Strict Inheritance and the Taxonomy of Lexical Types in DATR

James Kilbury
Seminar für Allgemeine Sprachwissenschaft
Heinrich-Heine-Universität Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf, Germany
e-mail: kilbury@ling.uni-duesseldorf.de

Abstract

DATR is a representation language for the definition of semantic networks with defaults and multiple inheritance; its principal domain of application is the definition of lexical information in computational linguistics. DATR is restricted to the representation of defeasible information, so taxonomies stated in DATR cannot be used for the classification of new objects, which requires strict information. This paper proposes a minimal extension to DATR that partially captures the distinction between strict and defeasible information and treats strictness as a global property of static DATR theories. An algorithm is presented that classifies new objects on the basis of taxonomies stated in modified DATR.

1 Introduction

Within recent computational linguistics a great deal of attention has been directed at the role of defaults and nonmonotonicity in linguistic description and at the corresponding need for appropriate formalisms (cf Gazdar 1987, Daelemans et al. 1992). While default information has been recognized to involve all levels of language, primary interest is now focussed on the use of nonmonotonic inheritance networks for the representation of lexical information (cf Daelemans/Gazdar 1990, and *Computational Linguistics* 18:2&3 (1992) with a special issue on inheritance). The language DATR (cf Evans/Gazdar 1990, Langer/Gibbon 1992) was developed specially for the represention of lexical information in networks allowing both multiple and nonmonotonic inheritance and was intended to be used in conjunction with unification grammars (cf Shieber 1986). Besides DATR, other approaches to nonmonotonicity in linguistic description such as *default unification* (cf Bouma 1992) have been proposed. These alternative approaches are currently subject to discussion, the ultimate outcome of which is still unclear.

While certain differences between the various approaches to nonmonotonicity in linguistic description have been pointed out, it has generally received less attention that the proposed formalisms have in common their inability to capture the distinction between *strict* and *defeasible* information. The latter has the status of a *default* or *standard as-*

sumption and can be overridden by other information, while the former cannot. As Horty and Thomason (1988) have stressed, "the importance of representing defeasible information in an inheritance system has been widely recognized, but it is not enough for a system to represent *only* defeasible information [because] without the ability to represent strict information as well, the system cannot represent definitional relations among concepts."

Whether or not it is "enough" for a system to represent only defeasible information depends, of course, on the particular task that the system is expected to fulfill. One possible task involves the compact representation of information about objects in a static network, and defeasible information appears to be adequate here. But genuine classification of new objects constitutes a different task and one which does indeed require strict information.

The distinction between strict and defeasible information is closely related to the much older distinction between *essential* and *accidental* properties, which is central to Aristotelian philosophy and to grammatical theory influenced by the latter. The specification of essential properties of a class constitutes strict information, while defeasible information can only involve accidental properties. Aristotle's methods, which he developed in *Categoriae* [Categories] and applied to zoological taxonomy in such works as *Historia animalium* [The History of Animals] and *De partibus animalium* [On the Parts of Animals], were adopted by the grammarians Dionysius Thrax (ca. 100 B.C.) and Apollonius Dyscolus for their taxonomy of word classes in Greek, and their classification served in turn as the basis for Priscian's (5th century A.D.) classification for Latin. Mediaeval modistic grammar embodies two layers of Aristotelian influence since, as Robins (1990: 84) notes, it integrates the grammatical tradition of Priscian with the [Aristotelian] system of scholastic philosophy.[1] Thus, Thomas of Erfurt (ca. 1350) follows a descriptive method for all word classes according to which he first describes the essential modes of signifying and then the accidental modes (cf Bursill-Hall in Thomas of Erfurt 1972: 53-54).[2]

Definitional relations involving strict information about essential properties are needed in particular when further classes are added to a given taxonomy in order to classify new individuals. *Distinctive* properties of classes must be essential. Within the subclass VERTEBRATE of ANIMAL, BIRD and MAMMAL are distinguished from FISH and REPTILE by being *warm blooded*; MAMMAL is distinguished from BIRD by the es-

