

Sign-Based Construction Grammar

An informal synopsis*

Ivan A. Sag
Stanford University

August 19, 2007

1 Introduction

This paper is intended as an introduction to some of the central notions of Sign-Based Construction Grammar (SBCG). SBCG is an attempt to adapt ideas developed over a twenty year period of research in Head-Driven Phrase Structure Grammar (HPSG¹) to the analysis of constructional phenomena of the sort studied in the tradition of Berkeley Construction Grammar (BCG²). My goal is to expand the empirical coverage of HPSG, while at the same time putting BCG on a firmer theoretical footing.

I take a language to be an infinite set of **signs** and the job of the grammarian to provide a recursive enumeration of the signs that constitute the language (Chomsky 1955, 1957, 1965). ‘Sign’ is understood in a sense close to that of Saussure (1916). However, the Saussurian sign simply relates ‘form’ and ‘meaning’ (*signifiant, signifié*). Saussure’s dichotomy of sign properties is here divided up into features of phonology, (morphological) form, syntax, semantics, and context.

Signs are modeled as **feature structures** and feature structures are of two basic kinds:

- atoms³ (*acc(usative)*, +, *bumble-bee*, . . .),
- functions (as explained below).

*This paper was originally intended to be a chapter of a comprehensive book on that topic (and may still be, some day). I thank Chuck Fillmore, Paul Kay, and Laura Michaelis for useful discussion of both content and exposition and for comments on an earlier version of this paper. Discussions with Farrell Ackerman, Emily Bender, Dan Flickinger, Mark Gawron, Adele Goldberg, Andreas Kathol, Stefan Müller, and Frank Van Eynde have also been extremely useful. None of the views expressed here should necessarily be attributed to any of these scholars, however; nor should they be held responsible for any mistakes that might be found herein.

¹See Pollard and Sag 1987, Pollard and Sag 1994, Ginzburg and Sag 2000, and Richter 2004, among others.

²See Fillmore et al. 1988, Fillmore and Kay 1996, Kay and Fillmore 1999, Kay 2002a, and Michaelis and Lambrecht 1996, among others.

³The set of atoms includes, for analytic convenience, an infinite set of indices.

But the interesting feature structures are the functions. A functional feature structure maps each feature in its domain (some proper subset of the set of features) to an appropriate value. Since feature structures map some features to atoms and other features to functions, a general characterization of this kind of feature structure is that it is “a function mapping features to feature structures.” The particular features and values assumed here are given in the Appendix.

As in LFG and HPSG, SBCG makes a strict distinction between **model objects** and **descriptions** of those objects. Like HPSG, the most important model objects are signs, (the formal representations of actual words and phrases (including sentences)). Another, distinct kind of model object in SBCG is the **construct**. As in Generalized Phrase Structure Grammar (GPSG – see Gazdar et al. 1985), constructs are in essence local trees that are licensed by some construction of the grammar (e.g. a grammar rule or schema) or by a lexical entry. More formally, a construct is a feature structure with a MOTHER (MTR) feature and a DAUGHTERS (DTRS) feature. The value of the MTR feature is a sign and the value of the DTRS feature is a nonempty list of signs. (Lists of feature structures are a kind of functional feature structure.⁴) Signs and constructs, as already noted, are feature structures – they are part of the language model. Constructions are descriptions that license classes of linguistic objects (signs or constructs) – they are part of the grammar (the description of the language model).

The linguistic objects proposed here are classified in terms of a system of **types** whose members will be subject to general constraints (‘type constraints’) that will play an important role in SBCG. Every feature structure of a type t must satisfy the constraints of all the supertypes of t , plus any additional constraints imposed on instances of t itself.

Most relevant will be constraints on various subtypes of the type *construct*, which will have the general form shown in (1):

$$(1) \quad x\text{-cxt} \Rightarrow [\dots]$$

The item on the left of the arrow is the name of some type of construct (‘ x ’ here), the item on the right side of the arrow specifies properties that each construct of that type must have, and the arrow represents a conditional (‘if-then’) relation. Expressions that look like (1) are thus constraints that can be read as: ‘Members of a particular construct class (must) have the specific properties indicated’.

An expression of the form (1) corresponds to what I will call a **combinatoric construction**. Each combinatoric construction specifies general properties associated with a class of constructs (configurations of a mother and one or more daughters). Put differently, each such

⁴A common way of encoding lists is in terms of the features FIRST and REST. The singleton list containing the feature structure F would be as shown in (i), where *elist* stands for the empty list:

$$(i) \quad \begin{bmatrix} \text{FIRST} & F \\ \text{REST} & \textit{elist} \end{bmatrix}$$

The doubleton list $\langle G, F \rangle$ would then be encoded as in (ii):

$$(ii) \quad \begin{bmatrix} \text{FIRST} & G \\ \text{REST} & \begin{bmatrix} \text{FIRST} & F \\ \text{REST} & \textit{elist} \end{bmatrix} \end{bmatrix}$$

expression defines the distinctive properties of a mode of combination that is part of the grammar of a language – the properties that define a way of putting expressions together to ‘construct’ other, more complex expressions.

Unlike standard presentations of HPSG, where the type constraints form part of the **signature** of a grammar, the type constraints of SBCG are an essential part of the **body** of the grammar. The grammar signature delineates the inventory of types, features (and their possible values) and a specification of which features ‘go with’ which types of feature structure. Against this background of possible feature structures, the constructions, by contrast, tell us which particular families of feature structures exist in a given language. This duality in the role of constructions is at the heart of SBCG: The grammar signature establishes a large space of feature structures out of which the inventory of constructions selects the signs that constitute the language.

A second kind of construction that will be introduced below involves lexical classes. **Lexical-class constructions** have the general form shown in (2), where *lex* stands for any subtype of the type *lexical-sign* (i.e. any subtype of either *lexeme* or *word*):

$$(2) \quad lex \Rightarrow [\dots]$$

There is no formal difference between the two kinds of construction just illustrated. The only difference is the nature of the type name that serves as the antecedent of the conditional constraint. The rest of this essay will provide further details about the various kinds of signs and constructs on the model side and types and constructions on the description side.

2 Feature Structures

Grammatical objects of all kinds (including signs, case values, parts of speech, and constructions) are modeled as **feature structures**. I make the further assumption that feature structures are either atoms (like *pl(ural)*, *acc(usative)*, +, etc.) or else functions from features to feature structures.⁵ This is a simple, but powerful way of modeling linguistic objects, one that is already familiar from much work in phonology, where speech segments are often modeled in this way. For example the following:⁶

$$(3) \quad \left[\begin{array}{ll} \text{CONTINUANT} & - \\ \text{VOICED} & - \\ \text{ANTERIOR} & + \\ \text{CORONAL} & - \\ \text{SONORANT} & - \\ \text{CONSONANTAL} & + \\ \text{VOCALIC} & - \end{array} \right]$$

⁵Carpenter 1992. See also Sag et al. 2003.

⁶This is a [t] in the feature system of Chomsky and Halle (1968).

Similarly, the fundamental tenet of ‘X-Bar Theory’⁷ is that familiar atomic categories like NP or VP are to be reanalyzed as functions, e.g. as in (4):⁸

$$(4) \quad \begin{bmatrix} \text{NOUN} & + \\ \text{VERB} & - \\ \text{BAR} & 2 \end{bmatrix}$$

Note that the functional nature of this kind of analysis can be obscured by linguists’ tendency to write the value of a feature before the feature’s name, e.g. [–CORONAL] or [+VERB] or to use other notations, e.g. X^2 (Harris 1946) or \bar{X} (Chomsky 1970). Yet it is clear that the analytic intent is preserved by regarding such objects as functions whose domain is a set of features and whose range is a set of feature values (e.g. the set $\{+, -\}$ in the system of Chomsky and Halle 1968 or that of Chomsky 1974). The use of functions to model linguistic objects is thus nothing out of the ordinary, though notational idiosyncrasy and the fact that proponents of movement-based approaches fail to present their modeling assumptions explicitly often obscures this fact.

