Lexical Rules in Constraint-based Grammars

Ted Briscoe*
University of Cambridge

Ann Copestake†
CSLI, Stanford University

Lexical rules have been used to cover a very diverse range of phenomena in constraint-based grammars. Examination of the full range of rules proposed shows that Carpenter's (1991) postulated upper bound on the length of list-valued attributes such as subcat in the lexicon cannot be maintained, leading to unrestricted generative capacity in constraint-based formalisms utilizing HPSG-style lexical rules. We argue that it is preferable to subdivide such rules into a class of semi-productive lexically-governed genuinely lexical rules, and a class of fully productive unary syntactic rules.

We develop a restricted approach to lexical rules in a typed default feature structure (TDFS) framework (Lascarides et al., 1996; Lascarides and Copestake, 1999) which has enough expressivity to state, for example, rules of verb diathesis alternation, but which does not allow arbitrary manipulation of list-valued features. An interpretation of such lexical rules within a probabilistic version of a TDFS-based linguistic (lexical and grammatical) theory allows us to capture the semi-productive nature of genuinely lexical rules, steering an intermediate course between fully generative or purely abbreviatory rules.

We illustrate the utility of this approach with a treatment of dative constructions within a linguistic framework which borrows insights from the constraint-based theories: HPSG, UCG (Zeevat et al., 1987) and Construction Grammar (Goldberg, 1995). We end by outlining how our approach to lexical rules allows for a treatment of passive and recursive affixation, which are generally assumed to require unrestricted list manipulation operations.

Introduction

In lexicalist approaches to grammar, lexical rules are a crucial component of the overall theory and more and more generalizations have come to be stated within them. For example, in the development of HPSG (Pollard & Sag, 1987; 1994) from GPSG (Gazdar *et al.*, 1985) several syntactic metarules concerning the location of empty categories in unbounded dependency constructions are restated as lexical rules. One example is the Subject Extraction rule which licenses subject extractions from sentential complements. Inflectional morphological rules, such as Plural Formation, and rules of verb alternation, such as Passive, are also stated as lexical rules by Pollard & Sag. Recently, Bouma & van Noord (1994) have proposed a lexical rule of Adjunct Introduction which can recursively add adverbial categories to the *subcat* list of a verbal category. There are three main problems with the treatment of lexical, or what might be better termed unary, rules as a homogeneous class.

* Computer Laboratory, University of Cambridge, Pembroke Street, Cambridge CB2 3QG, UK. E-mail: `ejb@cl.cam.ac.uk`
† Center for the Study of Language and Information, Stanford University, Ventura Hall, Stanford, CA 94305, USA. E-mail: `aac@csli.stanford.edu`

Firstly, Carpenter (1991) demonstrates that if lexical rules are able to perform arbitrary manipulations (deletion, addition and permutation) of potentially unbounded lists, any recursively enumerable language can be generated, even if the non-derived lexicon and grammar only generate context-free languages. However, once we are committed to treating rules such as Passive and Adjunct Introduction in a homogeneous way, restrictions which prevent lexical rules from increasing generative capacity, such as constraining the use of category variables, bounding the length of the *subcat* list, or limiting recursive application, cannot be imposed. Subdividing lexical and unary syntactic rules allows such restrictions to be naturally maintained in the lexicon.

Secondly, within theories like HPSG, which utilize constraint logics over (typed) feature structures ((T)FSs) as the lexical and grammatical representation language, the formal status of lexical rules is unclear. They do not have a straightforward interpretation as logical constraints and are normally treated as metalevel conditional generalizations concerning the set of admissible lexical entries. This leads to immediate disadvantages, such as non-declarativity, potential non-termination in application, and the need for the grammar writer to specify what stays unchanged as well as what changes in a derived entry. Our approach to lexical rules improves this situation by formalizing them in terms of default unification utilizing existing operations in the typed default feature structure (TDFS) representation language (Lascarides *et al.*, 1996; Lascarides and Copestake, 1999).

Thirdly, most lexical, though not unary, rules are semi-productive. In HPSG, lexical rules are interpreted as generative rules encoding putatively exceptionless conditional generalizations concerning the existence of derived lexical entries. Exceptions and irregularities must be marked explicitly in lexical entries. Whilst this approach is defensible for rules of inflectional morphology, generalizations about unbounded dependency constructions and adjunct introduction, it is less plausible when we come to consider morphological rules of derivation or most verb diathesis alternations. These latter are semi-productive; that is, subject to blocking (preemption by synonymy or by lexical form), to arbitrary lexical gaps, and to varying degrees of conventionalization (see Jackendoff, 1997a for a recent discussion). It is technically possible to reinterpret (some) lexical rules as (lexical) redundancy rules using constraint-based techniques (e.g. Sanfilippo, 1993). However, this approach reduces lexical rules to a purely abbreviatory device even more thoroughly than Jackendoff's (1975) original proposal. We will argue in §6 that the pure redundancy rule interpretation is not optimal, even for semi-productive lexical rules, as such rules are utilized in the production and interpretation of nonce usages and neologisms.

Throughout the paper, we assume a linguistic framework based on typed feature structures (TFSs) like HPSG, but allowing for defeasible specification in typed default feature structures (TDFSs) (e.g. Lascarides *et al.*, 1996). We distinguish between lexical rules and other unary rules and present an account of lexical rules compatible with this framework and potentially other constraint-based (T)FS frameworks. We argue that this account satisfactorily addresses the issues of generative power, formal interpretation and semi-productivity. In §1 we discuss the use and interpretation of HPSG-style lexical rules in more detail. In §2 we present the TDFS framework for lexical and grammatical representation, which extends monotonic theories of lexical organization with default inheritance and defeasible specification. In §3, we formalize lexical rules in terms of default unification and demonstrate that this leads to a more restricted operation fully defined in terms of the non-monotonic conditional logic of TDFSs. In §4 we show that, despite this restricted capacity, a linguistically insightful account of English dative constructions can be formulated which treats them as (mostly) derived by lexical rule. This draws on insights from Dowty's (1989) theory of proto-roles, Sanfilippo's (1990, 1992, 1993) treatment of verb alternations in UCG (Zeevat *et al.*, 1987) and Goldberg's (1995)

$$\left[\begin{array}{l} \textbf{base} \\ \text{PHON} : \boxed{1} \\ \text{3RDSNG} : \boxed{2} \\ \text{SYN} : \left[\begin{array}{l} \text{LOC} : \left[\begin{array}{l} \text{SUBCAT} : \boxed{3} \end{array}\right] \end{array}\right] \\ \text{SEM} : \left[\begin{array}{l} \text{CONT} : \boxed{4} \end{array}\right] \end{array}\right] \mapsto \left[\begin{array}{l} \textbf{3rdsng} \\ \text{PHON} : \mathit{f3rdsng}(\boxed{1}, \boxed{2}) \\ \text{SYN} : \left[\begin{array}{l} \text{LOC} : \left[\begin{array}{l} \text{SUBCAT} : \boxed{3} \end{array}\right] \end{array}\right] \\ \text{SEM} : \left[\begin{array}{l} \text{CONT} : \boxed{4} \end{array}\right] \end{array}\right]$$

Figure 1
HPSG Third Singular Verb Formation Lexical Rule

$$\left[\begin{array}{l} \textbf{base} \\ \text{PHON} : \boxed{1} \\ \text{3RDSNG} : \boxed{2} \end{array}\right] \mapsto \left[\begin{array}{l} \textbf{3rdsng} \\ \text{PHON} : \mathit{f3rdsng}(\boxed{1}, \boxed{2}) \end{array}\right]$$

Figure 2
Abbreviated form of lexical rule

analysis in Construction Grammar. In §5 we argue that the dative alternation is a semi-productive process in English not fully reducible to an abbreviatory convention, and provide an account of lexical rule application within a probabilistic version of the TDFS framework which captures the variable acceptability of different verbs in the dative construction. Thus, we modify and extend the analysis of dative in Lascarides *et al.* (1996) by providing a more adequate and restrictive formalization of this type of semi-productive lexical rule. In §6 we consider the extent to which our approach to lexical rules will generalize to other types of putatively lexical processes.

1 Lexical Rules in HPSG

Lexical rules in HPSG are interpreted as conditional relationships between lexical entries represented as constraints on TFSs (e.g. Pollard & Sag, 1994:395fn1). Since lexical entries themselves are FS descriptions, not FSs *per se*, this interpretation makes lexical rules metalevel statements in an informally-defined language. For example, the lexical rule for Third Singular Verb Formation (Pollard & Sag, 1987:210-213) is given in Figure 1. In this notation, coindexation must be interpreted as a copying operator and not as a specification of reentrancy within a single TFS description, and the description language must be extended with a conditional implication operator relating FSs. A linguistic infelicity of this notation is that it requires explicit specification of what is unchanged under lexical rule application as well as of the changes to the derived entry. The informal interpretation of such a rule is that for lexical entries which unify with the antecedent description, a derived entry can be created by copying elements as specified into the consequent description. However, in the HPSG literature, these explicit copying statements are often left unstated, with the notation being interpreted to mean that features which are omitted have their values copied over, giving the notation shown in Figure 2.

A number of modifications of this original proposal within the HPSG framework have been proposed. Copestake & Briscoe (1992) and Copestake (1992) represent lexical rules as TFSs containing 1 and 0 attributes representing input and output descriptions of the lexical rule, respectively. This enables lexical rules to be encoded in a type hierarchy and for relationships between rules to be expressed via type inheritance. The interpretation of lexical rules is analogous to that of grammar rules (e.g. Shieber, 1992), and such rules can be thought of as equivalent to unary grammar rules. Calcagno (1995) develops an algorithm for improving the notation for lexical rules by eliminating the need

$$\begin{bmatrix} \textbf{3rdsng-lr} \\ \text{IN} : \begin{bmatrix}\textbf{base}\end{bmatrix} \\ \text{OUT} : \begin{bmatrix} \textbf{3rdsng} \\ \text{PHON} : \mathit{f3rdsng}(\boxed{1}, \boxed{2}) \\ \text{3RDSNG} : \boxed{2} \end{bmatrix} \end{bmatrix}$$

Figure 3
Reformulated Third Singular Verb Formation Lexical Rule

to specify what is copied from input to output. Meurers (1995) also develops a similar algorithm but, although he augments the description language to allow lexical rules to be written in this abbreviated notation, he interprets the result as a relational constraint on lexical entries.[1] Thus, in Meurers' notation, the rule in Figure 1 would be written without coindexation as a TFS using IN/OUT attributes as in Figure 3, and appropriate coindexation would be algorithmically generated from both the rule specification and the type system to recover an expanded rule effectively equivalent to that in Figure 1 (though the entire values of SYN and SEM would be carried over). In Meurers' formulation the coindexation generated is interpreted as genuine reentrancy since the lexicon is defined utilizing 'junk slot' encoding (Ait-Kaci, 1984) of the set of possible words, as shown below:

$$\text{word} \rightarrow (L_1 \vee \ \ldots \ \vee L_n \vee \boxed{1} \begin{bmatrix} \text{STORE} : < \begin{bmatrix} \textbf{lex-rule} \\ \text{OUT} : \boxed{1} \end{bmatrix} > \end{bmatrix}$$

This constraint states that a word must be subsumed by one of the basic descriptions or by a description derived via one or more lexical rule applications (i.e. any description tagged $\boxed{1}$). This latter step integrates the interpretation of lexical rules into the underlying constraint logic of TFS descriptions by structure sharing information between descriptions of basic and derived lexical entries.

In view of the unrestricted generative power of conventional HPSG-style lexical rules (Carpenter, 1991), naive generative application of recursive or cyclic rules can lead to non-termination during parsing. Bouma & van Noord (1994) and Johnson & Dorre (1995) propose techniques for delayed evaluation of lexical rules so that they apply 'on demand' at parse time. Meurers & Minnen (1997) present a computational framework for efficient application of Meurers' (1995) formalization of lexical rules. In their covariation approach, a finite-state machine for the application of lexical rules is derived by computing possible 'follow' relations between the set of rules. Next pruned finite-state machines are associated with classes of actual lexical entries representing the restricted set of rules which can apply to those entries. Finally, entries themselves are extended with information common to all their derived variants. These techniques achieve most of the advantages of lexicon expansion in the face of recursive rules and cyclic rule interactions which preclude a full off-line expansion.

Since their inception, the productivity of some types of lexical rule has been an issue (e.g. Jackendoff, 1975). Given the interpretations discussed above, lexical rules predict that derived entries will exist without exception for any basic entry subsumed by the a lexical rule input description. Pollard & Sag (1987:210f) treat the blocking of regular rules of morphological affixation by making the application of affixation functions sensitive

1 Riehemann (1993) was the first to propose this interpretation of lexical rules, though she chose to represent them directly in the type inheritance hierarchy.

to specification of irregular affixes in base entries. Thus the rule of Third Singular Verb Formation in Figure 1 contains a function *f3rdsng* which combines the value of the attribute 3RDSNG with that of PHON to create the PHON of the output sign. If the base sign specifies an irregular form for third singular the function will simply return this form as opposed to the form produced by regular suffixation with *+s*. However, preemption by synonymy with an irregularly derived lexical entry is only one form of semi-productivity exemplified by lexical rules. Preemption by synonymy can also apply in cases where the blocking form is basic and does not involve affixation, such as *glory* blocking **gloriosity* (see Aronoff, 1976). Identity of lexical form between a derived and underived entry or a more irregular derived entry can serve to block the more regular derived entry, as with *sticker* which has a highly preferred meaning derived via object, as opposed to subject, *er*-nominalization (compare *bumper sticker* to *?a sticker of offensive posters to billboards*). There are many other types of semi-productivity, such as that created by 'accidental' gaps or idiosyncratic exceptions, for example, ?*thinkable* (compare *unthinkable*), as well as cases whose status as 'systematic' exceptions or gaps is unclear. For example, verbs such as *cost*, *weigh* or *last* which take measure phrase NP objects do not undergo Passive. These should probably be blocked from undergoing the lexical rule by enriching its structural input with restrictions on the semantic nature of the object NP. However, symmetric predicates, such as *resemble* and *equal*, which also appear not to passivize easily, are a more complex case (e.g. Wasow, 1980). Firstly, there are symmetric predicates that passivize relatively easily, such as *meet* – *Kim was usually met by someone from the Russian Embassy at the safe house*. Secondly, it is possible to create acceptable examples for the verbs standardly cited in the literature – *The difficulty of the solution was only equalled by the obduracy of the research team*. And thirdly, there is a general pragmatic reason to believe passive might be (largely) redundant with symmetric predicates since in most uses the subject and object can be reversed without changing meaning. Nevertheless, the formulation of passive given in Pollard & Sag (1987:215) does not address either type of exception (under the assumption that these verbs are all of type transitive).

Jackendoff (1975) and others have proposed that lexical rules be interpreted as redundancy statements which abbreviate the statement of the lexicon but which are not applied generatively. This conception of lexical rules has been utilized in a constraint-based framework by Sanfilippo (1993) who adds a list-valued attribute to the input description of a lexical rule which encodes the name of the lexical rule, so that the rule input specification will only unify with lexical entries which also specify the rule name as a member of this list-valued attribute. We illustrate this approach in slightly modified form with respect to the causative-inchoative verb alternation in Figure 4.[2] The effect is that a verb will explicitly specify the alternations that apply to it, as values for the feature ALTS, out of a set of alternations which are possible for its type, and that lexical rules may only be applied when the lexical sign has the appropriate value for that alternation instantiated.

