Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research Report
RR-92-04

Feature-Based Lexicons:

An Example and a Comparison to DATR

John Nerbonne

February 1992

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Daimler Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Krupp-Atlas, Mannesmann-Kienzle, Philips, Sema Group Systems, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Intelligent Communication Networks
- Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

Feature-Based Lexicons: An Example and a Comparison to DATR

John Nerbonne

DFKI-RR-92-04

This is the text of a talk prepared for the ASL Lexikon Workshop, Wandlitz, November 1991. It is to appear in a collection of workshop contributions, edited by Dorothee Reimann.

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-90020).

Feature-Based Lexicons: An Example and a Comparison to DATR*

John Nerbonne
nerbonne@dfki.uni-sb.de

Deutsches Forschungszentrum für Künstliche Intelligenz
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, Germany

Abstract

A FEATURE-BASED lexicon is especially sensible for natural language processing systems which are feature-based. Feature-based lexicons offer the advantages: (i) having a maximally transparent (empty) interface to feature-based grammars and processors; (ii) supplying exactly the EXPRESSIVE CAPABILITY exploited in these systems; and (iii) providing concise, transparent, and elegant specification possibilities for various lexical relationshhips, including both inflection and derivation. The development of TYPED feature description languages allows the use of INHERITANCE in lexical description, and recent work explores the use of DEFAULT INHERITANCE as a means of easing lexical development.

TDL is the implementation of a TYPE DESCRIPTION LANAGUAGE based on HPSG feature logics. It is employed for both lexical and grammatical specification. As a lexical specification tool, it not only realizes these advantages, but it also separates a linguistic and a computational view of lexical contents and supplies a development environment for lexicon engineering.

The most important competitor for feature-based lexical work is the very competent special purpose tool DATR, whose interface to feature-based systems is, however, inherently problematic. It is argued that feature-based systems (such as TDL) and DATR look compatible because of their common mathematical interpretation as graph description languages for directed graphs, but that this masks radically different modeling conventions for the graphs themselves.

The development of TDL is continuing at the German Artificial Intelligence Center (Deutsches Forschungszentrum für Künstliche Intelligenz—DFKI) in the natural language understanding project DISCO.

Keywords: Natural Language, Lexicon, Inheritance, Default Inheritance

*Thanks are due Roger Evans, Dafydd Gibbon, Hans-Ulrich Krieger and the audience at the ASL Lexikon Workshop, Wandlitz, Nov. 1991 for discussions and criticism of the ideas presented here. Needless to say, these people do not all agree with what is said.

Feature-Based Lexicons: An Example and a Comparison to DATR

Contents

1 Feature-Based Grammars

Feature-based grammars are employed nearly universally for the description of syntax and semantics in computational linguistics. We assume familiarity with this work here (Shieber 1986 is an excellent standard introductory reference to feature-based grammars) and shall employ the more particular feature theory of Head-Driven Phrase Structure Grammar (Pollard and Sag 1987 and Carpenter to appear 1992). This feature formalism appears to be sufficiently powerful for the encoding of arbitrary syntactic and semantic information in all of the linguistic theories currently in use. Pollard and Sag 1992 provides an in-depth treatment of several very complex areas of syntax and semantics, and Nerbonne et al. in preparation 1992 contains a number of applications to lexical, syntactic and semantic problems in German grammar. Here we review four aspects of the HPSG formalism we shall employ in lexical description. All of these are realized in TDL, the typed feature structure description language whose design and implementation we take up below.

1.1 Four Properties of Feature Description Languages

A first point is standard in all formalisms: feature description languages provide treatment for coreference (or reentrancy, as it is sometimes called because of its graph-theoretic interpretation). Two different attributes may be specified as having the same value, even when that value is unknown. For example, we might specify subject verb agreement in the following fashion, where the boxed numbers are just "tags" that identify the values as being the same:

$$\begin{bmatrix} \text{AGR}\ \boxed{1} \\ \text{SUBJECT}\ [\ \text{AGR}\ \boxed{1}\ ] \end{bmatrix}$$

(If descriptions of this sort are interpreted as graph descriptions, the need for this sort of specification demonstrates that the class of graphs we are interested in are not simply trees, but objects of the more general class of DIRECTED GRAPHS.) The point of coreference is the propagation of any

specification of the coreferred value. Thus above, any information about the AGR value accrues immediately to the SUBJECT|AGR value and vice versa.

Second, let us note that we shall have occasion to employ AVM's with DISJUNCTIVE value specifications. These are descriptions of objects whose value is included in one of the disjuncts. In the following example, we describe objects whose AGR|PER value must be FIRST or THIRD:

$$[\ \text{AGR|PER}\ \{\text{FIRST, THIRD}\}\ ]$$

In order to link particular choices with formal elements, we make extensive use of DISTRIBUTED DISJUNCTIONS, investigated by Backofen et al. 1990 and Dörre and Eisele 1989. This technique was developed because it (normally) allows more efficient processing of disjunctions, since it obviates the need to expand them to disjunctive normal form. It adds no expressive power to a feature formalism (assuming it has disjunction), but it abbreviates some otherwise prolix disjunctions:

$$\left[\begin{array}{l}\text{PATH1}\ \{_{\$1}\ a, b\} \\ \text{PATH2}\ \{_{\$1}\ \alpha, \beta\} \\ \text{PATH3}\ [\ \ldots\ ]\end{array}\right] = \left\{\left[\begin{array}{l}\text{PATH1}\ a \\ \text{PATH2}\ \alpha \\ \text{PATH3}\ [\ \ldots\ ]\end{array}\right], \left[\begin{array}{l}\text{PATH1}\ b \\ \text{PATH2}\ \beta \\ \text{PATH3}\ [\ \ldots\ ]\end{array}\right]\right\}$$

The two disjunctions in the feature structure on the left bear the same name '$1', indicating that they are a single alternation. The sets of disjuncts named covary, taken in order. This may be seen in the right-hand side of the equivalence. Two of the advantages of distributed disjunctions may be seen in the artificial example above. First, covarying but nonidentical elements can be identified as such, even if they occur remotely from one another in structure, and second, features structures are abbreviated. The amount of abbreviation depends on the number of distributed disjunctions, the lengths of the paths PATH1 and PATH2, and—in at least some competing formalisms—on the size of the remaining structure (cf. [PATH3] above).[1] Third, we employ a typed version of feature logic which allows the use of RECURSIVE TYPE SPECIFICATIONS of a kind found in HPSG (and UCG—cf. Moens et al. 1989), but generally not elsewhere. Types restrict the attributes on a given feature term (to an appropriate subset), as well as restricting the values which an attribute may have (to an appropriate type). In HPSG the type *sign* has an attribute (path) SYNTAX|LOCAL|SUBCAT which is restricted in value to lists of signs. This attribute encodes SUBCATEGORIZATION information, which is lexically based in HPSG, much as it is in Categorial Grammar (Bach 1988). Grammatical heads specify the syntactic and semantic restrictions they impose on their complements. For example, verbs and verb phrases bear a feature SUBCAT whose content is a (perhaps ordered) set of feature structures representing their unsatisfied subcategorization requirements. Thus the feature structures associated with transitive verbs include the information:

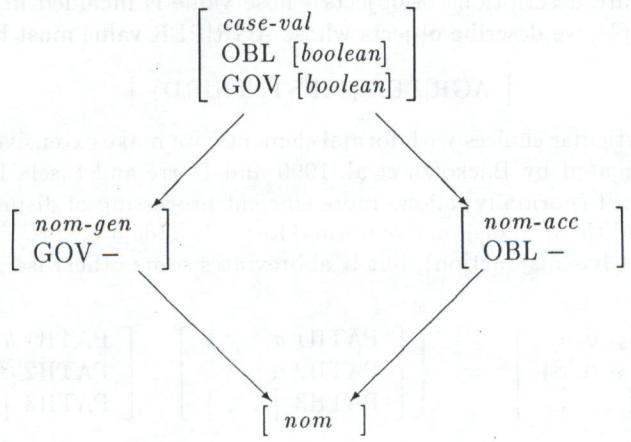
$$\left[\begin{array}{l}\textit{trans-verb} \\ \text{SYN|LOC|SUBCAT}\ \langle\left[\begin{array}{l}\textit{np} \\ \text{CASE ACC}\end{array}\right], \left[\begin{array}{l}\textit{np} \\ \text{CASE NOM}\end{array}\right]\rangle\end{array}\right]$$

(where *np* is the type of noun phrase signs, and *trans-verb* the type of transitive verb sign).

In order to appreciate the point about recursive specification, let us regard the subcategorization list as represented in [FIRST, REST] form (so that every SUBCAT either is null or occurs in [FIRST, REST] form). Then, the important point is to note that we have a type *list*, one of whose attributes, REST is restricted to values of type *list*, including the empty list. This is a recursive type specification. In general, SUBCAT is restricted to taking values which are of the type *list(sign)*—and this attribute occurs within signs. A similar recursion obtains when we define the type *tree* as a *lexical-sign* or a *phrasal sign* whose attribute DTRS (daughters) is a list of signs of the type *tree*. We shall employ recursive type specifications in a proposal for the representation of derivational relationships.

[1] Cf. Backofen et al. 1990 for a discussion of a third advantage of distributed disjunctions, namely a normal increase in processing efficiency. Cf. Krieger and Nerbonne 1992 for comments on the use of coreference in combination with disjunction.

Fourth, we shall employ inheritance extensively in order to describe lexical relationships succinctly (and thereby improve maintainability and modifiability). One of the greatest virtues of inheritance is that one can readily visualize relationships, e.g.:



As the diagram suggests, we specify inheritance relationships to indicate that information is shared between the bequeathing and the inheriting node. The feature value at any given node is just the unification of its local specifications with those it inherits. The diagram indicates, e.g., that the type *nom* inherits from both *nom-gen* and *nom-acc*. As such, it is heir to the feature, feature-value and type specifications these provide. An instance of the type *nom* must therefore minimally be defined to bear the inherited featues OBL and GOV, both with the value –:

$$\begin{bmatrix} \textit{case-val} \\ \text{OBL} \ - \\ \text{GOV} \ - \end{bmatrix}$$

And this is just the unification of the specifications of the bequeathing nodes, *nom-gen* and *nom-acc*:

$$\begin{bmatrix} \textit{case-val} \\ \text{OBL} \ - \\ \text{GOV} \ [\textit{boolean}] \end{bmatrix} \quad \begin{bmatrix} \textit{case-val} \\ \text{OBL} \ [\textit{boolean}] \\ \text{GOV} \ - \end{bmatrix}$$

1.2 Default Inheritance

The best inheritance mechanisms for lexical information have been Flickinger et al. 1985's work on "structured lexicons", Evans and Gazdar 1990's work on DATR, and Pollard and Sag 1987's Chapter 8 on lexical inheritance. Both Flickinger's work and DATR aim to supplement feature-based grammars, but both require an explicit translation step to convert lexical information into grammatical features. Furthermore, they are both hampered in expressive power, so that they accommodate some sorts of information poorly, even information which is standardly found in feature systems, e.g., disjunction, negation, and complex feature structures used as values.

Most work on feature structures, on the other hand, has failed to allow the use of DEFAULTS or OVERWRITING, which is crucial for a practical lexical tool.[2] The key advantage of default specifications is that they allow the description of SUBREGULARITIES, classes of items whose properties are largely, but not perfectly regular. In a system with default inheritance, these may be regarded not as anomalous, but rather as imperfectly regular, or regular within limits. We shall employ default inheritance regularly, perhaps most crucially in the specification of derivational relations.

Default specifications are appropriate for the description of the innumerable linguistic phenomena which are partially regular—or, which at the present stage of our linguistic knowledge, cannot be succinctly described in a completely regular way. It may ultimately be a philosophical

[2]But cf. Pollard and Sag 1987, p.194, Note 4; Sag and Pollard 1987, p.24; and Shieber 1986, pp.59-61.

point whether the former class is neglibly small, but we can be certain the the latter will remain large for many years, even decades to come. Thus, even if it turns out that the use of default specification has NO scientific significance, its utility in grammar development—for both current theory and practice—should not be underestimated. Some examples of exceptional behavior for which default specification is appropriate include: irregular and missing inflectional forms, irregular derivational form or meaning (Krieger and Nerbonne 1992), irregular syntax (Flickinger 1987, pp.64-66); irregular subcategorization specifications (Flickinger and Nerbonne 1991 or 1992). (Cf. Flickinger et al. 1985; Gazdar 1987; and Gazdar 1990 for further cases supporting the use of defaults in lexical specifications.) The use of defaults has seemed suspicious within the context of feature systems because these were developed (in part) to allow monotonic processing of linguistic information, and the use of defaults leads to nonmonotonicity.

But, as Bouma 1990, p.169 points out, the use of lexical defaults is a fairly harmless form of nonmonotoncity, since the lexicon is nonmonotonic only with respect to lexical development—the syntactic use of information specified via lexical default leads to none of the problems associated with nonmonotonic reasoning; e.g., inferences about phrases never need to be retracted, and the NL system may be configured to be perfectly monotonic at run-time. If we employ default inheritance for the specification of lexical information, then the inheritance hierarchy as a whole does NOT correspond to a subsumption or subtyping hierarchy—information may be overwritten which renders subsumption invalid. Care needs to be taken that the two notions of hierarchy—the classes involved in the default inheritance relationship and the feature structure types defined there—not be confused (cf. Cook et al. 1990). The mechanism we shall employ for the default combination of lexical information is the DEFAULT UNIFICATION developed by Bouma 1990; we may employ this within the lexicon, even while eschewing its use for parsing and generation.

Besides TDL, which we report on below, cf. Russell et al. 1992 and Copestake 1992 for further information on the use of default unification-based inheritance in computational lexicology.

2 Lexical Tasks

Four significant tasks of the lexicon are support in the description of inflection, derivation, syntax and semantics. Here we illustrate useful techniques in feature-based lexical description without attempting to treat the matter exhaustively. Our point is the relative ease with which descriptions are generated, particularly when this is compared to well-known alternatives.