[1]St. Thomas Aquinas (1963: 23-24) relates essence to definition as follows: "Essentia enim uniuscuiusque rei est illud quod significat definitio eius. Hoc autem est idem cum re cuius est definitio, nisi per accidens, inquantum scilicet definito accidit aliquid quod est praeter definitionem ipsius; sicut homini accidit albedo praeter id quod est animal rationale et mortale; unde animal rationale et mortale est idem quod homo, sed non idem homini albo inquantum est album." [The essence of anything is that which its definition signifies. This is identical with the thing of which it is the definition, unless per accidens something is added to the thing defined over and above its definition. Thus whiteness is added to man, over and above the fact that he is a rational and mortal animal. Hence rational and mortal animal is the same as man; but whiteness, so far as it is white, is not the same as man. (English translation from 1947: 15)]

[2]"Modus significandi essentialis est, per quem pars orationis habet simpliciter esse, vel secundum genus, vel secundum speciem. Modus significandi accidentalis est, qui advenit parti post eius esse completum, non dans esse simpliciter parti, nec secundum genus, nec secundum speciem." [The essential mode of signifying is the one by means of which the part of speech simply possesses its essence either in accordance with its class or species. The accidental mode of signifying is what happens to the part of speech after its essence has been completed and does not naturally confer essence to the part of speech either in terms of its class or species.] (Thomas of Erfurt 1972: 148-149)

sential property *lactiferous* (giving milk), while REPTILE differs from FISH in having the property *pulmonate* (having a lung). Thus, the essential properties of BIRD make PENGUIN its subclass despite the standard assumption for BIRD that, as an accidental property, birds fly. Accidental properties of a class may be overridden by accidental or essential properties of a subclass if we assume, for example – disregarding penguins with pilot's licenses, that any penguin necessarily cannot fly. We might further conclude that the ability to swim is an essential property of PENGUIN, so that the introduction of a new class TURKEY for an hitherto unclassified individual would not lead us to specify this new class as a subclass of PENGUIN, although the turkey is likewise a bird that cannot fly.

A central point here is that a taxonomy defined exclusively in terms of defeasible information is of limited usefulness because it is arbitrary with respect to the classification of new objects (unless we have e.g. additional formal criteria to select a particular classification[3]). Nothing would prevent us from introducing SEAL as an anomalous subclass of FISH, or BAT under BIRD. In fact, new nodes for subclasses could in principle be added anywhere in the network. Other cultures may very well locate SEAL as a subclass of FISH rather than MAMMAL, but this merely means that their taxonomies select different essential properties and not that taxonomies constitute arbitrary groupings.

A computational linguist developing a taxonomy for the lexical classes or types of a language is confronted with the same problem. An adequate taxonomy should include strict definitional information that determines the assignment of an appropriate type to a new lexical item.[4] As noted above, the language DATR was developed with the particular aim of representing lexical information. While the nodes of a DATR theory build a network that can be interpreted as a lexical taxonomy, DATR is restricted in the sense discussed here in that it represents only defeasible information.

2 The individual objects of PATR-II

The individual objects we wish to describe and classify with DATR are *directed acyclic graphs* (DAGs), which represent *feature structures* and constitute the primary data structure of unification-based grammar formalisms such as PATR-II (cf Shieber 1986). The lexicon of a unification grammar assigns DAGs such as that represented by the matrix in (1) to the *lexemes* (i.e. lexical items) of a given language:

[3]Such formal criteria, in particular, minimization of the amount of information specified at a new node, are used by Light (1993) in defining an algorithm for the insertion of new nodes in multiple default inheritance networks. After showing that the problem is NP-complete, Light presents a computationally tractable approximation algorithm that proposes insertions (i.e. new node definitions) which are near optimal in terms of his weightings. The algorithm is of great interest in the context of this paper since Light applies it for the insertion of new nodes in DATR theories representing lexical information. Nevertheless, his heuristic of maximizing inheritance from a minimal number of nodes is logically independent of *definitional* relations and the problem of taxonomic classification as viewed here. A careful distinction between the two issues is a prerequisite for examining their connections and the relevance of Light's work for the aquisition of definitional information.