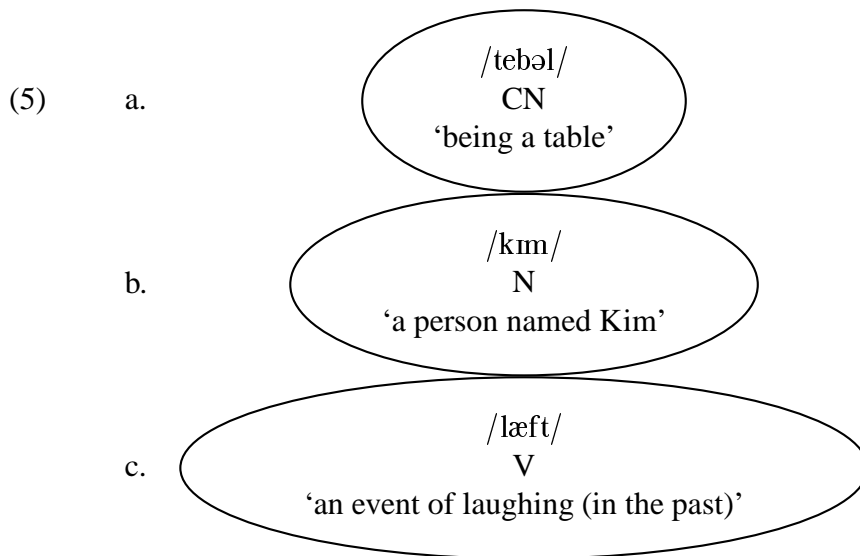
Building on the more explicit ideas pioneered by computational linguistic work of the late 1970s (e.g. Martin Kay’s Functional Unification Grammar [M. Kay 1979]) and the extensive subsequent work in GPSG, LFG, and HPSG,⁹ functions are used to model all grammatical objects. Grammatical categories, for example, are analyzed as complexes of various properties: nouns include specifications for the features CASE, NUMBER, and GENDER, verbs are specified for their inflection class (as [VFORM *finite*], [VFORM *present-participle*], etc.) and will have a ‘+’ or ‘–’ value for the feature AUXILIARY. These approaches take advantage of the full power of functions to model linguistic entities, unlike the phonological and X-bar illustrations given above, which represent a special case. In the examples from phonology and X-bar theory, the values of the features are all atomic. In the general case, values of features may either be atomic or correspond to complex feature structures. This allows for recursive embedding of feature structures within feature structures.

Signs are no exception. Saussure talked of signs as ‘associative bonds’ of sound concepts and semantic concepts. Adding in syntactic categories, which Saussure had little to say about, we arrive at a picture of signs like the one illustrated in (5)a-c (CN stands for common noun; N for non-common noun; V for verb):

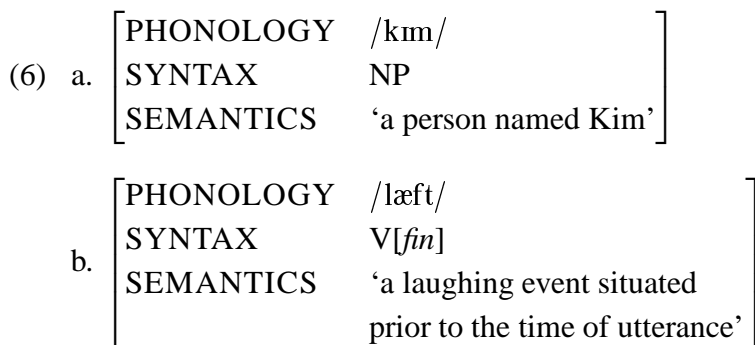
⁷Chomsky 1970, Jackendoff 1977, Kornai and Pullum 1990

⁸These are the distinctive features of the functional analysis of NPs proposed in Gazdar et al. 1985.

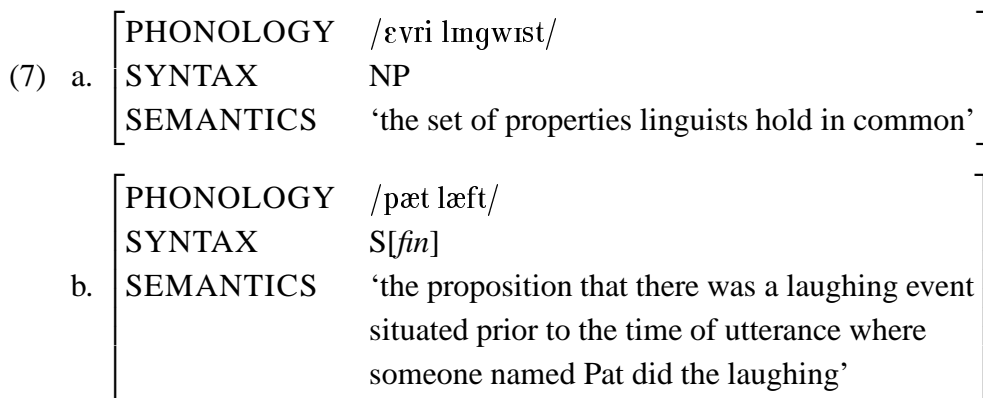
⁹Bresnan 1982, Gazdar et al. 1985, Pollard and Sag 1987, 1994, Carpenter 1992, Dalrymple et al. 1995, Richter 2004.



Signs, which are assumed to be the central objects of linguistic description, are less informally modeled as functions that specify a form, a meaning, contextual connections, and relevant syntactic information (including syntactic category and combinatoric potential). These functions can be represented in the form of attribute-value matrices, i.e. diagrams like the following:¹⁰



And, following work in HPSG (Pollard and Sag 1987, 1994), the notion of sign is extended to phrases, recognizing feature structures like (7) for complex linguistic expressions:



¹⁰The informal semantics assumed here is not meant to replace a more precise formulation.

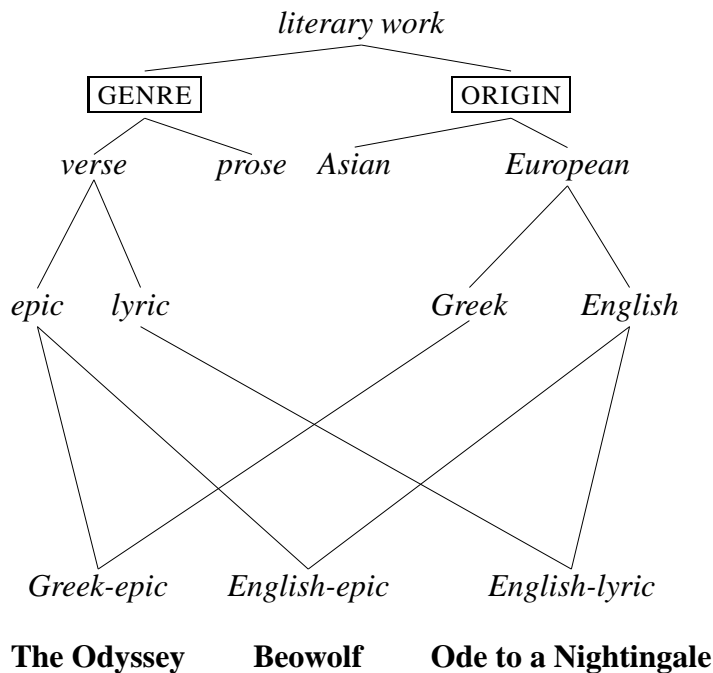
The non-atomic feature structures we use to model linguistic objects are **total** functions. That is, once an appropriate domain (a set of features) is established for a particular kind of feature structure, every feature structure of that kind assigns some appropriate value to every feature in that domain. The value assigned to any feature must also be a feature structure, hence that value is either an atom, or else it too is a function that must assign a value to every feature in *its* appropriate domain. A feature structure is thus always ‘complete’ in a simple, intuitive sense: every feature in a function’s domain is assigned a value in the appropriate range. Only atomic feature structures lack the property of containing features which require values. Thus, all feature structures can be thought of as ‘bottoming out’ with either atoms or indices.

It is important to notice that although feature structures themselves are complete, feature structure descriptions may be as partial as you like. This is crucial because most diagrams in this essay employ feature structure descriptions and partiality will be rampant. Lexical entries will be formulated as partial feature structure descriptions (typically being true of (or ‘satisfied by’) many feature structures), as will grammatical constructions of all kinds. Yet underlying all our concerns will be the set of feature structures that are specified by the theory we present. If some aspect of our theory goes awry, we should be able to figure out why by isolating certain (complete) feature structures that don’t satisfy the constraints of our theory, but should; or else other feature structures that shouldn’t satisfy our theory, but do. That is, an SBCG must neither overgenerate (by delimiting a set of feature structures that includes some that do not model expressions of the target language), nor undergenerate (by failing to provide descriptions of some feature structures that do model expressions of the target language).

Our feature structures have one more property that isn’t part of the basic theory of functions, as standardly presented – we assume that feature structures are organized in terms of a theory of linguistic **types**. A type is associated with a set of feature structures that have certain stated properties in common. One benefit derived from assigning feature structures to types is that we can thereby better organize the properties that classes of grammatical objects have and simplify their description in the process. Intuitively, it makes no sense (in English, anyway) to ask what case a verb has or whether a noun is an auxiliary noun; certain grammatical feature specifications are appropriate only for certain kinds of grammatical objects. This intuition is given formal expression in terms of the types that particular feature structures instantiate. Each feature structure instantiates a particular maximal (most specific) type. This type assignment, together with the general structure of the space of types, determines that the feature structure in question specifies values for a particular set of features and that each feature’s value is a particular kind of feature structure (possibly, a function of a particular type; possibly an atom, e.g. *nominative* or +).