2.1 Inflection

A central task in the description of inflectional morphology is the description of paradigms, such as that of the weak present indicative:

	sg	pl
1st	+ e, *kriege*	+ en, *kriegen*
2nd	+ st, *kriegst*	+ t, *kriegt*
3rd	+ t, *kriegt*	+ en, *kriegen*

2.1.1 A Distributed Disjunctive Treatment

Surely one of the most interesting insights of DATR is its treatment of the inflectional paradigm (even if this is not often commented on). In DATR a paradigm is characterized as a set of further specifications of an abstract lexeme (for an example, cf. §4 below). We can provide a very similar analysis using the distributed disjunctions introduced in §1 above:

$$\left[\begin{array}{l} \text{MORPH} \left[\begin{array}{l} \text{STEM } \boxed{2} \\ \text{ENDING } \boxed{3}\{_{\$1}\text{ ``e'', ``st'', ``t'', ``n'', ``t'', ``n''}\} \\ \text{FORM } \boxed{2}\&\boxed{3} \end{array}\right] \\ \\ \text{SYN|LOCAL|HEAD|AGR} \left\{_{\$1} \left[\begin{array}{l}\text{PER 1ST}\\ \text{NUM SG}\end{array}\right], \left[\begin{array}{l}\text{PER 2ND}\\ \text{NUM SG}\end{array}\right], \ldots, \left[\begin{array}{l}\text{PER 3RD}\\ \text{NUM PL}\end{array}\right]\right\} \end{array}\right]$$

This is a large disjunction, which in disjunctive normal form would have a first element whose ending is "e", and whose AGR|PER value is 1ST, etc. Thus a lexeme is characterized as the disjunction of its individual paradigmatic variants. In the case of lexemes with exceptional paradigms, default inheritance is employed to inherit what is common and overwrite what is exceptional. Cf. Krieger and Nerbonne 1992 for examples and discussion.

2.1.2 A Relational Treatment

If the feature formalism allows in additon to the equality constraints allowed in all theories, more general relational constraints as well, then other interesting possibilities for the description of inflection arise (the sketch here derives from Kathol 1991a.) Under Kathol's scheme one provides a single nondisjunctive feature description which describes the lexeme for weak (present indicative) verbs:

$$\left[\begin{array}{l} \text{MORPH} \left[\begin{array}{l} \text{STEM } \boxed{1} \\ \text{FORM } \boxed{2} \end{array}\right] \\ \text{SYN|LOCAL|HEAD|AGR} \left[\begin{array}{l} \text{PER } \boxed{3} \\ \text{NUM } \boxed{4} \end{array}\right], \end{array}\right]$$

Note that the feature term no longer contains the function symbol '&' "morphologically append" (and that feature terms can in general be free of function and relation symbols), but at the cost of requiring additionally that the following verbal-inflection relation holds:

$$\mathbf{R}_{\text{pres-weak-ind-verb-infl}}(\boxed{1}, \boxed{2}, \boxed{3}, \boxed{4})$$

where $\mathbf{R}_{\text{pres-weak-ind-verb-infl}}$ is constrained so that

$$\begin{array}{l} \boxed{2} = \boxed{1} \,\&\, \text{"e"} \;\text{ iff } \boxed{3} = \text{1ST} \wedge \boxed{4} = \text{SG} \\ \boxed{2} = \boxed{1} \,\&\, \text{"st"} \;\text{ iff } \boxed{3} = \text{2ND} \wedge \boxed{4} = \text{SG} \\ \boxed{2} = \boxed{1} \,\&\, \text{"t"} \;\text{ iff } (\boxed{3} = \text{3RD} \wedge \boxed{4} = \text{SG}) \vee (\boxed{3} = \text{2ND} \wedge \boxed{4} = \text{PL}) \\ \boxed{2} = \boxed{1} \,\&\, \text{"en"} \;\text{ iff } (\boxed{3} = \text{1ST} \vee \boxed{3} = \text{3RD}) \wedge \boxed{4} = \text{PL} \end{array}$$

Kathol's proposal is neat in dividing feature terms and relational constraints into separate bundles, and it also allows a kind of "inheritance", which we demonstrate with the modal verb *sollen*—this is like the weak verbs except in having a *emptyset* ending in 1st and 3rd singular.

$$\mathbf{R}_{\text{pres-modal-ind-verb-infl}}(\boxed{1}, \boxed{2}, \boxed{3}, \boxed{4})$$

where $\mathbf{R}_{\text{pres-modal-ind-verb-infl}}$ is constrained so that

$$\begin{array}{l} \boxed{2} = \boxed{1} \;\text{ iff } (\boxed{3} = \text{1ST} \vee \boxed{3} = \text{3RD}) \wedge \boxed{4} = \text{SG} \\ \boxed{2} = \boxed{5} \;\text{otherwise, with } \mathbf{R}_{\text{pres-weak-ind-verb-infl}}(\boxed{1}, \boxed{5}, \boxed{3}, \boxed{4}) \end{array}$$

The treatment using distributed disjunctions also allows the default inheritance of paradigmatic specifications, but it is not as neat (cf. Krieger and Nerbonne 1992 for details). We have not discussed the the lexicon/morphophonemic interface here, which in both the distributed disjunctive treatment and the relational treatment suggested by Kathol is constituted by references to a "morphologically append" function. The proper interface of lexical and morphological specification is a subject of ongoing research in DISCO.