[4]The work presented in this paper was carried out at the University of Düsseldorf (DFG SFB 282) within the research project "Simulation of Lexical Acquisition", the aim of which is to develop procedures using contextual information to assign lexical entries to new lexemes (i.e. unknown words) encountered during parsing (cf Kilbury/Naerger/Renz 1991)

(1)

$$\left[syntax : \left[\begin{array}{ll} bar : & zero \\ head : & \left[major : \left[\begin{array}{ll} n : & yes \\ v : & no \end{array} \right] \right] \end{array} \right] \right]$$

Within PATR-II, DAGs may be described with conjunctions of *path equations* such as those in (2) for the DAG F represented in (1), where each equation in (2) specifies the value at the end of an attribute path in the DAG:

```
(2)      <F syntax bar> = zero
         <F syntax head major n> = yes
         <F syntax head major v> = no
```

Path equations may alternatively stipulate that two paths of a *reentrant* DAG lead to the same node, whatever particular node that may be. DAGs themselves are *partial functions* assigning values to a subset of the set of paths A^* that can be built from a finite set A of attribute names (cf Shieber 1986: 12ff).

3 The representation language DATR

DATR (described in detail by Evans/Gazdar 1989a, 1989b, 1990) is a declarative language for the definition of semantic networks which allows for defaults as well as multiple inheritance. Its general properties are nonmonotonicity, functionality, and deterministic search.

A DATR *theory* (or network description) is a set of *axioms* (or sentences) which are related to each other by references. Together they define a hierarchical structure, a network. Both regularities and exceptions can be expressed, regularities using default inheritance, and exceptions, overriding.

DATR axioms consist of *node-path pairs* associated with a right-hand side. This can be a *value* (atomic or sequence), or an *evaluable* DATR expression if the value is to be inherited from another node, path, or node-path pair. The following DATR theory comprising three *node definitions* encodes familiar linguistic information to illustrate some relevant DATR features:

```
(3)   LEXICAL:      <syntax bar> == zero.

      NOUN:         <> == LEXICAL
                    <syntax head major n> == yes
                    <syntax head major v> == no.

      ADJ:          <> == NOUN
                    <syntax head major v> == yes.
```

The represented information can be retrieved with special DATR queries. These also consist of a node-path pair, whose evaluation returns the value sought. With the above DATR description the following examples show sensible DATR queries and their corresponding values:

```
(4)   NOUN:<syntax head major n> = ?   yes  (atomic value)
      NOUN:<syntax head major v> = ?   no   (atomic value)
      NOUN:<syntax bar> = ?            zero (inherited from node LEXICAL)
      ADJ:<syntax head major n> = ?    yes  (inherited from node NOUN)
      ADJ:<syntax head major v> = ?    yes  (atomic value)
      ADJ:<syntax bar> = ?             zero (inherited from node LEXICAL
                                             via node NOUN)
```

Seven inference rules and a default mechanism are given for the evaluation of DATR queries. Their semantics and properties are described by Evans and Gazdar (1989b, 1990).

Like the DAGs of PATR-II, node definitions of DATR may be viewed as partial functions mapping paths of attributes into values. Despite this and their syntactic similarities, DATR and PATR-II differ in their semantics, so that there is no obvious way of relating the two formalisms. In particular, one must explicate the relation of a node in a DATR theory to the DAG that this node describes or defines (cf Kilbury/Naerger/Renz 1991), but this is not the subject of the present paper.[5]

4 An extension to DATR for strict specification of information

Here we want to consider an extension to DATR that at least partially captures the distinction of strict and defeasible information. Any such extension should preserve the characteristic qualities of DATR, in particular its functionality, orthogonality, and *longest-defined-subpath-wins* principle (cf Evans/Gazdar/Moser 1991) that avoid conflicts of multiple inheritance such as the Nixon diamond. These properties make DATR considerably more restrictive than the mixed inheritance networks discussed by Horty and Thomason (1986).