In Sanfilippo's approach, a constraint-based encoding of categorial grammar (e.g. Zeevat *et al.*, 1987) is combined with Dowty's proto thematic role theory and proto-roles are interpreted as predicates holding between event and participant variables in a neo-Davidsonian semantic representation. The version of the rule given assumes proto-roles appropriate for movement verbs and could be used to relate lexical entries for *gallop* as in *John galloped the horse / The horse galloped*. For convenience here we are using an abbreviated version of the 'minimally-recursive' style of encoding for the semantics

2 Subscripted coindexing is used to abbreviate the path to the semantic index of the verbal argument category.

$$
\begin{bmatrix}
\textbf{intrans-verb} \\
\text{PHON} : \boxed{1} \\
\text{SYN} : \begin{bmatrix} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign}_{\boxed{i}} \end{bmatrix} \\
\text{SEM} : < \begin{bmatrix} \textbf{verb-rel} \\ \text{EVENT} : \textbf{e} \end{bmatrix}, \begin{bmatrix} \textbf{p-agt-cause-move} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \boxed{i} \end{bmatrix} > \\
\text{ALTS} : \boxed{4} \begin{bmatrix} \text{TRANS-ALT} : \textbf{caus-inchoat} \end{bmatrix}
\end{bmatrix} \mapsto
$$

$$
\begin{bmatrix}
\textbf{trans-caus-verb} \\
\text{PHON} : \boxed{1} \\
\text{SYN} : \begin{bmatrix} \text{RESULT} : \begin{bmatrix} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign}_{\boxed{i}} \end{bmatrix} \\ \text{ACTIVE} : \textbf{npsign}_{\boxed{j}} \end{bmatrix} \\
\text{SEM} : < \begin{bmatrix} \textbf{verb-rel} \\ \text{EVENT} : \textbf{e} \end{bmatrix}, \begin{bmatrix} \textbf{p-agt-cause} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \boxed{j} \end{bmatrix}, \begin{bmatrix} \textbf{p-pat-move} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \boxed{i} \end{bmatrix} > \\
\text{ALTS} : \boxed{4}
\end{bmatrix}
$$

Figure 4
The Causative-Inchoative Lexical Rule: Sanfilippo's approach

(MRS) described by Copestake *et al.* (1995). The semantics for the causative form of *gallop* described is equivalent in linearized notation to:

$$[\text{gallop}(e) \wedge \text{p-agt-cause}(e, x) \wedge \text{p-pat-move}(e, y)]$$

However, the rule can only apply if the transitive entry for *gallop* specifies **caus-inchoat** as the value of ALTS TRANS-ALT. An immediate problem arises, because as Pinker (1989) and others have argued, the rule is semi-productive rather than purely abbreviatory, in the sense that nonce usages are clearly interpreted conventionally as being novel (mis)applications of such rules. As in for example, *Kim subscribed his friend to* Byte *for a year* or *Don't fall my dolly over!*.

Although the proposals reviewed considerably clarify and enrich the original conception of lexical rules in HPSG, problems remain. The generative power of lexical rules places no limit on possible variations in derived words. A more restricted lexical rule formalism would encourage finer-grained distinctions between rule classes and development of substantive criteria for characterizing a rule as lexical (as opposed to unary syntactic). It would also allow an interpretation of lexical rules more consonant with constraint-based frameworks. Neither the interpretation of lexical rules as fully generative nor as purely abbreviatory is adequate. Although many lexical rules are subject to exceptions, gaps and variable degrees of conventionalization, most are semi-productive because they play a role in the production and interpretation of nonce usage, errors, and neologisms.

2 The Typed Default Feature Structure Formalism

Our alternative approach to lexical rules assumes the use of typed default feature structures (TDFSs) as the basic data structure. The TDFS formalism extends typed feature structure (TFS) formalisms, such as those described in Carpenter (1992) or King (1994), by allowing for structures which include default information. These are combined by typed (persistent) default unification. In most previous accounts (see e.g. Bouma, 1990; Carpenter, 1993; Copestake, 1993; Russell *et al.*, 1993), default unification is an asymmetric operation which combines two ordinary (T)FSs, one of which is treated as default and one non-default, to produce a normal TFS. In contrast, TDFSs mark default information explicitly within each structure, and are combined with a symmetric operation, which retains the information about the defeasibility of particular pieces of information. There are two advantages of the TDFS framework over previous approaches which are important for our treatment of lexical rules:

1. Unlike asymmetric default unification, the unification operation on TDFSs is order-independent. Thus inheritance hierarchies may be defined in which some inherited information may be overridden by more specific information, without the results being dependent on evaluation order.
2. Structures can be represented as having *persistent* defaults: in particular, parts of the semantics of lexical signs can be marked as defeasible, in a way that allows the information to be overridden either by sentential or discourse context.

Lascarides *et al.* (1996) provide an initial formalization of the TDFS framework, Lascarides & Copestake (1999: henceforth L&C) describe the version of default unification assumed in the informal outline of the TDFS formalism which follows.

$$\begin{bmatrix} \mathrm{F} : \mathbf{a} \\ \mathrm{G} : \top \end{bmatrix} \stackrel{<}{\sqcap}_c \begin{bmatrix} \mathrm{F} : \boxed{1}\ \mathbf{b} \\ \mathrm{G} : \boxed{1} \\ \mathrm{H} : \mathbf{c} \end{bmatrix} = \left\{ \begin{bmatrix} \mathrm{F} : \mathbf{a} \\ \mathrm{G} : \mathbf{b} \\ \mathrm{H} : \mathbf{c} \end{bmatrix}, \begin{bmatrix} \mathrm{F} : \boxed{1}\ \mathbf{a} \\ \mathrm{G} : \boxed{1} \\ \mathrm{H} : \mathbf{c} \end{bmatrix} \right\}$$

$$\begin{bmatrix} \mathrm{F} : \mathbf{a} \\ \mathrm{G} : \top \end{bmatrix} \stackrel{<}{\sqcap}_s \begin{bmatrix} \mathrm{F} : \boxed{1}\ \mathbf{b} \\ \mathrm{G} : \boxed{1} \\ \mathrm{H} : \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathrm{F} : \mathbf{a} \\ \mathrm{G} : \top \\ \mathrm{H} : \mathbf{c} \end{bmatrix}$$

Figure 5
Credulous and skeptical asymmetric default unification

2.1 Asymmetric default unification

In L&C, a symmetric unification operation on TDFSs is defined in terms of an asymmetric default unification operation, so we begin by describing the latter operation. We will also use a variant of this operation in our formalization of lexical rules in §3. Carpenter (1993) gives an elegant definition of asymmetric default unification of untyped feature structures in terms of maximal incorporation of information. Because of conflicts in defaults, this may give rise to multiple extensions. There are two alternative approaches to this: the operation can either return all extensions as a disjunction or return their generalization. This gives unification-based analogues of credulous and skeptical default logics. Here and below, we use $\sqcap$ to represent unification (i.e. conjunction of information), $\sqsupseteq$ for subsumption (i.e. is more general than), $\top$ is the most general type and $\perp$ indicates inconsistency.

Definition
Credulous Default Unification ($\stackrel{<}{\sqcap}_c$)
The result of credulously adding the default information in G to the strict information in F is given by:

$$F \stackrel{<}{\sqcap}_c G = \{F \sqcap G' : G' \sqsupseteq G \text{ is maximally specific such that } F \sqcap G' \text{ is defined } \}$$

where $\sqcap$ is ordinary unification

Definition
Skeptical Default Unification ($\stackrel{<}{\sqcap}_s$)

$$F \stackrel{<}{\sqcap}_s G = \sqcup\{F \sqcap G' : G' \sqsupseteq G \text{ is maximally specific such that } F \sqcap G' \text{ is defined } \}$$

The result is the **generalization** ($\sqcup$) of the result of the credulous default unification operation.

Figure 5 gives an example (here and in the examples below we assume that **a**, **b** and **c** are pairwise incompatible and pairwise generalize to $\top$). Carpenter makes use of a characterization of FSs as sets of *atomic feature structures*: that is, FSs which either consist of a single path with an associated atomic value (which would correspond to a type in a TFS framework), or of a pair of (possibly identical) paths which are shared. Any FS can be described in terms of the set of atomic FSs which are strictly more general than itself. Carpenter's definition of default unification can thus be understood in terms of incorporation of maximal subsets of the set of atomic FSs into which the default FS can be decomposed.

2.2 Well-formedness constraints

Because $\stackrel{<}{\sqcap}_s$ and $\stackrel{<}{\sqcap}_c$ are ultimately defined in terms of monotonic subsumption, we can create variant definitions corresponding to variant notions of subsumption. This is particularly relevant when we come to consider typed rather than untyped FSs. We can consider the subsumption ordering on well-formed typed feature structures, where well-formed TFSs are a subset of TFSs in general. There are a variety of proposals for well-formedness conditions for typed FSs: here we assume conditions on appropriate features based on Carpenter (1992). We will not give these in detail here, but briefly, we assume that every type, t, has a set of appropriate features associated with it, $AppFeat(t)$. For a TFS F to be well-formed, every node, q, in the structure must have an associated type, t and the features labelling the edges which come out of the node q directly must be equal to $AppFeat(t)$. For example, if the type **ne-list** has the appropriate features HD and TL, and has no subtypes, then the FS $\left[\text{SYN} : \textbf{intrans}\right]$ will not unify with a FS of type **ne-list**.

So, assume $\sqsupseteq_t$ indicates the partial order in the collection of well-formed TFSs, and $\sqcap_t$ indicates the corresponding greatest lower bound, and so on. Then we can define $\stackrel{<}{\sqcap}_t$, a straightforward variant on Carpenter's $\stackrel{<}{\sqcap}_s$, but defined on typed FSs.

Definition
Skeptical Typed Default Unification ($\stackrel{<}{\sqcap}_t$)
The result of skeptically adding the default information in the TFS G to the strict information in the TFS F is given by:

$$F \stackrel{<}{\sqcap}_t G = \sqcup_t \{F \sqcap_t G' : G' \sqsupseteq_t G \text{ is maximally specific such that } F \sqcap_t G' \text{ is defined }\}$$

In what follows, we will omit the subscript t and in general assume well-formedness of TFSs.

2.3 Symmetric default unification

As we mentioned above, we want to make a distinction between default and non-default information in a typed default FS (TDFS), and to allow the default information to 'persist' through a series of default unification operations. This is done by making a TDFS be a dual structure, of which the first component is a standard TFS indicating non-default information and the second is a structure known as a *tail*. A tail is a set of pairs, each consisting of an atomic FS and a type which indicates specificity. Atomic FSs associated with more specific types in the tail will be preferred over those associated with more general types in the case of conflict. As L&C discuss in detail, if we want to maintain associativity of default unification (in order to guarantee order-independence) we must maintain some of the unification 'history'. The tail does this and may thus include conflicting elements which were introduced by different TDFSs. However, the atomic FSs in the tail are required to be consistent with the indefeasible TFS. We use the notation I/T to indicate a TDFS with indefeasible component, I, and tail, T. An example of a TDFS is shown in (1).

(1) $$\begin{bmatrix} \textbf{t} \\ \text{F} : \begin{bmatrix} \textbf{v} \\ \text{G} : \top \end{bmatrix} \\ \text{H} : \textbf{a} \end{bmatrix} \Big/ \begin{array}{l} \{\langle \begin{bmatrix} \text{F} : \begin{bmatrix} \text{G} : \textbf{b} \end{bmatrix} \end{bmatrix}, \textbf{t} \rangle, \\ \langle \begin{bmatrix} \text{F} : \begin{bmatrix} \text{G} : \textbf{c} \end{bmatrix} \end{bmatrix}, \textbf{t}' \rangle, \\ \langle \begin{bmatrix} \text{F} : \begin{bmatrix} \text{G} : \boxed{1} \end{bmatrix} \\ \text{H} : \boxed{1} \end{bmatrix}, \textbf{t}'' \rangle\} \end{array}$$

The default unification operation, $\stackrel{<>}{\sqcap}$, combines two TDFSs. It simply involves taking the unification of their indefeasible components, and the union of their tails with all elements inconsistent with the indefeasible result removed. The following definition of the operation is slightly modified from L&C ($\wp_{fs}(T)$ is an operation which extracts the atomic FS components of tail elements).

Definition
Symmetric default unification $\overset{<>}{\sqcap}$
Let $F_1 =_{def} I_1/T^1$ and $F_2 =_{def} I_2/T^2$ be two TDFSs, and let $F_{12} =_{def} F_1 \overset{<>}{\sqcap} F_2$. Furthermore, assume $F_{12} =_{def} I_{12}/T^{12}$. Then I_{12} and T^{12} are calculated as follows:

1. The Indefeasible Part:
   $I_{12} = I_1 \sqcap I_2$
   That is, the indefeasible TFS is the unification of the indefeasible parts of the arguments.
2. The Tail T^{12}:
   $T^{12} =_{def} \text{Filter}(I_{12}, (T^1 \cup T^2))$
   where $\text{Filter}(I_{12}, T)$ includes only the elements of T where the atomic FS $\wp_{fs}(T)$ is compatible with I_{12}. That is:

$$\text{Filter}(F, T) = \{T' \in T : \wp_{fs}(T') \sqcap F \neq \bot\}$$

Of course, it will sometimes be necessary to know which default information 'wins'. That is, we require an operation, DefFS, which will give us a consistent default TFS, given the tail and the indefeasible structure. In the case of conflicts, the result of DefFS is dependent on the types which introduced the default information, since information associated with more specific types is preferred over that from more general types (the analogue of the 'penguin principle'). L&C therefore define the notion of a specificity partition of a tail, such that all the elements which are maximally specific (according to the partial order of their types in the type hierarchy) are in the first partition, all the next most specific elements are in the second partition and so on. That is, the partition of a tail T is $\langle \mu_1, \ldots, \mu_n \rangle$, where the types associated with the tail elements in μ_1 are strictly more specific than the types in μ_2, and so forth.

DefFS proceeds by using credulous asymmetric default unification to combine all the possible elements of the most specific partition with the indefeasible structure, then incorporating the next most specific partition into the result of this, and so on. Each step may result in multiple TFSs. Incorporation of information is defined by the credulous default unification operation, $\overset{<}{\sqcap}_{cs}$, which is a slight modification of $\overset{<}{\sqcap}_{c}$, as defined by Carpenter, because it has to allow for the non-default argument being a set of TFSs and the default argument being a set of atomic FSs which may be mutually incompatible.

Definition
Credulous Default Unification $\overset{<}{\sqcap}_{cs}$ on Sets
Let $\mathcal{F}_1$ be a *set* of TFSs $\{F_1, \ldots, F_n\}$, and $\mathcal{G}_2$ a set of atomic FSs. Then

$$\mathcal{F}_1 \overset{<}{\sqcap}_{cs} \mathcal{G}_2 = \{F_1 \overset{<}{\sqcap}_{ca} \mathcal{G}_2, \ldots, F_n \overset{<}{\sqcap}_{ca} \mathcal{G}_2\}$$

where

$$F_1 \overset{<}{\sqcap}_{ca} \{G_1, \ldots, G_n\} = \{F_1 \sqcap F_2 : \begin{array}{l} F_2 \text{ is the unification of a maximal subset of } \{G_1, \ldots, G_n\} \\ \text{such that } F_1 \sqcap F_2 \text{ is defined}\} \end{array}$$

Definition

The Operation DefFS

Let F be a TDFS I/T. Then

$$DefFS(F) = \sqcup((I \overset{<}{\sqcap}_{cs} \wp_{fs}(\mu_1)) \overset{<}{\sqcap}_{cs} \ldots \overset{<}{\sqcap}_{cs} \wp_{fs}(\mu_n))$$

So multiple TFSs may result from the credulous asymmetric default unification steps, but a final generalization operation takes only the information that they all have in common. Thus, in the absence of type precedence, conflicting defaults are treated skeptically overall.

In the example below, we illustrate DefFS on the TDFS shown in (1). On the assumption that the type **t** is strictly more specific than **t**′ which is strictly more specific than **t**″, each partition in the tail has a single element. In the example, the tail elements are shown in specificity order. The most specific will unify with the indefeasible structure, but the other elements conflict with the result (on the assumption that **a**, **b** and **c** are incompatible).

$$\text{DefFS}\left(\begin{bmatrix} \mathbf{t} \\ \text{F} : \begin{bmatrix} \mathbf{v} \\ \text{G} : \top \end{bmatrix} \\ \text{H} : \mathbf{a} \end{bmatrix} \Big/ \begin{array}{l} \{\langle \begin{bmatrix} \text{F} : \begin{bmatrix} \text{G} : \mathbf{b} \end{bmatrix} \end{bmatrix}, \mathbf{t} \rangle, \\ \langle \begin{bmatrix} \text{F} : \begin{bmatrix} \text{G} : \mathbf{c} \end{bmatrix} \end{bmatrix}, \mathbf{t}' \rangle, \\ \langle \begin{bmatrix} \text{F} : \begin{bmatrix} \text{G} : \boxed{1} \end{bmatrix} \\ \text{H} : \boxed{1} \end{bmatrix}, \mathbf{t}'' \rangle\} \end{array} \right) = \begin{bmatrix} \mathbf{t} \\ \text{F} : \begin{bmatrix} \mathbf{v} \\ \text{G} : \mathbf{b} \end{bmatrix} \\ \text{H} : \mathbf{a} \end{bmatrix}$$

The tail notation is somewhat cumbersome, especially since we are often concerned with TDFSs where the default information is mutually consistent. We can adopt an abbreviated notation for cases where there are no conflicts in which default information (i.e., the atomic FSs in the tail) is indicated by means of a slash notation, following the appropriate path in the indefeasible structure. For example, the TDFS shown in (2a) is represented in the abbreviated notation as shown in (2b).

