GOSSE BOUMA, FRANK VAN EYNDE AND DAN FLICKINGER

CONSTRAINT-BASED LEXICA

1. INTRODUCTION

As the field of generative linguistics has developed, the lexicon has taken on an increasingly important role in the description of both idiosyncratic and regular properties of language. Always viewed as a natural home for exceptions, the lexicon was given relatively little work in the early years of transformational grammar. Then Chomsky proposed in 1970 (Chomsky, 1970) that similarities in the structure of deverbal noun phrases and sentences could be expressed in terms of a lexical relationship between the verb and its nominalization. Jackendoff (1975) characterized further lexical regularities in both morphology and semantics, and Bresnan (1976, 1982) pioneered the development of a syntactic framework (Lexical Functional Grammar) in which central grammatical phenomena such as passivization could be explained within the lexicon. A parallel line of work by Gazdar (1981) called Generalized Phrase Structure Grammar sought to provide a nontransformational syntactic framework, by employing metarules over a context-free grammar. Gazdar et al. (1985) constrained the power of those metarules by restricting them to lexically-headed phrase structure rules. Pollard and Sag (1987, 1994) built on the work in GPSG, outlining the more radically lexicalist framework of Head-driven Phrase Structure Grammar (HPSG), abandoning construction-specific phrase structure rules in favor of a small number of rule schemata interacting with a more richly articulated lexicon to capture relevant syntactic generalizations. As a well-known and widely used constraint-based grammar formalism[1] HPSG will serve us well in this chapter by providing a precise linguistic framework within which we can organize the relevant data and examine the technical devices available for analysis of that data. For those who are not familiar with the notation, we first provide a brief introduction to this framework.

[1] Much of the current research in HPSG is being carried out in connection with computer-based implementations that provide valuable testbeds for the formal analyses being developed. A good collection of these implementations can be found at the following URL: http://ling.ohio-state.edu/HPSG/Implementation.html

1.1. *The HPSG framework*

The fundamental concept for linguistic representation in HPSG is the *sign*, generalizing de Saussure's idea of the minimal independent unit relating form and meaning in language. A sign in HPSG is a collection of multiple kinds of properties or information, including phonological, syntactic, semantic, and contextual constraints, represented as a typed attribute-value matrix (AVM) where each attribute's value is also assigned a type, potentially with additional constraints. Since such constraints on a type can introduce allowable or appropriate features for that type, they are called *appropriateness conditions*, or more generally *feature declarations*. Words are represented as signs, and phrases are also signs, with many but not all of the same attributes or features shared between words and phrases. Both words and phrases have at least the features PHON for phonology and SYNSEM for syntactic/semantic constraints. PHON has as its value a list of phonology descriptions, which we will in this chapter replace with the less precise but more convenient orthographic equivalents. SYNSEM has as its value another AVM structure of type *synsem*, with appropriate attributes LOC and SLASH, to distinguish constraints on local sisters in a phrase from constraints on unbounded dependency constructions. LOC again takes an AVM as value, with appropriate features CAT for syntactic (category) constraints and CONT for semantic content. CAT attributes include HEAD, identifying those properties of a sign which are shared with the phrase it heads; VALENCE for subcategorization constraints; and ARG-ST to represent the proper domain for several general constraints on argument structure. Finally, HEAD features are appropriately distinguished for the various types of lexical signs (e.g. VFORM for verbs or CASE for nouns), and the VALENCE attributes include SUBJ for subject, SPR for specifier, and COMPS for complements. This architecture for the sign is illustrated in Figure 1, showing the lexical entry for the base form of the verb *like*. The other two HEAD features AUX and INV illustrated here will be introduced and explained in section 5.

In this notation each attribute or feature, such as SUBJ, is shown in small caps, with the value to its right. List values are represented in angle brackets,[2] set values in curly brackets, and atomic values are given in italics. Notice here the use of boxed numbers in front of the values of certain attributes, to indicate that the structure for a given value is shared in two (or more) attributes. For example, the semantic index introduced by the subject noun phrase of *like* is coreferenced with the ARG1 attribute in the *like* relation in the CONT of the sign, expressing the linking of the syntactic subject's index with the appropriate thematic role in the semantics.

[2] The empty list is represented as $< \;\; >$, and a list with one member as $<[\,]>$.

$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}} \\
\text{PHON} & \left\langle \textit{like} \right\rangle \\
\text{SYNSEM} & \left[\begin{array}{ll}
\text{LOC} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{verb}} \\ \text{VFORM} & \textit{base} \\ \text{INV} & - \\ \text{AUX} & - \end{array}\right] \\
\text{VAL} & \left[\begin{array}{ll}
\text{SUBJ} & \left\langle \boxed{1} \left[\begin{array}{ll} \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \textit{noun} \\ \text{VAL} & \left[\begin{array}{ll} \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{array}\right] \end{array}\right] \\ \text{CONT} & \left[\text{INDEX } \boxed{3}\right] \end{array}\right] \end{array}\right] \right\rangle \\
\text{SPR} & \langle\rangle \\
\text{COMPS} & \left\langle \boxed{2} \left[\begin{array}{ll} \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{l} \textit{noun} \\ \text{CASE } \textit{acc} \end{array}\right] \\ \text{VAL} & \left[\begin{array}{ll} \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{array}\right] \end{array}\right] \\ \text{CONT} & \left[\text{INDEX } \boxed{4}\right] \end{array}\right] \end{array}\right] \right\rangle
\end{array}\right] \\
\text{ARG-ST} & \left\langle \boxed{1}, \boxed{2} \right\rangle
\end{array}\right] \\
\text{CONT} & \left[\begin{array}{ll} \text{RESTR} & \left[\begin{array}{ll} \text{RELN} & \textit{like} \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \boxed{4} \end{array}\right] \end{array}\right]
\end{array}\right] \\
\text{SLASH} & \langle\rangle
\end{array}\right]
\end{array}\right]
$$

Figure 1. HPSG sign for *like*

As noted above, the constraints in such a lexical entry interact closely with those of the phrase structure schemata, exemplified by the Head-Subject rule schema shown in Figure 2, which combines a head such as a verb phrase with its subject.

Like lexical signs, phrasal signs contain a SYNSEM attribute which describes the mother node of the phrase; in addition, phrases introduce attributes

$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{5} \oplus \boxed{6} \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \boxed{2} \end{bmatrix} \\ \text{SLASH} & \boxed{3} \end{bmatrix} \\ \text{NON-HD-DTR} & \begin{bmatrix} \text{PHON} & \boxed{5} \\ \text{SYNSEM} & \boxed{4} \end{bmatrix} \\ \text{HD-DTR} & \begin{bmatrix} \text{PHON} & \boxed{6} \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \left\langle \boxed{4} \right\rangle \\ \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \boxed{2} \end{bmatrix} \\ \text{SLASH} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 2. HPSG Head-Subject rule schema

for the daughter nodes in the phrase, each of which is a sign (either word or phrase). In this example, the rule introduces the NON-HD-DTR and HD-DTR attributes for the subject daughter and the head daughter, respectively. What the rule requires is that the SYNSEM of the subject daughter be identified with the head daughter's SUBJECT's value; it also incorporates the Head-Feature Principle, and requires that the head daughter have picked up all of its complements before combining with its subject. As with lexical signs, phrases provide a specification of their syntactic and semantic properties in the SYNSEM attribute, using reentrancies (identities in attribute values) to relate properties of the phrase to those of its immediate daughters. The PHON feature of the phrase illustrates a further use of reentrancies, here encoding the constraint that the phonology of the phrase is the result of appending the phonology of the non-head daughter with that of the head daughter.

While this rich architecture enables us to express precise, detailed properties of signs, and to express generalizations about these properties within phrases, the notation can be unnecessarily cumbersome. So in the remainder of this chapter much more abbreviated feature structures (AVMs) will be used, selecting only certain attributes of interest, and often suppressing outer attributes where the intention is clear. For a more complete presentation of the notation and the underlying HPSG framework, see Pollard and Sag (1994).

1.2. *Capturing lexical regularities*

A central challenge in a lexicalist framework like HPSG is to develop strategies for eliminating the massive redundancy present in fully specified lexical entries. Some aspects of this redundancy have always been recognized as properly lexical, such as the inflected forms of verbs and nouns, as well as words derived from other words by affixation. But other sources of redundancy in the lexicon, while motivated linguistically, are more specific to work in the HPSG framework, including modern treatments of unbounded dependencies (extraction phenomena) and argument attraction in verb clusters.

In addition to the explanatory benefits to the theory of grammar afforded by the elimination of redundancy in the lexicon, there are also significant practical benefits for the developer of an implementation of the grammar. Hand-coding of fully-specified lexical entries would be a difficult and time-consuming task, one prone to human error. Moreover, maintenance of such a redundant, detailed lexicon would prove to be difficult if not impossible, often requiring consistent adjustments to dozens or hundreds of entries.

To capture the lexical regularities found in inflection, derivation, and valence alternation, linguists have proposed a number of technical devices within constraint-based frameworks. The simplest of these is known as *underspecification*, where an attribute is given as its value a less specific type T, to indicate that any of the (more specific) subtypes of T can unify with that attribute's value. For example, a verb like *say*, which subcategorizes for a sentential complement, underspecifies that complement's attribute AUX to be *boolean*, thus accepting clauses headed either by auxiliary verbs ([AUX *plus*]) or by main verbs ([AUX *minus*]).