The structure of the space of types is a hierarchy. In fact, the types are inter-related in terms of a **multiple inheritance hierarchy**. A type B **inherits from (is a subtype of)** another type A, if and only if the set of feature structures corresponding to B is a subset of the set of feature structures corresponding to A. In a *multiple* inheritance hierarchy, a type can inherit from more than one immediate supertype. That is, type hierarchies like (8) are allowed:

(8)



Multiple-inheritance hierarchies are useful for analyzing cross-classifying properties of a set of objects, whether they are literary works, words, or constructs.

In SBCG the type hierarchy takes over the inheritance functions that **constructional inheritance** did in some earlier traditions of construction grammar.¹¹ For example, Fillmore (1999) treats in terms of constructional inheritance the various syntactic environments and semantic interpretations that the subject-auxiliary inversion (SAI) pattern can appear in, as partially illustrated in the following:

- (9) a. **Has he left?**
- b. **Am I tired!**
- c. Never **will I harm you.**
- d. What **did Merle know?**
- e. **May you live long and prosper!**
- f. **Had he been on time,** he wouldn't have gone hungry.

In SBCG, the auxiliary-initial clausal pattern is declared as a type (of construct), and various constructions, such as those exemplified above, mention this type in the consequent clause (i.e. after the arrow). For example, the inverted-exclamative construction illustrated in (9b) has *inverted-exclamative-construct* (*inv-excl-cxt*) as its antecedent and its consequent makes mention of the more general construct type *auxiliary-initial-construct* (*ai-cxt*), as illustrated in (10):

¹¹See, for example, Fillmore et al. 1988, Fillmore and Kay 1996, Fillmore 1999, Kay and Fillmore 1999, Kay 2002a, Kay 2002b, Michaelis and Lambrecht 1996.

$$(10) \quad \text{inv-excl-cxt} \Rightarrow \begin{bmatrix} \text{ai-cxt} \\ \dots \end{bmatrix}$$

The type hierarchy, including the defining specifications of each type, is defined by a grammar's **signature**,¹² which includes:

1. a set of grammatical types, organized into a multiple inheritance hierarchy,
2. a set of features, and
3. appropriateness declarations, stating which features are appropriate for (feature structures of) each type and what type of value each feature must have. Since no feature appears in more than one type, both the specification of what features go with a specified type and values these features must take can be expressed in a single statement such as the following (explained more fully in section 4 below):¹³

$$\text{cxt:} \quad \begin{bmatrix} \text{MTR} & \text{sign} \\ \text{DTRS} & \text{list(signs)} \end{bmatrix}$$

All three components of a grammar signature may contain both universal and language particular elements. Together, these components define a space of well-formed feature structures. But only a subset of these are licensed by the **constructions** of the language (discussed in section 4 below), which, together with the lexicon, constitute the body of the grammar.

3 Signs

In the following sections, we introduce the specific features whose values serve to distinguish the signs of a language from one another.

3.1 PHONOLOGY and FORM

I will have little to say here about morphology, and nothing at all about phonology, but it is fully intend for phonological and morphological entities to be part of linguistic signs. I will display the sequences of (morphological) objects that analyses associate with the signs that are being discussed, leaving it to a largely autonomous set of constraints to characterize the relation between the phonological and morphological aspects of signs.

There are thus two distinct sign-level features: PHONOLOGY (PHON) and FORM:

¹²A grammar signature is so named by analogy with a musical key or time signature; it lays out the way in which the particulars of the grammar (or the musical piece) are to be interpreted. For a more precise presentation, see Pollard and Sag 1994 or, for a much more detailed and fully formalized presentation: Richter 2004. Sag, Wasow and Bender's (2003) textbook is a very accessible introduction to an English grammar with these components.

¹³The Appendix contains a summary of the appropriateness declarations assumed in this paper.

- (11) a. The value of the feature PHON is a phonological phrase, i.e. a feature structure of type ϕ -*phr(ase)*.
- b. The value of the feature FORM is a sequence of morphological objects (formatives); these are the elements that will be phonologically realized within the sign's PHON value.

Here I leave open the precise characterization of formatives, though we will assume that they include **lexical formatives** and **affixes**. As will be explained below, morphological functions will take as input both a formative and a lexeme identifier, the latter indicating the particular (fine-grained) meaning associated with the lexeme. This allows us to accommodate morphological operations that make such distinctions as the following:

- (12) a. *lie/lay/lain* 'rest, recline' vs. *lie/lie/lie* 'tell falsehoods'
- b. *can/could* 'be able to' vs. *can/canned* 'put into cans'
- c. *fly/flew* (basic sense) vs. *fly/flied* (various derived senses, e.g. in baseball)
- d. *sell/sold* vs. *cell/celled*
- e. *write/wrote/written* vs. *right/righted/righted*

Morphological functions provide a convenient way of expressing 'elsewhere' conditions in morphological realization. For present purposes, I will simplify our presentation of signs by substituting conventional orthography for lists of formatives.

3.2 ARGUMENT-STRUCTURE

The basic purpose of the ARGUMENT-STRUCTURE (ARG-ST) feature is to encode the combinatoric potential of a lexical expression by listing its potential syntactico-semantic **arguments**. For verbs, the order of elements on the ARG-ST list corresponds in the main to that of the 'Accessibility Hierarchy' of Keenan and Comrie (1977), where the first NP is the verb's subject, the second NP (if there is one) is the verb's direct object, etc. This 'rank-based' encoding of grammatical relations, as shown by Keenan and Comrie and over a quarter century of research in relation-based syntactic theory, is independently motivated by cross-linguistic generalizations (e.g. relative clause accessibility), as well as by rank-based phenomena (binding, 'advancements', etc.) internal to the grammars of many diverse languages. The rank-based encoding also eliminates the need for an inventory of features like SUBJECT, OBJECT, OBJ2 (SECOND-OBJECT), or COMP (COMPLEMENT) to name particular grammatical 'functions'. In a 'nominative-accusative' language like English, the verb's subject is identified both as its XARG member (see the discussion of the feature XARG in section 3.3.1 below) and as the first member of its ARG-ST list.¹⁴

¹⁴I assume, with Manning (1996) and Manning and Sag (1998), that in a syntactically ergative language, the verb's XARG member is identified with the second member of its ARG-ST list.

Variable polyadicity of a given lexeme, e.g. active vs. passive vs. middle verbs, causative vs. inchoative verbs, or oblique-argument vs. ditransitive verbs, involves differences in the ARG-ST list. These differences can come about in two distinct ways in a SBCG: by derivational construction (as in passivization) or by lexical underspecification (as in so-called *spray/load* alternations).

A lexical item like *donate*, which is a transitive verb, has an ARG-ST list with three members:¹⁵

(13) $\langle \text{NP}, \text{NP}, \text{PP} \rangle$

Lexemes, especially verbal lexemes (see below), fall into diverse classes, as determined in part by the length of their ARG-ST list and the constraints imposed on particular arguments. Only lexical signs (lexemes or words) specify a value for ARG-ST, as guaranteed by the appropriateness declarations of the grammar signature.

ARG-ST lists are also the locus of the constraints of binding theory.¹⁶ For example, a reflexive on an ARG-ST list must be coindexed with a preceding element, if there is one (Principle A); personal pronominals must not be (Principle B).

3.3 SYNTAX

The values of the feature SYNTAX are feature structures of type *syntax-object* (*syn-obj*). These are functions that specify values for the three features CATEGORY, VALENCE, and MARKING, which we discuss in turn.

3.3.1 CATEGORY

The values of the feature CATEGORY are complex grammatical categories, treated here as feature structures of type *category* (*cat*).¹⁷ The various subtypes of *cat* will specify values for appropriate features. For example, the signature of the grammar of English assumed here includes the following information:

(14) a. The immediate subtypes of the type *category* are: *noun*, *verb*, *preposition* (*prep*), *adjective* (*adj*), . . .

¹⁵Some abbreviations:

$$\text{NP} = \left[\text{SYN} \begin{bmatrix} \text{CAT} & \textit{noun} \\ \text{VAL} & \langle \rangle \end{bmatrix} \right] \quad \text{PP} = \left[\text{SYN} \begin{bmatrix} \text{CAT} & \textit{prep} \\ \text{VAL} & \langle \rangle \end{bmatrix} \right]$$

$$\text{CP} = \left[\text{SYN} \begin{bmatrix} \text{CAT} & \textit{comp} \\ \text{VAL} & \langle \rangle \end{bmatrix} \right] \quad \langle \rangle = \text{the empty list.}$$

¹⁶This follows a tradition that begins with the Relational Grammar proposals of Johnson (1977). See also Pollard and Sag (1992, 1994) and Manning and Sag (1998).