(2) a
$$\begin{bmatrix} \mathbf{t} \\ \text{F} : \begin{bmatrix} \mathbf{v} \\ \text{G} : \top \end{bmatrix} \\ \text{H} : \top \end{bmatrix} \Big/ \begin{array}{l} \{\langle \begin{bmatrix} \text{H} : \mathbf{a} \end{bmatrix}, \mathbf{t} \rangle, \\ \langle \begin{bmatrix} \text{F} : \begin{bmatrix} \text{G} : \boxed{1} \end{bmatrix} \\ \text{H} : \boxed{1} \end{bmatrix}, \mathbf{t} \rangle\} \end{array}$$

b
$$\begin{bmatrix} \mathbf{t} \\ \text{F} : \begin{bmatrix} \mathbf{v} \\ \text{G} : /\boxed{1}\mathbf{a} \end{bmatrix} \\ \text{H} : /\boxed{1} \end{bmatrix}$$

We will use this abbreviated notation in the examples which follow. We refer the reader to L&C and Lascarides *et al.* (1996) for formal specification and detailed motivation of the TDFS formalism.[3]

2.4 Persistent defaults

Symmetric default unification can be used to extend the functionality of any linguistic framework based on, or embedded in, TFSs in two ways. Firstly, the organization of the lexicon can be extended naturally to allow for default inheritance networks of constraints. Default inheritance has been widely exploited in lexical descriptions both within and outside the TFS framework (e.g. Daelemans *et al.*, 1992). However, the TDFS

3 Note, however, that although L&C conjecture that DefFS is worst-case exponential, Malouf (1999) has developed an algorithm which is efficient even for cases where the tail is large in proportion to the indefeasible structure and where complex reentrancies are involved. Malouf's results so far suggest that practical complexity varies from linear to quadratic, and that, overall, recoding a monotonic system to use defaults can improve performance, chiefly because it reduces the complexity of the type hierarchy.

$$\begin{bmatrix} \textbf{transitive-pp-verb} \\ \text{PHON} : \textbf{climb} \\ \text{SEM} : < \begin{bmatrix}\textbf{climb-rel}\\ \text{EVENT} : \textbf{e}\end{bmatrix}, \begin{bmatrix}\textbf{p-agt-move}\\ \text{EVENT} : \textbf{e}\\ \text{ARG} : \textbf{x}\end{bmatrix}, \begin{bmatrix}\textbf{direction-rel/up-rel}\\ \text{EVENT} : \textbf{e}\\ \text{ARG} : \textbf{y}\end{bmatrix} > \end{bmatrix}$$

Figure 6
Lexical entry for *climb*

approach extends formalized and implemented proposals for default inheritance, such as DATR (Evans & Gazdar, 1989, 1996) or any version based on asymmetric unification, in that the lexicon now consists of a set of TDFSs, as opposed to a set of (T)FSs; that is, it permits default specification to persist 'beyond the lexicon' for exploitation during syntactic, semantic or discourse level operations. This is the most significant property of the TDFS framework for our use of lexical rules, so we will give a simple example of its potential utility.

Figure 6 gives a possible lexical entry for *climb* which exploits persistent default specification to capture the defeasible nature of aspects of the meaning of this verb. In this sense and pattern of realizations, *climb* can combine with a locative PP specifying the goal and/or directional path of movement. By default, the directional path is specified to be upwards. The preposition heading the PP may strengthen, override or leave default this specification, as illustrated in (3).

(3) a John climbed up the shaft
b John climbed along the shaft
c John climbed around the shaft
d John climbed through the shaft
e John climbed down the shaft

In addition, lexical rules which create an intransitive form of the verb by removing the requirement for the PP arguments must leave the default direction specification unmodified, and in all cases the lexically specified default remains defeasible during discourse processing in a context with which it is inconsistent, as illustrated in (4).

(4) a The shaft was very deep. John climbed for several hours, before reaching the bottom.
b The shaft had two inspection hatches: the first was five feet above the second. John climbed through the shaft entering from the first inspection hatch and exiting via the second.

The fact that defeasible components of meaning interact subtly with both lexical rules affecting grammatical realization and discourse context supports a framework in which defeasible semantic specifications can be explicitly represented at the lexical level. Copestake & Briscoe (1995) argue at some length for this position, and Lascarides & Copestake (1995) and Lascarides *et al.* (1995) show how lexical defaults interact appropriately with nonmonotonic discourse reasoning within the formal framework of DICE (Lascarides & Asher, 1991, 1993).

$$\begin{bmatrix}\textbf{base}\end{bmatrix} \mapsto \begin{bmatrix} \textbf{3rdsg} \\ \text{PHON} : \begin{bmatrix}\text{FUNCTION} : \textbf{f3rdsng} \\ \text{ARGS} : < \top, \boxed{1} > \end{bmatrix} \\ \text{3RDSNG} : \boxed{1} \end{bmatrix}$$

Applied to:

$$\begin{bmatrix} \textbf{base} \\ \text{PHON} : \begin{bmatrix}\text{FUNCTION} : \\ \text{ARGS} : < \textbf{walk} > \end{bmatrix} \\ \text{3RDSNG} : \\ \text{SYN LOC SUBCAT} : < \textbf{NP} > \end{bmatrix}$$

is equivalent to:

$$\begin{bmatrix} \textbf{3rdsg} \\ \text{PHON} : \begin{bmatrix}\text{FUNCTION} : \textbf{f3rdsng} \\ \text{ARGS} : < \top, \boxed{1} > \end{bmatrix} \\ \text{3RDSNG} : \boxed{1} \end{bmatrix} \stackrel{<}{\sqcap}_t \begin{bmatrix} \textbf{base} \\ \text{PHON} : \begin{bmatrix}\text{FUNCTION} : \\ \text{ARGS} : < \textbf{walk} > \end{bmatrix} \\ \text{3RDSNG} : \\ \text{SYN LOC SUBCAT} : < \textbf{NP} > \end{bmatrix}$$

$$= \begin{bmatrix} \textbf{3rdsg} \\ \text{PHON} : \begin{bmatrix}\text{FUNCTION} : \textbf{f3rdsng} \\ \text{ARGS} : < \textbf{walk}, \boxed{1} > \end{bmatrix} \\ \text{3RDSNG} : \boxed{1} \\ \text{SYN LOC SUBCAT} : < \textbf{NP} > \end{bmatrix}$$

Figure 7
Reinterpreted Third Singular Verb Formation Lexical Rule

3 Lexical Rules in the TDFS Formalism

Lexical rules differ from (default) inheritance in several ways. Firstly, a lexical rule relates lexical entries whose types are unordered, and secondly, such rules often apply recursively. Nevertheless, there are similarities between default unification and the process of creating a new lexical entry, conditionally, given the presence of another entry satisfying certain properties. In particular, asymmetric default unification presents itself as a way of implementing the requirement that information should carry over from input to output description in a lexical rule, provided that it is consistent with the output type and any further constraints specified in the rule. Furthermore, if we wish to express lexical rules in terms of the TDFS formalism outlined in §2, it makes sense to explore the utility of the nonmonotonic component developed.

Asymmetric default unification can provide the foundation for the carrying over of consistent information between input and output descriptions in lexical rules. For example, the second formulation of the rule of Third Singular Verb Formation (without coindexation) given in §1 can be reinterpreted as an application of asymmetric default unification, as illustrated in Figure 7 (an explicit encoding of function application has been introduced for clarity). Under this new interpretation of the notation (and assuming for the moment that the structures have empty tails), if the input description subsumes a lexical structure, then the result is given by asymmetrically default unifying the lexical structure with the output description. The effect is that information in the input which is inconsistent with the output description is lost, whether because it conflicts with a path specification or because the path itself is inconsistent with the output type. On the other hand, conflicting or new information in the output specification survives. The typed variant of Carpenter's definition of asymmetric default unification which was presented in §2.2 will support this approach to lexical rules.

3.1 The definition of a TDFS lexical rule

All lexical rules are of the following form, where I_a/T_a and I_b/T_b are TDFSs:

$$I_a/T_a \mapsto I_b/T_b$$

The interpretation of the rule is as follows:

Given a lexeme, I_1/T_1, where $I_1 \sqsubseteq I_a$, then the result of applying the lexical rule is a TDFS, I_0/T_0, such that:

1. $I_0 = I_b \overset{<}{\sqcap}_t I_1$
   The indefeasible output is given by asymmetrically default unifying the indefeasible part of the output specification with the indefeasible part of the input lexeme.
2. $T_0 = \mathrm{Filter}(I_0, (T_b \cup T_1))$
   The output tail is the union of the output specification's tail T_b and the input tail, T_1, minus any tail elements which are incompatible with the new indefeasible information.

Note that this definition parallels the definition of $\overset{<>}{\sqcap}$, with $\overset{<}{\sqcap}_t$ replacing $\sqcap$. Note also that although as we have described them here lexical rules apply to TDFSs, it should be clear that their application to ordinary TFSs is just the special case where there are no default components in any of the feature structures involved.

3.2 Maximal incorporation of information in a type hierarchy

When we extend atomic FSs as defined by Carpenter to a typed framework, if a TFS has a path π with value **t**, then $[\ \pi : \mathbf{t}\ ]$. is an atomic FS in the decomposition of the TFS. However, if **t** has supertypes **r**, **s** and so on, in order to get maximal incorporation of information, we must also include the atomic FSs $[\ \pi : \mathbf{r}\ ], [\ \pi : \mathbf{s}\ ]$ and so on (see L&C, section 6.3). This is important in our use of asymmetric default unification to define lexical rules, as we illustrate with the following example.

Suppose we describe the inchoative/causative alternation using the lexical rule shown in (5).

(5) $[\ \textbf{intrans-verb}\ ]/\{\} \mapsto [\ \textbf{trans-caus-verb}\ ]/\{\}$

This is intended to apply to a range of semantic subclasses and the specific semantic class should be retained after application of the lexical rule: for example, if the input is a movement verb then the result should also be a movement verb. If semantic subclasses cross-classify with types such as **trans-verb** and **intrans-verb**, this could be implemented using multiple inheritance: i.e. the type **intrans-movt-verb** would inherit from **intrans-verb** and **movt-verb**, and the type **trans-caus-movt-verb** would inherit from **trans-caus-verb** and **movt-verb** as shown in Figure 8. Assume we apply the rule in (5) to a lexeme of type **intrans-movt-verb**. If the asymmetric default unification operation used in the definition of lexical rules did not consider the supertypes of this type, the application of the rule would give an output of type **trans-verb**, because the value of the type on the input is incompatible with **trans-verb**. However, if we take account of all the supertypes of the type which are compatible with the output, the result is the desired one: **trans-caus-movt-verb**. This is illustrated in Figure 9.

3.3 Reversibility and backformation

Given the arguments in Lascarides *et al.* (1996) about the desirability of the symmetric form of default unification on the grounds of order-independence, it may seem surpris-

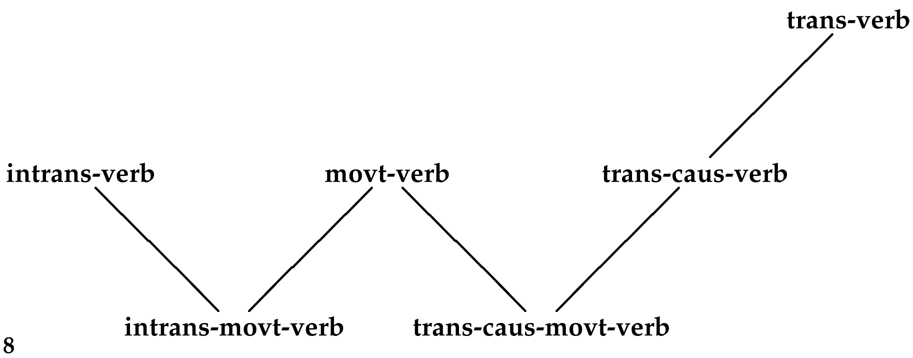


Figure 8
Multiple inheritance for movement verbs

Rule: **intrans-verb** $\mapsto$ **trans-caus-verb**

$$\begin{bmatrix} \textbf{intrans-verb} \\ \text{PHON} : \textbf{phon} \\ \text{SYN} : \begin{bmatrix} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign} \end{bmatrix} \\ \text{SEM} : < \begin{bmatrix} \textbf{verb-rel} \\ \text{EVENT} : \textbf{e} \end{bmatrix}, \begin{bmatrix} \textbf{p-agt-cause} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{x} \end{bmatrix} > \end{bmatrix} \mapsto$$

$$\begin{bmatrix} \textbf{trans-caus-verb} \\ \text{PHON} : \textbf{phon} \\ \text{SYN} : \begin{bmatrix} \text{RESULT} : \begin{bmatrix} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign} \end{bmatrix} \\ \text{ACTIVE} : \textbf{npsign} \end{bmatrix} \\ \text{SEM} : < \begin{bmatrix} \textbf{verb-rel} \\ \text{EVENT} : \textbf{e} \end{bmatrix}, \begin{bmatrix} \textbf{p-agt-cause} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{x} \end{bmatrix}, \begin{bmatrix} \textbf{p-pat} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{y} \end{bmatrix} > \end{bmatrix}$$

Input: $$\begin{bmatrix} \textbf{intrans-movt-verb} \\ \text{PHON} : \textbf{gallop} \\ \text{SYN} : \begin{bmatrix} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign} \end{bmatrix} \\ \text{SEM} : < \begin{bmatrix} \textbf{gallop-rel} \\ \text{EVENT} : \textbf{e} \end{bmatrix}, \begin{bmatrix} \textbf{p-agt-cause-move} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{x} \end{bmatrix} > \end{bmatrix}$$

Output: $$\begin{bmatrix} \textbf{trans-caus-movt-verb} \\ \text{PHON} : \textbf{gallop} \\ \text{SYN} : \begin{bmatrix} \text{RESULT} : \begin{bmatrix} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign} \end{bmatrix} \\ \text{ACTIVE} : \textbf{npsign} \end{bmatrix} \\ \text{SEM} : < \begin{bmatrix} \textbf{gallop-rel} \\ \text{EVENT} : \textbf{e} \end{bmatrix}, \begin{bmatrix} \textbf{p-agt-cause-move} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{x} \end{bmatrix}, \begin{bmatrix} \textbf{p-pat-move} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{y} \end{bmatrix} > \end{bmatrix}$$

Figure 9
Causative-inchoative revisited

ing to suggest that an order-dependent operation be the basis for the formalization of lexical rules. But it is clear that lexical rule application must be order-dependent, for example to distinguish *truthfulness* from **truthnessful*. However, bidirectionality and reversibility are required in order to model both analysis and generation, and also to deal with backformation. In fact, although conventional HPSG-style lexical rules appear to be straightforwardly interpretable as bidirectional, there is no guarantee that any given input can be recovered from knowledge of the rule and of its output, unless all the information in the input is copied over to the output. The most that can be said is that by reversing the rule a result will be obtained which is unifiable with the original input. Of course, lexical rules could be written in such a way that all information about the input can be recovered, by copying all the information over (for example to a DTR-style attribute in the output).

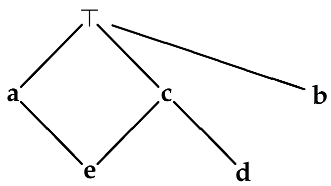
Reversibility of TDFS lexical rules is defined somewhat differently. Given a rule $FS_1 \mapsto FS_2$, and an input I which results in output O, we can recover I' defined as the unification of FS_1 with F, the most general FS such that $F \sqcap FS_2 = O$. By the definition of default unification, $I \sqsubseteq F$, and since TDFS lexical rules apply under subsumption, $I \sqsubseteq FS_1$. Thus $I \sqsubseteq I'$, that is, we can guarantee that the result of the 'reverse' application subsumes or is equal to the input (which is actually a stronger result than we obtain for conventional lexical rules). With both conventional and TDFS lexical rules, therefore, additional assumptions about the relationship between the input and the output, at least with respect to phonology/orthography and semantics, would have to be made if we wanted to guarantee that a lexical entry will be fully retrievable.