In addition to underspecification, standard HPSG offers two primary means of expressing the sharing of properties or constraints in the lexicon: one is the use of types arranged in inheritance hierarchies, presented in section 2; and the other is the specification of lexical rules which relate pairs of lexical entries, presented in section 3. In more recent work, a third means of capturing lexical regularities has emerged, employing relational constraints on types; these are presented in section 4. Having introduced the various methods for capturing lexical regularities, we then apply them in section 5 to the analysis of English subject-auxiliary inversion.

2. Inheritance and the Hierarchical Lexicon

Inheritance is a knowledge representation technique familiar from work in artificial intelligence as well as (object-oriented) software engineering. Inheritance is particularly useful for storing concisely large amounts of knowledge related to a given set of objects. The objects are classified according to a hierarchically organized collection of *classes*. Each class comes with a number of properties. If class C_2 is a subclass of C_1, all properties of C_1 are inherited (by default) by C_2. Multiple inheritance allows a class to be a subclass of more than one superclass, in which case the class inherits the (non-conflicting) properties of all its super-classes. The power of inheritance lies in the fact that properties which hold for a class of objects need only be declared once. Apart from leading to more concise representations, inheritance, and the object-oriented approach that comes with it, also helps to ensure consistency and facilitates maintenance of large knowledge bases.

The use of inheritance as a technique to organize the lexicon has been a dominant trend in constraint-based grammar formalisms. The early work within HPSG on inheritance can be found in Flickinger et al. (1985), Pollard and Sag (1987; chpt. 8), and Flickinger (1987). Here, the lexicon is presented as a hierarchically organized collection of *frames* (or *classes*). In such a set-up, the lexical entry for a specific lexical item need only specify the idiosyncratic (unpredictable) properties of the item and the frame (or set of frames) from which it inherits.

Consider, for instance, the AVM for the base form of the verb *like* presented in Figure 1. This AVM contains several pieces of information that are common to verbs, to transitive verbs, to base forms, etc. Using frames, we may therefore decompose this entry as shown in Figure 3 (leaving out redundant path-prefixes, and abbreviating the AVMs for NPs).

The base verb *like* inherits from the frames for *main*, *base* and *trans*, which all inherit from the more general frame *verb-wrd*, which itself is a specialization of *word*. The HEAD feature of a *verb-wrd* is required to be of type *verb*. For the type *verb* the features VFORM, INV, and AUX are appropriate. Furthermore, the constraints on *verb-wrd* restrict the value of the valence features SUBJ, SPR, and COMPS, in terms of ARG-ST. The frames *main*, *base*, and *trans* define *like* as a main verb,[3] a base form, and a transitive verb, respectively. The only information idiosyncratic to *like* is its phonology and its content. This seems reasonable, as the relationship between phonology

[3] We assume here that main verbs, as opposed to auxiliaries, always have one element on their SUBJ-list. See section 5 for details.

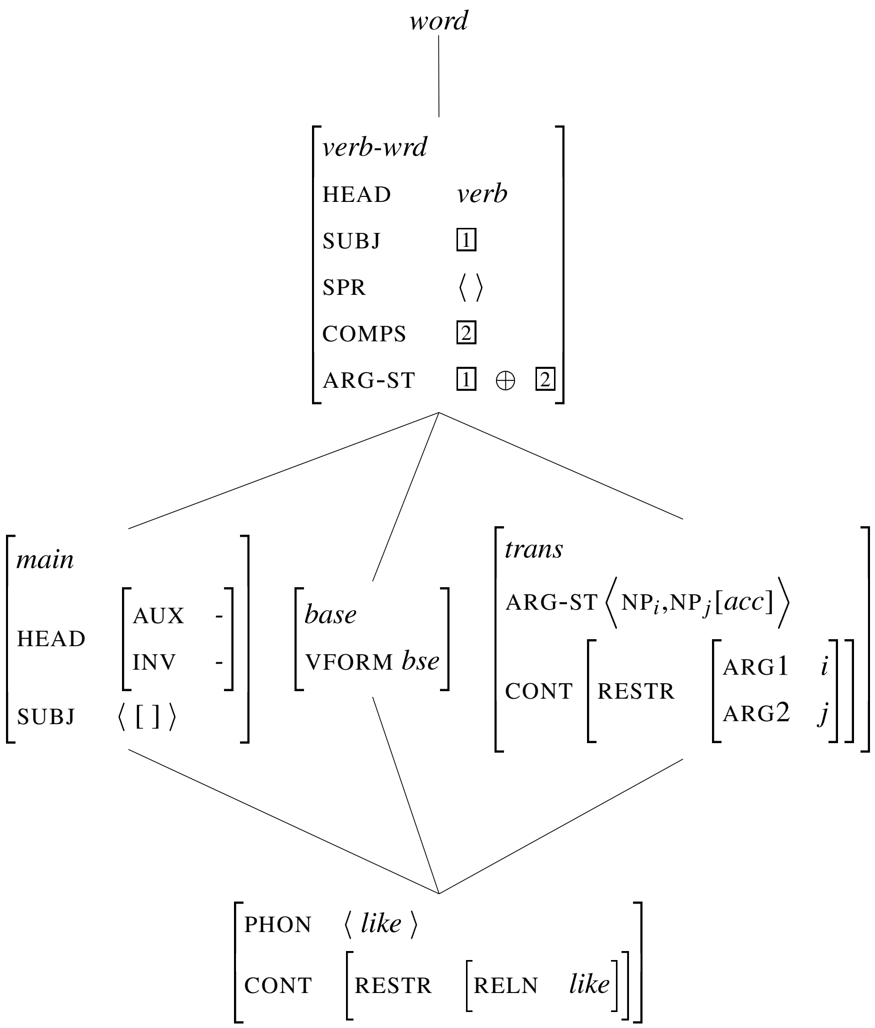


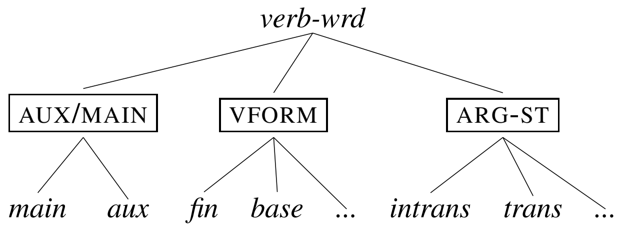
Figure 3. Fragment of the verbal inheritance hierarchy

and semantics is arbitrary, and cannot be predicted from general properties of the language.

A more complete picture of the lexical hierarchy would typically contain several other frames, such as *noun-wrd*, *determiner-wrd*, *adjective-wrd*, etc., which also inherit directly from *word*. Furthermore, next to *main*, there would be a frame *aux*, several frames for other verb forms and a (large) number of alternative frames for the various subcategorization types.

There are certain regularities within the hierarchy which are not made explicit in Figure 3. A specific lexical entry will always be either a main verb or an auxiliary, will belong to exactly one subcategorization type, and will have a specific morphological form. To make the structure of the hierarchy more explicit, Pollard and Sag (1987) introduce *partitions*. The idea is that whereas an entry may inherit from more than one frame, it will always inherit from exactly one frame in a given partition. In the inheritance hierarchy below, partitions appear in boxes:

(1)



The frame for *verb-wrd* has a number of subframes, which can be grouped in three partitions, AUX/MAIN, VFORM and ARG-ST. A given verb form will normally inherit from a member of each of these partitions, but not from both *main* and *aux*, for instance.

Lexical inheritance is naturally connected to the use of types. The type system in HPSG is defined hierarchically in terms of subsumption: if T' is a subtype of T, all features declared appropriate for T are appropriate for T' as well, and furthermore if the value of a feature f is required to be of type S on T, the value of f on T' can only be required to be of type S', where S' is a subtype of S. Since a subtype T' of T must satisfy all the constraints T satisfies, it is said that T subsumes T'. For example, given that *word* and *phrase* are both subtypes of *sign*, all features appropriate for *sign* are appropriate for *word* and *phrase* as well.