¹⁷Note that ‘CATEGORY’ denotes a feature and *category* denotes a type. Features are represented in CAPITALS and types in *lower case italics*. The common supertype of the types that can serve as values for the feature CATEGORY could have been named something different, e.g. *part-of-speech*.

- b. CASE is used to distinguish the cases of nouns; the possible values of CASE (in English) are *nominative (nom)* and *accusative (acc)* and *oblique (obl)*.¹⁸
- c. VFORM (VF) is used to specify the morphosyntactic category of a verb; the possible values of VF are *finite (fin)*, *base*, *present-participle (prp)*, *past-participle (psp)*, and *passive-participle (pas)*.
- d. AUXILIARY (AUX) is used to specify whether a verb is also an auxiliary; the possible values of AUX are + and –.

This partial signature countenances complex grammatical categories like those shown in (15), but none like the ones pictured in (16):¹⁹

$$\begin{array}{l}
 (15) \quad \text{a. } \left[\left[\begin{array}{l} \textit{noun} \\ \text{CASE} \quad \textit{nom} \\ \dots \end{array} \right] \right] \quad \text{b. } \left[\left[\begin{array}{l} \textit{verb} \\ \text{VF} \quad \textit{fin} \\ \text{AUX} \quad + \\ \dots \end{array} \right] \right] \quad \text{c. } \left[\left[\begin{array}{l} \textit{verb} \\ \text{VF} \quad \textit{prp} \\ \text{AUX} \quad - \\ \dots \end{array} \right] \right] \\
 \\
 (16) \quad \text{a. } * \left[\left[\begin{array}{l} \textit{noun} \\ \text{VF} \quad \textit{fin} \\ \dots \end{array} \right] \right] \quad \text{b. } * \left[\left[\begin{array}{l} \textit{verb} \\ \text{AUX} \quad - \\ \text{CASE} \quad \textit{nom} \\ \dots \end{array} \right] \right] \quad \text{c. } * \left[\left[\begin{array}{l} \textit{noun} \\ \text{AUX} \quad - \\ \text{VF} \quad \textit{prp} \\ \dots \end{array} \right] \right]
 \end{array}$$

I should make clear that I am using attribute-value matrix (AVM) notation to formulate our feature structure **descriptions**. The objects these formulas describe are **functions** of the appropriate kind. For example, a nominal category is a function whose domain includes CASE, but not AUX or VF, while a verbal category is a function whose domain includes AUX and VF, but not CASE. Note, that whenever a particular feature structure is being illustrated, rather than a description, double outermost brackets are used, as in (15)–(16).

Lexical descriptions are typically minimal, specifying perhaps a FORM value, a syntactic category and a meaning. But the set of possible feature structures that are licensed by any given lexical description is circumscribed by the constraints of the grammar signature, which require that certain feature must have a value and that the value must have certain properties. For example, the lexical entry licensing the proper noun *Dale* says nothing about the value of

¹⁸The *obl* CASE value is not overtly distinguished from *acc*. But the distinction between the two CASE values is useful for distinguishing direct objects from other nominal arguments. Note that genitive nominal expressions are not distinguished in terms of CASE. This is because case is a property of head nouns and the Modern English 's is a phrasal clitic that appears in final position of a genitive NP, rather than as an inflection on the head noun:

- (i) [[The man on the radio's] voice] . . .
- (ii) *[[The man's on the radio] voice] . . .

Genitive NPs will be treated more fully in work now in progress.

¹⁹[[Doubled brackets]] are henceforth used to display feature structures, i.e. objects of the model; [single brackets] will be used to display descriptions, objects in the grammar, including constructions.

the feature CASE. But any given occurrence of the proper noun *Dale* is modeled by a feature structure where the CASE value is resolved. In *Dale left*, it is resolved as nominative; in *Find Dale!*, it is resolved as accusative.

Although the functional entities in our models are always total functions, essentially all of the business of grammar is conducted in terms of descriptions of about classes of these entities that are partial (or underspecified). For example, the descriptions in (17) describe more than one feature structure, and hence underspecify the indicated ambiguities:

- (17) a. [FORM < I , forgot , how , good , beer , tastes >]
 ([[*how good*] [*beer tastes*]] vs. [*how* [[*good beer*] *tastes*]])
- b. [FORM < flying , planes , can , be , dangerous >]
 (*flying* is an adjective modifying *planes* or else a gerund whose object is *planes*)
- c. [PHON [ϕ -phrase
 /aylbili:vinyu:/]]
 (*I'll believe in you* vs. *I'll be leavin' you*)

There are three other CAT features that need to be introduced:²⁰

- (18) a. SELECT is used to let an expression select what it can modify or combine with as a 'marker'. The value of SELECT is a possibly empty list of signs. If an expression's SELECT value is nonempty, then it is either a modifier (adjective, adverb, etc.) or a 'marker' (complementizer, determiner, etc.).
- b. EXTERNAL-ARGUMENT (XARG) is used to specify the argument of an argument-taking expression that is visible from outside its local domain (i.e. from outside the phrase it projects). The value of XARG is either a sign or the distinguished atom *none*.²¹ The external argument of a clause is its subject; an NP's external argument is its prenominal genitive NP, if there is one (the XARG value of the NP is *none*, otherwise).

²⁰The feature SELECT was originally proposed by Van Eynde (1998) as a generalization of the two features MOD and SPEC that were employed by Pollard and Sag (1994). See also Van Eynde 2003, 2004, 2006. The fundamental insights of the SELECT analysis here are indeed those of Van Eynde, despite minor differences of execution that might seem to indicate otherwise. For example, Van Eynde follows the basic feature inventory and more complex feature geometry of Pollard and Sag, which I have been concerned with streamlining, e.g. by eliminating the features HEAD and LOCAL. Similarly, the reason that SELECT is treated as list-valued is to provide more uniformity in the treatment of constraints than Pollard and Sag were able to achieve.

²¹Sag and Pollard (1991), who first introduced this feature assumed its value is an index, an assumption preserved in ongoing computational grammar development work using the English Resource Grammar and the Grammar Matrix. For more information, see <http://lingo.stanford.edu/> and <http://www.delph-in.net/>. See also Copestake et al. 2005.

- c. LEXICAL-ID (LID) is used to individuate lexical items semantically; the value of LID is either a frame specifying the (fine-grained) meaning of a lexeme (e.g. *book-frame*) or else the distinguished atom *none*.

The features SELECT and XARG are discussed in more detail in section 4 below.

3.3.2 VALENCE

The basic function of the feature VAL(ENCE) is to encode the degree of saturation of any linguistic expression, i.e. which of its arguments it has yet to combine with syntactically. VAL is thus closely related to the feature ARG-ST – these features both take a possibly empty list of signs as their value.

A lexeme’s VAL list, whose members I will refer to as the lexeme’s **valents**, is exactly the same as its ARG-ST list, as long as no covert or nonlocal argument realization takes place. Although phrases have no ARG-ST in SBCG, a verb phrase like *persuaded me to go*, which contains as constituents all but the first of the verb’s syntactico-semantic arguments (i.e. its subject), has the following singleton VAL list:

(19) ⟨ NP ⟩

Similarly, the clause *My dad persuaded me to go*, which contains all the verb’s syntactico-semantic arguments, has an empty VAL list:

(20) ⟨ ⟩

The lexical head of the clause is the verb, and the phrases that it projects gradually ‘saturate’ the verb’s VAL list by ‘removing elements from it’.²² Clauses, NPs, pronouns, and proper names have an empty VAL list because they are already saturated, i.e. they need not, indeed cannot combine with subjects or complements.

Discrepancies between a lexeme’s ARG-ST list and its VAL list can arise in several ways, e.g. via long-distance dependencies, which will not concern us here. Of direct relevance, however, is the phenomenon of **null instantiation**, which arises when a lexeme undergoes a derivational construction that ‘removes’ an element from a lexeme’s VAL list.²³

²²This way of looking at things, which has its origin in the argument cancellation of Categorical Grammar (Ajdukiewicz 1935; see also http://en.wikipedia.org/wiki/Categorical_grammar) involves a ‘bottom-up’ procedural metaphor where one starts with a predicator and builds successively larger phrases of which that predicator is the lexical head. It is important to recognize, however, that the constructions of an SBCG, like the rules of a Context-Free Phrase Structure Grammar, are static constraints defining well-formed local tree configurations. The importance of this fundamental design property has been discussed by many over the years, including Kaplan and Bresnan 1982, Sag et al. 1986, Fenstad et al. 1987, Jackendoff 1997, 2002, Pullum and Scholz 2005, and Sag and Wasow to appear.

