only need to be made in one or two nodes, not in thousands of individual entries; (iii) uniformity—several levels of linguistic description can be encoded in the same way and be made subject to the same rules of inference; (iv) modularity—multiple inheritance allows different taxonomies to apply for different levels of description; and (v) interaction—where a lexical property at one level of description (e.g. syntactic gender) depends on a lexical property at another level of description (e.g. the phonology of a word-final vowel), then this can be stated.

The work that has been done to date suggests that while default inheritance is essential for lexical networks, full unrestricted multiple inheritance is probably more of a hindrance than a help. It looks as if some version of orthogonal or prioritized inheritance will be sufficient for lexical knowledge representation. Moore and Kaplan (in Whitelock et al. 1987, pp. 62–63) have noted that lexical defaults amount to default specification (as opposed to the conjectural defaults of standard AI knowledge representation) and that they often substitute for (large) finite specifications. Likewise, Thomason (1991) has referred to lexical defaults as "a priori in a sense" or "at least stipulative or conventional" and he goes on to point out that any nonmonotonic lexical application can be converted to a monotonic one, albeit at the cost of scale. These considerations provide some limited grounds for optimism with regard to the tractability and mathematical probity of (future) languages for lexical representation. However, two cautionary notes are in order: firstly, the inheritance relation itself is not the sole source of intractability, and, secondly, existing work on inheritance lexicons has been almost wholly based on familiar European languages.

Acknowledgments
We are grateful to Rich Thomason for relevant conversation and comments.

References

Aït-Kaci, Hassan (1984). *A Lattice-Theoretic Approach to Computation Based on a Calculus of Partially Ordered Types.* Doctoral dissertation, University of Pennsylvania, Philadelphia, PA.

Allgayer, Jürgen; Jansen-Winkeln, Roman; Reddig, Carola; and Reithinger, Norbert (1989). "Bidirectional use of knowledge in the multi-modal NL access system XTRA," *IJCAI-89*, 1492–1497.

Alshawi, Hiyan; Carter, David; van Eijck, Johan; Moore, Robert; Moran, Douglas; Pereira, Fernando; Pulman, Stephen; and Smith, Arnold (1989). "Final Report: Core Language Engine," Technical Report, Project No. 2989, SRI, Cambridge.

Andry, Francois; Fraser, Norman M.; McGlashan, Scott; Thornton, Simon; and Youd, Nick (1991). "Making DATR work for speech: Lexicon compilation in SUNDIAL," *Computational Linguistics*, 18, 3.

Appelt, Douglas, and Konolige, Kurt (1988). "A practical nonmonotonic theory of speech acts." In *ACL Proceedings, 26th Annual Meeting*, 170–178.

Archangeli, D. (1988). "Aspects of underspecification theory," *Phonology*, 5, 183–207.

Barnett, Jim; Knight, Kevin; Mani, Inderjeet; and Rich, Elaine (1990). "Knowledge and natural language processing," *Communications of the ACM*, 33, 8, 50–71.

Bobrow, Robert J., and Webber, Bonnie Lynn (1980a). "PSI-KLONE," *Third Biennial Conference of the CSCSI/SCEIO*, 131–142.

Bobrow, Robert J., and Webber, Bonnie Lynn (1980b). "Knowledge representation for syntactic/semantic processing," *AAAI-80*, 316–323.

Bouma, Gosse (1992). "Feature structures and nonmonotonicity," *Computational Linguistics*, 18(2), 183–203.

Brachman, Ronald J. (1990). "The future of knowledge representation," *AAAI-90*, 1082–1092.

Brachman, Ronald J., and Schmolze, James G. (1985). "An overview of the KL-ONE knowledge representation system," *Cognitive Science*, 9, 191–216.

Briscoe, Ted; Copestake, Ann; and Boguraev, Bran (1990). "Enjoy the paper: Lexical semantics via lexicology." In *COLING-90*, 42–47.

Briscoe, Ted, and Copestake, Ann (1991). "Sense extensions as lexical rules." In *Proceedings of the IJCAI Workshop on Computational Approaches to Non-Literal Language*. Sydney.

Cahill, Lynne, and Evans, Roger (1990). "An application of DATR: the TIC lexicon," *ECAI-90*, 120–125.

Calder, Jo (1989). "Paradigmatic morphology." In *ACL Proceedings, Fourth European Conference*, 58–65.

Calder, Jo, and Bird, Steven (1991). "Defaults in underspecification phonology." In *Default Logics for Linguistic Analysis*, edited by Hans Kamp, 129–139, DYANA, ESPRIT deliverable R2.5.B, Edinburgh.