The lexical hierarchy can be thought of similarly: if *verb-wrd* inherits from *word*, all constraints on *word* must hold for *verb-wrd* as well. Therefore, one can try to define the lexical hierarchy as a hierarchy of types. For instance, we could introduce a type *verb-wrd*, with *main, aux, fin, base, intrans, trans, etc.* as subtypes. Each of these subtypes is subsumed by *verb-wrd* and introduces additional constraints. If two types are elements of the same *partition*, i.e. *aux* and *main* are both elements of the AUX/MAIN partition, this means that they are incompatible (i.e. not unifiable). If two types belong to different partitions, they will in general be unifiable, and give rise to a uniquely typed most general unifier. *Aux* and *fin*, for instance, belong to different partitions, and thus their unification must give rise to something of type *aux-fin*. A procedure for translating the lexical hierarchy of Pollard and Sag (1987) into a type hierarchy is given in Carpenter (1992b). As lexical types frequently introduce constraints which refer to deeply embedded parts of an AVM, it is convenient to assume a formal framework in which types may be the antecedent of implicational constraints. For instance, the information expressed by the frame *main* in Figure 3 may be expressed as an implicational constraint for the type *main*.

$$\text{(2)}\ \ main \rightarrow \begin{bmatrix} \text{SYNSEM} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{INV} & - \\ \text{AUX} & - \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle [\ ] \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

This implication states that every AVM of type *main* also has to satisfy the constraint on the right-hand side of the arrow.[4]

One of the reasons why inheritance is such a powerful tool is the fact that it can be extended with a notion of nonmonotonicity. Nonmonotonic inheritance means that if class C_2 with property P_2 inherits from class C_1 with property P_1, where P_1 and P_2 are incompatible, inheritance of the conflicting information from P_1 is blocked. Thus, the property P_1 is only a default for classes inheriting from C_1, as it may be suppressed on subclasses by providing a conflicting property. The classical example involves birds and penguins. Birds have the property that they can fly. Penguins are birds, but cannot fly. The *cannot-fly* property suppresses the *fly* property in this case, even though **penguin** is a subclass of **bird**.

Nonmonotonicity appears to be a natural requirement for constraint-based, hierarchical lexica. One of the typical properties of generalizations about lexical items is that they have exceptions. This is true in particular of derivational and inflectional morphology, but may hold for other dimensions of grammatical information as well.[5] For instance, English verbs in general take an ordinary NP as subject. If one includes such a property in the definition of **verb**, however, 'weather'-verbs such as *rain* must be treated as exceptions as they take only expletive NP's as subject. Also, *subject raising* verbs are exceptional, as the question whether they select an ordinary NP or an expletive as subject depends on the restrictions imposed by the verb heading their VP-complement. Proposals for nonmonotonic extensions of constraint-based grammars can be found in Bouma (1992), Carpenter (1992c), and Lascarides et al. (1996). Most implemented systems restrict themselves to monotonic

[4] See Götz and Meurers (1997) for discussion.

[5] See Thomason (1997) for a recent overview.

inheritance, sometimes extended with more or less ad-hoc extensions to cope with some aspects of nonmonotonicity.

3. LEXICAL RULES

Lexical rules are often considered to be the principal mechanism for capturing generalizations within the lexicon. In HPSG, as in almost any other grammatical framework, lexical rules have been proposed to account for inflection and derivation (section 3.1), and valence alternations of various kinds (section 3.2). Some of the more problematic aspects of using lexical rules in a constraint-based setting are discussed in section 3.3.

3.1. *Inflection and Derivation*

Pollard and Sag (1987) propose an account of inflection based on lexical rules. Their rule for third person singular inflection in English is as follows.[6]

(3)
$$\begin{bmatrix} \textit{base} & \\ \text{PHON} & \boxed{1} \\ \text{3RDSNG} & \boxed{2} \\ \text{SYNSEM} & \begin{bmatrix} \text{VAL} & \boxed{3} \\ \text{CONT} & \boxed{4} \end{bmatrix} \end{bmatrix} \Rightarrow \begin{bmatrix} \textit{3rdsng} & \\ \text{PHON} & f_{3rdsng}(\boxed{1},\boxed{2}) \\ \text{SYNSEM} & \begin{bmatrix} \text{VAL} & \boxed{3} \\ \text{CONT} & \boxed{4} \end{bmatrix} \end{bmatrix}$$

The function f_{3rdsng} produces the third person singular inflected phonological form of a given base form. It takes this base form as argument, as well as the value of the feature 3RDSNG. The latter is used to account for exceptional forms. For regular verbs, the value of this feature would be unspecified, in which case the inflected form is the base form + *-s*. If a value for 3RDSNG is provided (such as *has* for the base form *have*), this value is returned as result by f_{3rdsng}. The valence properties, as well as the semantics, are shared between input and output. The fact that the output must be specified as [VFORM *fin*] and selects for a third person singular subject is accounted for by assigning the output the type *3rdsng*. The constraints that come with this type are assumed to provide the relevant information.

[6] In this example, as well as in the examples of lexical rules given below, we assume the architecture of (lexical) signs given in section 1.1. We believe that the differences between our version of these rules and the original versions is of little or no importance for the issue at hand.

Regular instances of derivational morphology can be captured with lexical rules similar to those used to account for inflectional morphology. Sag and Wasow (1997), for example, present the following lexical rule for agent-nominalization (*run-runner*).

(4)
$$\begin{bmatrix} \textit{verb} & \\ \text{PHON} & \boxed{1} \\ \text{ARG-ST} & \left\langle \text{NP}_i\, ,.. \right\rangle \\ \text{CONT} & \begin{bmatrix}\text{RESTR} & \begin{bmatrix}\text{AGENT} & i\end{bmatrix}\end{bmatrix}\end{bmatrix} \Rightarrow \begin{bmatrix} \textit{noun} & \\ \text{PHON} & f_{er}(\boxed{1}) \\ \text{CONT} & \begin{bmatrix}\text{INDEX} & i\end{bmatrix}\end{bmatrix}$$

Agent-nominalization takes as input a verb, and produces a noun. The function f_{er} attaches the suffix *-er* to the stem of the verb. The index of the agent role of the verb is token-identical to the index of the corresponding noun.

3.2. *Valence Alternation*

Apart from inflection and derivation, there is a third class of lexical rules to be found in the literature, usually referred to as valence alternation rules. These rules are used to account for the fact that some word classes appear systematically with different, but related, subcategorization requirements. One way to think about valence alternation rules is as a special kind of derivational rules: they produce derived lexical entries, but the morphophonological effects are limited (i.e. often the phonological form of the input and output is identical, in some cases the relationship follows an inflectional paradigm).

Following a proposal in Pollard and Sag (1994; pp 145 ff.), one might account for the extraposition of the subject clause in (5b) by means of the lexical rule in (6).

(5) a. That I do not know his favourite recipes bothers me.

b. It bothers me that I do not know his favourite recipes.

(6)
$$\begin{bmatrix} \text{SUBJ} & \left\langle \text{S}[\textit{comp}] \right\rangle \\ \text{COMPS} & \boxed{1} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{SUBJ} & \left\langle \text{NP}[\textit{it}] \right\rangle \\ \text{COMPS} & \boxed{1} \oplus \left\langle \text{S}[\textit{comp}] \right\rangle \end{bmatrix}$$

This lexical rule replaces the subject *that*-clause by an expletive *it*-NP, while appending the *that*-clause to COMPS.[7]

Also passive is often discussed under the rubric of valence alternation. The lexicalist analysis of the alternation in

(7) a. Kim will read the book

b. The book was read by Sandy

assumes that the passive participle *read* selects for a subject that corresponds to the object of the base form *read*, and that the *by*-phrase corresponds to the subject of the base form. This can be accounted for by a passive lexical rule, adopted from Borsley (1996; p. 197).[8]

$$(8)\quad \begin{bmatrix} \text{PHON} & \boxed{1} \\ \text{HEAD} & \begin{bmatrix}\text{VFORM} & \textit{bse}\end{bmatrix} \\ \text{SUBJ} & \langle\ \boxed{2}\ \text{NP}_i\ \rangle \\ \text{COMPS} & \langle\ \boxed{3} \mid \boxed{4}\ \rangle \\ \text{ARG-ST} & \langle\ \boxed{2} \mid \boxed{5}\ \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} \text{PHON} & \textit{fpass}(\boxed{1}) \\ \text{HEAD} & \begin{bmatrix}\text{VFORM} & \textit{passp}\end{bmatrix} \\ \text{SUBJ} & \langle\ \boxed{3}\ \rangle \\ \text{COMPS} & \boxed{4}\ (\oplus \langle\ \boxed{6}\ \text{PP}[\textit{by}]_i\ \rangle\ ) \\ \text{ARG-ST} & \boxed{5}\ (\oplus \langle\ \boxed{6}\ \rangle\ ) \end{bmatrix}$$

Note that apart from making a change to the valence features SUBJ and COMPS, the rule also affects ARG-ST. This will account for the fact that binding possibilities differ for actives and passives. Furthermore, passive morphology is assumed to be added by the function *fpass*.

While the use of lexical rules for phenomena such as extraposition and passive is relatively uncontroversial, there have also been proposals for using lexical rules to account for phenomena (such as extraction and the selection of adjuncts) that are traditionally accounted for outside the lexicon. For instance, following a proposal in Pollard and Sag (1994; p. 378), Sag and Fodor (1994) and Sag (1997) propose an account of extraction in which traces are eliminated in favour of the following lexical rule:

$$(9)\quad \begin{bmatrix} \text{COMPS} & \langle \ldots \begin{bmatrix}\text{LOC} & \boxed{3}\end{bmatrix} \ldots \rangle \\ \text{SLASH} & \boxed{2} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{COMPS} & \langle \ldots\ \ldots \rangle \\ \text{SLASH} & \boxed{2} \cup \left\{\boxed{3}\right\} \end{bmatrix}$$

7 See Keller (1995), Van Eynde (1996) and Bouma (1996), for alternative accounts, arguing that extraposition should be analyzed as a nonlocal dependency. The proposal of Van Eynde does not use lexical rules.