Thus, idiosyncratic information in the input lexical entry, which is overridden in the output, cannot be recovered. This actually seems to be a desirable property when we consider how backformations may be modelled. For example, *self-destruct* rather than *self-destroy* is the backformation from *self-destruction* (see Bauer, 1986:232). It is thus apparently linguistically-unmotivated to assume that the input to a lexical rule is completely recoverable from a derived form and the asymmetry of TDFS lexical rules arguably captures some of the markedness of backformation without precluding the possibility that individual reversed rules faithfully model the effects of backformations.

3.4 Discussion

Unlike the versions of lexical rules described by Riehemann (1993), Copestake & Briscoe (1995), and Meurers (1995), this interpretation cannot be incorporated into a monotonic constraint-based formalism, since the operation of lexical rules is essentially non-monotonic, in that the incorporation of additional information into the input may result in loss of information from the output. This treatment of lexical rules therefore requires that there be an interface between the lexicon and the syntactic component so that operations are carried out in a predefined order. This is not a disadvantage from our perspective, since the use of defaults to encode information which does not persist beyond the lexicon requires such an interface in any case. The cost of the fully monotonic version of lexical rules is that the complete history of rule application has to be accessible, either explicitly, as with Riehemann's approach, or implicitly (e.g. via junk slots). However, our current approach does mean that some potential applications of lexical rules are precluded, which we discuss further in §6.

In contrast with Meurer's notation, it is not possible to straightforwardly compile out our lexical rules into equivalent rules which do not use defaults, even when TFSs rather than TDFSs are considered. Consider the abstract example shown in Figure 10. To state this using conventional lexical rules would be extremely complex, since it would require a whole series of rules to deal with the possible values of F, each with negative conditions to prevent them applying incorrectly. Contrast this to Meurer's approach,



$$\begin{bmatrix} \mathbf{t} \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{r} \\ \mathrm{F} : \mathbf{a} \end{bmatrix}$$

	Input	Result
1	$\begin{bmatrix} \mathbf{t} \\ \mathrm{F} : \mathbf{b} \end{bmatrix}$	$\begin{bmatrix} \mathbf{r} \\ \mathrm{F} : \mathbf{a} \end{bmatrix}$
2	$\begin{bmatrix} \mathbf{t} \\ \mathrm{F} : \mathbf{c} \end{bmatrix}$	$\begin{bmatrix} \mathbf{r} \\ \mathrm{F} : \mathbf{e} \end{bmatrix}$
3	$\begin{bmatrix} \mathbf{t} \\ \mathrm{F} : \mathbf{d} \end{bmatrix}$	$\begin{bmatrix} \mathbf{r} \\ \mathrm{F} : \mathbf{e} \end{bmatrix}$

Figure 10
Abstract lexical rule example

where all inputs would give a result where F had the value **a**. Essentially the difference is that whereas our approach can be paraphrased as "transfer all information for a feature F which does not conflict with information on the output", Meurer's definition can be paraphrased "transfer all values for a feature F unless any information is stated about F". The example shown in Figure 9 illustrates one case for which our approach leads to desirable results. Another example arises when inflection is implemented using lexical rules, as with the example of the third person singular rule (Figure 7). With conventional lexical rules it is impossible to carry over information about the type of the input to the output (unless the rule is totally monotonic in operation, in which case it is equivalent to the use of simple subsumption-based inheritance). Therefore lexical types cannot persist after inflectional rules are applied, unless the rule is split so that one subrule applies to each type (see, for example, Copestake (1992) for further discussion). This class of problems is avoided with our current approach.

In other respects, this version of lexical rules is intermediate in expressivity between simple inheritance and conventional lexical rules. It is more powerful than simple inheritance, because the 'input' and 'output' types of lexical rules do not have to be in a fixed inheritance hierarchy, and recursive application of rules is possible. Consider for example, an alternative encoding, using simple inheritance, of the inchoative-causative rule shown in Figure 9. This requires that a type exists **caus-or-inch-verb** which has **intrans-verb** and **trans-caus-verb** as subtypes. However, in order to encode further alternations, these types themselves have to have subtypes corresponding to each rule. For example, if the goal PP in *John galloped the horse to the barn / The horse galloped to the barn* is encoded as an argument, then **intrans-verb** and **trans-caus-verb** both have to have subtypes for the goal PP and this configuration has to be replicated for all subtypes of **intrans-verb**, such as **intrans-movt-verb**. This quickly leads to highly complex type hierarchies, since the configuration of the types encodes 'reachability', as well as inheritance. Furthermore, encoding lexical rules by inheritance does not allow for any notion of semi-productivity of a rule. Individual exceptions to a rule can be encoded, for example, if *run* is encoded as an **intrans-verb** and not specified to be of the higher type **caus-or-inch-verb**, then causative forms of *run* will not be generated. But this leaves it as an accident that, at least for this class of movement verbs, it is generally the causative rather than the inchoative form which is impossible or marked. We will discuss how we can exploit the asymmetry of TDFS lexical rules to allow for semi-productivity in §5.

Our approach to lexical rules is, on the other hand, less powerful than conventional HPSG-style lexical rules because it is not possible to arbitrarily 'rearrange' material between the input and the output structures: a value which is present on a particular path in the input can either be unified with a value on the same path in the output structure, or be overridden, but it cannot be moved to a new position within the output feature structure. This approach therefore shares some of the restrictions of simple inheritance with respect to encoding potentially recursive operations, and we discuss the implications of this further in §6.

It is clear that the notion of lexical rules that we have presented encodes something like 'type reachability' rules for lexical types. This is an inherently more restricted notion than that of HPSG-style lexical rules which can also encode arbitrary operations on list-valued features. A consequence of this more restricted notion is that lexical rules cannot be used to rearrange the order of list-valued features and cannot be applied recursively in a manner which makes such lists unbounded. In fact, since lexical rules simply relate lexical types predefined in the lexicon, they cannot increase the generative capacity of the overall system in which they are embedded.[4]

In §4, nevertheless, we show that an insightful account of one rule that would be encoded via *subcat* list manipulations can be stated in this more restrictive framework. In §6 we consider the extent to which our approach can capture other putative lexical rules and argue for a clearer distinction between lexical rules and (unary) syntactic rules.[5]

4 Dative Constructions

Lascarides *et al.* (1996) present an account of the dative alternation which illustrates the utility of the TDFS framework for encoding defeasible lexical semantic entailments in terms of Dowty's (1988) proto thematic roles, and the interaction of these with the dative alternation (e.g. Green, 1974). This account drew heavily on Goldberg's (1995) analysis of dative in Construction Grammar (e.g. Fillmore *et al.*, 1988), and attempted to integrate her insights and general approach into a more formally-explicit constraint-based framework. In this section, we extend this account by embedding the account in a syntactic framework based on UCG (Zeevat *et al.*, 1987), as integrated with Dowty's approach to thematic roles and extended by Sanfilippo (1990, 1992, 1993), and augmented with linking theory (Chang, 1995).[6] This allows us to utilize TDFS lexical rules in an insightful way to express an alternation otherwise naturally treated in terms of rearrangement of a list-valued *subcat* feature. In §7 we go on to propose a probabilistic interpretation of TDFS lexical rules which allows us to capture the semi-productivity of the dative alternation, and other lexical rules.

Following Goldberg (1995) we argue that there is a family of dative constructions which exhibit the same syntactic properties and related semantic properties, exemplified in (6).

(6) a Mary gave Joe a present
b Joe painted Sally a picture
c Mary promised Joe a new car
d He tipped Bill two pounds
e The medicine brought him relief
f The music lent the party a festive air
g Jo gave Bob a punch
h He blew his wife a kiss
i She smiled herself an upgrade

The core dative construction involves a volitional agent and willing recipient and carries the entailments that the agent causes the recipient to receive the object denoted by the patient/affected-object argument, as in (6a). Under this interpretation, (6b) involves a shift in meaning where the recipient may or may not receive the affected-object, but the

4 We omit a formal proof as this would require more detailed specification of the syntactic component than is warranted in the rest of the paper.
5 An alternative method of exploiting the TDFS formalism to encode rules was mentioned in L&C (p87) and has been explored by Malouf (1999). This technique uses the TDFS description language to allow a very succinct statement of rules which use coindexation to relate their input and output: effectively paths in the input and output structure can be specified to be coindexed by default. The expanded rules have the same properties as the lexical rule variants described by Copestake (1992) or Riehemann (1993). Thus, in terms of the current paper, this encoding is a way of improving the representation of syntactic rules.
6 Nevertheless, the general approach to lexical rules is equally compatible with HPSG, Combinatory Categorial Grammar (e.g. Steedman, 1997), Tree Adjoining Grammar (e.g. Joshi, 1987) or indeed any grammatical theory embeddable in the T(D)FS representation language.

$$\left[\begin{array}{l} \textbf{dative-verb} \\ \text{PHON} : \textbf{phon} \\ \text{SYN} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l}\text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign}\end{array}\right] \\ \text{ACTIVE} : \textbf{npsign}\end{array}\right] \\ \text{ACTIVE} : \textbf{npsign}\end{array}\right] \\ \text{SEM} : < \left[\begin{array}{l}\textbf{verb-rel}\\ \text{EVENT} : \textbf{e}\end{array}\right], \left[\begin{array}{l}\textbf{p-agt-caus/p-agt-cause-transfer}\\ \text{EVENT} : \textbf{e}\\ \text{ARG} : \textbf{x}\end{array}\right], \\ \qquad\qquad \left[\begin{array}{l}\textbf{p-pat/p-pat-aff-obj}\\ \text{EVENT} : \textbf{e}\\ \text{ARG} : \textbf{y}\end{array}\right], \left[\begin{array}{l}\textbf{p-obl/p-obl-recip}\\ \text{EVENT} : \textbf{e}\\ \text{ARG} : \textbf{z}\end{array}\right] > \end{array}\right]$$

Figure 11
The Dative Type Constraint

agent acts with this intention, whilst (6c) involves a similar shift as the act of transfer is intended to take place at some point in the future and may not in fact occur. (6d), unlike the previous examples, does not have an oblique counterpart. The remaining examples all appear to involve metaphorical or idiomatic extensions to the core dative construction.

We represent the difference in the basic (abstract) meaning of the dative construction in (6a) and (6b) in terms of entailments associated with proto thematic roles, so that 'agent' becomes **p-agt-cause-transfer** in (6a) and **p-agt-cause-make-intend-transfer** in (6b). There are lexical rules which relate the dative construction with the alternative oblique complementation patterns involving *to* and *for* PP arguments, respectively, which alter the proto-agent role of a 'creation' verb such as *paint* from **p-agt-cause-make** to **p-agt-cause-make-intend-transfer**. We abbreviate the different entailments (willingness and successful transfer) concerning the first object in (6a) and (6b) as 'recipient' and 'benefactive', respectively. It should only be necessary to state the form of the dative construction once. Furthermore, it should not be necessary to say that verbs of creation, such as *paint* are lexically ambiguous between two- and three- place predicates; rather it is participation in the dative construction itself which creates the third benefactive argument for these inherently two-place predicates. We also assume without explicit argument here that the *for* PP variant is produced by a lexical rule introducing the optional PP.

In the TDFS framework we can state the semantics of the dative construction independently of specific lexical heads (or arguments) as a type constraint on the set of dative lexical items. The type constraint expressed as a TDFS in Figure 11 achieves this and utilizes default specification of the proto-roles on the arguments so that specific verbs can override the core entailments. This constraint stipulates that dative verbs will necessarily have a **p-agt-cause** role, but, by default, this will be specialized to **p-agt-cause-transfer**, thus expressing the generalization that the dative construction usually implies that some transfer has taken place. Similarly, the role on the proto-patient is **p-pat-aff-obj** by default, since the object will usually be affected, and **p-obl-recip** captures the entailment that the oblique argument usually corresponds to a recipient. (We assume that the second object is treated as an oblique (indirect object) argument in the framework, though nothing particularly rests on this assumption.)

Most verbs will not be directly specified in the lexicon as being of type dative, but will become associated with this type via the lexical rules relating oblique-transfer verbs and transitive-creation verbs to it. The type constraints for these source types are given

$$\left[\begin{array}{l} \textbf{oblique-transfer-verb} \\ \text{PHON} : \textbf{phon} \\ \text{SYN} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign} \end{array}\right] \\ \text{ACTIVE} : \textbf{npsign} \end{array}\right] \\ \text{ACTIVE} : \textbf{pptosign} \end{array}\right] \\ \text{SEM} : < \left[\begin{array}{l} \textbf{verb-rel} \\ \text{EVENT} : \textbf{e} \end{array}\right], \left[\begin{array}{l} \textbf{p-agt-cause-transfer} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{x} \end{array}\right], \left[\begin{array}{l} \textbf{p-pat-aff-obj} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{y} \end{array}\right], \left[\begin{array}{l} \textbf{p-obl-recip} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{z} \end{array}\right] > \end{array}\right]$$

Figure 12
The Oblique Transfer Type Constraint

$$\left[\begin{array}{l} \textbf{trans-create-verb} \\ \text{PHON} : \textbf{phon} \\ \text{SYN} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign} \end{array}\right] \\ \text{ACTIVE} : \textbf{npsign} \end{array}\right] \\ \text{SEM} : < \left[\begin{array}{l} \textbf{verb-rel} \\ \text{EVENT} : \textbf{e} \end{array}\right], \left[\begin{array}{l} \textbf{p-agt-cause-make} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{x} \end{array}\right], \left[\begin{array}{l} \textbf{p-pat-aff-obj} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{y} \end{array}\right] > \end{array}\right]$$

Figure 13
The Transitive Creation Type Constraint

The Dative Lexical Rule

oblique-transfer-verb $\mapsto$ **dative-verb**

Create-to-Benef-Dative Lexical Rule

$$\textbf{trans-create-verb} \mapsto \left[\begin{array}{l} \textbf{dative-verb} \\ \text{SEM} : < \left[\begin{array}{l} \textbf{verb-rel} \\ \text{EVENT} : \textbf{e} \end{array}\right], \left[\begin{array}{l} \textbf{p-agt-cause-make-intend-transfer} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{x} \end{array}\right], \\ \qquad \left[\begin{array}{l} \textbf{p-pat-aff-obj} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{y} \end{array}\right], \left[\begin{array}{l} \textbf{p-obl-benef} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \textbf{z} \end{array}\right] > \end{array}\right]$$

Figure 14
Dative Lexical Rules

in Figure 12 and Figure 13, respectively. (We ignore the issue of how the information represented at these types might be factored between supertypes to capture further generalizations concerning verb classes; see, for example, Sanfilippo, 1993.) The two lexical rules required are given in Figure 14.