Cardelli, Luca (1984). "A semantics of multiple inheritance." In *Semantics of Data Types*, edited by G. Kahn, D. B. McQueen, and G. Plotkin, 51–67. New York: Springer.

Carpenter, Bob, and Pollard, Carl (1991) "Inclusion, disjointness, and choice: The logic of linguistic classification." In *ACL Proceedings, 29th Annual Meeting*, 9–16.

Chomsky, Noam, and Halle, Morris (1968). *The Sound Pattern of English*. New York: Harper and Row.

Copestake, Ann, and Briscoe, Ted (1991). "Lexical operations in a unification-based framework." In *Proceedings, ACL-SIGLEX Workshop on Lexical Semantics and Knowledge Representation*, Berkeley, 188–197.

Daelemans, Walter M. P. (1987a). "A tool for the automatic creation, extension and updating of lexical knowledge bases." In *ACL Proceedings, 3rd European Conference*, 70–74.

Daelemans, Walter M. P. (1987b). *Studies in language technology: An object-oriented computer model of morphophonological aspects of Dutch*. University of Leuven. Doctoral dissertation.

Daelemans, Walter M. P. (1988). "A model of Dutch morphophonology and its applications," *AI Communications*, 1(2), 18–25.

De Smedt, Koenraad (1984). "Using object-oriented knowledge-representation techniques in morphology and syntax programming," *ECAI-84*, 181–184.

De Smedt, Koenraad, and de Graaf, Josje (1990). "Structured inheritance in frame-based representation of linguistic categories." In *Proceedings of the Workshop on Inheritance in Natural Language Processing*, edited by Walter Daelemans and Gerald Gazdar, 39–47. Tilburg, The

Netherlands: ITK.

Emele, Martin C., and Zajac, Rémi (1990). "Typed unification grammars." In *COLING-90*, 293–298.

Emele, Martin C.; Heid, Ulrich; Momma, Stefan; and Zajac, Rémi (1990). "Organizing linguistic knowledge for multilingual generation." In *COLING-90*, 102–107.

Etherington, David W. (1988). *Reasoning with Incomplete Information*. London/Los Altos: Pitman/Morgan Kaufmann.

Etherington, David W.; Borgida, Alex; Brachman, Ronald J.; and Kautz, Henry (1989). "Vivid knowledge and tractable reasoning: preliminary report," *IJCAI-89*, 1146–1152.

Evans, Roger (1987). "Towards a formal specification for defaults in GPSG." In *Proceedings, Workshop on Natural Language Processing, Unification, and Grammar Formalisms*, University of Stirling, 3–8.

Evans, Roger, and Gazdar, Gerald (1989a). "Inference in DATR." In *ACL Proceedings*, 4th European Conference, 66–71.

Evans, Roger, and Gazdar, Gerald (1989b). "The semantics of DATR." In *Proceedings, Seventh Conference of the Society for the Study of Artificial Intelligence and Simulation of Behaviour*, edited by Anthony G. Cohn, 79–87.

Evans, Roger, and Gazdar, Gerald (1990). "The DATR Papers, Volume 1," Cognitive Science Research Paper CSRP 139, University of Sussex, Brighton.

Evans, Roger; Gazdar, Gerald; and Moser, Lionel (In press). "Prioritised multiple inheritance in DATR." In *Default Inheritance in the Lexicon*, edited by Ted Briscoe, Ann Copestake, and Valeria de Paiva. Cambridge: Cambridge University Press.

Fahlman, Scott (1979). *NETL: A System for Representing and Using Real-World Knowledge*. Cambridge, MA: The MIT Press.

Flickinger, Daniel P.; Pollard, Carl J.; and Wasow, Thomas (1985). "Structure-sharing in lexical representation." In *ACL Proceedings, 23rd Annual Meeting*, 262–267.

Flickinger, Daniel P. (1987). *Lexical Rules in the Hierarchical Lexicon*. Doctoral dissertation, Stanford University.

Flickinger, Daniel P., and Nerbonne, John (In press). "Inheritance and complementation: A case study of easy adjectives and related nouns," *Computational Linguistics*, 18(3).

Fraser, Norman M., and Hudson, Richard A. (1992). "Inheritance in word grammar," *Computational Linguistics*, 18(2), 133–158.

Gazdar, Gerald (1987). "Linguistic applications of default inheritance mechanisms." In *Linguistic Theory and Computer Applications*, edited by Peter Whitelock, Mary McGee Wood, Harold L. Somers, Rod L. Johnson, and Paul Bennett, 37–67. London: Academic Press.