²³In many varieties of the Romance languages, clitic pronouns have been reanalyzed as inflectional affixes (see Miller and Monachesi 2003). This leads to special kind of null instantiation, where ARG-ST elements are realized morphologically, rather than syntactically. For a detailed analysis of this phenomenon in French, broadly compatible with the framework developed here, see Miller and Sag 1997.

3.3.3 MARKING

The feature MARKING (MRKG), introduced by Pollard and Sag (1994) and refined in crucial ways by Van Eynde,²⁴ is used to distinguish expressions like *that Kim laughed*, *whether Morgan was awake*, and *Hillary's* from their respective 'unmarked' counterparts *Kim laughed*, *Morgan was awake*, and *Hillary*. The MRKG value is *unmarked* (*unmk*) in the case of all unmarked signs, but I will assume various other MRKG values, such as those in (21):

- (21) *that* *that*-clauses, e.g. *that Kim laughed*
whether *whether*-clauses, e.g. *whether Morgan left, whether to leave*
than compared phrases, e.g. *than in Rome, than Pat*
as equated phrases, e.g. *as in Rome, as I could*
of some *of*-phrases, e.g. *of Rome*
det 'determined' nominal signs, i.e. *the table, Prince, we*

Some prepositions also lead a life as markers, as in the case of *than* and *as* (Hankamer 1973) and certain uses of *of*.²⁵

An element that specifies a MRKG value other than *unmk* is called a 'marker'; all such elements also specify a nonempty value for the feature SELECT. Not all marked phrases, however, contain such an element, for example genitive NPs, proper nouns, and pronouns are all marked *det*. The MRKG value of a marker is passed up to its mother via constraints on the Head-Functor Construction introduced in section 4 below. The marking value of a head is passed up to its mother via constraints on the Head-Complement Construction.

Note that the feature inventory introduced here allows variable-grain category description. That is, we may describe traditional categories (NP, VP, S, etc.) in terms of the descriptions in (22):

- (22) $\left[\begin{array}{l} \text{CAT} \text{ } verb \\ \text{VAL} \text{ } \langle \rangle \end{array} \right] \text{ (S)}$
- $\left[\begin{array}{l} \text{CAT} \text{ } verb \\ \text{VAL} \text{ } \langle \text{NP} \rangle \end{array} \right] \text{ (VP)}$
- $\left[\begin{array}{l} \text{CAT} \text{ } noun \\ \text{VAL} \text{ } \langle \rangle \\ \text{MKG} \text{ } det \end{array} \right] \text{ (NP)}$

But in addition, it is possible to describe very fine-grained categories, e.g. (23), which is the category of any determined NP whose head noun is the idiomatic lexical item *advantage* (the only lexeme whose LID value is *advantage₂-frame*):

²⁴See Van Eynde 2003, 2004, 2006. Van Eynde's MARKING values are more complex than those assumed here, in large part because of his need to analyze complex morphological and agreement patterns absent in English. I leave open the possibility of modifying the theory of MARKING to incorporate further of Van Eynde's insights.

²⁵See Abeillé et al. 2006 for an analogous distinction in French.

$$(23) \left[\begin{array}{l} \text{CAT} \left[\text{LID } \textit{advantage}_2\text{-frame} \right] \\ \text{VAL} \langle \rangle \\ \text{MKG } \textit{det} \end{array} \right]$$

A structure like that shown in (23) can then appear as a valent of a particular lexeme (*take*, accounting for the range of expressions that includes *take advantage of NP*, *take unfair/egregious/maximum/the very fullest*,... *advantage of NP*, as well as passive versions like *Fullest advantage was not taken of NP*).

Multigrain descriptions of this kind are a central feature of Construction Grammar, enabling grammatical descriptions that ‘scale up’ to deal with large data sets that include the full range of structures from the most idiomatic to the most productive in an internally consistent manner (Fillmore et al. 1988, Kay and Fillmore 1999). Mainstream generative grammar has failed to treat, indeed, has given the appearance of being uninterested in this problem (See, for example, Chomsky 1995).

3.4 SEMANTICS

For present purposes, it will suffice to introduce the following two features of our semantic objects which serve as the values of the feature SEM:

- (24) a. INDEX is used to individuate the referent of an expression. Its value is an index, essentially a variable assigned to an individual (in the case of an NP) or a situation (in the case of a VP or clause).²⁶
- b. FRAMES is used to specify the predications that together determine the meaning of a sign. The value of FRAMES is a (possibly empty) list of frames.

A frame may be understood as an elementary scene in which certain roles are specified and particular participants are assigned to them. For example, in an eating frame the participants are an actor, who does the eating, and the food, which gets eaten. Representing frames in the form of feature structures, each role is denoted by a feature and the corresponding participant by an index. In addition, there is often an event index, which denotes the entire scene. The frame as a whole is represented as a feature structure *type* and the frames are inter-related as one branch of the multiple inheritance type hierarchy.²⁷ In the case of most (if not all) verbs, an event index will be encoded as a SITUATION (SIT) feature, whose value is a situational index, and there will often be an ACTOR feature, whose value is an individual index, as shown in (25):²⁸

²⁶Event nominals will probably also require treatment in terms of situational indices.

²⁷Following the approach to semantic relations explored first by Pollard and Sag 1994: Ch.8) and further developed by Davis (2001)).

²⁸This corresponds to a Davidsonian event variable, which is, roughly, a referential index pointing to the entire scene described by an elementary predication.

$$(25) \left[\begin{array}{l} \textit{sem-obj} \\ \text{INDEX} \quad s \\ \text{FRAMES} \quad \left\langle \left[\begin{array}{l} \textit{laugh-fr} \\ \text{ACTOR} \quad i \\ \text{SIT} \quad s \end{array} \right] \right\rangle \end{array} \right]$$

The SEM value of a proper noun like *Pat* will be a naming frame, with specifications for the features NAME and NAMED, as sketched in (26):

$$(26) \left[\begin{array}{l} \textit{sem-obj} \\ \text{INDEX} \quad i \\ \text{FRAMES} \quad \left\langle \left[\begin{array}{l} \textit{name-fr} \\ \text{NAME} \quad \text{Pat} \\ \text{NAMED} \quad i \end{array} \right] \right\rangle \end{array} \right]$$

It is important to understand that SBCG is in fact compatible with most approaches to semantic analysis. For a discussion of filler-gap constructions couched in SBCG that uses a more familiar model-theoretic conception of English semantics, see Sag to appear a.

3.5 CONTEXT

I will have relatively little to say about CONTEXT (CNTXT) values here. For the sake of completeness, the reader might want to envisage a theory of CONTEXT values (feature structures of type *context*) like the one developed by Pollard and Sag (1994),²⁹ based on such features as CONTEXTUAL-INDICES (C-INDS) and BACKGROUND (BCKGRND), and UTTERANCE-LOCATION (UTT-LOC). The context-objects in their theory look like (27):

$$(27) \left[\begin{array}{l} \textit{context} \\ \text{C-INDS} \quad \left[\begin{array}{l} \text{SPEAKER} \quad \textit{index} \\ \text{ADDRESSEE} \quad \textit{index} \\ \text{UTT-LOC} \quad \textit{index} \\ \dots \end{array} \right] \\ \text{BCKGRND} \quad \textit{set}(\textit{proposition}) \end{array} \right]$$

The various contextual indices specify contextual elements that ground an account of indexical and deictic expressions formulated in the style of David Kaplan's (1989) seminal work. The propositions specified in the BCKGRND value correspond to the set of utterance felicity conditions, which any part of the utterance sign may in principle contribute to. That is, as a

²⁹See also Green 1996, Green 2000; Ginzburg and Sag 2000, Ginzburg and Cooper 2004, Kay 1997, Fillmore 1985.

mother sign is constructed from a sequence of daughter signs, its BCKGRND must include all elements in the BCKGRND sets of the daughter signs.