These rules can be stated quite simply with reference to the type system for verbs. The first rule, which is the core (recipient) dative rule, is simply stipulated as a 'reachability' relationship between the two types **oblique-transfer-verb** and **dative-verb**. However, when this rule is applied to a lexical entry of type **oblique-transfer-verb**, the specifications of the proto-roles as **p-agt-cause-transfer** and **p-obl-recip** in the result will be indefeasible, in contrast to their defeasible status in **dative-verb**, because they are indefeasible in the basic type. We show the input and output TDFSs for a transfer verb (*give*) undergoing this rule in Figure 15. In contrast, the benefactive dative rule specializes the dative type so that the proto-role entailments stipulated override the defaults on the type **dative-verb**. Thus, **p-agt-cause-make-intend-transfer** overrides **p-agt-cause-**

Input:

$$\begin{bmatrix}\textbf{oblique-transfer-verb}\\ \text{PHON}: \textbf{give}\\ \text{SYN}: \begin{bmatrix}\text{RESULT}: \begin{bmatrix}\text{RESULT}: \begin{bmatrix}\text{RESULT}: \textbf{ssign}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{ACTIVE}: \textbf{pptosign}\end{bmatrix}\\ \text{SEM}: < \begin{bmatrix}\textbf{give-rel}\\ \text{EVENT}: \textbf{e}\end{bmatrix}, \begin{bmatrix}\textbf{p-agt-cause-transfer}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{x}\end{bmatrix}, \begin{bmatrix}\textbf{p-pat-aff-obj}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{y}\end{bmatrix}, \begin{bmatrix}\textbf{p-obl-recip}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{z}\end{bmatrix} >\end{bmatrix}$$

Output:

$$\begin{bmatrix}\textbf{dative-verb}\\ \text{PHON}: \textbf{give}\\ \text{SYN}: \begin{bmatrix}\text{RESULT}: \begin{bmatrix}\text{RESULT}: \begin{bmatrix}\text{RESULT}: \textbf{ssign}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{SEM}: < \begin{bmatrix}\textbf{give-rel}\\ \text{EVENT}: \textbf{e}\end{bmatrix}, \begin{bmatrix}\textbf{p-agt-cause-transfer}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{x}\end{bmatrix}, \begin{bmatrix}\textbf{p-pat-aff-obj}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{y}\end{bmatrix}, \begin{bmatrix}\textbf{p-obl-recip}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{z}\end{bmatrix} >\end{bmatrix}$$

Figure 15
give

Input:

$$\begin{bmatrix}\textbf{trans-create-verb}\\ \text{PHON}: \textbf{paint}\\ \text{SYN}: \begin{bmatrix}\text{RESULT}: \begin{bmatrix}\text{RESULT}: \textbf{ssign}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{SEM}: < \begin{bmatrix}\textbf{paint-rel}\\ \text{EVENT}: \textbf{e}\end{bmatrix}, \begin{bmatrix}\textbf{p-agt-cause-make}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{x}\end{bmatrix}, \begin{bmatrix}\textbf{p-pat-aff-obj}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{y}\end{bmatrix} >\end{bmatrix}$$

Output:

$$\begin{bmatrix}\textbf{dative-verb}\\ \text{PHON}: \textbf{paint}\\ \text{SYN}: \begin{bmatrix}\text{RESULT}: \begin{bmatrix}\text{RESULT}: \begin{bmatrix}\text{RESULT}: \textbf{ssign}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{ACTIVE}: \textbf{npsign}\end{bmatrix}\\ \text{SEM}: < \begin{bmatrix}\textbf{paint-rel}\\ \text{EVENT}: \textbf{e}\end{bmatrix}, \begin{bmatrix}\textbf{p-agt-cause-make-intend-transfer}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{x}\end{bmatrix},\\ \qquad \begin{bmatrix}\textbf{p-pat-aff-obj}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{y}\end{bmatrix}, \begin{bmatrix}\textbf{p-obl-benef}\\ \text{EVENT}: \textbf{e}\\ \text{ARG}: \textbf{z}\end{bmatrix} >\end{bmatrix}$$

Figure 16
paint

transfer and **p-obl-benef** overrides **p-obl-recip**. The effect is that the result does not imply that transfer of the affected object was necessarily successful. Figure 16 shows the input and output for a creation verb (*paint*) undergoing this rule. Thus these rules correct predict that while (6a) implies successful transfer, (6b) only implies that transfer was intended.

The verb types shown so far do not make explicit the linking between proto-role

$$\left[\begin{array}{l} \textbf{dative-verb} \\ \text{PHON} : \textbf{phon} \\ \text{SYN} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign}\,\boxed{1} \end{array}\right] \\ \text{ACTIVE} : \textbf{npsign}\,\boxed{2} \end{array}\right] \\ \text{ACTIVE} : \textbf{npsign}\,\boxed{2} \end{array}\right] \\ \text{SEM} : < \left[\begin{array}{l} \textbf{verb-rel} \\ \text{EVENT} : \textbf{e} \end{array}\right], \left[\begin{array}{l} \textbf{p-agt-caus/p-agt-cause-transfer} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \boxed{1} \end{array}\right], \\ \qquad \left[\begin{array}{l} \textbf{p-pat/p-pat-aff-obj} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \boxed{2} \end{array}\right], \left[\begin{array}{l} \textbf{p-obl/p-obl-recip} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \boxed{2} \end{array}\right] > \end{array}\right]$$

Figure 17
Incoherent Linked Dative TDFS

arguments and the semantic indices of syntactic arguments. In HPSG and Construction Grammar, such linking is provided on a construction-by-construction basis in the type definitions for each construction/sign. The standard way of linking arguments in a (T)FS framework is to make the semantic indices of the syntactic arguments reentrant with the indices representing the arguments of the (proto) thematic roles. However, if we adopted this approach, default unifying linked versions of **oblique-transfer-verb** with **dative-verb** would result in the incoherent structure shown in Figure 17 because the reentrancies in the input type description are consistent with the output type description, and are thus incorrectly preserved.

There are ways in which we could extend the formalism to avoid this problem, for instance by allowing specification of inequalities (Carpenter, 1992) which could be used to explicitly prevent inappropriate coindexation (see Lascarides and Copestake (1999) for inequalities in the TDFS framework). However, for the purposes of linking, this restriction on the expressivity of lexical rules is a virtue rather than a vice: Pinker (1989) argues on quite independent grounds that linking rules should apply after lexical rules so that all lexical entries are subject to the same linking constraints. By factoring linking constraints out of the type constraints on specific constructions we can eliminate redundancy and explicitly express the appropriate linking generalizations. Chang (1995) follows Wechsler (1991) and Davis (1996) in assuming that linking generalizations are captured via constraints specified as TFSs, but makes the assumption that linking applies after lexical rule application. She shows how linking generalizations can be captured within the TDFS framework assuming the linguistic framework of UCG / Construction Grammar and proto thematic roles outlined above. By reorganizing the semantic type hierarchy to take account of the distinction between internal and external causation (see, Levin and Rappaport Hovav, 1995), Chang is able to define eleven partly defeasible verbal linking constraints, each specifying the link between one thematic role and one argument position, correctly linking monadic, dyadic and triadic predicates which may undergo causative-inchoative, passive and/or dative alternations by persistent default unification of all linking constraints with each distinct basic and derived verbal type. Thus, our approach to lexical rules is similar to that of Pinker (1989) in that all basic and derived lexical entries are subject to a few general linking constraints which coindex syntactic arguments with appropriate proto-roles. However, in common with Goldberg (1995), we adopt the position that such rules are not fully reducible to operations on semantic representations, but rather concern the interplay of syntax and semantics in (bounded dependency) constructions.

Verbs such as *promise* in (6c) are treated as a subclass of transfer verbs, which we call future transfer verbs, defined by a subtype of **oblique-transfer-verb** altering the proto role entailments analogously to the benefactive case already discussed, so that no entailment of actual transfer is made. If the source type description is modified in this fashion, the main dative lexical rule will apply and produce an output TDFS in which the dative type defeasible proto role entailments are overwritten to something like **p-agt-cause-intend-fut-transfer** and **p-pat-benef**, respectively. Thus, all that is required is an extra source type description for this semantic subclass of verbs to produce correct behaviour.

The verb *tip* in (6d) is an example of a small class of verbs (also including *envy*), which show that dative variants can exist without an oblique 'source'. In an approach which generated derived entries strictly from basic source entries, these cases would be problematic. However, in an approach such as ours where lexical rules relate independently-defined (specialized) type descriptions, there is no prediction that output types of lexical rules cannot be basic types for some verbs. We can simply treat *tip* and similar cases as basic dative verbs and alter the defeasible proto role entailments of the dative type as appropriate. Thus, we make no prediction that the set of dative verbs will be a subset of the set of oblique verbs.

Goldberg (1995) argues that (6e,f) are licensed by a metaphorical extension of the transfer relation by which causal events are viewed as transfers. Causing an effect in an entity is understood as transferring that effect to it. We capture this by altering the entailments associated with verbs such as *lend* by the proto-roles specified by **oblique-transfer-verb** using a lexical rule which creates an entry of a sister type **met-oblique-transfer-verb** which specifies different proto-roles. The dative lexical rule would also apply to this subtype, capturing the fact that this metaphorical extension in the dative construction parallels a similar extension of the same verb set in the oblique *to* prepositional phrase construction.

In contrast, (6g,h) provide evidence that certain quasi-idiomatic expressions need to be associated directly with **dative-sign** as they have no counterparts in such oblique expressions. We assume that (quasi-)idioms are best represented as subtypes of lexical signs in which not only the syntactic head but also other arguments are severely constrained in terms of lexical selection. Thus Goldberg claims the quasi-idiomatic expressions in (6g,h) are licensed by a metaphor which involves understanding actions intentionally directed at another person as being entities transferred to that person. As a first approximation, we might represent this process in terms of the subtype of **dative-verb** shown in Figure 18, in which we have overridden the default proto-role specifications with entailments specific to the metaphor which we assume also serve to constrain the range of acceptable arguments to the (transfer) verb.

Finally, (6i) demonstrates that reflexive datives can sometimes be formed from intransitive verbs. The semantic restrictions on the source for such constructions are not entirely clear to us, however, all the putative similar examples to the attested (6i) seem innovative and much less conventionalized than the core dative examples. It would be straightforward to introduce a lexical rule mapping intransitive verbs to the dative construction. However, the relative productivity of this putative rule should be represented as being much lower than the two rules discussed above. Nothing in the representation we have presented so far equips us to deal with the issue of (semi-)productivity which also arises with the two rules introduced earlier; for example, famously *donate* does not undergo the dative alternation although it is uncontroversially an oblique transfer verb. In the next section we consider this issue in detail.

$$\left[\begin{array}{l} \textbf{dative-met-verb} \\ \text{PHON} : \textbf{phon} \\ \text{SYN} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l} \text{RESULT} : \left[\begin{array}{l} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign} \end{array}\right] \\ \text{ACTIVE} : \textbf{npsign} \end{array}\right] \\ \text{ACTIVE} : \textbf{npsign} \end{array}\right] \\ \text{SEM} : < \left[\begin{array}{l} \textbf{verb-rel} \\ \text{EVENT} : \mathbf{e} \end{array}\right], \left[\begin{array}{l} \textbf{p-agt-caus/p-agt-cause-met-transfer} \\ \text{EVENT} : \mathbf{e} \\ \text{ARG} : \mathbf{x} \end{array}\right], \\ \qquad\qquad \left[\begin{array}{l} \textbf{p-pat-transferred-ev} \\ \text{EVENT} : \mathbf{e} \\ \text{ARG} : \mathbf{e}' \end{array}\right], \left[\begin{array}{l} \textbf{p-obl-met-recip} \\ \text{EVENT} : \mathbf{e} \\ \text{ARG} : \mathbf{z} \end{array}\right] > \end{array}\right]$$

Figure 18
Metaphorical Dative Type Constraint

5 Semi-productivity and Probabilistic TDFS Grammar

The search for a fully productive statement of verb alternations such as dative has led to an increasingly semantic perspective on such rules. However, whilst a reasonable case can be made that the conditions on the rules introduced in Figure 14 express necessary conditions for their application, they do not capture sufficient conditions because of the existence of semantically similar non-alternating verbs, such as *donate* as opposed to *give*, or *create* as opposed to *paint*, as (7a,b) illustrate.

(7) a *The president donated the club a trophy
b *The architect created them a bridge

Pinker (1989) argues that so-called broad semantic classes of the type identified in the dative rules given above (i.e., creation or transfer verbs) provide necessary conditions for lexical rule application, but that narrow class lexical rules should be specified breaking down such rules into a number of fully-productive subcases. In the attempt to define such subcases Pinker is forced to make subtle and often unintuitive distinctions, and to claim that the meaning components involved are features of Universal Grammar to which the grammars of any language may be sensitive. For example, in attempting to differentiate the dative alternating and non-alternating subclasses in (8a,b), Pinker (1989:110-111) characterizes those in (8a) as 'verbs of continuous imparting of force in some manner causing accompanied motion' and those in (8b) as 'verbs of continuous causation of motion... in a deictically-specified direction'

(8) a *John dragged/pulled Bill the computer / the computer to Bill
b John brought/carried/took Bill the computer / the computer to Bill
c John pushed Bill his beer / ?calculator / *computer
d John slid Bill his beer / ?calculator / *computer

Furthermore, the continuous nature of force imparted seems crucial to the acceptability of dative with the first class: so (8c) is more acceptable, the lighter the affected object and the more plausible it is to construct a scenario in which *push* is synonymous with *slide*, as in (8d).

Notwithstanding the subtle distinctions which Pinker makes and the often very tiny size of the narrow verb classes he identifies, there remain exceptions to his generalizations. For example, in British English, which is generally slightly more liberal with respect to the dative alternation than American English, the examples in (9a,b) are pairs, classified in the same narrow class by Pinker, where one is acceptable in the dative and one is not.

(9) a John designed / *created them a bridge
b I picked / *indicated her a dress

Such dialectal disparities would be less problematic if they applied uniformly to narrow classes but instead they appear to be insensitive to Pinker's classifications. In addition, Pirelli *et al.* (1994) and Nicholls (1995) document cases where Pinker's narrow classes do not generalize to equivalent alternations in Italian and French, respectively; demonstrating that cross-linguistic disparities are similarly insensitive to putative narrow classes. More generally Pinker's notion of a fully productive narrow-class lexical rule is falsified by many examples cited in Boguraev and Briscoe (1989), Levin (1992) and Schütze (1997). We conclude that the programme of treating all exceptions to dative as systematic, as opposed to accidental or idiosyncratic (Wasow, 1980), fails for dative movement, and will, we suspect, fail for most verb diathesis alternations. Therefore, it is not possible to characterize such rules as a fully productive generative operation.

Within the generative tradition of work on lexical rules, the only alternative to treating such rules as fully productive generalizations is to treat them as redundancy rules of some form (e.g. Jackendoff, 1975 and see the discussion in §1). However, this approach does not account for the *semi*-productive nature of such rules. For example, Pinker notes with respect to the dative alternation that a variety of recent nonce verbs readily undergo dative because they are clearly members of the communication subclass of transfer verbs, as (10a) illustrates.

(10) a John faxed / xeroxed / emailed his colleagues a copy of the report
b Sun donated us a bunch of computers
c John explained me the problem
d He stripped him the ball

A redundancy rule only relates existing lexical entries to achieve abbreviation in the statement of the lexicon. It cannot be generatively applied to nonce usages. Similarly, the examples in (10b,c,d) are all attested uses of the dative which violate putative (narrow class) morphological or semantic constraints on its application from Pinker (1989:155-7). The existence of creative or analogous application to nonce usages and attested exceptions to the narrow class rules makes the redundancy rule approach unsatisfactory, at least, when formalized as a purely abbreviatory device (see §1 above).

The search for narrow classes and full productivity seems futile for rules of verb alternation because such rules are inherently semi-productive in the same manner that derivational rules are often characterized as semi-productive (e.g. Bauer, 1983; Biber, 1985). Instead, we argue, following Goldberg (1995), who in turn is influenced by theories of semi-productivity developed for morphology, that rules of verb alternation are sensitive to both type and token frequency effects which determine language users' assessments of the degree of acceptability of a given derived form and also their willingness to apply a rule in producing or interpreting a novel form. Bauer (1983:71f), in supporting the view that lexical rules should be treated as fully productive generative rules analogous to those employed in syntactic description, argues that it is this greater 'item-familiarity' of lexical items which allows judgements of relative novelty / conventionality to be built up. He points out that there are simply too many combinatoric possibilities at the sentential level for the frequency of particular combinations to be assessed with any confidence by a language user. However, in the case of words and, we might add, idioms the range of possibilities, though large, is not so great that judgements of novelty based on frequency of use cannot be acquired. Bauer argues, therefore, that accounting for semi-productivity is an issue of performance, not competence. There is considerable evidence that language users are very sensitive to the relative frequency of variant forms and senses of lexical items. Such assumptions underlie influential theories of language variation and change (e.g. Labov, 1972) and psycholinguistic accounts of preferences and misinterpretation during language comprehension (e.g. Kelly and Martin, 1994).