8 $\langle H|T \rangle$ denotes the list with head H and tail T.

This rule removes an element from COMPS, while adding the local features of the removed element to SLASH. This accounts for the fact that *read* in (10a) selects for an object complement, whereas *read* in (10b) instead introduces a *slashed*-category, which will licence the initial WH-phrase *which book*.

(10) a. Kim believes Sandy has read the book.

b. Which book does Kim believe Sandy has read?

Following a proposal in Miller (1992), a number of authors (van Noord and Bouma, 1994; Manning et al., 1997; Kim and Sag, 1995) have also proposed a lexical rule which allows heads to select for their adjuncts. The proposed lexical rule adds adjuncts to the COMPS list of verbs. One of the striking aspects of such a rule is that it must be able to apply to its own output (as a verb may combine with more than a single adjunct).

3.3. *Formal and Computational Aspects*

One problematic aspect of lexical rules is that they are heavily underspecified. Roughly speaking, the input specification of a lexical rule contains information specifying to which lexical entries the rule applies (i.e. passive applies to [VFORM *bse*] lexical entries). The output specification contains the information which holds for the output, but not for the input (i.e. the result of passive is [VFORM *passp*]). The interpretation of such linguistic rule specifications is that all information in the input which is not mentioned in the output specification (i.e. for which the output specification does not provide explicit information) is to be included in the result of applying the rule to a given lexical entry. A different way of expressing this is that, by default, information in the input is assumed to be included in the output. Proposals for a formalization of the kind of default mechanism this would require can be found in Bouma (1992) and Lascarides et al. (1996).

A rather different solution to solving the default copying problem is implicit in attempts to integrate lexical rules with the hierarchical approach to the lexicon. In the approach of Flickinger et al. (1985) lexical rules relate frames to frames. The lexical rule for passive, for instance, takes as input a lexical entry inheriting from *base* and *transitive*, and has as output an entry which inherits from *passive*. The value of PHON, the valence features, and the semantics of the output are defined explicitly in terms of the input. All other information relevant to the output, but not idiosyncratic to the specific entry to which the rule applies, is added to the output by means of *inheritance* (i.e. from frames that are supertypes of *passive*) and *not copied* between input and output.

While most lexical rules have some effect on the phonological form of a word, there are also a number of rules where the phonological form of the input is simply identical to that of the output. This is true, for instance, for the extraposition (6) and extraction (9) lexical rules presented above. In many implemented grammar formalisms (e.g. ALE, Carpenter (1992a)) such lexical rules are treated as unary syntax rules, which happen to give rise to words, instead of phrases. Apart from questions about the linguistic adequacy of such an approach, one may question the effects it will have on processing. Unary rules may, if they can be applied recursively, give rise to non-branching derivations of arbitrary depth. Termination of the parse-process is typically not guaranteed in such cases.

In fact, Carpenter (1991) proves that even the simplest grammar formalism employing complex symbols for representing syntactic valence is in principle undecidable, if it includes a lexical rule component. His proof rests on the observation that linguistic adequacy suggests that lexical rules must be able to insert, delete, and permute elements of a valence-list such as COMPS, and furthermore, that lexical rules must be able to apply recursively to their own output. Taken together, these two assumptions lead to the conclusion that lexical rules can be used to perform arbitrary computations by manipulating a list-valued feature. A formalism containing such a rule component is therefore able to recognize all recursively enumerable languages, and thus is undecidable. Note that recursive rule application is necessary in any case for the adjunct lexical rule, and also for the extraction lexical rule, if multiple extractions are to be accounted for (as may be the case for the Scandinavian languages and for so-called *easy*-constructions in English).

Even if termination cannot be guaranteed in general, it is often possible for a given grammar and set of lexical rules to obtain a parser that will be able to deal with unary rule application. In van Noord and Bouma (1994), for instance, it is argued that a recursive lexical rule such as the adjuncts lexical rule can be dealt with if one employs a logic programming technique known as *delayed evaluation*. A variant of this approach is proposed in Meurers and Minnen (1997). They argue that standard lexical rules can be compiled automatically into a constraint on lexical entries, in which all possible rule interactions are encoded. Furthermore, information shared between rule inputs and outputs can be propagated ‘upward’, from input to output, before any of the actual rule applications are carried out.

As the result of (1991) illustrates, lexical rules are procedural devices not unlike transformations in transformational grammar. A lexical rule may perform arbitrary, unconstrained operations on feature structures, apply any number of times and in any order (often leading to spurious ambiguity), which gives rise to a number of formal and computational problems. Given the procedural, transformational flavour of lexical rules, it is not surprising that several authors have argued for approaches in which the use of lexical rules is kept at a minimum, or even completely eliminated. All of these proposals have in common that they make use of implicational constraints and relations to constrain the value of certain features.

4. Alternatives for Lexical Rules

Constraint-based grammar formalisms have in general been very successful in eliminating those parts of linguistic theory which are difficult to understand in declarative terms. In syntax, for instance, declarative alternatives for transformations, feature percolation mechanisms, or indexing operations are included in practically every constraint-based grammar formalism. Constraint-based approaches to phonology (Bird, 1995; Mastroianni and Carpenter, 1994) have proven that the notion of a phonological rule as applying to one representation and producing another, is superfluous in a system where (possibly complex) constraints apply to (underspecified) phonological representations. Computational semantics, finally, has shown that rules for quantifier storage and retrieval can be eliminated in a formalism using underspecified semantic representations and constraints (Alshawi, 1992; Copestake et al., 1997). There are several proposals in the literature in which constraint-based alternatives for lexical rules are developed. These are reviewed below.

4.1. *Constraint-based approaches to inflection*

Kathol (1994) and Krieger and Nerbonne (1993) outline an approach to inflection in which the phonological form of an inflected word is defined by means of a relational constraint (Kathol) or distributed disjunction (Krieger and Nerbonne) defined on PHON and the relevant agreement features. For instance, Kathol presents the following definition of adjectives in German:

$$(11)\ \textit{adj-word} \rightarrow \begin{bmatrix} \text{PHON} & \boxed{3} \\ \text{STEM} & \boxed{1} \\ \text{HEAD} & \begin{bmatrix} \text{INFL} & \boxed{2} \end{bmatrix} \end{bmatrix} \wedge r_{adj}(\boxed{1},\boxed{2},\boxed{3})$$

This implicational constraint states that all AVM's of type *adj-word* must satisfy the r_{adj} relation, which is a relation between the values of STEM (the uninflected stem form of an adjective), INFL (a bundle of inflectional features), and PHON (the inflected form of the adjective). The definition of r_{adj} is basically a table, containing entries such as the following:

(12) a. $r_{adj}(\boxed{1}, \begin{bmatrix} \text{NUM} & sg \\ \text{CASE} & gen \vee acc \\ \text{DECL} & weak \end{bmatrix}, \langle \boxed{1}, \texttt{en} \rangle)$

b. $r_{adj}(\boxed{1}, \begin{bmatrix} \text{NUM} & sg \\ \text{GEN} & fem \\ \text{CASE} & nom \vee acc \end{bmatrix}, \langle \boxed{1}, \texttt{e} \rangle)$

A given lexical entry such as (13) can now satisfy the constraint in (11), among others, as shown in (14a,b).

(13) $\begin{bmatrix} \textit{adj-word} & \\ \text{STEM} & rot \end{bmatrix}$

(14) a. $\begin{bmatrix} \textit{adj-word} & \\ \text{PHON} & \langle rot,en \rangle \\ \text{STEM} & rot \\ \text{HEAD} & \begin{bmatrix} \text{INFL} & \begin{bmatrix} \text{NUM} & sg \\ \text{CASE} & gen \vee acc \\ \text{DECL} & weak \end{bmatrix} \end{bmatrix} \end{bmatrix}$

b. $\begin{bmatrix} \textit{adj-word} & \\ \text{PHON} & \langle rot,e \rangle \\ \text{STEM} & rot \\ \text{HEAD} & \begin{bmatrix} \text{INFL} & \begin{bmatrix} \text{NUM} & sg \\ \text{GEN} & fem \\ \text{CASE} & nom \vee acc \end{bmatrix} \end{bmatrix} \end{bmatrix}$

Krieger and Nerbonne argue that paradigmatic information can be incorporated into the lexical entries (following the inheritance-based approach to inflection presented in Evans and Gazdar (1989)), thus opening the possibility for dealing in a principled manner with exceptional cases. The constraint-based approach to inflection can also be extended to related phenomena. Miller and Sag (1997), for instance, present a constraint-based analysis of French cliticization.

The constraint-based approach sees inflection as a constraint, which will instantiate the value of PHON relative to a set of agreement features, or, alternatively, constrains the possible values for the agreement features relative to a given PHON value. The difference between this approach and the lexical rule-based one sketched in section 3 is that no rule-application is involved, and thus thorny issues such as the relationship between input and output, or order of rule-application, do not arise. Another advantage of this approach is that it is immediately compatible with a hierarchical approach to the lexicon, whereas the interaction of inheritance and lexical rules is much more complicated.