Gazdar, Gerald (In press). "Ceteris paribus." In *Aspects of Computational Linguistics: Syntax, Semantics, Phonetics*, edited by Christian Rohrer and Hans Kamp. Berlin: Springer-Verlag.

Gazdar, Gerald; Klein, Ewan; Pullum, Geoffrey K.; and Sag, Ivan A. (1985). *Generalized Phrase Structure Grammar*, Oxford/Cambridge: Blackwell/Harvard.

Gibbon, Dafydd (1990a). "Underspecification in phonology." In *The DATR Papers*, Volume 1, edited by Roger Evans and Gerald Gazdar, 99–100. University of Sussex, Brighton: COGS.

Gibbon, Dafydd (1990b). "Prosodic association by template inheritance." In *Proceedings, Workshop on Inheritance in Natural Language Processing*, Walter Daelemans and Gerald Gazdar, 65–81. Tilburg, The Netherlands: ITK (Institute for Language Technology and AI).

Gibbon, Dafydd (1991). "ILEX: a linguistic approach to computational lexica." In Ursula Klenk, ed. *Computatio Linguae: Aufsä tze Zur algorithmischen und quantitativen Analyse der Sprache*. Zeitschrift für Dialektologie und Linguistik, Beiheft 7, 32–53.

Hetzron, Robert (1975). "Where the grammar fails," *Language*, 51, 859–872.

Horty, John F.; Thomason, Richmond H.; and Touretzky, David S. (1990). "A skeptical theory of inheritance in nonmonotonic semantic networks," *Artificial Intelligence*, 42(2–3), 311–348.

Hudson, Richard A. (1984). *Word Grammar*. Oxford: Blackwell.

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

Jackendoff, Ray (1975). "Morphological and semantic regularities in the lexicon," *Language*, 51, 639–671.

Jacobs, Paul S. (1986). "Knowledge structures for natural language generation." In *COLING-86*, 554–559.

Jacobs, Paul S. (1987). "A knowledge framework for natural language analysis." In *IJCAI-87*, 2, 675–678.

Joshi, Aravind; Webber, Bonnie L.; and Weischedel, Ralph (1984). "Default reasoning in interaction." In *Proceedings, AAAI Non-Monotonic Reasoning Workshop*, New York, 144–150.