The frequency with which a given word form is associated with a particular lexical entry (i.e. sense or grammatical realization) is often highly skewed; Church and Mercer (1993) point out that a model of part-of-speech assignment in context will be 90% accurate (for English) if it simply chooses the lexically most frequent part-of-speech for a given word. In the LOB corpus, there are about 18 times as many instances of *believe* in the most common subcategorization class (sentential complement) as in the 4 least common classes combined, and other multiple complement taking verbs show similar strong skews (e.g. Briscoe *et al.*, 1990). In the absence of other factors, it seems very likely that language users utilize frequency information to resolve indeterminacies in both generation and interpretation. Such a strategy is compatible with and may well underlie the Gricean Maxim of Manner, in that ambiguities in language will be more easily interpretable if there is a tacit agreement not to utilize abnormal or rare means of conveying particular messages. We can model this aspect of language use as a conditional probability that a word form will be associated to a specific lexical entry, derived using a maximum likelihood estimator:[7]

$$\text{Prob(lexical-entry} \mid \text{word-form)} = \frac{freq(\text{lexical-entry with word-form})}{freq(\text{word-form})}$$

This proposal is not novel and is the analogue of proposals to associate probabilities with initial trees in, for example, a Lexicalized Tree Adjoining Grammar (Resnik, 1992; Schabes, 1992). However, it differs from recent proposals by, for example, Brew (1995) to associate probabilities with values on paths in a TFS formalism underlying HPSG, as the probabilistic information is much less fine-grained. We associate a single probability with each complete TDFS which represents a lexical entry. In a probabilistic grammar based on this approach, the probability of a derivation must depend in part on details of the grammatical approach adopted. In a categorial framework it may be there are only mutually exclusive schemata for combining lexical entries into phrasal and clausal signs, so the probability of a given derivation can be treated as the product of the probability of the lexical entries utilized in that derivation. In this case, for word-forms i in sentence:

$$\text{Prob(sent-interp)} = \prod_i (\text{lex-entry} \mid \text{word-form}_i)$$

In frameworks which incorporate alternative competing syntactic rule schemata or operations, it might be necessary to associate probabilities with such rules and treat the probability of a derivation as the combined product of the probability of the syntactic operations applied and the lexical entries utilized (e.g. Schabes, 1992). Under this formulation, the conditional probability of a lexical entry given a word form is independent of the larger context in which the word occurs (except to the extent that this is encoded in the lexical entry). This approximation ties the number of probabilistic parameters to be estimated by the language user to the size of the user's lexicon and is thus the probabilistic analogue of the item-familiarity approach described above. It say, in effect, that users can track the relative frequency of words/lexemes but not of (most) phrases or sentences.

We assume that lexical probabilities are acquired for both basic and derived lexical entries independently of any lexical rules used to create derived entries. Thus we make no claim that a derived entry will necessarily be less frequent than a basic one. It might seem that this assumption commits us to a 'full entry' theory of the lexicon (e.g. Aronoff, 1976; Jackendoff, 1997a) in which all possible words are present; that is, the consequences of lexical rules are precomputed. In the limit, the full entry theory cannot be correct because of the presence of recursive lexical rules such as derivational rules of *re-*, *anti-* or

7 In what follows we assume familiarity with the basic axioms of probability theory and with statistical estimation (e.g. Box and Tiao, 1973). We should emphasize that we are proposing a probabilistic version of a grammatical theory developed in the TDFS framework, not as a solution to practical engineering problems of parsing, but as a theoretical account of the 'item-familiarity' view of semi-productivity. We doubt that this theory is psycholinguistically accurate in the sense that language users literally compute probabilities and abide by the axioms of probability theory. However, probability theory provides a precise and clear framework in which to represent estimates of the relative likelihood of events.

$$\begin{bmatrix} \textbf{trans-caus-verb} \\ \text{PHON} : \textbf{fax} \\ \text{SYN} : \begin{bmatrix} \text{RESULT} : \begin{bmatrix} \text{RESULT} : \textbf{ssign} \\ \text{ACTIVE} : \textbf{npsign}_{\boxed{2}} \end{bmatrix} \\ \text{ACTIVE} : \textbf{npsign}_{\boxed{3}} \end{bmatrix} \\ \text{SEM} : < \begin{bmatrix} \textbf{fax-rel} \\ \text{EVENT} : \textbf{e} \end{bmatrix}, \begin{bmatrix} \textbf{p-agt-cause} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \boxed{3} \end{bmatrix}, \begin{bmatrix} \textbf{p-pat-aff-obj} \\ \text{EVENT} : \textbf{e} \\ \text{ARG} : \boxed{2} \end{bmatrix} > \end{bmatrix}$$

create/transfer-lexeme-fsm(trans(0.2), oblique-tr(0.3) . . .)

Figure 19
Lexeme for *fax*

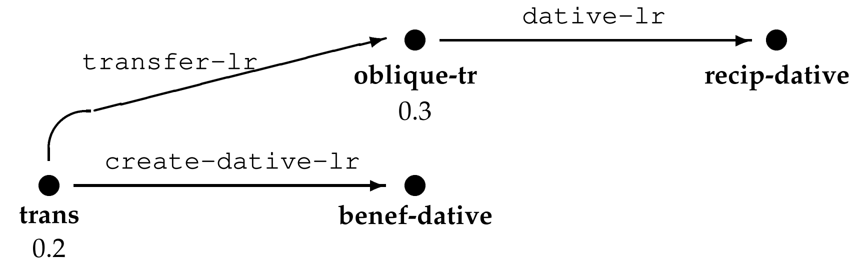


Figure 20
FSM for *fax*

great- prefixation in words such as *rereprogram*, *anti-anti-missile* or *great-great-grandfather*. Instead we adopt an intermediate position in which basic entries are augmented with a representation of the attested lexical rules which have applied to them and any such derived chains, where both the basic entry and these 'abbreviated' derived entries are associated with a probability. One way of formalizing and implementing this approach is to adopt the covariation technique of Meurers & Minnen (1997) discussed in §1, in which finite-state machines (FSMs) representing the possible lexical rules which can apply to each basic lexical entry are associated with equivalence classes of such entries and the entry is simplified to information common between the variants. If we assume a precompiled representation of this form, conditional probabilities that a word form will be associated with a particular (basic or derived) entry can be associated with states in the FSM, as illustrated in Figure 19.

In this representation, the states of the FSM, which have been given mnemonic names corresponding to their types, are each associated with a probability representing the relative likelihood that *fax* will be associated with the derived entry which results from applying the rule to the source entry (the probabilities shown here are purely for illustrative purposes). We call this representation the lexeme for a given word. Figure 20 shows part of the corresponding FSM explicitly. Note that there are states with no associated probabilities, reflecting possible but unattested usages. The topology of the FSM associated with a given word may be shared with other words, but the specific probabilities associated with the states representing lexical entries will be idiosyncratic, so that the each lexeme representation must minimally encode the unique name of the relevant FSM and a probability for each attested state / lexical entry as shown in Figure 19. If the derived form is irregular in some way, then the exceptional information can be stipulated at the relevant state, and the feature structure calculated by default-

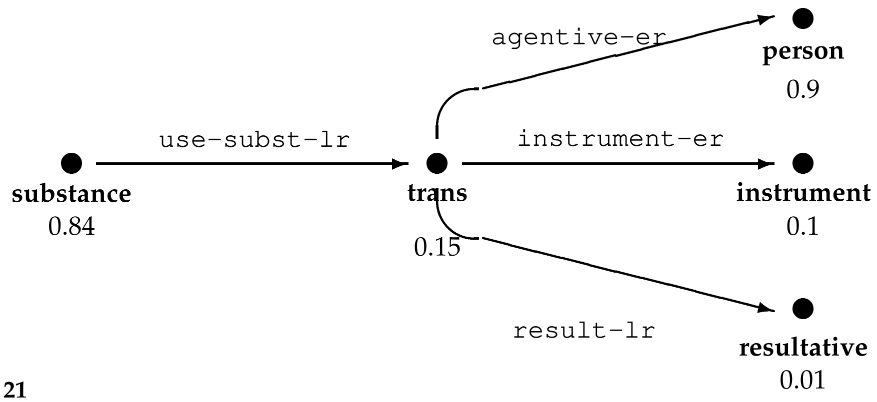


Figure 21
Lexeme for *lacquer*

unifying the specified information with the productive output of the lexical rule. For example, if *beggar* is treated as derived by the agentive *-er* rule (which is reasonable synchronically), then the irregular morphology can be stipulated and will override the predicted *begger*.

The resulting FSM is not equivalent to a Markov model because probabilities on states represent output probabilities and not transition probabilities in the machine. In addition, since the probabilities encode the relative likelihood that a given word *form* will associate with a particular lexical entry, the set of probabilities on states of a FSM will not be globally normalized. One FSM will represent the application of both rules of conversion (zero affixation) and rules of derivation to a given lexeme and the latter will change the form of the word, and thus participate in a different distribution. See for example, Figure 21, which is intended to cover the noun and verb *lacquer*, plus the derived form, *lacquerer* (with agentive and instrument readings taken as distinct). Thus, probabilities on states in FSMs are not required to sum to one, though conditional probabilities of the set of possible (attested and unattested) lexical entries for a given word form are.

One immediate problem with the proposed representation is that certain rules may apply cyclically or recursively creating an infinite set of entries. The FSM encoding devised by Meurers and Minnen is specifically developed as a form of precompilation compatible with this possibility. The majority of clearly recursive or cyclic rules in the literature are derivational, so it is clear from the word form how many times a rule has applied. We can extend the probabilistic encoding scheme to allow sets of probabilities to be encoded on states annotated with number of affixations (e.g. [q4, [anti,*p*1; anti-anti,*p*2;. . .]]). We assume for now that rules of conversion, such as most verb alternation rules, do not apply cyclically or recursively and discuss apparent exceptions in §6.

A second problem with the acquisition of reliable estimates of such probabilities for a language user (or implemented parser) is that many possibilities will remain unseen and will, therefore, be unattested. For instance, the fact that *donate* has not been seen in the dative construction may indicate the ungrammaticality of this realization or merely reflect lack of linguistic exposure to the appropriate dialect, register, or whatever. The simple maximum likelihood estimator shown above will assign zero probability to unseen events. There are a variety of methods for estimating probabilistic parameters or smoothing probability distributions which avoid assigning zero probability to unseen events. One standard approach assigns a hypothetical single observation to each unseen event in a distribution before normalizing frequencies to obtain probabilities. This captures the intuition that the more frequent the observation of some events in a distribution, the less likely it is that the unseen possibilities will occur. Thus, a rare word with only a few observations may be more likely to be seen in an alternative realization than a very frequent word which has been observed many times in some subset of the possible realizations licensed by the grammar. However, all unseen events will be assigned the same probability within each distinct distribution and this is at best a gross estimate of their actual distribution. (The technique is analogous to assuming a uniform prior distribution within the framework of Bayesian estimation.)

In the case of unattested derived lexical entries for a given word form, the relative productivity of the lexical rule(s) required to produce the derived entry are the most likely source of information to estimate the probability of an unattested derived entry given a word form.[8] Within the probabilistic framework presented above lexical rules are not directly associated with probabilities. Nevertheless we can represent the relative productivity of each lexical rule by calculating the ratio of possible to attested outputs for each rule (see Aronoff, 1976):

$$\text{Prod(lexical-rule)} = \frac{M}{N}$$

(where N is the number of attested lexical entries which match the lexical rule input and M is the number of attested output entries). This is a very simple estimate of productivity, and more complex accounts are considered below.

The estimate for degree of productivity of a rule can be combined with smoothing to obtain a variant enhanced smoothing method of the type discussed by Church & Gale (1994), capable of assigning distinct probabilities to unseen events within the same distribution. This can be achieved by estimating the held back probability mass to be distributed between the unseen entries using the basic smoothing method outlined above and then distributing this mass differentially by multiplying the total mass for unseen entries (expressed as a ratio of the total observations for a given word) by a different ratio for each lexical rule. This ratio is obtained by dividing the ratio representing the productivity of the lexical rule(s) by the sum of the ratios of the lexical rules required to construct all the unseen entries.

$$\text{Unseen-pr-mass(word-form)} = \frac{\text{number-of-unattested-entries(word-form)}}{freq(\text{word-form}) + \text{number-of-unattested-entries(word-form)}}$$

$$\text{Est-freq(lex-entry}_i \text{ with word-form}_j) = \text{Unseen-pr-mass(word-form}_j) \times \frac{Prod(lr_i)}{\sum Prod(lr_1), \ldots, Prod(lr_n)}$$

(where $lr_1 \ldots lr_n$ are the n lexical rules needed to derive the n unattested entries for word-form$_j$) This will yield revised ratios for each given word which can then be normalized to probabilities.

To make this clearer, consider the use of the probabilities to drive interpretation in the case of a nonce usage; for example, a language user faced with an unattested

8 An estimate of the relative productivity of a lexical rule would correspond to Goldberg's (1995) notion of type frequency, whilst the conditional probability of a lexical entry being associated with a specific word form corresponds to her token frequency.

realization (in their experience) of *fax* in a dative construction, such as *fax me the minutes of the last meeting*. Given the assumptions made in the lexeme representation in Figure 19, *fax* may undergo either the benefactive-dative or recipient-dative rules to yield a dative realization. These rules would produce either a deputive reading where, although the speaker is a beneficiary of the action, the recipient is unspecified (Allerton, 1984), or a reading where the speaker is also the recipient of the transfer action. Choosing between these rules in the absence of clear contextual information could be achieved by choosing the derivation (and thus interpretation) with highest probability. This would depend solely on the relative probability of the unseen derived entries created by applying these two rules to *fax*. This would be (pre)computed by applying the formulae above to a representation of the lexeme for *fax* in which ratios represent the number of observations of an entry for a given word form over the total number of observations of that word form, and unattested entries are noted and assigned one observation each

$$\text{create/transfer-lexeme-fsm}\big(\text{trans}(\tfrac{20}{100}), \text{oblique-tr}(\tfrac{30}{100}), \\ \text{recip-dative}(\tfrac{1}{100}), \text{benef-dative}(\tfrac{1}{100}), \ldots\big)$$

Now if we assume that the recipient dative rule can apply to 100 source entries and the resulting derived entries are attested in 60 cases, whilst the benefactive dative can apply to 1000 entries and the derived entries are attested in 100 cases, we can compute the revised estimates of the probabilities for the unseen entries for *fax* by instantiating the formula for estimated frequency as follows:

$$\text{Est-freq}(\textit{fax} \text{ with recipient-dative}) = \frac{2}{100} \times \Big(\frac{1}{\sum(\frac{60}{100}, \frac{100}{1000})} \times \frac{60}{100}\Big)$$

and similarly for the benefactive-dative case. The resulting ratios can then be converted to probabilities by normalizing them along with those for the attested entries for *fax*. In this case, the recipient reading will be preferred as the recipient dative rule is more productive.

This general approach could be refined in order to take account of Pinker's observations concerning narrow class rules, and already handles the possibility of specialized subcases of more general rules. For example, we could factor the computation of productivity between subtypes of the input type of a rule and derive more fine-grained measures of productivity for each narrow class a rule applies to (assuming the type system is not recursive in such a way that the subclasses of the input type are infinite). In the case of specialized subcases of lexical rules which apply to a narrower range of lexical items, but yield a more specific interpretation (such as the rules of Meat or Fur grinding, as opposed to general Grinding, proposed in Copestake & Briscoe, 1995; see §6), the relative productivity of each rule will be estimated in the manner described above, but the more specialized rule is likely to be more productive since it will apply to fewer entries than the more general rule. Similarly, in Figure 21, we assumed a `use-substance` lexical rule, but a more accurate estimation of probabilities might be obtained by considering specialized subclasses. This approach to deriving estimates of the productivity of lexical rules is applied to four denominal verb formation rules in Briscoe and Copestake (1996), where the probabilities of the basic and derived word forms are estimated from part-of-speech tagged textual corpora.

The probabilistic approach we have presented is part of a theory of language use or performance rather than one of competence or grammatical representation. As such it is not a part of the T(D)FS representation language, which is intended as a general formalism in which paradigmatic (lexical) and syntagmatic (syntactic and semantic) theories can be encoded or embedded. This probabilistic approach to lexical rules integrates neatly with extant proposals to control application of lexical rules efficiently within a constraint-based framework, such as those of Meurers & Minnen (1997). To our knowledge it is the first attempt to formalize relatively informal accounts of semi-productivity based on the 'item-familiarity' view of (morphological) lexical rules. As such it serves to highlight a potential difference between genuinely lexical rules and unary syntactic rules, such as Adjunct Introduction, because in the probabilistic framework presented here the latter rule applied during construction of a sentential derivation will not affect the probability of the derivation, whilst lexical rules may since they can output lexical entries with estimated conditional probabilities.