4.2. *Valence Alternation*

Kathol (1994) considers an analysis of (adjectival) passive where both the active and passive forms of a verb are instances of a *proto-verb*, which contains the information common to both. A *proto-verb* has subsorts for active and passive, in which the values for PROTO-SUBJ and PROTO-COMPS have been linked in appropriate ways to the values for SUBJ and COMPS respectively.

proto-verb

(15)
$$\left[\begin{array}{ll} \textit{active} & \\ \text{PR-ARGS} & \left[\begin{array}{ll} \text{PR-SUBJ} & \boxed{1} \\ \text{PR-COMPS} & \boxed{2} \\ \text{PR-ERG} & \boxed{3} \end{array}\right] \\ \text{SUBJ} & \boxed{1} \\ \text{COMPS} & \boxed{2} \\ \text{PR-ERG} & \boxed{3} \end{array}\right] \quad \left[\begin{array}{ll} \textit{passive} & \\ \text{PR-ARGS} & \left[\begin{array}{ll} \text{PR-SUBJ} & \boxed{1} \\ \text{PR-COMPS} & \boxed{2} \\ \text{PR-ERG} & \langle \boxed{3}\text{NP}_i \rangle \end{array}\right] \\ \text{SUBJ} & \langle \text{NP}_i \rangle \\ \text{COMPS} & \boxed{2} - \boxed{3} \end{array}\right]$$

The type for *passive* identifies the ergative argument object of transitive verbs with the actual subject. The proto-ergative argument corresponds to the (proto-)object of transitive verbs, and to the proto-subject of unaccusative verbs, and thus this subtype generalizes over both personal and impersonal passives. If the ergative argument is the (proto)-object, it will be a member of PR-COMPS, but not of COMPS. This account of passive covers both personal and impersonal passives.

The use of *proto* attributes is reminiscent of the use of a feature such as STEM, ROOT or MORPH-BASE, whose value is a full-blown feature structure corresponding to a word or lexeme, see Riehemann (1994), Manning and Sag (1995), and others. Yet, we believe there is an important difference. The approach of Kathol does not distinguish between stems, lexemes or roots, and derived or inflected words. Rather, his presentation suggests that the passive and active forms of a verb are specializations of a general type. Thus, even though one might argue that the *passive* type encodes the effect of rule-application, it is difficult to envisage something like successive, or even cyclic, rule application. The only way in which lexical rules could be 'applied' to the structures introduced by Kathol is by refining or specializing these structures, and thus, rule application essentially is restricted to monotonic application of constraints to a single structure. We believe that this is the hallmark of a declarative account.

Manning and Sag (1995) argue that argument structure (ARG-ST) is a level of representation that should be distinguished from valence (i.e. the features SUBJ, SPR and COMPS). In the default case, the relationship between the two may be transparent (i.e. ARG-ST is typically the `append` of SUBJ and COMPS, for verbs), but this is by no means the only possibility. They argue that passive and ergative constructions in a variety of languages can be analyzed in terms of a noncanonical relationship between ARG-ST and valence. Van Noord and Bouma (1997) argue that an account of reflexive binding in the context of argument composition verbs provides another argument for distinguishing between argument structure and valence.

Interestingly, the introduction of ARG-ST makes it possible to restate a number of lexical rules as constraints, more or less along the lines of Kathol's analysis of passive (where ARG-ST takes over the role of the *proto* features). Bouma (1997) and Bouma et al. (1997) argue that the adjuncts and extraction lexical rules may be eliminated. Instead, the effect of these rules is incorporated in the relational constraint which defines the mapping between valence and argument structure. For instance, adopting the lexicalist approach to unbounded dependencies outlined in Sag (1997), in which complement extraction simply amounts to realizing certain complements as *gaps*, Bouma et al. (1997) demonstrate that one may incorporate the effect of complement extraction as part of the constraint defining the mapping between argument structure and COMPS. The mapping constraint in (16) realizes the tail of ARG-ST as complements, with the exception of those arguments that are instantiated as *gaps*.

$$(16)\quad \begin{bmatrix} \textit{verb-wrd} & \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{COMPS} & \boxed{2} - \textit{list}(\textit{gap}) \\ \text{ARG-ST} & \langle \boxed{1} \rangle \oplus \boxed{2} \end{bmatrix}$$

5. A CASE STUDY : SUBJECT-AUXILIARY INVERSION

In the previous sections, we have presented and compared various methods for capturing lexical generalizations. For concreteness' sake they have been exemplified with treatments of inflection, derivation and valence alternation, but –overall– the emphasis was on the formal and computational properties of the different methods. In this section we take another –complementary– perspective. Starting from a specific phenomenon, i.e. Subject-Auxiliary Inversion in English, we present three ways in which the relevant lexical generalizations can be captured and evaluate them with respect to their empirical adequacy. The three methods are underspecification, lexical rules and constraint-based inheritance. Although we will plead in favour of the latter, we do not want to suggest that this choice is always the most appropriate one. The purpose of this section is not to show that one or another method is inherently superior to the other ones, but rather to argue that the choice between them is not only a matter of taste or of formal considerations, but also of empirical adequacy. As a background for the discussion we first sketch the HPSG treatment of the English auxiliaries.

5.1. *The English Auxiliaries*

Following a tradition which goes back to Ross (1969) and Pullum and Wilson (1977), HPSG treats the English auxiliaries as complement taking verbs. The auxiliary of the perfect, for instance, is analysed as a verb which takes a participial VP as its complement and a nominative NP as its subject (Figure 4).[9]

The auxiliary forms a *head-complement* phrase with the participial VP and the resulting finite VP forms a *head-subject* phrase with the pronoun. The fact that the subject of the auxiliary is understood as the subject of its participial complement is made explicit in its valence features: the first (and only) member of the auxiliary's SUBJ list is identical to the one (and only) member of the SUBJ list of its VP complement. In this respect, perfect *have* is similar to other subject raising verbs, such as *tend* and *seem*.[10] What makes the auxiliaries special, though, is their particular behavior with respect to

[9] The representation only mentions the head and valence features. For reasons of perspicuity we have adopted a tree style notation, using the PHON values as the leaves, but –if necessary– this can be converted straightforwardly into the more ornate AVM notation, which was adopted in Figure 2.

[10] The subject raising treatment is appropriate for most of the English auxiliaries, but not for all. The British stative *have*, for instance, as in *He has no money*, is a plain transitive verb. Other exceptions are discussed in Van Eynde (1998; p. 213-4).

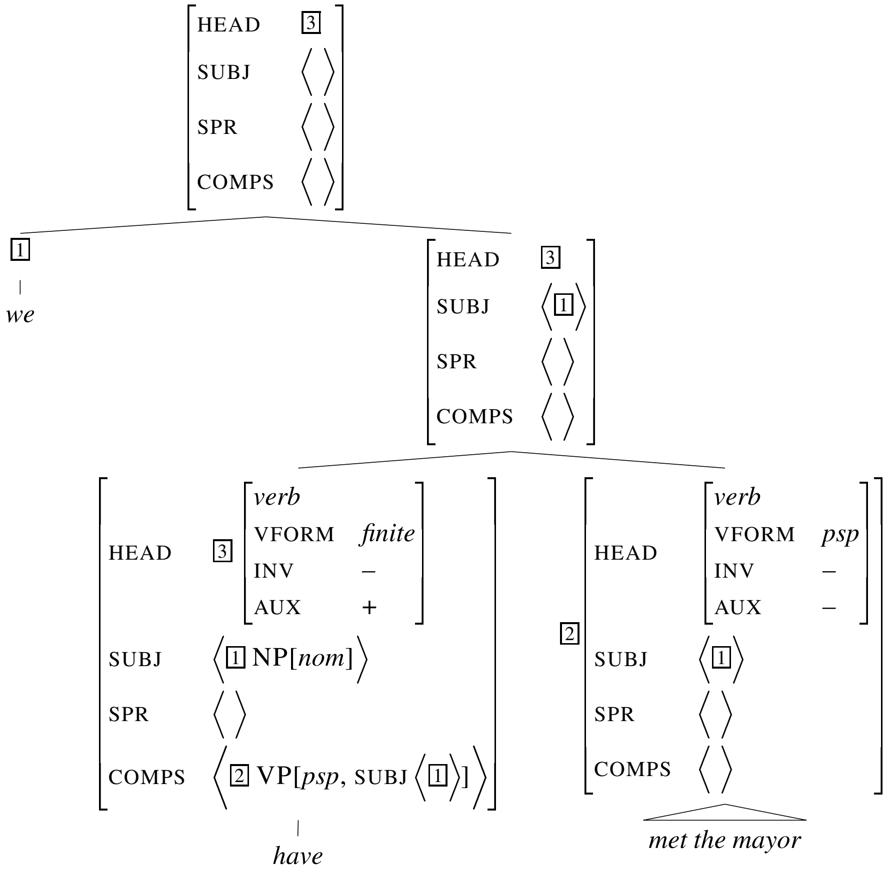


Figure 4. HPSG analysis of *We have met the mayor*

negation, inversion, contraction and ellipsis (the NICE properties). In this case study, it is the inversion property which we will focus on.

Roughly speaking, the auxiliaries are the only English verbs which can precede their subject. Compare, for instance, the auxiliary *have* with the main verb *put*.