It is of course possible to augment these assumptions about contextual features, incorporating, for example, features like TOPIC and/or FOCUS, as in Michaelis and Lambrecht 1996. Engdahl and Vallduvi (1994, 1996) propose an analysis of ‘information packaging’ formulated in terms of the CONTEXT values structured as shown in (28):

$$(28) \left[\begin{array}{l} \textit{context} \\ \text{INFO-STRUCTURE} \left[\begin{array}{l} \text{FOCUS} \quad \dots \\ \text{GROUND} \left[\begin{array}{l} \text{LINK} \quad \dots \\ \text{TAIL} \quad \dots \end{array} \right] \end{array} \right] \end{array} \right]$$

I will not explore such elaborations here.³⁰

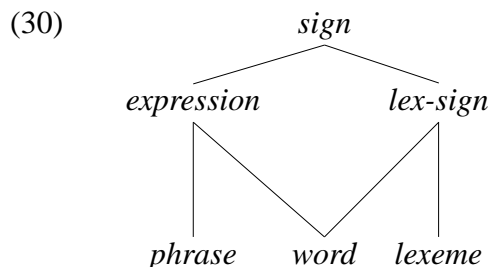
3.6 Signs: A Synthesis

Signs are analyzed as feature structures that specify values for the five features PHON, FORM, ARG-ST, SYN, SEM, and CONTEXT, whose values have now all been introduced:

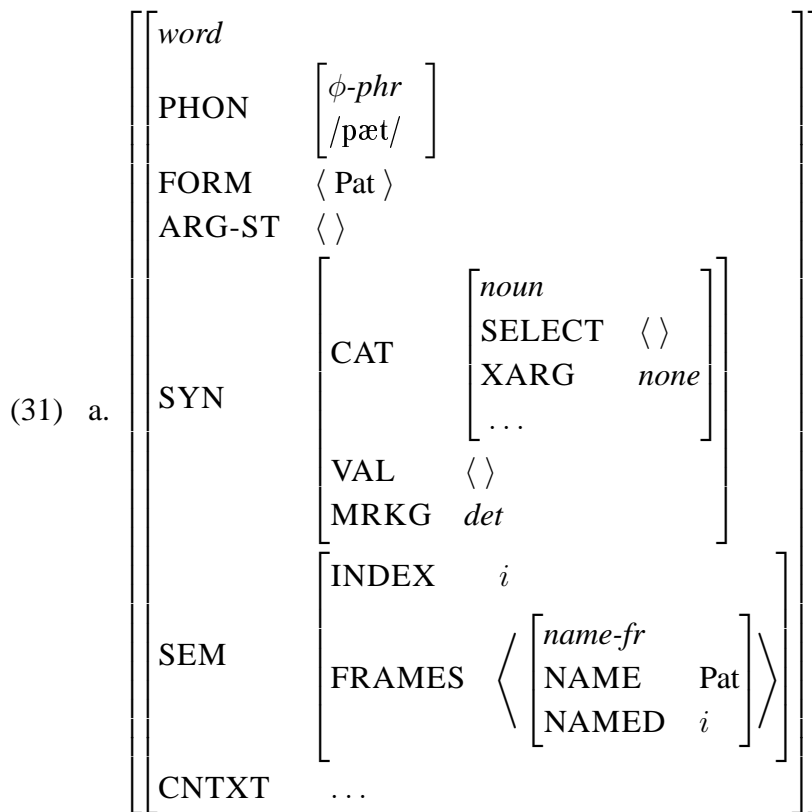
$$(29) \left[\begin{array}{l} \textit{sign} \\ \text{PHONOLOGY} \quad \textit{phonological-phrase} \\ \text{FORM} \quad \textit{list(formative)} \\ \text{ARG-ST} \quad \textit{list(sign)} \\ \text{SYNTAX} \quad \textit{syn-obj} \\ \text{SEMANTICS} \quad \textit{sem-obj} \\ \text{CONTEXT} \quad \textit{context} \end{array} \right]$$

The immediate subtypes of *sign* are *lexical-sign* (*lex-sign*) and *expression*. The immediate subtypes of *expression* are *word* and *phrase*, while those of *lexical-sign* are *word* and *lexeme*. The supertype relations of *word* thus reflect the fact that words share properties with phrases that lexemes lack (e.g. the ability to be the daughter of a phrasal construct, discussed in the next section) and that words share properties with lexemes that phrases lack (e.g. having an ARG-ST list). These cross-cutting properties of words are accommodated by treating feature structures of this type as both a kind of expression and a kind of lexical sign. The resulting **multiple-inheritance hierarchy** is sketched in (30):

³⁰Nor will we explore the ways in which a set of background propositions may be structured by relations such as unilateral entailment, as in a scalar model (Fillmore et al. 1988, Kay 1990, Israel 1996, Schwenter 1999, Schwenter and Vasishth 2000). Additionally, some contextual propositions may be mentioned in a sign type or construction, not for acceptance of their content’s being a felicity condition on the utterance, but as ‘context propositions’, whose content is acknowledged as being ‘on the floor’, although not necessarily accepted – perhaps specifically denied – by a conversational participant. See, for example Kay 1997.



We are now in a position to illustrate in more detail what the various signs introduced earlier will look like in SBCG. (Recall that double brackets indicate a function – i.e., an item of the language, rather than a function description – i.e. an item of the grammar.) The following diagrams display most of the main properties of the feature structures corresponding to the words *Pat*, *laughed*, and *laugh*:^{31,32}



³¹Further abbreviations:

$NP_i = NP \ \& \ [SEM|INDEX \ i]$ $NP[nom] = NP \ \& \ [SYN|CAT|CASE \ nom]$

³²The PHON values here are schematic, simply indicating a phonological phrase (*phi-phr*) includes a given phoneme sequence. The phonological phrase could also be modeled in terms of tree structure. See Bird and Klein 1994, Klein 2000, Asudeh and Klein 2002, and Haji-Abdolhosseini 2003.

b.

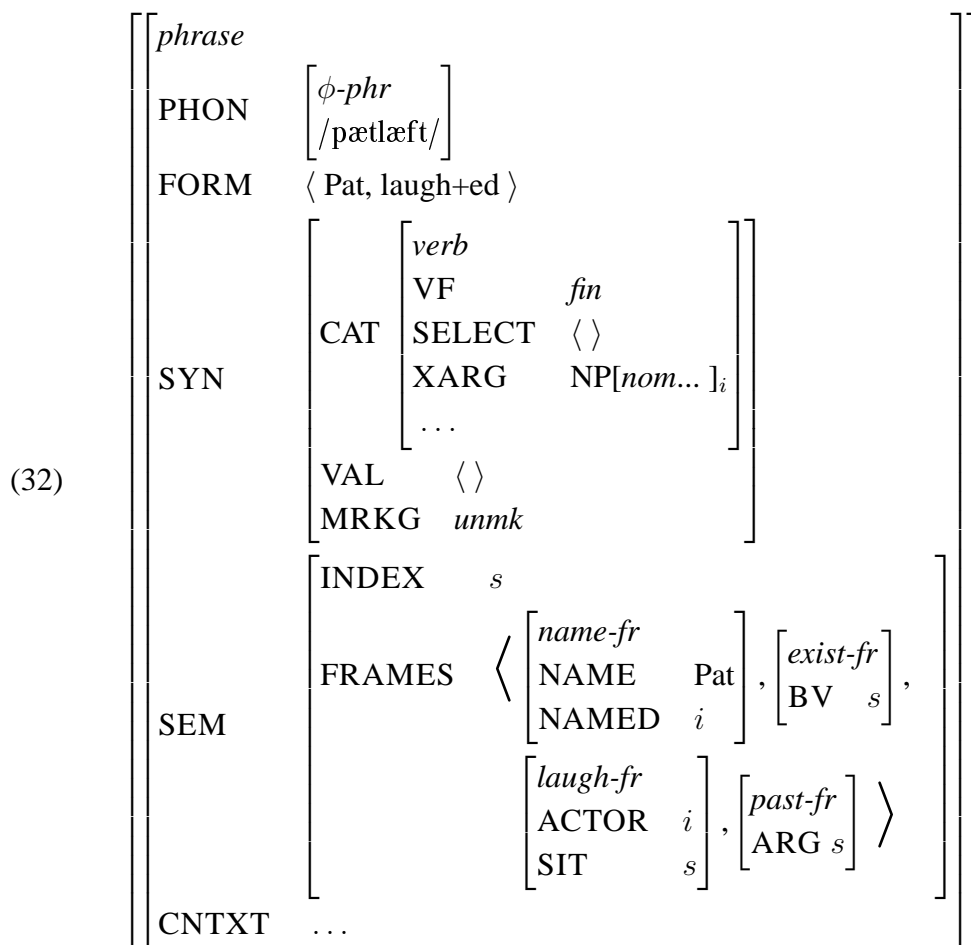
	<i>word</i>	
PHON	$\left[\begin{array}{l} \phi\text{-}phr \\ /læft/ \end{array} \right]$	
FORM	$\langle \text{laugh+ed} \rangle$	
ARG-ST	$\langle \text{NP}[\textit{nom}]_i \rangle$	
SYN	$\left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \textit{verb} \\ \text{VF} \quad \textit{fin} \\ \text{SELECT} \quad \langle \rangle \\ \text{XARG} \quad \text{NP}[\textit{nom}\dots]_i \\ \dots \end{array} \right] \\ \text{MRKG} \quad \textit{unmk} \\ \text{VAL} \quad \langle \text{NP}[\textit{nom}\dots]_i \rangle \end{array} \right]$	
SEM	$\left[\begin{array}{l} \text{INDEX} \quad s \\ \text{FRAMES} \quad \left\langle \left[\begin{array}{l} \textit{exist-fr} \\ \text{BV} \quad s \end{array} \right], \left[\begin{array}{l} \textit{laugh-fr} \\ \text{ACTOR} \quad i \\ \text{SIT} \quad s \end{array} \right], \left[\begin{array}{l} \textit{past-fr} \\ \text{ARG} \quad s \end{array} \right] \right\rangle \end{array} \right]$	
CNTXT	...	

c.