The general claim we make here is that if we assume that speakers choose well-attested high-frequency forms to realize particular senses and listeners choose well-attested high-frequency senses when faced with ambiguity, then much of the 'semi-productivity' of lexical rules is predicted. This improves on the control principle suggested in Copestake (1992), that lexical rules should only be applied if no interpretation was applicable which did not involve a lexical rule, since it allows for cases such as *turkey*, where the derived (meat) use is more frequent than the non-derived (animal) use in the corpora which we have examined. The two other control effects suggested in Copestake (1992) are both also superseded by the current proposal. One of these was to allow for blocking, which is discussed below. The other was that more specific lexical rules should be preferred over more general ones. We would expect that, in general, the more specialized rule will be more productive, as a natural consequence of applying to a smaller class, but the earlier proposal would have had the undesirable consequence that this was a fixed consequence, which could not be adjusted for cases where the generalization did not hold. Thus the grammar writer was, in effect, required to consider both competence and performance when stipulating a rule.

Blocking can be treated as a special case of this principle: if speakers use higher frequency forms to convey a given meaning, an extended meaning will not become conventionalized if a common synonym exists. This means that we do not have to stipulate a separate blocking principle in interpretation, since the blocked senses will not be attested or will have a very low frequency. And in generation, we assume that higher probability forms are preferred as a way of conveying a given meaning. It is necessary to allow for the possibility of *unblocking*, because of examples such as the following:

(11) In the case of at least one county primary school . . . they were offered (with perfect timing) saute potatoes, carrots, runner beans and roast cow.
(Guardian newspaper, May 16th 1990, in a story about mad cow disease.)

However, this is not the complete story, since we have not accounted formally for the extra implicatures that the use of a blocked form conveys, nor have we allowed for the generation of blocked forms (apart from in the circumstances where the generator's lexicon omits the synonym). Both these problems require an account of the interface with pragmatics (see Copestake & Lascarides, 1997 for one such account which integrates probabilistic information into pragmatic reasoning).

The method proposed above for estimating the probability of unattested but possible derived lexical entries for given lexical items is simple. Other more complex schemes could be developed, which, for example, took account of the average probability of the output of a lexical rule. This might be necessary, for example, to model the relative frequencies of *-er* vs *-ee* suffixation, since although the latter is more productive (by Baayen and Lieber's (1991) definition), tokens of the former are more frequent overall (Barker, 1996). However, we have presented a simple approach here, since we currently have no evidence that a more complex approach is justified, given that our main aim is to rank unseen senses by plausibility. Another problem is the need to ensure that classes have comparable frequency distributions. This could matter if there were competing lexical rules, defined on different but overlapping classes, since if one class has a high percentage of low frequency words compared to the other, the estimate of its productivity will tend to be lower. The productivity figure could be adjusted to allow for item frequency within classes. We will not discuss this further here, but see Baayen and Sproat (1996) for discussion of the related phenomenon of ambiguous derivational affixes.

Schütze (1997:133f) argues, in the context of a detailed critique of Pinker (1989), that accounts of lexical rules which do not include a quantitive component cannot form the basis for a satisfactory theory of the acquisition of lexical rules by language learners. The seed for the formation of a specific lexical rule must be comparison of the semantics and alternation / derivation behaviour of a class of lexical items, but since there will always be noise in the form of exceptions because of the inherent semi-productivity of the processes modelled by lexical rules, the induction of a rule must be based in part on quantitive reasoning concerning the degree of generalization obtained from, or equivalently number of exceptions to, a putative lexical rule. The probabilistic approach proposed here could, we think, form the basis for such reasoning (see Schütze (1997) for a detailed discussion of the learning of lexical rules.) Jackendoff (1997a:115f) also notes that the learning of semi-productive lexical rules must be grounded in the *prior* existence of basic and derived lexical entries in the child's lexicon. Jackendoff (1997a:124f) goes on to argue for a full entry theory of lexical organization in which the output of semi-productive lexical rules is entirely specified lexically. He suggests that the advantage of positing semi-productive rules remains because the 'informational-cost' of learning such semi-regular components of the lexicon is reduced. The attraction of the current proposal, integrated with Meurers and Minnen's (1997) partial precompilation approach, is that we can do justice to the facts of semi-productivity and also achieve an efficient and maximally non-redundant encoding of the lexicon.

6 Other Lexical Processes

Lexical rules should be able to account for processes of morphological inflection, derivation and conversion. Verb alternations are a class of morphological conversion rules which exhibit similar semi-productive behaviour to other processes of derivation and conversion. We have discussed dative in detail to demonstrate that a linguistically adequate account of one such rule is possible in the proposed framework. A similar approach to other alternations should be feasible within the framework presented. However the formalism introduced so far provides no mechanism for building up any recursive structure. There is no direct way of copying over information from the input into a different slot in the output structure. The mechanism proposed will therefore allow for at most one step of affixation as shown for the inflectional rule in Figure 7, because inputting the output to another rule would override the function specification. Similarly, the rules given in §4 rely on their semantic effects applying to known structures in the input.

At first sight, this restriction might appear to preclude a treatment of rules such as Passive which have been assumed to require manipulations of a list-valued SUBCAT feature (Pollard & Sag, 1987). However, since Passive is non-recursive, the appropriate effect can be expressed indirectly in our framework, by setting up a feature which is subsequently linked to the 'real' SUBCAT. We use ARGREAL (for argument realization) for this feature in the sketch of the account of Passive that follows.

We assume the following types:

Input:

$$\begin{bmatrix} \textbf{oblique-transfer-verb} \\ \text{PHON}: \textbf{give} \\ \text{SYN ARGREAL}: \begin{bmatrix} \text{LIST}: \begin{bmatrix} \text{HD}: \textbf{npsign} \\ \text{TL}: \begin{bmatrix} \text{HD}: \textbf{npsign} \\ \text{TL}: \begin{bmatrix} \text{HD}: \textbf{pptosign} \\ \text{TL}: \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{LAST}: \boxed{1} \end{bmatrix} \end{bmatrix}$$

Output:

$$\begin{bmatrix} \textbf{unlinked-passive} \\ \text{PHON}: \textbf{give} \\ \text{SYN ARGREAL}: \begin{bmatrix} \text{LIST}: \begin{bmatrix} \text{HD}: \boxed{2}\,\textbf{ppbysign} \\ \text{TL}: \begin{bmatrix} \text{HD}: \textbf{npsign} \\ \text{TL}: \begin{bmatrix} \text{HD}: \textbf{pptosign} \\ \text{TL}: \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{LAST}: \boxed{1} \begin{bmatrix} \text{HD}: \boxed{2} \\ \text{TL}: \textbf{elist} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Linked form:

$$\begin{bmatrix} \textbf{linked-passive} \\ \text{PHON}: \textbf{give} \\ \text{SYN}: \begin{bmatrix} \text{SUBCAT}: \boxed{3} \begin{bmatrix} \text{HD}: \textbf{npsign} \\ \text{TL}: \begin{bmatrix} \text{HD}: \textbf{pptosign} \\ \text{TL}: \boxed{1} \begin{bmatrix} \text{HD}: \boxed{2}\,\textbf{ppbysign} \\ \text{TL}: \textbf{elist} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{ARGREAL}: \begin{bmatrix} \text{LIST}: \begin{bmatrix} \text{HD}: \boxed{2} \\ \text{TL}: \boxed{3} \end{bmatrix} \\ \text{LAST}: \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 22
An example of Passive: **oblique-transfer-verb** is here assumed to be a subtype of **unlinked-active**

$$\begin{bmatrix} \textbf{unlinked-active} \\ \text{SYN}: \begin{bmatrix} \text{ARGREAL}: \textbf{difflist} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{unlinked-passive} \\ \text{SYN}: \begin{bmatrix} \text{ARGREAL}: \begin{bmatrix} \text{LIST}: \begin{bmatrix} \text{HD}: \boxed{1}\,\textbf{ppbysign} \end{bmatrix} \\ \text{LAST}: \begin{bmatrix} \text{HD}: \boxed{1} \\ \text{TL}: \textbf{elist} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{linked-active} \\ \text{SYN}: \begin{bmatrix} \text{ARGREAL}: \begin{bmatrix} \text{LIST}: \boxed{1} \end{bmatrix} \\ \text{SUBCAT}: \boxed{1} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{linked-passive} \\ \text{SYN}: \begin{bmatrix} \text{ARGREAL}: \begin{bmatrix} \text{LIST}: \begin{bmatrix} \text{TL}: \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{SUBCAT}: \boxed{1} \end{bmatrix} \end{bmatrix}$$

The Passive lexical rule simply states:

$$\textbf{unlinked-active} \mapsto \textbf{unlinked-passive}$$

The feature ARGREAL is encoded as a difference list on **unlinked-active**: that is, there are two features, LIST and LAST, such that the value of the LIST feature is a list in the usual HD/TL encoding and LAST is maintained as a pointer to the end of the list. Figure 22 shows an example of rule application to a lexical sign (equivalent to the sign for *give* shown in Figure 15 with a SUBCAT list instead of the categorial encoding).

As above, we assume that linking occurs after lexical rule application, and the types **linked-active** and **linked-passive** provide the appropriate reentrancy statements

$$\begin{bmatrix} \text{PHON} : \begin{bmatrix} \text{STEM} : < \textbf{tie} > \\ \text{PREFIXES} : < \textbf{re, re} > \end{bmatrix} \\ \text{SEM} : < \begin{bmatrix} \textbf{re-rel} \\ \text{EVENT} : \mathbf{e} \\ \text{ARG} : \mathbf{e}' \end{bmatrix}, \begin{bmatrix} \textbf{re-rel} \\ \text{EVENT} : \mathbf{e}' \\ \text{ARG} : \mathbf{e}'' \end{bmatrix}, \begin{bmatrix} \textbf{tie-rel} \\ \text{EVENT} : \mathbf{e}'' \end{bmatrix}, \ldots > \end{bmatrix}$$

Figure 23
reretie

between ARGREAL and SUBCAT. The effect is that the first element of the SUBCAT list in the linked active form will be the last element in the linked passive, which is realized as a PP(by) rather than an NP because of the constraints on the **unlinked-passive** type. An advantage of this approach to Passive is that linking generalizations can be identical for active and passive forms, since they are expressed with respect to the ARGREAL slot rather than SUBCAT. The feature ARGREAL has similarities with the argument structure feature used in more recent versions of HPSG, and we suspect it would be possible to combine the functionality of both features.

The formalism is therefore sufficiently expressive to encode some non-recursive list manipulation operations, if suitable pointers into the list, such as LAST, are set up lexically or on the output of rules which produce lists of sufficiently determinate structure. But this is inadequate for encoding rules which require that the entire semantics of the input (which may itself arise from previous rule applications) be modified by the semantics expressed by the rule. To take a simple example, consider prefixation by *re*, and assume that this is encoded by a lexical rule which can apply an arbitrary number of times, each time appending *re* to a list of prefixes in the PHON value, and adding a re-rel to the semantics, as sketched in Figure 23 which shows a possible representation for *reretie*.[9]

In order to allow rules of this type to be expressed, an extension to the formalism is required to allow the values of list-valued features in the output of the rule to be appended to the input values. Figure 24 illustrates how this could be used to express a lexical rule for *re-* prefixation, with $\oplus$ being used as the notation to indicate that the value of the feature in the input be appended to the right of the structure stated in the output. Figure 25 shows the equivalent rule in the conventional notation, using reentrancy to indicate copying. However, this extension is more restricted in expressivity than allowing arbitrary copying between input and output structures, and still does not make available the arbitrary list-manipulation operations which are possible in conventional HPSG-style lexical rules. This extension relies on the use of a flat representation such as minimal recursion semantics (MRS, Copestake *et al.*, 1995), since the reason we can simply use Append to construct the semantics of the output is that the semantics of any sign is always encoded as a list, without any embedding of structures. Thus we are exploiting the fact that semantic composition in the grammar as a whole relies on the Append operation.

The use of append in conjunction with a flat semantic representation is also adequate to express potentially recursive rules of regular sense extension, such as ‘grinding’ and ‘portioning’, as lexical rules (Copestake & Briscoe, 1992, 1995). In general, we believe that our formulation of lexical rules is expressive enough to capture inflectional, derivational and conversion processes, to model both systematic and idiosyncratic exceptions, and preemption by both synonymy and lexical form.

9 The semantic representation is not intended to be taken too seriously, but the argument applies to any representation where it is assumed that, for example, *reretie* is not equivalent to *retie*.

$$\textbf{verb} \mapsto \begin{bmatrix} \text{PHON} : \begin{bmatrix} \text{PREFIXES} : < \textbf{re} > \oplus \end{bmatrix} \\ \text{SEM} : < \begin{bmatrix} \textbf{re-rel} \\ \text{EVENT} : \mathbf{e} \\ \text{ARG} : \mathbf{e}' \end{bmatrix} > \oplus < \begin{bmatrix} \textbf{rel} \\ \text{EVENT} : \mathbf{e}' \end{bmatrix}, \ldots > \end{bmatrix}$$

Figure 24
Sketch of lexical rule for *re-* prefixation

$$\begin{bmatrix} \textbf{verb} \\ \text{PHON} : \begin{bmatrix} \text{PREFIXES} : \boxed{1} \end{bmatrix} \\ \text{SEM} : \boxed{2} \end{bmatrix} \mapsto \begin{bmatrix} \text{PHON} : \begin{bmatrix} \text{PREFIXES} : \begin{bmatrix} \text{HD} : \textbf{re} \\ \text{TL} : \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{SEM} : \begin{bmatrix} \text{HD} : \begin{bmatrix} \textbf{re-rel} \\ \text{EVENT} : \mathbf{e} \\ \text{ARG} : \mathbf{e}' \end{bmatrix} \\ \text{TL} : \boxed{2} \begin{bmatrix} \text{HD} : \begin{bmatrix} \textbf{rel} \\ \text{EVENT} : \mathbf{e}' \end{bmatrix} \\ \text{TL} : \textbf{list} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 25
Corresponding lexical rule using reentrancy notation for copying

A rule such as Adjunct Introduction (Bouma & van Noord, 1994), which adds adjunct categories to the *subcat* lists of verb entries recursively creating a potentially infinite set of derived entries, seem to us to be a clear example of a non-lexical unary syntactic rule. Firstly, it appears to be fully productive in that it is neither lexically-governed nor subject to idiosyncratic exceptions, blocking or other forms of semi-productivity. Secondly, its function is to add, or better interpolate, adjunct categories to the *subcat* list 'on demand', given the syntactic context, and this is best achieved by the syntactic component during syntactic analysis.

It is possible that an account of Adjunct Introduction could be formulated as a TDFS lexical rule via Append, though it is difficult to see how such a formulation could account naturally for argument-adjunct interpolation of the kind found in examples like: *United flies from New York daily to the Gulf Coast*. However, once we adopt the probabilistic approach to lexical rules, it becomes increasingly unnatural and unmotivated to attempt to interpret such processes as lexical. The item-familiarity theory of lexical productivity clearly should not extend to attempting to model whether a specific verb is more likely to appear with one or two adjuncts, because this is in no sense a definition of a 'possible lexeme'.

Furthermore, there are now clear theoretical advantages for creating a distinct class of unary non-lexical rules. In the TDFS framework an interface between the lexical component and syntactic-semantic component of the grammar is required so that some lexical default specification does not persist into the syntactic component (for example, defaults concerning grammatical agreement – see Lascarides and Copestake, 1999). Thus, such rules must necessarily apply after all genuinely lexical rules, and creating such a separation means that it should be possible to ensure that the two types of rule cannot interact in ways which lead to unrestricted generative capacity in the full system. The two main criteria for distinguishing such rules which we have identified so far are (semi-)productivity and the creation of (un)bounded list structures in the syntactic representation.