(17) a. Where have you put your car?

b. * Where put you your car?

This observation, though, has to be qualified in two ways. First, we should make a distinction between Subject-Aux inversion and stylistic inversion, for the latter is also allowed for main verbs.

(18) a. Down came the rain.

b. In the corner stood a black arm chair.

A syntactic characteristic of stylistic inversion is that it concerns the entire VP. As a consequence, if the verb has any complements or adjuncts, these dependents precede the subject as well.

(19) a. Into the room strode solemnly the men I had been waiting for.

b. *Into the room strode the men I had been waiting for solemnly.

Subject-Aux inversion, on the other hand, only concerns the verb, so that its complements and/or adjuncts follow the subject. In other words, while the subject is clause final in stylistic inversion, it is not in the case of Subject-Aux inversion (modulo ellipsis).

A second qualification concerns the form of the auxiliaries. In order to be invertible, the auxiliary has to be tensed or subjunctive: nonfinite forms and imperatives—with the exception of *don't*—cannot be involved in Subject-Aux inversion. For the sake of concreteness, Figure 5 gives a survey of the relevant forms.

The table only mentions the forms with [+AUX] uses. This explains the absence of *dares*, *needs* and *needed*, which are invariably used as main verbs, and hence non-invertible.

The contracted negatives are listed separately, since they are autonomous lexical units: neither their forms nor their meanings can be derived compositionally from the ones of the auxiliary and the negation marker *not*, as demonstrated at length in Zwicky and Pullum (1983). Some of the contracted negatives are rarely used, especially in American English; they have been put between square brackets.

Of the subjunctive forms, the one of the copula is certainly invertible.

(20) That is pretty much the kind of foreign policy the next President, be he Clinton or Dole, will provide. (TIME, 11/4/96, p.47)

The auxiliary *have* has also got a separate form for the subjunctive, but it is used so rarely that we have only found some non-inverted uses, as in

(21) Chapter 10 restates a general overview, lest it have been lost in the intervening furor. (Jackendoff 1977, p.27)

We assume, though, that the inverted use of this form is—in principle—not impossible. The auxiliary *do*, on the other hand, does not occur in the subjunctive: this form only exists for the transitive main verb *do*.

AUX	present	pres-neg	past	past-neg	subj	imp-neg
Central	can	can't	could	couldn't		
Modals	may	[mayn't]	might	mightn't		
	must	mustn't				
	shall	[shan't]	should	shouldn't		
	will	won't	would	wouldn't		
Marginal	dare	daren't	dared	[daredn't]		
Modals	need	needn't				
			ought	oughtn't		
			used	[use(d)n't]		
Do	do	don't	did	didn't		don't
	does	doesn't				
Have	have	haven't	had	hadn't	have	
	has	hasn't				
Copula	are	aren't	were	weren't	be	
	is	isn't	was	wasn't		
	am					

Figure 5. The invertible forms of the auxiliaries

As for the imperative forms, invertibility is harder to test, since they usually occur without subject. However, if there is one, it turns out that *don't* is the only form which may—and actually must—precede it.

(22) a. Don't you ever say that again!

b. *You don't ever say that again!

c. *Do you behave, please.

d. *Have you some patience, please.

e. *Be you quiet, please.

The sentences with *do*, *be* and *have* are grammatical if the subject *you* is dropped, but then we no longer have inversion, of course. Notice also that *be* and *have* show the typical [−AUX] property of requiring DO support in case of emphasis and negation.

Having defined in more precise terms what Subject-Aux Inversion involves, we can now address the question of its analysis.

Since HPSG is a monostratal framework, it does not make use of any movement transformations: there is, for instance, no operation which moves the auxiliary from a canonical d-structure position to a pre-subject s-structure position, as in GB. Instead, there is only one stratum of syntactic representation, and the order of the words in that representation corresponds to the surface word order. As a consequence, the inverted constructions are not derived from their non-inverted counterparts, but generated as they are. The task of spelling out what they have in common is left to the lexicon, especially to the AVMs of the invertible auxiliaries. How this can be modeled will be discussed in some detail in the rest of this section.

5.2. *Underspecification*

A good example of an underspecification analysis is the one of Pollard and Sag (1994, 40-43); in order to model inverted constructions they employ a phrase type in which the verb simultaneously combines with its subject and its complements (Figure 6).

While the structure is rather different from the one of the non-inverted clause,[11] the AVM of the auxiliary is not: the only difference concerns the value of its INV feature. As a consequence, if the value of this feature is left underspecified, it is possible to capture both the inverted and non-inverted uses of the auxiliary in terms of one AVM. Which value it takes in a specific context is determined by the properties of the phrase in which it appears: in a phrase of type *head-subject-complements*, for instance, the value of INV has to be *plus*, whereas in phrases of type *head-subject* and *head-complements* it has to be *minus*. This treatment also provides a straightforward way to prevent the inversion of the non-invertibles; if they are assigned a negative INV value, they will not be accepted as heads of a *head-subj-comps* phrase.

In spite of its elegance, though, this analysis cannot be maintained as it is, since it ignores the fact that the auxiliaries do not always have the same syntactic and semantic properties in their [+INV] and [−INV] uses. Some examples of such discrepancies are discussed in Gazdar et al. (1982). One concerns the contracted negative *aren't*, which is compatible with the first person singular in its [+INV] use, but not in its [−INV] use.

[11] On the reasons for assigning a flat [Verb-NP-XP] structure rather than a binary one, such as [Verb-[NP-XP]] or [[Verb-NP]-XP], see Gazdar et al. (1985, p. 73).

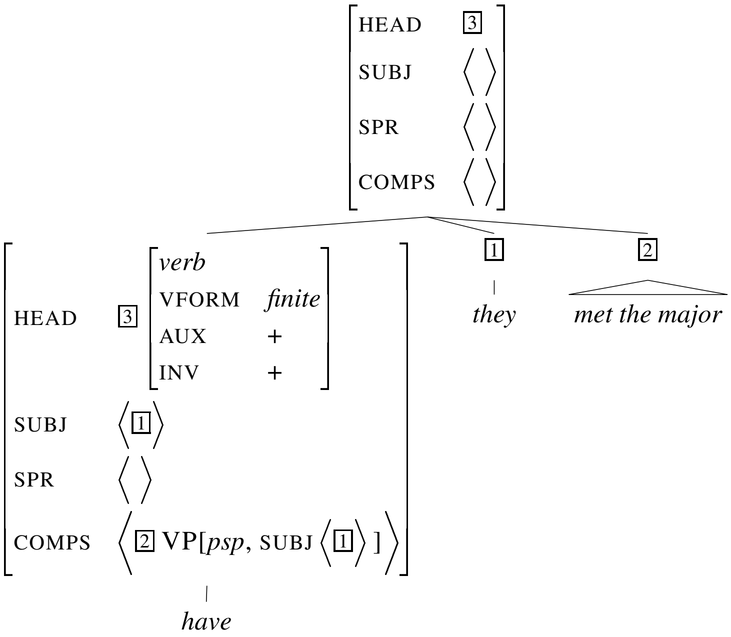


Figure 6. HPSG analysis of *Have they met the mayor?*

(23) a. Aren't I pretty?

b. * I aren't pretty.

Another concerns the meaning of *shall* in

(24) a. I shall go downtown.

b. Shall I go downtown?

While the [–INV] *shall* expresses futurity, its [+INV] counterpart "has a deontic sense, essentially equivalent to that of *should*." (o.c., p. 611)
A third example concerns the scope of the negation in *mightn't*.

(25) a. Mightn't Kim go?

b. Kim mightn't go.

Whereas the negation in the [+INV] *mightn't* has wide scope, the one of its [–INV] counterpart has narrow scope. Not mentioned in Gazdar et al. (1982), but equally telling is the case of *mustn't*.

(26) a. Mustn't I go?

b. He mustn't leave now.

While the most natural interpretation of [+INV] *mustn't* is the one with wide scope negation, as in the paraphrase *Isn't it the case that I must go?* (Palmer 1987, p. 127), its [–INV] counterpart can only have narrow scope negation. Yet another verb of this type is deontic *shouldn't*, which has narrow scope negation in its [–INV] use, and wide scope negation in its [+INV] use (Palmer 1987, p. 133).

In the words of Gazdar et al. (1982, p. 611): "Contrasts of this sort are exactly what one is led to expect once the lexicon distinguishes [+INV] and [–INV] forms." For the underspecification approach, such contrasts are an obvious problem, since they demonstrate that the AVMs of the [+INV] and [–INV] auxiliaries must be kept distinct.

5.3. *Valence alternation by lexical rule*

A treatment which postulates distinct AVMs for the [–INV] and [+INV] auxiliaries is the one which makes use of a lexical rule. For the sake of exposition, let us take the one of Warner (1993, pp. 83-5).[12]

$$
\begin{array}{l}
[\text{+AUX, +FIN, –BSE, SUBJ} \left\langle \boxed{1}\ \text{XP} \right\rangle, \text{COMPS} \left\langle \ldots \right\rangle] \quad \Rightarrow \\
[\text{+AUX, +INV, +FIN, –BSE, SUBJ} \left\langle \ \right\rangle, \text{COMPS} \left\langle \boxed{1}\ \text{XP}, \ldots \right\rangle]
\end{array}
$$

In other words, for every nonimperative finite auxiliary with an XP on its SUBJ list, there is a [+INV] counterpart in which the subject is the least oblique complement.