Let us introduce a connective `=+` for the specification of strict information in analogy to the connective `==` for defeasible information in so-called 'definitional' sentences of DATR. We shall interpret `=+` as being equivalent to `==` for the inference mechanism of DATR, so that `=+` does not constitute an extension *within* DATR; instead, the semantics of `=+` will be defined metalinguistically as the assertion that a given DATR theory Γ in which `=+` appears has the property of *strictness*. At the same time, `=+` marks the axioms Γ in that constitute strict specifications. Our intuition is that the strictness property should then be defined so as *not* to apply to DATR theories like that in (5):

```
(5)      F:<a b> =+ x.

         G:<> == F
           <a> == y. (where x is distinct from y)
```

The query *G:<a b> = ?* yields the value *y* because of the second axiom in the definition of node *G*, although the value of *F:<a b>* is specified strictly in the definition of *F*, and

[5] In the examples given here, the attribute paths of DAGs in PATR-II correspond directly to paths in DATR node definitions, but Kilbury/Naerger/Renz (1991) shows other, indirect ways in which this information can be encoded.

the first axiom in the definition of G states that G inherits from F with respect to paths which are extensions of $<>$, and $< a\ b >$ is an extension of $<>$. So we want this theory not to have the property of strictness.

Despite this suggestive example, it is anything but clear how the strictness property can be defined so as to capture our intuitions about strict information and otherwise be consonant with the design of DATR. Let us first introduce notation: we can write $(\alpha =+ u) \in M$ iff $M : \alpha =+ u$ is an axiom in some DATR theory Γ, $N \in \Gamma$ iff N is a node defined in Γ, and $\alpha \sqsubseteq \beta$ iff path β is an *extension* of path α, or $\alpha \sqsubset \beta$ iff β is a proper extension of α. Since a DATR node N is a partial function, we can use $\alpha \in DOM(N)$ to mean that some path α is in the domain of N. The definition of strictness will depend crucially on our concept of inheritance. We will write $N\colon \alpha \rightarrow M$ for N *inherits from* M *with respect to* α iff $N : \alpha == M$ (or $N : \alpha =+ M$) is an axiom, where $N : \alpha == M$ is equivalent to $N : \alpha == M : \alpha$ in DATR. We then define the reflexive transitive closure $\rightarrow^*$ of $\rightarrow$ as follows:

(6)

(1) $N\colon \alpha \rightarrow^* N$ for all nodes N;
(2) if $N\colon \alpha \rightarrow M$ then $N\colon \alpha \rightarrow^* M$;
(3) if $N\colon \alpha \rightarrow L$ and $L\colon \beta \rightarrow^* M$ where $\beta \sqsubseteq \alpha$, then $N\colon \alpha \rightarrow^* M$.

Strictness can be defined tentatively as the property of a DATR theory Γ formulated in (7):

(7)

$$\begin{aligned}&\forall M \in \Gamma, \forall \alpha \in DOM(M), \forall u : [((\alpha =+ u) \in M) \supset \\ &\qquad [\forall N \in \Gamma, \forall \beta \in DOM(N) : [(N\colon \beta \rightarrow^* M) \wedge (\beta \sqsubseteq \alpha) \supset (N\colon \alpha = M\colon \alpha)]]]\end{aligned}$$

Now consider the DATR theory stated in (8):

(8)

```
F:<a b> =+ x.

G:<a b> =+ y.

H:<> == F
  <a> == G
  <a b> == z.
```

This theory does not meet the strictness condition, and the query *H:<a b> = ?* returns the value *z*. As stated above, we do not intend any attempt to modify the inference mechanism of DATR so as to guarantee the precedence of strict before defeasible information, and the example suggests our reason for this decision. Whereas for theory (5) above we have the intuition that the query *G:<a b> = ?* should be evaluated to x rather than y, we have no corresponding intuition about an evaluation of *H:<a b> = ?* in (8) that would capture the notion of strict information. Once the longest-defined-subpath-wins principle of DATR is suspended, the door is open to conflicts of multiple inheritance, since there is no clear reason why the value returned from the query *H:<a b> = ?* should be either x or y if not z.