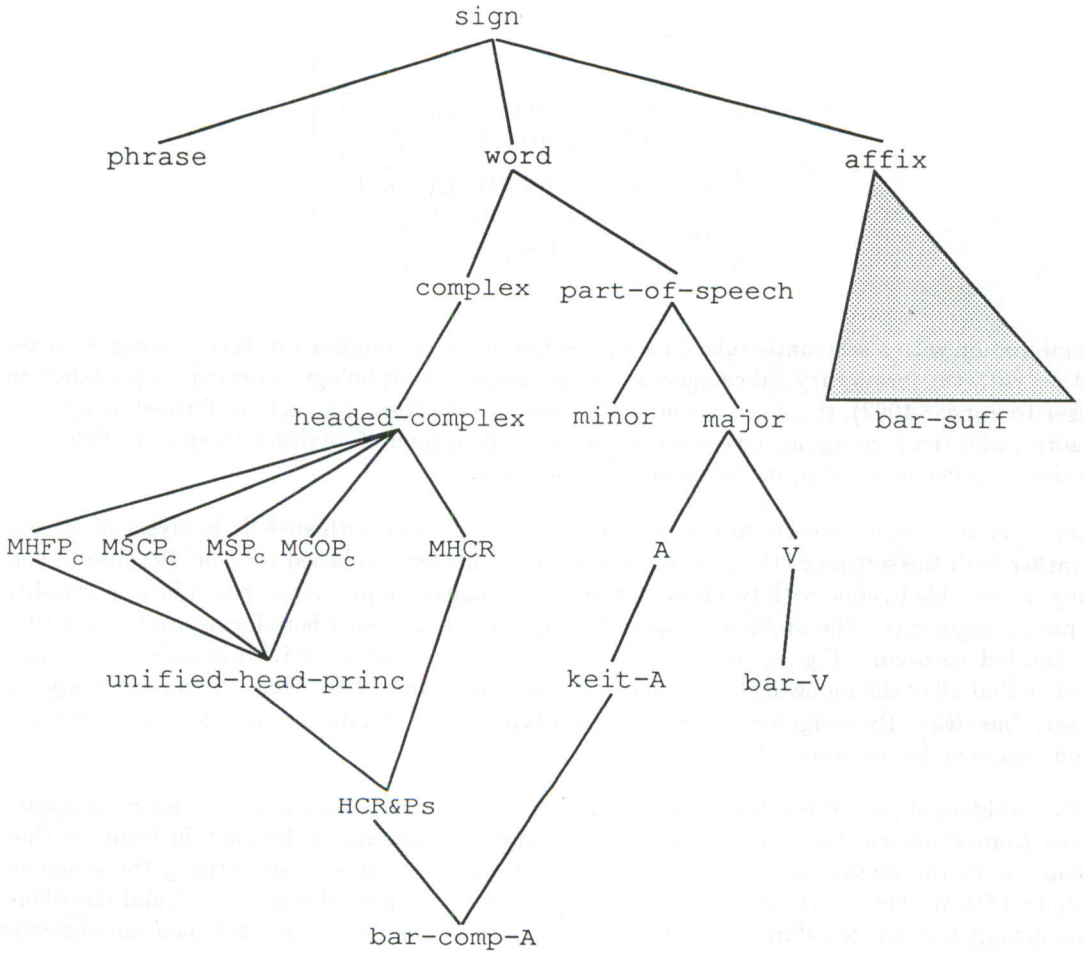


Figure 1: Structure of the inheritance network in case of *-bar* suffixation, including morphological constraints (morphological head feature principle—MHFP$_c$, etc.) and rule (MHCR). Note that we impose additional LOCAL constraints on certain classes, especially on *bar-comp-A*. Note further that, although the class of adjectives formed using *-bar* inherits from *A* (adjective) and from *HCR&Ps*, it does NOT inherit from either of its component morphs—*bar-V* or *bar-suffix*.

2.2 Derivation

One of the most exciting aspects of applying feature-based techniques to lexical problems is the opportunity it provides for the treatment of derivation, long a neglected child in compuational lexicology. Derivation, when it is described, is generally seen as the result of applying "lexical rules" to lexical entries. Thus entities of two sorts—rules and entries—encode derivational relationships. We can overcome this division in a feature-based treatment.[3]

By describing derivation FRAMES with typed subcategorization requirements, we return the content of derivation to the lexicon (rather dividing it between rules and affix contents). We assume a type hierarchy like that in Figure 1.

Given an elaborate type hierarchy, we may describe a derivational affix such as German *-bar* as morphologically subcategorizing for a verb stem of an appropriate sort. We foresee an entry along the following lines (cf. Krieger to appear 1992 for more detailed specification and comments):

[3] A more extended presentation of the work in this section may be found in Krieger and Nerbonne 1992, in the section on derivation, and in Krieger to appear 1992.

$$\text{bar} \equiv \left[\begin{array}{l} \textit{bar-suff} \\ \text{MORPH} \left[\begin{array}{l} \text{FORM "-bar"} \\ \text{SUBCAT bar-V}\boxed{1} \end{array}\right] \\ \text{SYN|LOC} \left[\text{ HEAD|MAJ A }\right] \\ \text{SEM} \left[\begin{array}{l} \text{OPERATOR } \Diamond \\ \text{SCOPE } \boxed{1} \end{array}\right] \end{array}\right]$$

In combination with a schematic rule allowing the formation of complex words consisting of affixes together with the forms they subcategorize for, and several morphological principles (sketched in Krieger to appear 1992), this form accounts for several properties of adjectives formed using *-bar*: category (adjective), component verb stem, and semantics (and cf. Krieger to appear 1992 for a discusion of other properties, including subcategorization).

But very interesting aspects of this proposal have to do, not with specific analyses of affixes, but rather with the setting of the proposal within a default feature-based lexicon. Because of this setting we are able to cope with two normally crippling analytical problems: sporadic applicability and partial regularity. The problem of sporadic applicability is seen where forms are missing that are expected to occur. E.g., given the usual restriction on *-bar* adjective formation, we might expect to find all of the following forms (derived from transitive verbs): **habbar*, **legbar*, **weißbar*, **ladbar*, **meidbar*. By assigning these verbs to a type inappropriate for *-bar* derivation, we can exclude these verbs, as desired.[4]

The problem of partial regularity may be illustrated with the form *kündbar* 'subject to layoff', derived from *kündigen* 'to lay off' is regular in syntax and semantics, but not in form, so that we enter it in the lexicon as inheriting from the *-bar* schema above, overwriting the irregular attribute FORM. Thus it is subject to the same treatment of partial regularity found elsewhere in the default lexicon. Similarly, we might wish to treat *wunderbar* 'wonderful' as a semantically irregular derivative of *wundern* 'to surprise', *sichtbar* as a formally irregular derivative of *sehen* 'to see'; *kostbar* 'valuable' as a semantically irregular derivative of *kosten* 'to try, taste'. *Eßbar* 'edible' from *essen* 'to eat' is probably one of a large class of derivatives whose meaning is a bit more specific that expected, applying to objects which not merely can be eaten, but which can be SAFELY eaten. Cf. *trinkbar* 'potable', *belastbar* 'loadable', *verzehrbar* 'consumable' and *betretbar* 'tresspassable'.

2.3 Syntax and Semantics

Since the entire point of departure for applying feature-based description to the lexicon is the employment of techniques borrowed from syntax and semantics, this point should require little support.

We illustrate syntax and semantics briefly nevertheless in order to emphasize the potential complexity of this area in its interface to inflection and derivation. Below is the disjunctive description of a verb lexeme which includes both its active and passive variants.

[4] There is a potential objection here, viz., that these words should be found in the lexicon as potential, but not actual words. Without taking a stand on the desirability of this position, let us note that the problem then becomes one of marking a feature [actual ±]—a problem of partial regularity.

$$\left[\begin{array}{ll} \text{MORPH} & \left[\begin{array}{l}\text{STEM}\\ \text{PREFIX } \{_{\$1}\ \text{“”}, \text{“ge”}\}\\ \text{ENDING } \{_{\$1}\ \{_{\$2}\ \text{“e”}, \ldots, \text{“n”}\}, \text{“n”}\}\end{array}\right] \\ \text{SYN|LOCAL} & \left[\begin{array}{ll}\text{SUBCAT} & \{_{\$1}\ \langle \text{NP[ACC]}_{\boxed{4}}, \text{NP[NOM]}_{\boxed{3}}\rangle,\\ & \quad \langle(\text{PP[VON]}_{\boxed{3}},)\text{NP[NOM]}_{\boxed{4}}\rangle\ \}\\ \text{HEAD|AGR} & \left\{_{\$2}\ \left[\begin{array}{l}\text{PER 1ST}\\ \text{NUM SG}\end{array}\right], \cdots, \left[\begin{array}{l}\text{PER 3RD}\\ \text{NUM PL}\end{array}\right]\right\}\end{array}\right] \\ \text{SEM} & \left[\begin{array}{l}\text{PRED}\\ \text{SOURCE } \boxed{3}\\ \text{THEME } \boxed{4}\end{array}\right]\end{array}\right]$$

where NP[ACC] abbreviates [*np*, CASE ACC], etc. The structure above provides for two alternate (sets of) forms, active and passive. This alternation is indicated by the disjunction name '$1'. The second, passive alternative correlates with a subcategorzation in which, e.g., an optional PP[von] phrase fills the same semantic argument slot as the active subject. The first, active alternative in $1 is simply the active paradigm sketched above.[5] The point of drawing attention to this sort of example here is the reminder that seemingly simple morphological markings such as passive affixation may have quite complex syntactic and semantic correlates, and that the lexicon must be prepared to specify these.

3 TDL—As Lexicon Definition Tool

TDL—a TYPE DESCRIPTION LANGUAGE—is an implementation of typed feature description language with type inheritance which allows the defintion of HPSG grammars. It supports typed feature-based reasoning (including multiple inheritance), separates a user interface from underlying implementation, and is accompanied by a graphical development environment.