Typical of this treatment is that the inverted subject is not treated as a subject daughter, but rather as the least oblique complement of the verb. Such an analysis was first proposed in Borsley (1989) for the treatment of Welsh VSO clauses, and was later extended to the treatment of VSO clauses in other languages, such as German, Japanese and Korean, cf. Pollard (1990) on German. For English, it implies that one no longer needs a separate phrase type for the inverted clauses: instead, they are now straightforward instances of the *head-complements* phrase type; furthermore, given the correlation between surface order and degrees of obliqueness, it follows –without further stipulation– that the inverted subject precedes the other complements.[13]

[12] In Warner's terminology, [+FIN,–BSE] stands for the nonimperative finite forms.

[13] This treatment is also adopted in Sag (1997; p.439).

At first sight, this approach looks better than the one of underspecification, for since it assigns separate AVMs to the inverted auxiliaries and their non-inverted counterparts, it is –at least in principle– better equipped to deal with their discrepancies. In practice, though, it does not fare any better, for since the information which is not explicitly mentioned in the lexical rule is considered to be identical for both the input and the output (see section 3.3), the prediction is that the inverted auxiliaries share all properties of the [–INV] ones, apart from the values of INV, SUBJ and COMPS. As a consequence, the rule incorrectly predicts that the inverted *aren't* is only used in the plural and the second person, that *shall I XP* has the same meaning as *I shall XP*, and that the inverted *mustn't*, *shouldn't* and *mightn't* have narrow scope negation.

These deficiencies can be repaired, of course: it is always possible to add whatever stipulations are needed to fit the facts, but then it is worth looking at the nature of those stipulations. The one which is needed for the negated modals will have to undo the constraint that the negation has narrow scope, and the one which is needed for *shall* will have to change its meaning; both of these stipulations are nonmonotonic. Moreover, the one which is needed for *aren't* involves the addition of a separate entry for the first person singular use, so that *aren't* is assigned three AVMs (the [–INV] one, the derived [+INV] one and the non-derived [+INV] one), which is one more than would be needed in a treatment which simply acknowledges that [+INV] and [–INV] *aren't* are different, equally basic entries.

What this suggests, is a radically different view on the relation between inverted and non-inverted auxiliaries. Instead of taking their similarities as the norm, and resorting to ad-hoc stipulations whenever there is a discrepancy, it would be preferable to take their differences as the starting point and to leave the capturing of their similarities to some specific constraints. This is the approach which will be adopted in the next paragraph.

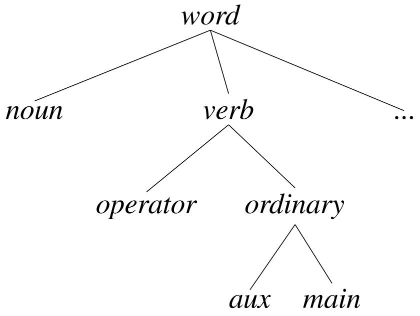
5.4. *Constraint-based inheritance*

In contrast to the other methods, which could be exemplified with existing proposals, this method has not yet been applied to the phenomenon of Subject-Aux inversion. The following is, hence, novel.

For a start, let us assume the (partial) hierarchy of lexical types in (27).

As pointed out in section 2, the lexical items of type *verb* belong to the part of speech *verb* and have an ARG-ST value which equals the append of their SUBJ and COMPS values (the Argument Realization Principle).

(27)



(28) *verb* $\rightarrow$ $$\left[\text{SYNSEM}\left[\text{LOC}\left[\text{CAT}\begin{bmatrix}\text{HEAD} & \textit{verb}\\ \text{SUBJ} & \boxed{1}\\ \text{SPR} & \langle\,\rangle\\ \text{COMPS} & \boxed{2}\\ \text{ARG-ST} & \boxed{1}\oplus\boxed{2}\end{bmatrix}\right]\right]\right]$$

The operator verbs are the inverted auxiliaries.[14] They are finite and have positive values for AUX and INV. Furthermore, since they have an empty SUBJ list, it follows –from constraint (27)– that their subject is treated as a complement, as in Warner's analysis.

(29) *operator* $\rightarrow$ $$\left[\text{SYNSEM}\left[\text{LOC}\left[\text{CAT}\begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{VFORM}\ \textit{finite}\\ \text{AUX}\ +\\ \text{INV}\ +\end{bmatrix}\\ \text{SUBJ} & \langle\,\rangle\end{bmatrix}\right]\right]\right]$$

The ordinary verbs, on the other hand, have a negative value for INV and a SUBJ list of length one, which implies that their subject is VP-external.[15]

(30) *ordinary* $\rightarrow$ $$\left[\text{SYNSEM}\left[\text{LOC}\left[\text{CAT}\begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{INV}\ -\end{bmatrix}\\ \text{SUBJ} & \langle[\,]\rangle\end{bmatrix}\right]\right]\right]$$

[14] This term is taken from Quirk et al. (1985).

[15] This does not exclude the possibility of stylistic inversion, for since the subject is clause final in that case, it can be treated as VP-external.

The ordinary verbs include both the main verbs, which are [– AUX], and the non-inverted auxiliaries, which are [+ AUX].[16]

Given this hierarchy, most of the auxiliaries have two AVMs, i.e. one of type *operator* and one of type *aux*. Since these AVMs are not related by lexical rule, but equally basic, there is no implication—neither explicit nor implicit—that they have the same properties. On the contrary, the initial assumption is that their properties are different. This does not lead to a high degree of redundancy, though, since the properties which they have in common can be captured in terms of constraints and multiple inheritance. The fact that the inverted *shouldn't* has the same argument structure as its non-inverted homonym, for instance, can be stated independently of the question of whether the first argument is realized as a subject or as a complement.

An unusual property of the present treatment is that it predicts more commonalities between the [– INV] auxiliaries and the main verbs than between the [– INV] auxiliaries and the operator verbs. As such, it is diametrically opposed to the predictions which are made in the underspecification treatment. Strange as this may seem at first, this might well be an asset, since it provides a natural explanation for the fact that a number of the auxilaries which are commonly used in non-inverted clauses, are hardly ever used as operators, especially in American English. Gazdar et al. (1985, p. 65) mentions *ought* and *might*, but the list can be extended with *dared* and *used*, and with the contracted negatives *mustn't, mightn't, needn't, daren't* and *oughtn't*. In the underspecification treatment, this discrepancy would have to be interpreted as a loss of generality, but within the present treatment, it is naturally interpreted as a gain of generality, since it amounts to the elimination of an ambiguity in the lexicon.

In sum, for the treatment of Subject-Auxiliary inversion in English, the combined use of multiple inheritance and implicational constraints appears to yield better results than either underspecification or lexical rules. As pointed out at the beginning of the section, this does not mean that this approach is intrinsically better than the other two. On the contrary, as a means for capturing lexical generalizations, underspecification will always be the most straightforward method, but there are cases in which it leads to overgeneralization, and in such cases the addition of finer-grained distinctions in the lexical hierarchy appears to yield better results.

[16] The non-inverted auxiliaries can still precede the subject, but only as the result of stylistic inversion, as in *By 'strategy' is meant the basic planning of the whole operation*, cf. Quirk et al. (1985, p. 1380-1).

6. Conclusions

Capturing lexical generalizations adequately is of crucial importance for all lexicalist grammar frameworks. In this paper we have reviewed the situation in HPSG, where inheritance and lexical rules have been used to capture such generalizations.

Lexical rules are traditionally used to account for a wide range of lexical relationships. The status of lexical rules within HPSG has always been somewhat unclear. Many analyses within HPSG presuppose the existence of certain lexical rules, but at the same time most linguists working within a declarative, non-transformational, theory such as HPSG feel uncomfortable if they have to defend analyses which, at least superficially, introduce a procedural and transformational aspect into the theory.

Inheritance is much more appealing from a declarative, constraint-based, perspective. By organizing the lexicon as an inheritance network generalizations can be stated as inheritance relationships, the various dimensions along which a word needs to be classified can be expressed using multiple inheritance, and exceptions can be expressed using nonmonotonic inheritance. Again, certain formal details remain to be resolved, especially with respect to nonmonotonicity.

Interestingly, inheritance is also immediately compatible with attempts to eliminate lexical rules by using underspecification and relational constraints. If the relationship between, say, an active verb and its passive counterpart, or the noninverted and inverted usage of an auxiliary is no longer captured by means of a lexical rule, it must be the case, if they are to be related at all, that both are ‘instances’ of a general, underspecified, lexical entry. Both plain underspecification (i.e. not assigning a value to certain crucial features) and relational constraints (defining the value of one or more features using a definite relation which typically has more than a single solution) are essential for defining such abstract lexical entries.