Our DATR theories representing taxonomies of lexical types are of a restricted form: While nodes in a DATR theory constituting a taxonomy may inherit from other nodes outside the taxonomy proper, those building the former are linked by axioms of the structure $N : \texttt{<>} == M$ so as to form a *tree*, a *heterogeneous unipolar tree-structured inheritance system* (cf Touretzky et al. 1987: 473). As we shall see below in §5, this tree skeleton of strict information is necessary in order for us to be able to use a taxonomy dynamically for the deterministic classification of new objects. We thus redefine the relation $N \rightarrow M$ for N inherits from M simply as $N : \texttt{<>} == M$ or $N : \alpha =+ M$ and build the corresponding reflexive transitive closure $\rightarrow^*$; strictness is then redefined correspondingly in (9):

(9)

$$\forall M \in \Gamma, \forall \alpha \in DOM(M), \forall u : [((\alpha =+ u) \in M) \supset [\forall N \in \Gamma : [(N \rightarrow^* M) \supset (N : \alpha = M : \alpha)]]]$$

This condition is easy to test algorithmically: For a given theory Γ we simply build the finite relation $\rightarrow^*$ and then, for each strict specification of $M : \alpha$ with =+ at some node M, check whether each node N such that $N \rightarrow^* M$ is defined so that $N : \alpha = M : \alpha$.

Other possible questions about the realization of strictness in DATR lose their relevance in our particular linguistic domain of application. Consider the following theory:

(10)
```
F:<a> =+ x.

G:<> == F.
  <a b> == y.
```

While we might wish strictness to exclude such theories, this would conflict with the longest-defined-subpath-wins principle of DATR, and so (10) is allowed to retain the strictness property in accordance with (9). But we have assumed an isomorphism between the paths of DATR node definitions and those of PATR-II feature structures, and we presuppose further that the latter are restricted by type definitions. As a consequence, the problem of path extension in (10) does not arise in our lexical taxonomies.

Now we can use the new connective =+ in stating a DATR theory that represents a highly simplified taxonomy of the lexical types of German:[6]

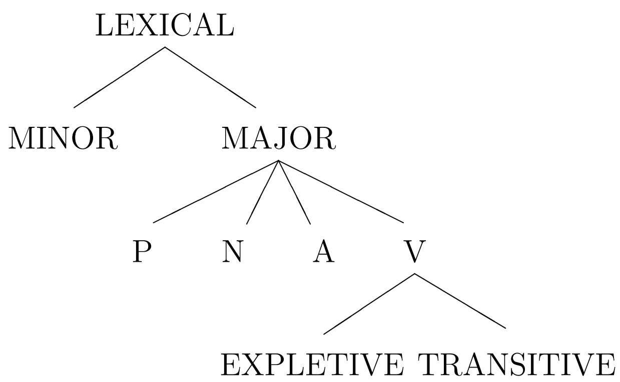
(11)
```
LEXICAL:     <syn bar> =+ zero.

MINOR:       <> == LEXICAL
             <syn head major> =+ no.

MAJOR:       <> == LEXICAL.
```

[6]The attribute *subcat* encodes the subcategorization (i.e. complements) of a verb as a list of corresponding DAGs with *first*, *rest*, and *nil*, while the values *expl* and *norm* distinguish expletive from normal subjects. The attribute name *syn* abbreviates *syntax*.

```
P:            <> == MAJOR                        % preposition
              <syn head major n> =+ no
              <syn head major v> =+ no.

N:            <> == MAJOR                        % noun
              <syn head major n> =+ yes
              <syn head major v> =+ no.

A:            <> == MAJOR                        % adjective
              <syn head major n> =+ yes
              <syn head major v> =+ yes.

V:            <> == MAJOR                        % verb
              <syn head major n> =+ no
              <syn head major v> =+ yes
              <syn subcat first syn head form> == norm
              <syn subcat rest> == nil.

EXPLETIVE:    <> == V
              <syn subcat first syn head form> =+ expl
              <syn subcat rest> =+ nil.

TRANSITIVE:   <> = V
              <syn subcat rest first syn head major n> =+ yes
              <syn subcat rest first syn head major v> =+ no
              <syn subcat rest first syn head case> == acc
              <syn subcat rest rest> == nil.
```