The sorts of type inference required in lexicon definition include unification, default unification or overwriting, appropriateness checking for both types and attributes, instantiation, subsumption and classification. TDL currently supports all of these except subsumption and classification. Unification is used to combine information from multiple ancestors—this must be compatible and fails otherwise; default unification is employed to combine inherited with local information (and simple overwriting is under examination for use in future versions); appropriateness checks are performed both on the values of typed attributes (those requiring specific types) and on the attributes specified within a type; and instantiation simply provides a concrete instance of a given type. The current version of TDL can only check on appropriateness if the type in question is a user-defined type. To check more generally, subsumption (subtyping) must be available. Finally, it would be useful to be able to provide the minimal type to which an untyped instance belongs. This process is known as classification. Diehl 1992 contains discussion of the inferential services provided by TDL.

TDL is implemented in CommonLisp and runs on MacIntoshes and Sun4's. A unique feature of TDL is its strict separation of user and implementer views which was achieved by defining the TDL langauge using Zebu, a LISP version of YACC, the UNIX public-domain compiler-compiler. The language was deliberately designed to resemble the specification languages popular in computational linguistics writings. Enforcing this separation allows one to provide the lexicon writer with a cleaner view of his specifications.

The TDL development environment includes several further utilities designed to immprove habitability, including a feature editor, which displays feature structures in attribute-value format; a "grapher", which provides a hierarchical view of inheritance, as well as facilities which allow

[5] Cf. Kathol 1991b for a more ambitious analysis, under which a single nondisjunctive participial form serves in both active and passive voices, so that passive is not an inflectional alternation.

partial views, e.g., of the hierarchy as it affects given feature values. The last is useful for debugging single components. Schäfer and Krieger 1992 is a terse guide to the facilities on the rather less comfortable Sun4 platform (lacking feature editor and grapher).

4 Comparison to DATR

DATR is perhaps the first lexicon definition tool with a useful notion of default inheritance AND and a rigorous declarative definition (Evans and Gazdar 1989, Evans and Gazdar 1990). It has been used in several applications in combination with feature-based grammars (Kilbury et al. 1991), and is used as a tool in interesting morphophonemic investigations (Reinhard 1989, Gibbon 1991, Gibbon and Ahoua 1991).

Like feature-based descriptions, DATR uses directed graphs to represent linguistic information (in the case of DATR, these are actually restricted to trees, as we shall see). Figure 2 presents a sample graph representing (some of) the information in the lexeme *come*, together with a describing theory (of which there are, in general, many).[6]

Given the fact that both feature formalisms and DATR have direct interpretation as graph description languages, it may seem surprising that their interface should be complicated. They even resemble each other superficially in the atomic case, each assigning values (graph nodes) to attributes (directed edges):

$$[\text{AUX } -] \qquad < \text{aux} > == -$$

But this superficial resemblance is deceptive; for example, the feature formalism statement requires that the path AUX ends at the node $-$ (there can be no continuation), while the DATR statement may describe a graph where the path $<$ aux $>$ leads to a node labeled $-$, but which may well have further paths leading from it—something the feature formalism statement is incompatible with.

4.1 Graphs and Trees

The fundamental differences stem from different design goals of the two formalisms. DATR is designed to allow the expression of defaults, which may be understood in the following way: we view an entire DATR theory as describing a graph (cf. Figure 2), where attributes correspond to edge labels, and values to node labels. The function assigning labels to nodes (values) is specified only partially, but from it we derive a complete labeling by allowing node labels to be "inherited" along directed edges where no node label is already present. These inherited node labels are default values. It is easy to see how they are overridden by additional specification (the condition on the inheritance along directed edges is that there be no such specification). It is crucial to note that this scheme cannot work (without complication) if two directed edges point to the same unlabeled node; in that case the node would potentially inherit conflicting labels. The DATR solution to this is simple and efective: no such graphs may be described, only trees.

Now several differences between DATR and feature formalisms become apparent. The prohibition against directed graphs in which distinct edges point to the same node is effectively a prohibition against what feature theories have called "reentrancy", and what is expressed in feature formalisms as path equality (PATR-II) or labeled values (HPSG) or variables (Smolka 1988). There cannot be a path equality statement in DATR with the same semantics as that in feature formalisms, since this would mean that genuine directed graphs—and not merely trees—were being described. What looks like a path equality relation, '==', is an assignment operator in DATR, assigning node labels to nodes at the end of named paths, and there are no path equalities—variables, coreferences or reentrancies—in DATR, as there are in feature formalisms.

A closely related point is the denotation of path expressions: both feature formalisms and DATR allow the expression of path equivalence, but in the case of feature formalisms this is interpreted as node (and thus subgraph) identity, while in the case of DATR this is interpreted as identity of node label (normally, linguistic form). We might summarize that feature paths denote subgraphs while DATR paths denote node values.

[6] The graph and theory are borrowed from a presentation by Roger Evans on DATR at the ACQUILEX Workshop on Default Inheritance in the Lexicon, April, 1991.

A DATR theory

VERB: < syn aux > == no
< syn cat > == V
< morph pres fin sing 3 > == ("< morph root >" *s*)
< morph past > == ("< morph root >" *ed*)
< morph pres part > == ("< morph root >" *ing*).

COME : <> == **VERB**
< morph root > == *come*
< morph past > == *came*
< morph past part > == < morph root > .