	<i>lexeme</i>	
PHON	$\left[\begin{array}{l} \phi\text{-}phr \\ /læf/ \end{array} \right]$	
FORM	$\langle \text{laugh} \rangle$	
ARG-ST	$\langle \text{NP}[\dots]_i \rangle$	
SYN	$\left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \textit{verb} \\ \text{SELECT} \quad \langle \rangle \\ \text{XARG} \quad \text{NP}[\dots]_i \\ \dots \end{array} \right] \\ \text{MRKG} \quad \textit{unmk} \\ \text{VAL} \quad \langle \text{NP}[\dots]_i \rangle \end{array} \right]$	
SEM	$\left[\begin{array}{l} \text{INDEX} \quad s \\ \text{FRAMES} \quad \left\langle \left[\begin{array}{l} \textit{laugh-fr} \\ \text{ACTOR} \quad i \\ \text{SIT} \quad s \end{array} \right] \right\rangle \end{array} \right]$	
CNTXT	...	

Note that the feature structure in (31c) must have a determinate value for every appropriate feature, including VF. In fact, any choice of VF value satisfies the requirements imposed by the lexical entry for *laugh*.

The sign corresponding to the (sentential) phrase *Pat laughed* is given below. It should be noted that the diagram presents a single *sign*, which by definition has no daughters; it does not show a construct, which corresponds to a local tree. Our model of a phrase or sentence is a single sign, which carries all the linguistically relevant information about that phrase or sentence. Trees come into the picture only as record of the derivational history.



I will return to the various syntactic and semantic details presupposed here in the process of presenting the theory in which these signs are embedded.³³ At the risk of tedium, let me repeat that the objects of our grammatical description are not trees; nor are they sets of trees put into correspondence by constraints or transformations. Rather, signs are constructed by a single, recursive process that builds words from one or more lexemes and phrases (phrasal signs) from one or more expressions. Each step of this process of ‘construction’ (which is representable as

³³Note that since feature structures are total functions, it follows that all signs have a value for FORM, SYN, SEM, and CONTEXT. Similarly, all lexical signs must in addition specify a value for ARG-ST. Crucially, however, a lexical or phrasal description may specify minimal information.

$$(34) \left[\begin{array}{l} \textit{phrase} \\ \text{PHON} \left[\begin{array}{l} \phi\text{-}phr \\ /pætto:/ \end{array} \right] \\ \text{FORM} \langle \text{Kim, the} \rangle \\ \text{SYN} \left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \textit{verb} \\ \text{VF} \quad \textit{fin} \\ \text{SELECT} \quad \langle \rangle \\ \text{XARG} \quad \text{NP}[\textit{nom...}]_j \\ \dots \end{array} \right] \\ \text{VAL} \quad \langle \rangle \\ \text{MRKG} \quad \textit{that} \end{array} \right] \\ \text{SEM} \left[\begin{array}{l} \text{INDEX} \quad s \\ \text{FRAMES} \left\langle \left[\begin{array}{l} \textit{name-fr} \\ \text{NAME} \quad \text{Bo} \\ \text{NAMED} \quad i \end{array} \right], \left[\begin{array}{l} \textit{sneeze-fr} \\ \text{ACTOR} \quad i \\ \text{SIT} \quad s \end{array} \right], \left[\begin{array}{l} \textit{past-fr} \\ \text{ARG} \quad s \end{array} \right] \right\rangle \end{array} \right] \\ \text{CNTXT} \quad \dots \end{array} \right]$$

Given what has been said so far, (34) is a well-formed feature structure of type *sign*. Each feature (PHON, FORM, SYN, SEM, and CNTXT) has a value of an appropriate type and each of these values is a function that also specifies well-formed values for all and only the appropriate features, and so forth. However, even though this is the case, the problems with the feature structure in (34) are numerous:

- (35) a. This is a finite clause whose FORM value is $\langle \text{Kim} , \text{the} \rangle$, yet *Kim the* is not a well-formed English clause..
- b. The FORM value $\langle \text{Kim} , \text{the} \rangle$ has been phonologically realized as $/pætto:/$.
- c. The meaning of the sentence is (roughly) that a person named Bo sneezed at some time in the past, yet that meaning cannot be expressed by uttering *Kim the*.

...

Clearly, we need more grammar than just the inventory of types and the feature declarations that are specified in the grammar's signature. Unwanted signs like (34) are ruled out by populating the grammar with an inventory of **constructions** – and a general principle requiring well-formed signs (other than lexemes or unanalyzable words) to be licensed by some (combinatoric) construction. I will follow tradition and distinguish the combinatoric constructions from the **lexicon**, which is a set of lexical entries, each of which is a description of a family of lexical signs.

A construction in SBCG is not simply “any conventionalized pairing of form and meaning”, as assumed by Goldberg (1995: 4) and much previous work in Construction Grammar. In SBCG, a combinatoric construction specifies a class of constructs – feature structures that contain a mother sign and a list of daughter signs (that is, local trees with signs at the nodes). To be sure, these constructions will induce conventionalized pairings of form and meaning in the mother sign, but the mechanism by which they achieve this is circumscribed with particular reference to mothers and their daughters, in a way described below. And lexical-class constructions introduce no structure at all; hence, SBCG is strictly local.³⁵

Signs in SBCG are licensed by a general grammatical principle which can be formulated as follows:

(36) **The Sign Principle:**

Every sign must be lexically or constructionally licensed, where:

- a. a sign is lexically licensed only if it satisfies some lexical entry, and
- b. a sign is constructionally licensed only if it is the mother of some construct.

This principle of course presupposes that feature structure well-formedness must involve all the other constraints of the grammar, in particular those of the grammar signature. The Sign Principle thus specifies a further condition that must be satisfied by feature structures that are of type *sign*, i.e. lexemes, words, and phrases. The goal of the next two sections is to lay out the assumptions that underlie this simple formulation of grammatical theory.

4.1 Lexical Constructions and Lexical Entries

The properties of a word – a verb, say – are partly determined by a lexical entry (a lexeme-description in the lexicon), partly by lexeme-class constructions, and in addition by one or more of a set of derivational and inflectional constructions. Following tradition, derivational constructions allow lexemes to be formed from other lexemes; inflectional constructions allow words to be constructed from lexemes.³⁶ The morphological base, semantic frame, and valence list of *laughed*, for example, are specified in the lexical entry for the lexical item *laugh* (on which, see below), but its inflected form, and the added constraints that (i) its VF value must be *fin* and (ii) the CASE value of its subject (its first, and only, ARG-ST member) must be *nom*, are determined by the preterite construction, one of a handful of inflectional constructions that survive in the grammar of Present-Day English.

³⁵It should also be noted at this point that signs and constructs are not the only objects in an SBCG model of a language. Signs are the most complex kind of feature structure, but they exist side by side with their simpler counterparts, such as frames, semantic-objects, categories, and even atoms. For a more detailed inventory of the types of feature structure that an SBCG might assume, see the Appendix.

³⁶Our approach to morphology here is realizational (Matthews 1991, Anderson 1992), perhaps closest in spirit to the approach developed by Stump (2001) and related work. Lexical affixes are not signs; rather affixation effects (as well as more complex morphological processes) are determined by the morphological functions associated with specific lexical constructions.

Derivational and inflectional constructions fit uniformly into a two-level mode, one that is articulated in terms of a mother and its daughter(s).³⁷ For example, the verbal word whose form is *laughed* is constructed from the verbal lexeme whose form is *laugh*, in accordance with the Preterite Construction. The resulting mother-daughter configuration is a **construct**.

In order to express constructional generalizations in a systematic way, it is useful, as indicated above, to model constructs as feature structures of the form sketched in (37):

$$(37) \quad \left[\begin{array}{ll} \text{MTR} & \textit{sign} \\ \text{DTRS} & \textit{nelist}(\textit{sign}) \end{array} \right]$$

MOTHER (MTR) is used to specify the sign that is constructed in a given construct; the value of MTR is a sign. The feature DAUGHTERS (DTRS) specifies the more basic sign(s) which must be licensed in order for the mother to be; the value of DTRS is a nonempty list (*nelist*) of signs. It is feature structures of this kind that are licensed by particular combinatoric constructions. These licensed constructs in turn give rise to licensed signs, as per the Sign Principle in (36).³⁸

What then is a construction? In SBCG, a construction is a constraint defining the properties that are common to all instances of a given feature structure type. That is, a construction is a constraint of the form shown in (38), where each τ is a type name and D is a description that all instances of τ must satisfy:

$$(38) \quad \tau \Rightarrow D$$

Each combinatoric construction licenses a grammatically distinct class of constructs.