7 Conclusions

Both Goldberg (1995) and Jackendoff (1997a,b) contrast the lexical rule approach to bounded dependencies with one which treats each construction independently and characterizes relations between constructions, somewhat vaguely, in terms of 'inheritance'. Jackendoff (1997b:556f) makes the point that lexical rules in lexicalist frameworks are expressive enough to describe bounded constructions and any idiosyncratic meanings they convey, so they can be used to capture relations amongst such constructions. He argues against this approach though, because he suggests that the elements of the constructions related by such rules are often not lexical. Copestake and Briscoe (1995) make the same point with respect to examples of systematic metonymy, where such semi-productive 'lexical' rules apply to noun phrase constructions. Jackendoff also argues, however, that constructions need to be treated as a kind of phrasal lexical item whose (idiosyncratic) meaning is learnt like that of a lexical item (1997b:554). For us, the defining characteristic of a lexical rule is that it requires some listing of properties to accurately express its behaviour whether this be because it is lexically-governed, has exceptions, is underspecified in its effects or whatever. Thus idioms must be lexically specified, though they are best treated as particularly idiosyncratic phrases / constructions, rather than lexical items.

For Jackendoff (1997a:115f) the crucial distinction is not lexical / non-lexical but productive / semi-productive rule. Jackendoff's definition of a productive rules encompasses a rule such as plural noun formation or third singular verb formation (see §1 above), despite the existence of irregular derived forms, because he argues that such a rule's output need not be listed. It is semi-productive only to the extent that certain aspects of its output can be overridden or blocked by lexical specification of *exceptions*. He reserves the term semi-productive for rules, such as denominal verb formation (*shelf* $\mapsto$ *shelve*), where the exact output of the rule is underspecified and the existence of the derived words is not guaranteed. Thus, the precise meaning of the denominal verb is partly systematic ('to put x in/on y') and partly idiosyncratic and unpredictable (e.g. *to saddle (a horse)* means 'to put a saddle on a horse's back', whilst *to shelve a book* means 'to put a book on a shelf'), and the phonological form is not always identical with or entirely predictable from the nominal form (e.g. *shelve*). Furthermore, there are many nouns which do not have corresponding denominal forms (*mustard* vs. *butter*, *teapot* vs. *knife*, etc.). Jackendoff argues that the nature of the exceptions to the latter type of rule requires (full) listing in the lexicon of the derived forms, whilst the former does not.

In our approach, lexical rules are those which require some element of lexical/listed specification, whether it be the listing of irregular forms which override aspects of the rule output or of idiosyncratic aspects of the resulting meaning, or the unattested status of the derived entry. The approach to lexical rules we have advocated, integrating a restrictive default-based formalization with partial precompilation and a probabilistic account of 'item-familiarity' and semi-productivity, is capable of expressing inflectional, derivational and conversion rules whose domain is (within) that of a bounded dependency construction (i.e. includes 'alternation' rules relating bounded constructions). This approach reintegrates construction-based generalizations with more traditional lexical rules, provides a very general means for encoding semi-productivity, and makes a principled distinction between lexical and unary syntactic rules which should allow the generative power of the overall grammar to be restricted. To summarize: lexical rules cannot perform arbitrary operations on unbounded lists; they are unidirectional, but have limited reversibility properties appropriate to account for backformation; they involve no extension to the underlying logic of the TDFS framework; they require statement of what changes, not what stays the same; they are subject to type constraints and can exploit the default inheritance hierarchy to capture generalizations; they can be semi-productive and are predicted to be sensitive to blocking, exceptions, conventionalization, and so forth; and they allow a linguistically elegant, and accurate account of the dative construction/alternation, subsuming the insights emerging from recent detailed analyses of this specific lexical rule.

Furthermore, we outlined ways in which this approach can be extended straightforwardly to deal with rules apparently involving more complex *subcat* list manipulations, and with recursive processes of derivation and conversion and their associated semantics, without sacrificing these desirable properties. It follows from our approach that some putative lexical rules should be treated as unary syntactic rules. A potential advantage of this division of labour is that it may even be possible to develop a separate treatment of unary syntactic rules which does not utilize category-valued variables over list-valued features. In any case, if these rules only apply to the output of the lexicon, this will avoid the increase in generative capacity identified by Carpenter (1991), resulting from the interaction of recursion, arbitrary list operations and unbounded lists, by keeping list-valued features bounded during lexical rule application, and only allowing unbounded additions to or limited modification of such features during syntactic processing.

Acknowledgments

We would like to thank Tony Kroch, Mark Liberman, Geoff Nunberg, Mark Steedman and, especially, Annie Zaenen for helpful input and advice. The content and structure of the paper is, we hope, much improved on the basis of 3 anonymous referees' insightful comments on an

earlier draft. All the ideas and mistakes, nevertheless, remain our responsibility.

References

Ait-Kaci, Hassan. (1984). A lattice-theoretic approach to computation based on a calculus of partially ordered type structures. University of Pennsylvania, Doctoral Dissertation.

Aronoff, Mark. (1976). *Word Formation in Generative Grammar*. Linguistic Inquiry Monograph 1. MIT Press, Cambridge, Mass.

Baayen, Harald and Rochelle Lieber. (1991). Productivity and English derivation: a corpus-based study. *Linguistics* 29: 801–843.

Baayen, Harald and Richard Sproat. (1996). Estimating lexical priors for low-frequency morphologically ambiguous forms. *Computational Linguistics* 22.2: 155–166.

Barker, Chris. (1995). "Episodic *-ee* in English: thematic relations and new word formation." In *Semantics and Linguistic Theory V,* edited by Mandy Simons and Teresa Galloway. Cornell University, Ithaca, NY, 1–18.

Bauer, Laurie. (1983). *English Word-Formation*. Cambridge University Press, Cambridge, England.

Bird, Stephen and Ewan Klein. (1994). Phonological analysis in typed feature systems. *Computational Linguistics* 20.3: 455–491.

Boguraev, Bran and Ted Briscoe. (1989). "Utilizing the LDOCE grammar codes." In *Computational Lexicography for Natural Language Processing,* edited by Bran Boguraev and Ted Briscoe. Longman, London, 85–116.

Bouma, Gosse. (1992). Feature structures and nonmonotonicity. *Computational Linguistics* 18.2: 183–204.

Bouma, Gosse and Gertjan van Noord. (1994). Constraint-based categorial grammar. *Proceedings of the 32nd Annual Meeting of the Association of Computational Linguistics*, Las Cruces, NM: 147–154.

Box, George E.P and George C. Tiao. (1973). *Bayesian Inference in Statistical Analysis*. Addison-Wesley, Reading, MA.

Brew, Chris. (1995). Stochastic HPSG. *Proceedings of the 7th European Conference of the Association of Computational Linguistics*, Dublin, Ireland.

Briscoe, Ted and Ann Copestake. (1996). Controlling the application of lexical rules. *Proceedings of the ACL SIGLEX Workshop on Breadth and Depth of Semantic Lexicons*, Santa Cruz: 7–19.

Briscoe, Ted, Ann Copestake and Bran Boguraev. (1990). Enjoy the paper: lexical semantics via lexicology. *Proceedings of the 13th Int. Conf. on Comp. Ling.*, Helsinki: 42–47.

Calcagno, Michael. (1995). Interpreting lexical rules. *Proceedings of the Conference on Formal Grammar*, Barcelona.

Carpenter, Bob. (1991). The generative power of categorial grammars and head-driven phrase structure grammars with lexical rules. *Computational Linguistics* 17.3: 301–314.

Carpenter, Bob. (1992). *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge, England.

Carpenter, Bob. (1993). "Skeptical and credulous default unification with application to templates and inheritance." In *Inheritance, Defaults and the Lexicon,* edited by Ted Briscoe, Ann Copestake and Valeria de Paiva. Cambridge University Press, Cambridge, England, 13–37.

Chang, Nancy. (1995). A constraint-based approach to linking. Cambridge University, MPhil Dissertation.

Church, Ken and Robert Mercer. (1993). Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics* 19.1: 1–24.

Church, Ken and William Gale. (1994). Enhanced good-turing and cat-cal estimation. *Speech and Language Processing* 10.

Copestake, Ann. (1992). The representation of lexical semantic information. Doctoral dissertation, University of Sussex, Cognitive Science Research Paper CSRP 280.

Copestake, Ann. (1993). "Defaults in lexical representation." In *Inheritance, Defaults and the Lexicon,* edited by Ted Briscoe, Ann Copestake and Valeria de Paiva. Cambridge University Press, Cambridge, England, 223–245.

Copestake, Ann and Ted Briscoe. (1992). "Lexical operations in a unification based framework." In *Lexical Semantics and Knowledge Representation. Proceedings of the first SIGLEX Workshop, Berkeley, CA,* edited by James Pustejovsky and Sabine Bergler. Springer-Verlag, Berlin, 101–119.

Copestake, Ann and Ted Briscoe. (1995). Semi-productive polysemy and sense extension. *Journal of Semantics* 12: 15–67.

Copestake, Ann, Dan Flickinger, Robert Malouf, Suzanne Riehemann, Ivan Sag. (1995). Translation using minimal recursion semantics. *Proceedings of the The Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI95)*, Leuven, Belgium.

Copestake, Ann and Alex Lascarides. (1997). Integrating symbolic and statistical representations: the lexicon-pragmatics interface. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL 97)*, Madrid: 136–143.

Daelemans, Walter, Konrad de Smedt and Gerald Gazdar. (1992). Inheritance in natural language processing. *Computational Linguistics* 18.2: 205–218.

Davis, Antony. (1996). Lexical semantics and linking in the hierarchical lexicon. Stanford University, Doctoral Dissertation.

Dowty, David. (1989). "On the semantic content of the notion 'thematic role'." In *Property Theory, Type Theory and Natural Language Semantics,* edited by Gennaro Chierchia, Barbara Partee and Ray Turner. Reidel, Dordrecht, The Netherlands, 69–129.

Evans, Roger and Gerald Gazdar. (1989). Inference in DATR. *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics (EACL-1989)*, Manchester, England: 66–71.

Evans, Roger and Gerald Gazdar. (1996). DATR: a language for lexical knowledge representation. *Computational Linguistics* 22.2: 167–216.

Fillmore, Charles, Paul Kay and Mary O'Connor. (1988). Regularity and idiomaticity in grammatical constructions. *Language* 64: 501–538.

Goldberg, Adele. (1995). *Constructions*. Chicago University Press, Chicago, IL.

Green, Georgia. (1974). *Semantics and Syntactic Regularity*. Indiana University Press.

Jackendoff, Ray. (1975). Morphological and semantic regularities in the lexicon. *Language* 51(3): 639–71.

Jackendoff, Ray. (1997a). *The Architecture of the Language Faculty*. MIT Press, Cambridge, MA.

Jackendoff, Ray. (1997b). Twistin' the night away. *Language* 73(3): 534–559.

Johnson, Mark and Jochen Dorre. (1995). Memoization of coroutined constraints. *Proceedings of the 33rd Annual Meeting of the Association of Computational Linguistics*, Cambridge, MA: 100–107.

Joshi, Aravind. (1987). "An introduction to tree-adjoining grammars." In *Mathematics of Language,* edited by Alexis Manaster-Ramer. John Benjamins, Amsterdam, 87–115.

Kelly, Martin and Steven Martin. (1994). Domain-general abilities applied to domain-specific tasks: sensitivity to probabilities in perception, cognition and language. *Lingua* 92: 105-140.

King, Paul. (1994). "An expanded logical formalism for head-driven phrase structure grammar." Arbeitspapiere des SFB 340/59, University of Tübingen.

Labov, William. (1972). *Sociolinguistic Patterns*. University of Pennsylvania Press, Philadelphia.

Lascarides, Alex and Nicholas Asher. (1991). Discourse relations and defeasible knowledge. *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL-91)*, Berkeley, California: 55–63.

Lascarides, Alex and Nicholas Asher. (1993). Temporal interpretation, discourse relations and common sense entailment. *Linguistics and Philosophy* 16: 437–493.

Lascarides, Alex, Ted Briscoe, Nicholas Asher and Ann Copestake. (1995). Order independent and persistent typed default unification. *Linguistics and Philosophy* 19.1: 1–89.

Lascarides, Alex and Ann Copestake. (1995). "The pragmatics of word meaning." In *Semantics and Linguistic Theory V,* edited by Mandy Simons and Teresa Galloway. Cornell University, Ithaca, NY, 204–221.

Lascarides, Alex and Ann Copestake. (1999). Default representation in constraint-based frameworks. *Computational Linguistics* 25.1: 55–106.

Lascarides, Alex, Ann Copestake and Ted Briscoe. (1996). Ambiguity and coherence. *Journal of Semantics* 13: 41–65.

Levin, Beth. (1992). *Towards a Lexical Organization of English Verbs*. University of Chicago Press, Chicago, IL.

Levin, Beth and Malka Rappaport Hovav. (1995). *Unaccusativity at the Syntax-Lexical Interface*. MIT Press, Cambridge, MA.

Malouf, Robert. (1999). "Practical default inheritance in constraint-based grammars." Paper presented at Ohio State University.

Meurers, Detmar. (1995). "Towards a semantics for lexical rules as used in HPSG." Acquilex Workshop on Lexical Rules, Cambridge.

Meurers, Detmar and Guido Minnen. (1997). A computational treatment of HPSG lexical rules as covariation in lexical entries. *Computational Linguistics* 23.4: 543–596.

Nicholls, Diane. (1995). "Can fully productive lexical rules be defined and can they apply cross-linguistically?." Acquilex-II Working Paper 79: `http://www.cl.cam.ac.uk/Research/NL/acquilex/`.

Pinker, Steven. (1989). *Learnability and Cognition: The Acquisition of Argument Structure*. MIT Press, Cambridge, MA.

Pirelli, Vito, Nilda Ruimy and Simonetta Montemagni. (1994). "Lexical regularities and lexicon compilation." Acquilex-II Working Paper 36: `http://www.cl.cam.ac.uk/Research/NL/acquilex/`.

Pollard, Carl and Ivan Sag. (1987). *An information-based approach to syntax and semantics: Volume 1 fundamentals*. CSLI Lecture Notes 13, CSLI Publications, Stanford CA.

Pollard, Carl and Ivan Sag. (1994). *Head-driven phrase structure grammar*. Chicago University Press, Chicago.

Resnik, Philip. (1992). Probabilistic Lexicalized Tree Adjoining Grammar. *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France.

Riehemann, Suzanne. (1993). Word formation in lexical type hierarchies. M.Phil thesis, University of Tübingen, Germany.

Russell, Graham, Afzal Ballim, John Carroll and Susan Warwick-Armstrong. (1993). "A practical approach to multiple default inheritance for unification-based lexicons." In *Inheritance, Defaults and the Lexicon,* edited by Ted Briscoe, Ann Copestake and Valeria de Paiva. Cambridge University Press, Cambridge, England, 137–147.

Sanfilippo, Antonio. (1990). Grammatical Relations, Thematic Roles and Verb Semantics. PhD thesis. Centre for Cognitive Science, University of Edinburgh.

Sanfilippo, Antonio. (1992). "Verbal diathesis, knowledge acquisition, lexicon construction and dictionary compilation." Acquilex-II Working Papers 1: `http://www.cl.cam.ac.uk/Research/NL/acquilex/`.

Sanfilippo, Antonio. (1993). "LKB encoding of lexical knowledge from machine-readable dictionaries." In *Inheritance, defaults and the lexicon,* edited by Ted Briscoe, Ann Copestake and Valeria de Paiva. Cambridge University Press, Cambridge, England, 190–222.

Schabes, Yves. (1992). Stochastic lexicalized tree adjoining grammar. *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France.

Schütze, Hinrich. (1997). *Ambiguity resolution in language learning: computational and cognitive models*. CSLI Lecture Notes 71, CSLI Publications, Stanford, CA.

Shieber, Stuart. (1992). *Constraint-based grammar formalisms*. MIT Press, Cambridge, MA.

Steedman, Mark. (1987). Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory* 5: 403–439.

Wasow, Tom. (1980). "Major and minor rules in lexical grammar." In *Lexical Grammar,* edited by Tuen Hoekstra, Harry van der Hulst and Michael Moortgat. Foris, Dordrecht, 285–312.

Wechsler, Steven. (1991). Argument structure and linking. Stanford University, Doctoral Dissertation.

Young, Mark and Bill Rounds. (1993). A logical semantics for nonmonotonic sorts. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL-93)*, Columbus, Ohio: 209–215.

Zeevat, Henk, Ewan Klein and Jo Calder. (1987). "An introduction to unification categorial grammar." In *Categorial Grammar, Unification Grammar and Parsing: Working Papers in Cognitive Science 1,* edited by Nicholas Haddock, Ewan Klein and Glyn Morrill. Centre for Cognitive Science, University of Edinburgh, 195–222.