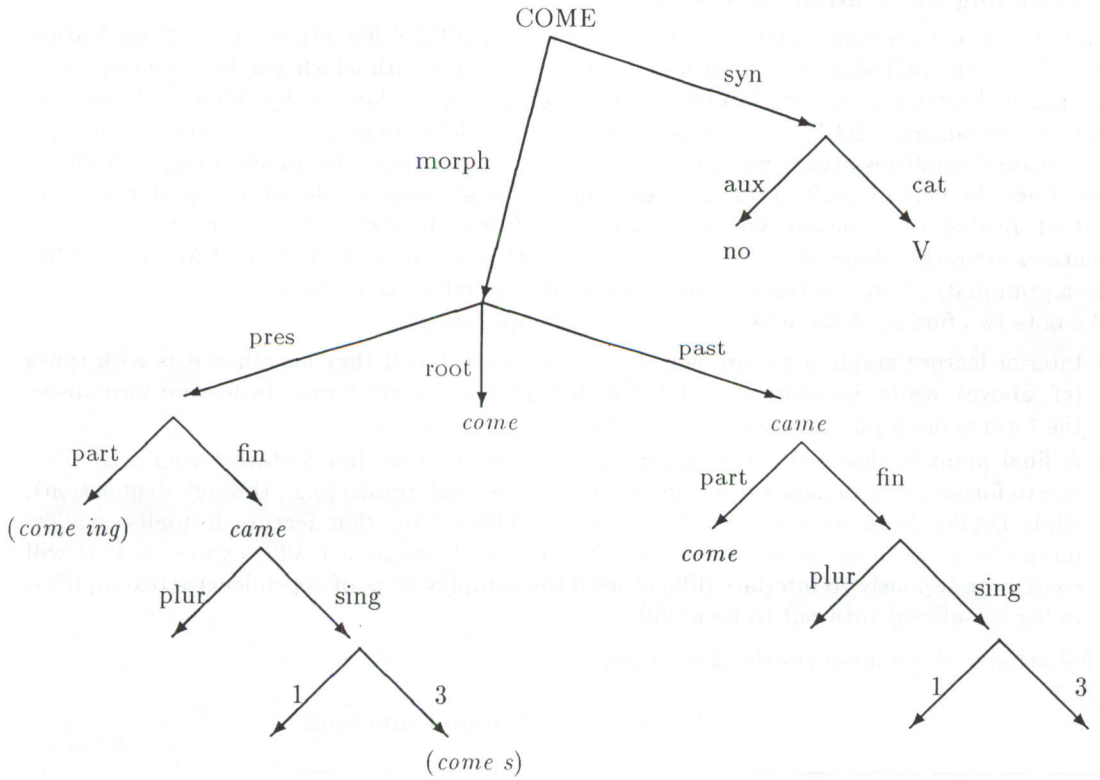


Figure 2: A DATR theory describing a graph modeling the English lexeme *come* (the VERB graph is not shown). All nodes are at least implicitly labeled. To find the implicit label of a given node, just follow edges back to the nearest labeled node—the value. Thus the value of the attribute path mor|pres|fin is *come* (a node is nearest itself), and the value at mor|pres|fin|plur is also *come*. Please note that because of this mechanism, DATR graphs must be trees—i.e., given a node, we must be able to ask from which UNIQUE nearest node is there an edge directed to it. This ensures that implicit values are uniquely assigned. We also note that this inheritance mechanism exploits the relative depth—number of edges—between nodes, and that lists are the only complex values, used here to model strings of morphemes still subject to morphophonemic processing.

A corollary of the last point concerns the complexity of values. As we've just noted, feature formalisms allow one to denote feature graphs, while DATR expressions always denote node values—of more limited complexity (lists seem to be all that is allowed).

Returning to the question of trees vs. directed graphs, it may not be superfluous to add that the use of genuine directed graphs (and not merely the special case of trees) is crucial in feature formalisms because the confluence of graph edges models the sharing of linguistic properties, which is frequently exploited in feature-based description (cf. above). DATR must use trees so that a unique value for a node's ancestor is guaranteed. The default inheritance mechanism in DATR depends on this.

It is also worth noting that feature-based theories have a profoundly different response available to the problem of inheriting nonidentical values: rather than forbid the inheritance, or require that one value be given up in favor of the other, the feature-based theory may work with the unification (or default unification; cf. above) of the values.

4.2 Modeling Conventions and Other

Beyond the need to accommodate default inheritance, DATR differentiates itself from feature theories due to the radically different MODELING CONVENTIONS with which graphs are interpreted. As Figure 2 illustrates, the use DATR makes of graphs is fundamentally different from that in feature formalisms. DATR uses relative depth to model default preference (cf. Figure 2), while feature formalisms attach no significance to the relative depth (number of edges) between nodes. Since the default preferences involved concern the forms of words, we group attributes in DATR where they tend toward syncretism (sharing a form). In feature formalisms the modeling conventions are rather different, leading one to group attributes (e.g., PER and NUM under AGR) where a grammatical process (agreement) makes common reference to them.

We note two further differences for the sake of completeness:

- Interior feature graph nodes are not always decorated, but if they are, then it is with types (cf. above), while the values associated with DATR nodes are forms. In feature formalisms the form is often just another value (cf. examples in 2.1).
- A final point is clear even if its import is not, and that is that feature formalisms allow one to formulate statements which generalize over several graphs (e.g., through disjunction), while DATR disallows this sort of statement. This means that feature formalisms allow more concise expressions and are more difficult to evaluate than DATR expressions. It will contribute seriously to interface difficulties if the complex sorts of dependencies (exemplified in Sec.2.3 above) turn out to be useful.

The following table summarizes the differences:

	DATR	Feature Formalisms
genuine dags? (edges directed at same node)	no	yes
value at path	node	subgraph
denotation of node labels	forms	type (or no labels)
complex values?	lists	feature structures
significance of depth	default preference	none
disjunction, etc.	no	yes

4.3 Reconciliation?

Could one revise DATR to make it more compatible with feature structure work? More recent work (Evans et al. 1992) attempts to relax the DATR prohibition against multiple inheritance, even while retaining the basic denotational rules (under which paths denote the values of the nodes they lead to) and one can charitably imagine the introduction of variables to DATR which would allow the expression of coreference (but note that this would have to be a second node decoration—it could not replace form).

The significance of depth is a much more fundamental problem, since it has to do with the modeling of linguistic phenomena which need not turn out to be the same. In feature formalisms, attributes are specified as in common (as attributes of a common node) when there are rules which treat them as identical. This is why PER and NUM are grouped under AGR (to simplify the statement of subject-verb agreement), and why AGR and CASE are grouped under N-AGR (simplifying the statement of DET-$\bar{\text{N}}$ agreement), etc. DATR suggests a (potentially) completely orthogonal organization in which attributes are specified as in common in case they have similar morphological forms.

It is an open question whether ALL linguistic generalizations tend to follow the lines of morphological form. Surely they need not, as is shown by the case of the German first and third person plural verb forms—which are identical for all verbs in all tenses and moods. This has not led to sugestions for collapsing these attributes in general because there are generalizations, e.g., about person in reflexive and pronominal NP's, which treat the identical FORMS as representing distinct INFORMATION. But a single case cannot decide the value of a sweeping generalization.

Even if this question can be answered affirmatively, there is a fundamental problem in reformulating the specifications of a rich feature formalism into a weaker DATR: distinctions must be lost. And if these distinctions are warranted, then the weaker formalism will be inadequate. The problems of relative expressive capacity may have been minor in the early days of feature formalism work (when the formalisms—e.g., PATR-II—were not much more expressive than DATR), but there have been steady amplifications on the feature-based side since then.

Before concluding this section we should note the work of Kilbury et al. 1991, who show how to construct an interface from DATR to feature formalisms (using the example of PATR-II). This work may be seen as foundering on exactly the factors noted above, conflicting modeling conventions and relative expressive capacity. The proposal is quite general: they advocate constructing an interface by using DATR as a metalanguage in which one can describe and constrain (the syntax of) PATR-II equations. This is unproblematic because expressions in the PATR-II language are just trees, and it does not suffer fatally from the incompatible semantics of the formalisms because it simply ignores the semantics of the feature specifications.

In this way they finesse both the question of modeling conventions and and expressive capacity. This may be the best one can do, but it is unsatisfying for two reasons: first, it ignores underlying semantics at the cost of having to write specifications about PATR-II syntax, including bracketing, use of colon, etc., an untidy level of indirection. But one is forced to something like this when attempting to cast the specifications of a richer language in the forms of a weaker. Second, in using DATR as a metalanguage, massive redundancy is introduced into lexical specifications, which now specify not only form values for the attribute '< person >', but attributes in which the value PATR-II 'PERSON' figure (as DATR ' 'PERSON' '). Formally, these have nothing to do with one another. The name of the DATR attribute and (part of) the value of a completely distinct attribute look similar to human eyes, but are formally unrelated. And this, too, is forced given the different semantics of the expressions in the two formalisms.

5 Conclusions

The purpose of the extended comparison to DATR was, of course, to suggest that, rather than take DATR as basic and attempt to do the work of feature specifications within it, we take the opposite tack, beginning with feature structures and attempt to borrow the most valuable insights of DATR into them—the use of defaults and the treatment of the paradigm as an abstract specification (lexeme), each of whose further complete specifications is a form within the paradigm. Section 2

demonstrates how these DATR virtues are readily obtainable within approaches which combine default inheritance with feature-based specification.