The inheritance tree formed by the nodes in (11) is represented graphically in (12):

(12)



The intransitive verb *schlafen* 'sleep' is assigned the type *V* because it takes a normal subject and no further complements, whereas *regnen* 'rain' gets the subtype *EXPLETIVE* of *V* because it takes the expletive subject *es* 'it' but otherwise shares the specifications of

V. The verb *sehen* 'see' has the subtype *TRANSITIVE* of *V* because it shares the strict specifications of *V* but takes a second nominal complement in accusative case.

Now we can add further node definitions to the taxonomy to cover additional verb types. The node *DATIVE* for *helfen* 'help' is added as a subtype of *TRANSITIVE* but overrides the defeasible specification of the latter with a strict specification of dative case for the object. *BITRANSITIVE* for *nennen* 'call' is likewise added as a subtype of *TRANSITIVE* but as a sister of *DATIVE* because of its two complements in accusative case. The type *VCOMP* for *versuchen* 'attempt' inherits from V but is a subtype neither of *EXPLETIVE* nor *TRANSITIVE* because of its normal subject and verbal complement, so that the corresponding node is added directly under *V*. Thus, the new node definitions of (13) may be added to the DATR theory of (11):

```
(13)   DATIVE:         <> == TRANSITIVE
                       <syn subcat rest first syn head case> =+ dat.

       BITRANSITIVE:   <> == TRANSITIVE
                       <syn subcat rest first syn head cas> =+ acc
                       <syn subcat rest rest first syn head cas> == acc
                       <syn subcat rest rest rest> =+ nil.

       VCOMP:          <> == V
                       <syn subcat rest first syn head major n> =+ no
                       <syn subcat rest first syn head major v> =+ yes
                       <syn subcat rest first syn head form> == infinitive
                       <syn subcat rest rest> =+ nil.
```

5 A procedure for type assignment with DATR taxonomies

The preceding discussion illustrates a procedure for the introduction of new type nodes in taxonomic DATR theories which relies on the distinction between strict and defeasible information, although the distinction is not embodied in the inference mechanism of DATR itself.[7] Let us say that *a DAG* Φ *meets the specifications of type node* N or that Φ *is of type* N iff the following condition holds:

(14)

$$\forall \alpha \in DOM(N), \forall \beta \in DOM(\Phi): \\ [(\alpha \sqsubseteq \beta) \wedge (\sim \exists \gamma \in DOM(N): [\alpha \sqsubset \gamma \sqsubseteq \beta]) \supset (N\!: \alpha = \Phi: \beta)]$$

Moreover, we say that N *characterizes* Φ iff (15) holds:

(15)

$$\forall \alpha \in DOM(\Phi): [\Phi: \alpha = N\!: \alpha]$$

[7]Related work on automatic classification within the context of general knowledge representation has been going on in artificial intelligence for years; cf the literature cited in Carpenter (1992: 85)

We now can informally state an algorithm that assigns a given individual DAG Φ a unique type node within a DATR type taxonomy Γ that has the strictness property:

(16)

1. Start at the root node in Γ ;
2. stop at the current node N if Φ meets the specifications of N and N characterizes Φ ; otherwise
3. descend from the current node to the unique daughter whose strict specifications are met by Φ, if such a node already exists, and continue at (2); otherwise
4. add and stop at a new daughter node N with specifications from Φ that override those defeasible specifications of the current mother node M that are not met by Φ.

If a new node is added to Γ, the algorithm thereby produces a new taxonomy Γ' which also has the strictness property. Consequently, strictness is *invariant* with respect to the algorithm.