By combining relational constraints with inheritance, a lexical organization arises in which generalizations which hold for a class of items (say the information common to all verbs, to all transitive verbs, or to all operator verbs) are expressed using inheritance, whereas generalizations relating different forms of the same abstract lexical entry are expressed using relational constraints.

References

Alshawi, H. (ed.): 1992, *The Core Language Engine*. ACL-MIT press.

Bird, S.: 1995, *Computational phonology*. Cambridge: Cambridge University Press.

Borsley, R. D.: 1989, 'An HPSG approach to Welsh'. *Journal of Linguistics* **25**, 333–354.

Borsley, R. D.: 1996, *Modern Phrase Structure Grammar*. Oxford: Blackwell.

Bouma, G.: 1992, 'Feature Structures and Nonmonotonicity'. *Computational Linguistics* **18**(2), 183–204.

Bouma, G.: 1996, 'Extraposition as a nonlocal dependency'. In: G.-J. Kruijff, G. Morrill, and D. Oehrle (eds.): *Proceedings Formal Grammar*. Prague, pp. 1–14.

Bouma, G.: 1997, 'Valence alternations without lexical rules'. In: J. Landsbergen, J. Odijk, K. van Deemter, and G. V. van Zanten (eds.): *CLIN 1996, Papers from the seventh CLIN Meeting 1996*. Eindhoven, pp. 25–40.

Bouma, G., R. Malouf, and I. Sag: 1997, 'Satisfying Constraints on Adjunction and Extraction'. ms. Groningen University and Stanford University.

Bresnan, J.: 1976, 'On the form and functioning of transformations'. *Linguistic Inquiry* **7**, 3–40.

Bresnan, J.: 1982, 'The Passive in Lexical Theory'. In: J. Bresnan (ed.): *The Mental Representation of Grammatical Relations*. MIT Press, pp. 3–86.

Carpenter, B.: 1991, 'The Generative Power of Categorial Grammars and Head-Driven Phrase Structure Grammars with Lexical Rules'. *Computational Linguistics* **17**(3), 301–313.

Carpenter, B.: 1992a, 'The Attribute Logic Engine User Guide'. Technical report, Laboratory for Computational Linguistics, Carnegie Mellon University, Pittsburgh.

Carpenter, B.: 1992b, *The Logic of Typed Feature Structures*. Cambridge: Cambridge University Press.

Carpenter, B.: 1992c, 'Skeptical and Creduluous Default Unification with applications to templates and Inheritance'. In: T. Briscoe, A. Copestake, and V. de Paiva (eds.): *Default Inheritance within Unification-Based Approaches to the Lexicon*. Cambridge: Cambridge University Press.

Chomsky, N.: 1970, 'Remarks on nominalization'. In: *Readings in English Transformational Grammar*. Waltham, Mass.: Ginn.

Copestake, A., D. Flickinger, and I. Sag: 1997, 'Minimal Recursion Semantics: An Introduction'. Ms. Stanford University.

Evans, R. and G. Gazdar: 1989, 'Inference in DATR'. In: *Proc. of the 4th European ACL*. pp. 66–71.

Flickinger, D. P.: 1987, 'Lexical Rules in the Hierarchical Lexicon'. Ph.D. thesis, Stanford University.

Flickinger, D. P., C. Pollard, and T. Wasow: 1985, 'Structure Sharing in Lexical Representation'. In: *23th Annual Meeting of the Association for Computational Linguistics*.

Gazdar, G.: 1981, 'Unbounded Dependencies and Coordinate Structure'. *Linguistic Inquiry* **12**, 155–84.

Gazdar, G., E. Klein, G. Pullum, and I. Sag: 1985, *Generalized Phrase Structure Grammar*. Oxford: Basil Blackwell.

Gazdar, G., G. Pullum, and I. Sag: 1982, 'Auxiliaries and related phenomena in a restrictive theory of grammar'. *Language* **58**, 591–638.

Götz, T. and W. D. Meurers: 1997, 'Interleaving universal principles and relational constraints over typed feature logic'. In: *Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the EACL*. Madrid, Spain.

Huddleston, R.: 1974, 'Further remarks on the analysis of auxiliaries as main verbs'. *Foundations of Language* **11**, 215–229.

Jackendoff, R.: 1975, 'Morphological and semantic regularities in the lexicon'. *Language* **51**, 639–671.

Kathol, A.: 1994, 'Passive without Lexical Rules'. In: J. Nerbonne, K. Netter, and C. Pollard (eds.): *German in Head-driven Phrase Structure Grammar*. Stanford, pp. 237–272.

Keller, F.: 1995, 'Towards an Account of Extraposition in HPSG'. In: *Proceedings of the EACL*. Dublin.

Kim, J.-B. and I. A. Sag: 1995, 'The Parametric Variation of English and French Negation'. In: *Proceedings of the West-Coast Conference on Formal Grammar 1995*.

Krieger, H.-U. and J. Nerbonne: 1993, 'Feature-Based Inheritance Networks for Computational Lexicons'. In: T. Briscoe, V. de Paiva, and A. Copestake (eds.): *Inheritance, Defaults, and the Lexicon*. Cambridge: Cambridge University Press, pp. 90–136.

Lascarides, A., T. Briscoe, N. Asher, and A. Copestake: 1996, 'Order Independent and Persistent Typed Default Unification'. *Linguistics and Philosophy* **19**(1), 1–89.

Manning, C. and I. Sag: 1995, 'Dissociations between Argument Structure and Grammatical Relations'. Draft, Stanford University, July 1995.

Manning, C., I. Sag, and M. Iida: 1997, 'The Lexical Integrity of Japanese Causatives'. In: R. Levine and G. Green (eds.): *Readings in HPSG*. Cambridge: Cambridge University Press. to appear.

Mastroianni, M. and B. Carpenter: 1994, 'Constraint-based Morpho-Phonology'. In: *Proceedings of the first meeting of the ACL-SIG in Computational Phonology*. Las Cruces, NM, pp. 13–24.

Meurers, W. D. and G. Minnen: 1997, 'A computational treatment of lexical rules in HPSG as covariation in lexical entries'. *Computational Linguistics* **23**(4), 543–568.

Miller, P.: 1992, *Clitics and Constituents in Phrase Structure Grammar*. New York: Garland.

Miller, P. and I. Sag: 1997, 'French Clitic Movement without Clitics or Movement'. *Natural Language and Linguistic Theory*. to appear.

Palmer, F.: 1987, *The English verb*. London: Longman. Second edition.

Pollard, C.: 1990, 'On Head Non-Movement'. In: *Appendix to the Proceedings of the Symposium on Discontinuous Constituency*. ITK Tilburg.

Pollard, C. and I. Sag: 1987, *Information Based Syntax and Semantics, Volume 1*. Center for the Study of Language and Information Stanford.

Pollard, C. and I. Sag: 1994, *Head-driven Phrase Structure Grammar*. Center for the Study of Language and Information Stanford.

Pullum, G. and D. Wilson: 1977, 'Autonomous syntax and the analysis of auxiliaries'. *Language* **53**(4), 741–788.

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik: 1985, *A comprehensive grammar of the English language*. London/New York: Longman.

Riehemann, S.: 1994, 'Morphology and the Hierarchical Lexicon'. ms.

Ross, J.: 1969, 'Auxiliaries as Main Verbs'. In: W. Todd (ed.): *Studies in Philosophical Linguistics*, Series One. Evanston (Ill.): Great Expectations, pp. 77–102.

Sag, I.: 1997, 'English Relative Clause Constructions'. *Journal of Linguistics*. to appear.

Sag, I. A. and J. D. Fodor: 1994, 'Extraction without traces'. In: *West Coast Conference on Formal Linguistics 13*. Stanford.

Sag, I. A. and T. Wasow: 1997, *Syntactic Theory: A Formal Introduction*. Stanford CA: CSLI Publications. To appear.

Thomason, R.: 1997, 'Nonmonotonicity in Linguistics'. In: *Handbook of Logic and Language*. Dordrecht: North Holland, pp. 777–831.

Van Eynde, F.: 1996, 'A monostratal treatment of it extraposition without lexical rules'. In: W. Daelemans, G. Durieux, and S. Gillis (eds.): *CLIN 1995, Papers from the sixth CLIN Meeting 1995*. Universitaire Instelling Antwerpen, Antwerpen.

Van Eynde, F.: 1998, 'Tense, Aspect and Negation'. In: F. van Eynde, and P. Schmidt (eds.): *Linguistic Specifications for Typed Feature Structure Mechanisms. Studies in Machine Translation and Natural Language Processing, vol. 10* . Luxembourg: Office for Official Publications of the European Communities, pp. 209–280.

van Noord, G. and G. Bouma: 1994, 'Adjuncts and the Processing of Lexical Rules'. In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*. Kyoto, pp. 250–256.

van Noord, G. and G. Bouma: 1997, 'Dutch Verb Clustering without Verb Clusters'. In: P. Blackburn and M. de Rijke (eds.): *Specifying Syntactic Structures*. Stanford: CSLI Publications, pp. 123–153.

Warner, A. (ed.): 1993, *English auxiliaries. Structure and History*. Cambridge U.P.

Zwicky, A. and G. Pullum: 1983, 'Cliticization vs. inflection: English *n't*'. *Language* **59**, 502–513.