Of course, the opportunities within feature-based theories are much richer, and here we remind only of the opportunities they provide for specifying derivational relations using type requirements and for encoding the fairly intricate dependencies required in the case of the passive.

Acknowledgements

This short report discusses the lexicon work in the DFKI DISCO Project and was presented to the ASL Lexicon Workshop as a suggestion for lexical design. The DISCO work is led by Hans-Ulrich Krieger, who has been active in all phases of its conception, design and development. Kader Diagne, Stephan Diehl, Ralf Flassig, John Nerbonne, Klaus Netter, Hannes Pirker, Ulrich Schaefer, Florian Theissing und Christoff Weyers have contributed variously to theoretical foundations, linguistic content, and implementation (including that of the lexicon itself, the TDL development environment, and interfaces to morphological and syntactic processing). Rolf Backofen, Stephan Busemann, Harald Trost and Hans Uszkoreit have provided useful suggestions and criticisms about many aspects of the work. The DISCO Projekt is supported by a research grant, ITW 9002 0, to project manager Prof. Dr. Hans Uszkoreit, from the Bundesministeriums für Forschung und Technologie.

References

Bach, E. 1988. Categorial Grammars as Theories of Language. In *Categorial Grammars and Natural Language Structures*, ed. R. T. Oehrle, E. Bach, and D. Wheeler, 17–34. Dordrecht and Boston: Reidel.

Backofen, R., L. Euler, and G. Görz. 1990. Towards the Integration of Functions, Relations and Types in an AI Programming Language. In *Proceedings of GWAI-90.* Berlin. Springer.

Bouma, G. 1990. Defaults in Unification Grammar. In *Proceedings of the 28th Annual Meeting of the ACL*, 165–172. Association for Computational Linguistics.

Carpenter, B. to appear, 1992. *The Logic of Typed Feature Structures.* Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press.

Cook, W., W. Hill, and P. Canning. 1990. Inheritance is not Subtyping. In *ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL).*

Copestake, A. 1992. Defaults in the LRL. In *Default Inheritance within Unification-Based Approaches to the Lexicon*, ed. T. Briscoe, A. Copestake, and V. de Paiva. Cambridge: Cambridge University Press.

Diehl, S. 1992. Typed Feature Structures: DISCO-TDL—A Type Defintion Language. Technical Report in preparation, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken.

Dörre, J., and A. Eisele. 1989. Determining Consistency of Feature Terms with Distributed Disjunctions. In *Proceedings of GWAI-89 (15th German Workshop on AI)*, ed. D. Metzing, 270–279. Berlin. Springer-Verlag.

Evans, R., and G. Gazdar. 1989. Inference in DATR. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics*, 66–71.

Evans, R., and G. Gazdar. 1990. The DATR Papers. Technical Report SCRP 139, School of Cognitive and Computing Sciences, University of Sussex.

Evans, R., G. Gazdar, and L. Moser. 1992. Prioritised Multiple Inheritance in DATR. In *Default Inheritance within Unification-Based Approaches to the Lexicon*, ed. T. Briscoe, A. Copestake, and V. de Paiva. Cambridge: Cambridge University Press.

Flickinger, D. 1987. *Lexical Rules in the Hierarchical Lexicon.* PhD thesis, Stanford University.

Flickinger, D., and J. Nerbonne. 1991 or 1992. Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns. *Computational Linguistics* 18 or 19.

Flickinger, D., C. Pollard, and T. Wasow. 1985. Structure-Sharing in Lexical Representation. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics.*

Gazdar, G. 1987. Linguistic Applications of Default Inheritance Mechanisms. In *Linguistic Theory and Computer Applications*, ed. P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, and P.Bennett. London: Academic Press.

Gazdar, G. 1990. An Introduction to DATR. In *The DATR Papers*, ed. R. Evans and G. Gazdar, 1–14. No. SCRP 139Cognitive Science Research Reports. University of Sussex: School of Cognitive and Computing Sciences.

Gibbon, D. 1991. ILEX: A Linguistic Approach to Computational Lexica. Technical Report 5, Universität Bielefeld, Bielefeld.

Gibbon, D., and F. Ahoua. 1991. DDATR: un logiciel de traitement d'heritage par defaut pour la modélisation lexicale. Technical report, Institute de Linguistique Appliquée, Université Nationale de Côte d'Ivoire, March.

Kathol, A. 1991a. Agreement in HPSG Revisited. unpublished paper.

Kathol, A. 1991b. Verbal and Adjectival Passives in German. In *MIT Working Papers in Linguistics*, Vol. 14. MIT.

Kilbury, J., P. Naerger, and I. Renz. 1991. DATR as a Lexical Component for PATR. In *Proceedings of the 6th Annual Meeting of the European Chapter of the Association for Computational Linguistics.*

Krieger, H.-U. to appear, 1992. Lexicon-Driven Derivation in HPSG. In *HPSG in German*, ed. J. Nerbonne, K. Netter, and C. Pollard, xxx–yyy. Stanford: CSLI.

Krieger, H.-U., and J. Nerbonne. 1992. Feature-Based Inheritance Networks for Computational Lexicons. In *Default Inheritance within Unification-Based Approaches to the Lexicon*, ed. T. Briscoe, A. Copestake, and V. de Paiva. Cambridge: Cambridge University Press. also DFKI Research Report RR-91-31.

Moens, M., J. Calder, E. Klein, M. Reape, and H. Zeevat. 1989. Expressing Generalizations in Unification-Based Grammar Formalisms. In *Proceedings of the 4th Meeting of the European Chapter of the Association for Computational Linguistics*, 174–181.

Nerbonne, J., K. Netter, and C. Pollard (ed.). in preparation, 1992. *HPSG in German.* Lecture Note Series. Stanford: CSLI.

Pollard, C., and I. Sag. 1987. *An Information-Based Theory of Syntax and Semantics, Vol.I.* Stanford: CSLI.

Pollard, C., and I. Sag. 1992. An Information-Based Theory of Syntax and Semantics, Vol.II. unpublished manuscript in preparation.

Reinhard, S. 1989. Prosodic Inheritance and Morphological Generalizations. In *European ACL*, xxx–yyy.

Russell, G., A. Ballim, J. Carroll, and S. Warwick-Armstrong. 1992. A Practical Approach to Multiple Default Inheritance for Unification-Based Lexicons. In *Default Inheritance within Unification-Based Approaches to the Lexicon*, ed. T. Briscoe, A. Copestake, and V. de Paiva. Cambridge: Cambridge University Press.

Sag, I., and C. Pollard. 1987. Head-Driven Phrase Structure: An Informal Synopsis. Technical Report CSLI-87-89, Center for the Study of Language and Information, Stanford University.

Schäfer, U., and H.-U. Krieger. 1992. TDL *extra-light User's Guide: Franz Allegro Common LISP Version.* DISCO.

Shieber, S. 1986. *An Introduction to Unification-Based Approaches to Grammar.* Stanford University: Center for the Study of Language and Information.