Note that the underspecification of a node such as *MAJOR* in (11) forces continued descent according to step (3) or (4) of (16); future work must formally establish the uniqueness property referred to in step (3). The conditions still need to be defined under which the specifications of new nodes are to be made strict or defeasible; this question will also be the subject of further research.

6 Conclusion

The technique described allows us to assign lexical types represented by DATR nodes to individual DAGs associated with lexemes. We were able to obtain this result by extending a highly restricted subclass of DATR theories to reflect the distinction of strict and defeasible information. Further work is needed to investigate the relation of definitional information stated in terms of strict specifications to heuristically defined node-insertion algorithms for networks with only defeasible information.

DATR was never intended as a general-purpose knowledge-representation language, and it remains to be investigated whether the technique presented here is also useful in other domains.

References

Bouma, Gosse (1992): Feature Structures and Nonmonotonicity, *Computational Linguistics* 18: 183-203.

Carpenter, Bob (1992): *The Logic of Typed Feature Structures.* Cambridge: CUP.

Daelemans, Walter / Gazdar, Gerald (eds, 1990): *Proc. of the Workshop on Inheritance in Natural Language Processing.* ITK Tilburg, The Netherlands.

Daelemans, Walter / Gazdar, Gerald / De Smedt, Koenraad (1992): Inheritance in Natural Language Processing, *Computational Linguistics* 18: 205-218.

Evans, Roger / Gazdar, Gerald (1989a): Inference in DATR, *Proc. of the 4th Conference of the European Chapter of the ACL*, 66-71.

Evans, Roger / Gazdar, Gerald (1989b): The Semantics of DATR, A. Cohn (ed.) *AISB89, Proc. of the 7th Conference of the Society for the Study of Artificial Intelligence and Simulation of Behaviour*, 79-87. London: Pitman.

Evans, Roger / Gazdar, Gerald (eds, 1990): *The DATR Papers: February 1990 (= Cognitive Science Research Paper* 139). School of Cognitive and Computing Sciences, University of Sussex, Brighton, England.

Evans, Roger / Gazdar, Gerald / Moser, Lionel (1991): Prioritised Multiple Inheritance in DATR. University of Sussex.

Gazdar, Gerald (1987): Linguistic Application of Default Inheritance Mechanisms, Peter J. Whitelock et al. (eds) *Linguistic Theory and Computer Applications*, 37-67. London: Academic Press.

Horty, John F. / Thomason, Richmond H. (1988): Mixing Strict and Defeasible Inheritance, *Proc. of AAAI-88*.

Kilbury, James / Naerger, Petra / Renz, Ingrid (1991): DATR as a Lexical Component for PATR, *Proc. of the 5th Conference of the European Chapter of the ACL*, 137-142.

Langer, Hagen / Gibbon, Dafydd (1992): DATR as a Graph Representation Language for ILEX Speech Oriented Lexica (= Report ASL-TR-43-92/UBI), Universitt Bielefeld.

Light, Marc (1993): Classification in Feature-based Default Inheritance Hierarchies (= Technical Report 473), University of Rochester.

Robins, R. H. (^{3}1990): *A Short History of Linguistics.* London and New York: Longman.

Shieber, Stuart M. (1986): *An Introduction to Unification-based Approaches to Grammar.* Stanford, Calif.: CSLI.

St. Thomas Aquinas (1947): *Compendium of Theology*, tr. by Cyril Vollert. London: Herder. [English translation]

St. Thomas Aquinas (1963): *Compendium Theologiae*, tr. by Hans Louis Fäh and ed. by Rudolf Tannhof. Heidelberg: Kerle. [Latin text and German translation]

Thomas of Erfurt (1972): *Grammatica Speculativa*, tr. and ed. with commentary by G. L. Bursill-Hall. London: Longman. [Latin text and English translation]

Touretzky, David S. / Horty, John F. / Thomason, Richmond H. (1987): A Clash of Intuitions: The Current State of Nonmonotonic Multiple Inheritance Systems, *Proc. of IJCAI 87*, 473-482.