Kameyama, Megumi (1988). "Atomization

in grammar sharing." In *ACL Proceedings, 26th Annual Meeting*, 194–203.
Kameyama, Megumi; Ochitani, Ryo; and Peters, Stanley (1991). "Resolving translation mismatches with information flow." In *ACL Proceedings, 29th Annual Meeting*, 193–200.
Kaplan, Ronald M. (1987). "Three seductions of computational psycholinguistics." In *Linguistic Theory and Computer Applications*, edited by Peter Whitelock, Mary McGee Wood, Harold L. Somers, Rod L. Johnson, and Paul Bennett, 149–188. London: Academic Press.
Karttunen, Lauri (1986). "D-PATR: A development environment for unification-based grammars." In *COLING-86*, 74–80.
Kempen, Gerard (1987). "A framework for incremental syntactic tree formation." In *IJCAI-87*, 2, 655–660.
Kilbury, James; Naerger, Petra; and Renz, Ingrid (1991). "DATR as a lexical component for PATR." In *ACL Proceedings, 5th European Conference*, 137–142.
Lascarides, Alex, and Asher, Nicholas (1991). "Discourse relations and defeasible knowledge." In *ACL Proceedings, 29th Annual Meeting*, 55–62.
Mercer, Robert E. (1988). "Using default logic to derive natural language presuppositions." In *Proceedings, Seventh Biennial Conference of the CSCSI/SCEIO*, 14–21.
Mercer, Robert E., and Reiter, Raymond (1982). "The representation of presuppositions using defaults." In *Proceedings, Fourth Biennial Conference of the CSCSI/SCEIO*, 103–107.
Moens, Marc; Calder, Jo; Klein, Ewan; Reape, Mike; and Zeevat, Henk (1989). "Expressing generalizations in unification-based grammar formalisms." In *ACL Proceedings, 4th European Conference*, 174–181.
Moore, Robert C. (1985). "Semantical considerations on nonmonotonic logic," *Artificial Intelligence*, 25(1), 75–94.
Perrault, C. Raymond (1990). "An application of default logic to speech act theory." In *Intentions in Communication*, edited by Philip Cohen, Jerry Morgan, and Martha Pollack, 161–185. Cambridge, MA: The MIT Press.
Pollard, Carl, and Sag, Ivan A. (1987). *Information-Based Syntax and Semantics*, Volume 1. Stanford/Chicago: CSLI/Chicago University Press.
Porter, Harry H. (1987). "Incorporating inheritance and feature structures into a logic grammar formalism." In *ACL Proceedings*, 25th Annual Meeting, 228–234.
Pullum, Geoffrey K. (1979). *Rule Interaction and the Organization of a Grammar*. New York: Garland.
Pustejovsky, James (1989). "Current issues in computational lexical semantics." In *ACL Proceedings, Fourth European Conference*, xvii–xxv.
Pustejovsky, James (1991). "The generative lexicon," *Computational Linguistics*, 17(4), 409–441.
Quillian, M. (1968). "Semantic memory." In *Semantic Information Processing*, edited by Marvin Minsky, 227–270. Cambridge, MA: The MIT Press.
Reinhard, Sabine (1990). "Verarbeitungsprobleme nichtlinearer Morphologien: Umlautbeschreibung in einem hierarchischen Lexikon." In *Lexikon und Lexicographie*, edited by B. Rieger and B. Schaeder, 45–61. Hildesheim: Olms Verlag.
Reinhard, Sabine, and Gibbon, Dafydd (1991). "Prosodic inheritance and morphological generalisations." In *Proceedings, Fifth Conference of the European Chapter of the Association for Computational Linguistics*, 131–136.
Russell, Graham; Ballim, Afzal; Carroll, John; and Warwick-Armstrong, Susan (In press). "A practical approach to multiple default inheritance for unification-based lexicons," *Computational Linguistics*, 18(3).
Schubert, Lenhart K., and Hwang, Chung Hee (1989). "An episodic knowledge representation for narrative texts." In *Proceedings, First International Conference on Principles of Knowledge Representation and Reasoning*, 444–458.
Selman, Bart, and Levesque, Hector J. (1989). "The tractability of path-based inheritance," *IJCAI-89*, 1140–1145.
Shieber, Stuart M. (1986a). "A simple reconstruction of GPSG." In *COLING-86*, 211–215.
Shieber, Stuart M. (1986b). *An Introduction to Unification-Based Approaches to Grammar*. Chicago: University of Chicago Press.
Simmons, Robert F. (1973). "Semantic networks: Their computation and use for understanding English sentences." In *Computer Models of Thought and Language*, edited by Roger C. Schank and Kenneth M. Colby, 63–113. San Francisco: Freeman.
Smolka, G. (1988). "A feature logic with subsorts," *LILOG Report 33*, IBM Deutschland, Stuttgart.
Steels, Luc (1978). "Introducing conceptual grammar," Working Paper 176, MIT AI Laboratory, Cambridge, MA.

Steels, Luc, and De Smedt, Koenraad (1983). "Some examples of frame-based syntactic processing." In *Een Spyeghel voor G. Jo Steenbergen*, edited by Fr. Daems and L. Goossens, 293–305. Leuven: Acco.

Thomason, Richmond H. (1991). "Inheritance in natural language," Unpublished manuscript, Intelligent Systems Programme, Pittsburgh.

Thomason, Richmond H., and Touretzky, David S. (1992). "Inheritance theory and networks with roles." In *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, edited by John Sowa, 231–266. San Mateo: Morgan Kaufmann.

Touretzky, David S. (1986). *The Mathematics of Inheritance Systems*. London/Los Altos: Pitman/Morgan Kaufmann.

Vogel, Carl, and Popowich, Fred (1990). "Head-driven phrase structure grammar as an inheritance hierarchy." In *Proceedings, Workshop on Inheritance in Natural Language Processing*, edited by Walter Daelemans and Gerald Gazdar, 104–113. ITK, Tilburg.

van der Linden, Erik-Jan (1992). "Incremental processing and the hierarchical lexicon," *Computational Linguistics*, 18(2), 219–235.

Weischedel, Ralph M. (1989). "A hybrid approach to representation in the Janus natural language processor." In *ACL Proceedings, 27th Annual Meeting*, 193–202.

Whitelock, Peter; McGee Wood, Mary; Somers, Harold L.; Johnson, Rod L.; and Bennett, Paul (1987). *Linguistic Theory and Computer Applications*. London: Academic Press.

Zajac, Rémi (1992). "Inheritance and constraint-based grammar formalisms," *Computational Linguistics*, 18(2), 159–182.