Smolka, G. 1988. A Feature Logic with Subsorts. Technical Report 33, WT LILOG–IBM Germany.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
D-6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all publisched papers so far can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-90-15
Harald Trost: The Application of Two-level Morphology to Non-concatenative German Morphology
13 pages

RR-90-16
Franz Baader, Werner Nutt: Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification
25 pages

RR-90-17
Stephan Busemann:
Generalisierte Phasenstrukturgrammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung
114 Seiten

RR-91-01
Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, Gert Smolka: On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations
20 pages

RR-91-02
Francesco Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, Werner Nutt: The Complexity of Existential Quantification in Concept Languages
22 pages

RR-91-03
B.Hollunder, Franz Baader: Qualifying Number Restrictions in Concept Languages
34 pages

RR-91-04
Harald Trost: X2MORF: A Morphological Component Based on Augmented Two-Level Morphology
19 pages

RR-91-05
Wolfgang Wahlster, Elisabeth André, Winfried Graf, Thomas Rist: Designing Illustrated Texts: How Language Production is Influenced by Graphics Generation.
17 pages

RR-91-06
Elisabeth André, Thomas Rist: Synthesizing Illustrated Documents: A Plan-Based Approach
11 pages

RR-91-07
Günter Neumann, Wolfgang Finkler: A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures
13 pages

RR-91-08
Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist: WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation
23 pages

RR-91-09
Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds
31 pages

RR-91-10
Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages
31 pages

RR-91-11
Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches
37 pages

RR-91-12
J.Mark Gawron, John Nerbonne, Stanley Peters:
The Absorption Principle and E-Type Anaphora
33 pages

RR-91-13
Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming
17 pages

RR-91-14
Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I
27 pages

RR-91-15
Bernhard Nebel, Gert Smolka:
Attributive Description Formalisms ... and the Rest of the World
20 pages

RR-91-16
Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations
18 pages

RR-91-17
Andreas Dengel, Nelson M. Mattos:
The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18
John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein:
A Diagnostic Tool for German Syntax
20 pages

RR-91-19
Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20
Christoph Klauck, Ansgar Bernardi, Ralf Legleitner
FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21
Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22
Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23
Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24
Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25
Karin Harbusch, Wolfgang Finkler, Anne Schauder:
Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26
M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger:
Integrated Plan Generation and Recognition
- A Logic-Based Approach -
17 pages

RR-91-27
A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer:
ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28
Rolf Backofen, Harald Trost, Hans Uszkoreit:
Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-29
Hans Uszkoreit: Strategies for Adding Control Information to Declarative Grammars
17 pages

RR-91-30
Dan Flickinger, John Nerbonne:
Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns
39 pages

RR-91-31
H.-U. Krieger, J. Nerbonne:
Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32
Rolf Backofen, Lutz Euler, Günther Görz:
Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

RR-91-33
Franz Baader, Klaus Schulz:
Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures
33 pages

RR-91-34
Bernhard Nebel, Christer Bäckström:
On the Computational Complexity of Temporal Projection and some related Problems
35 pages

RR-91-35
Winfried Graf, Wolfgang Maaß: Constraint-basierte Verarbeitung graphischen Wissens
14 Seiten

RR-92-03
Harold Boley:
Extended Logic-plus-Functional Programming
28 pages

RR-92-04
John Nerbonne: Feature-Based Lexicons: An Example and a Comparison to DATR
15 pages

RR-92-05
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Michael Schulte, Rainer Stark:
Feature based Integration of CAD and CAPP
19 pages

DFKI Technical Memos

TM-91-01
Jana Köhler: Approaches to the Reuse of Plan Schemata in Planning Formalisms
52 pages

TM-91-02
Knut Hinkelmann: Bidirectional Reasoning of Horn Clause Programs: Transformation and Compilation
20 pages

TM-91-03
Otto Kühn, Marc Linster, Gabriele Schmidt:
Clamping, COKAM, KADS, and OMOS: The Construction and Operationalization of a KADS Conceptual Model
20 pages

TM-91-04
Harold Boley (Ed.):
A sampler of Relational/Functional Definitions
12 pages

TM-91-05
Jay C. Weber, Andreas Dengel , Rainer Bleisinger:
Theoretical Consideration of Goal Recognition Aspects for Understanding Information in Business Letters
10 pages

TM-91-06
Johannes Stein: Aspects of Cooperating Agents
22 pages

TM-91-08
Munindar P. Singh: Social and Psychological Commitments in Multiagent Systems
11 pages

TM-91-09
Munindar P. Singh: On the Semantics of Protocols Among Distributed Intelligent Agents
18 pages

TM-91-10
Béla Buschauer, Peter Poller, Anne Schauder, Karin Harbusch: Tree Adjoining Grammars mit Unifikation
149 pages

TM-91-11
Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann:
Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel:
ODA-based modeling for document analysis
14 pages

TM-91-15
Stefan Bussmann: Prototypical Concept Formation An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang:
Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

DFKI Documents

D-91-01
Werner Stein , Michael Sintek: Relfun/X - An Experimental Prolog Implementation of Relfun
48 pages

D-91-02
Jörg P. Müller: Design and Implementation of a Finite Domain Constraint Logic Programming System based on PROLOG with Coroutining
127 pages

D-91-03
Harold Boley, Klaus Elsbernd, Hans-Günther Hein, Thomas Krause: RFM Manual: Compiling RELFUN into the Relational/Functional Machine
43 pages

D-91-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1990
93 Seiten

D-91-06
Gerd Kamp: Entwurf, vergleichende Beschreibung und Integration eines Arbeitsplanerstellungssystems für Drehteile
130 Seiten

D-91-07
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
TEC-REP: Repräsentation von Geometrie- und Technologieinformationen
70 Seiten

D-91-08
Thomas Krause: Globale Datenflußanalyse und horizontale Compilation der relational-funktionalen Sprache RELFUN
137 Seiten

D-91-09
David Powers, Lary Reeker (Eds.):
Proceedings MLNLO'91 - Machine Learning of Natural Language and Ontology
211 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-10
Donald R. Steiner, Jürgen Müller (Eds.):
MAAMAW'91: Pre-Proceedings of the 3rd European Workshop on „Modeling Autonomous Agents and Multi-Agent Worlds"
246 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-11
*Thilo C. Horstmann:*Distributed Truth Maintenance
61 pages

D-91-12
Bernd Bachmann:
$H^{iera}C_{on}$ - a Knowledge Representation System with Typed Hierarchies and Constraints
75 pages

D-91-13
International Workshop on Terminological Logics
Organizers: Bernhard Nebel, Christof Peltason, Kai von Luck
131 pages

D-91-14
Erich Achilles, Bernhard Hollunder, Armin Laux, Jörg-Peter Mohren: KRIS: Knowledge Representation and Inference System
- Benutzerhandbuch -
28 Seiten

D-91-15
Harold Boley, Philipp Hanschke, Martin Harm, Knut Hinkelmann, Thomas Labisch, Manfred Meyer, Jörg Müller, Thomas Oltzen, Michael Sintek, Werner Stein, Frank Steinle:
µCAD2NC: A Declarative Lathe-Worplanning Model Transforming CAD-like Geometries into Abstract NC Programs
100 pages

D-91-16
Jörg Thoben, Franz Schmalhofer, Thomas Reinartz:
Wiederholungs-, Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile
134 Seiten

D-91-17
Andreas Becker:
Analyse der Planungsverfahren der KI im Hinblick auf ihre Eignung für die Abeitsplanung
86 Seiten

D-91-18
Thomas Reinartz: Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe
107 Seiten

D-91-19
Peter Wazinski: Objektlokalisation in graphischen Darstellungen
110 Seiten

Feature-Based Lexicons: An Example and a Comparison to DATR

John Nerbonne

RR-92-04
Research Report