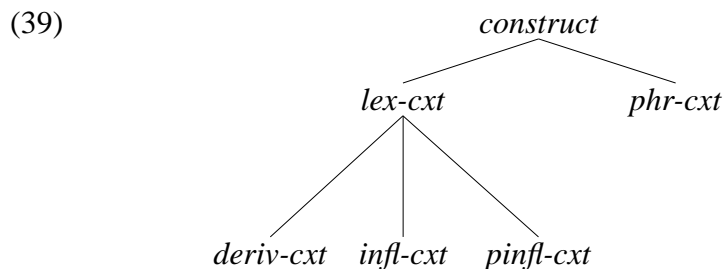
The immediate subtypes of *construct* are *lexical-construct* (*lex-cxt*) and *phrasal-construct* (*phr-cxt*). Lexical constructs (following Sag et al. (2003, Ch.16)) are further classified in terms of the subtypes *derivational-construct* (*deriv-cxt*), *inflectional-construct* (*infl-cxt*), and *postinflectional-construct* (*pinfl-cxt*).³⁹ This hierarchy of construct types is sketched in (39):

³⁷The approach to lexical constructions adopted here is based on ideas developed originally (to the best of our knowledge) by Copestake 1992. See also Orgun 1996, 2000, Sag, Wasow and Bender 2003: Ch.16, and Koenig 1999, among others.

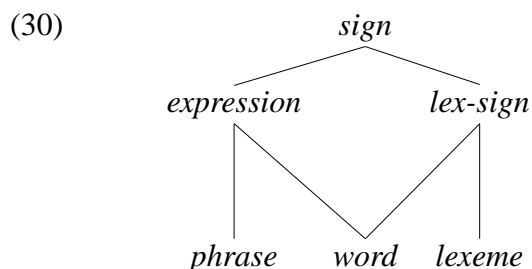
³⁸The terms ‘construct’ and ‘construction’ thus have a meaning here that is distinct from the way these terms have been used in the tradition of BCG. In this body of research, a construct was any fleshed-out ‘feature structure tree’ (See Kay 2002b) – of any degree of configurational complexity, including single node feature structure trees. In the present approach, a linguistic object – word, phrase or sentence – is modeled by a single sign, not by a derivational tree (which is the rough equivalent in the present architecture of a feature structure tree in BCG). In addition, the notion ‘construct’ is restricted to local trees and constructions are correspondingly restricted either to licensing constructs so localized or to licensing lexeme classes.

A construction is the grammatical constraint (analogous to a ‘rule’) that licenses a particular set of constructs. The major theoretical changes, are thus: (1) replacing trees-with-feature-structures-at-the-nodes (feature structure trees) with feature structures simpliciter and (2) imposing locality on constructions and constructs by making each in its respective domain correspond intuitively to a local tree.

³⁹For an alternative view, see Runner and Aranovich 2003.



The lexical signs (but not the phrases) depicted in (30) and repeated, function as the mothers and daughters of lexical constructs.



That is, lexical constructs, which are discussed in the remainder of this section, are constrained by the following type declaration, specified in the grammar's signature:

(40) $lex-cxt$: [DTRS list($lex-sign$)]

(The daughters of a lexical construct are all of type $lex-sign$, i.e. they are words or lexemes.)

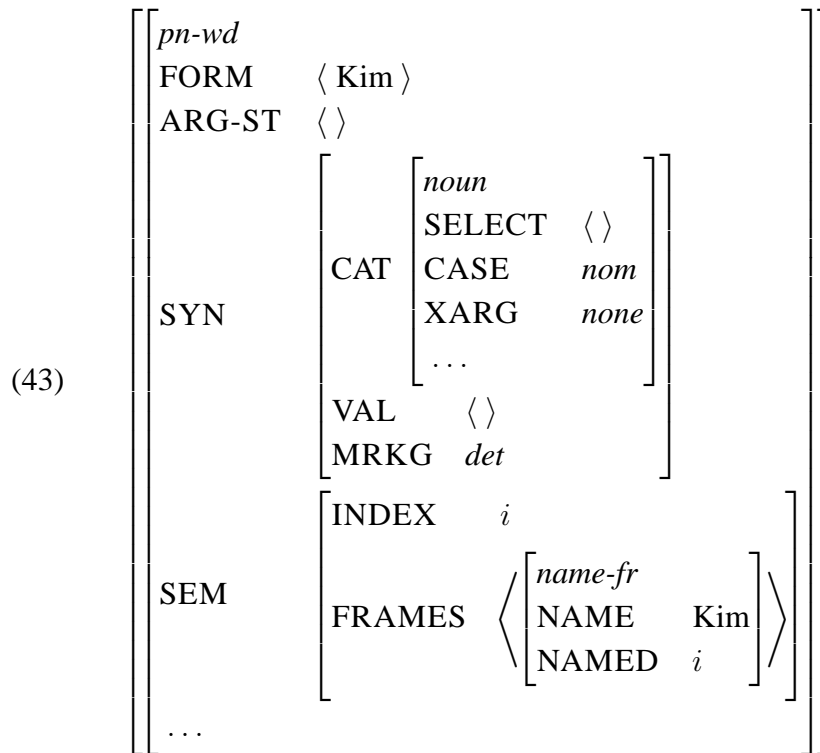
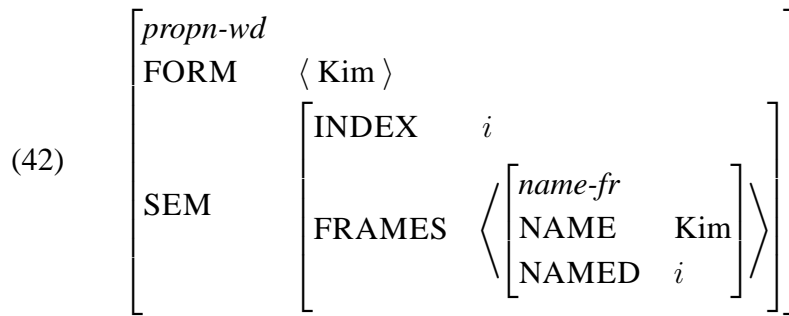
4.1.1 Lexical Entries

Let us begin with lexical entries. As in most feature-based theories of grammar, a lexical entry is specified as a constraint relating form, syntactic category, and meaning. In SBCG, lexical entries contain varying degrees of information about all aspects of lexemes. A lexical entry is thus a feature structure description that describes a class of lexical signs – lexemes or words.

But lexical entries can be quite spartan. This is because the grammar contains various type constraints that impose conditions on feature structure well-formedness. These are the lexical-class constructions mentioned above. The following lexical-class construction, for example, defines once and for all the distinctive properties that proper nouns have in common ($pn-wd$ stands for *proper-noun-wd*):

(41) $pn-wd \Rightarrow$ $\left[\text{SYN} \left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \text{noun} \\ \text{SELECT} \langle \rangle \\ \text{XARG} \text{ none} \end{array} \right] \\ \text{VAL} \langle \rangle \\ \text{MRKG} \text{ det} \end{array} \right] \right] \right]$

Once constraints like this are in place, a lexical entry can be reduced to a specification of (1) a FORM value, (2) a SEM value (3) a lexical type and (4) any further irregularities. For example, a lexical entry like (42) licenses a feature structure like (43):⁴⁰



Because the feature structure in (43) is of type *pn-wd*, for example, it must obey the type constraint sketched in (41).

Notice in addition that there must be a value for CASE in (43) – (either *nom* or *acc*), since (43) is a total function (as are all the functions within it), not a constraint on functions. Consequently, a diagram just like (43), except for having *nom* replaced by *acc*, would have served just as well to illustrate a construct satisfying the lexical entry in (42).

⁴⁰I assume here that the naming predication of a proper name is part of its SEM value. An alternate view, developed by Pollard and Sag (1987, 1994), treats these predications as part of the set of background conditions specified within CNTXT values. Again, SBCG is broadly compatible with a wide range of approaches to semantic and pragmatic analysis.

The basic intuition behind the theoretical and terminological innovations presented here (which distinguish SBCG from earlier work in Construction Grammar), is that constructions license the building of signs out of lexemes (which are in turn licensed by lexical entries) or else out of one or more distinct signs (in the case of combinatoric constructions). The constructions thus form a recursive system similar to a Context-Free Grammar. Crucially, combinatoric constructions (like CFG rules) are static constraints on mother-daughter configurations (constructs) and thus provide a declarative, order-independent, characterization of grammar. These design properties of SBCG fit well with the findings of modern psycholinguistics, as argued by Sag and Wasow to appear.⁴¹

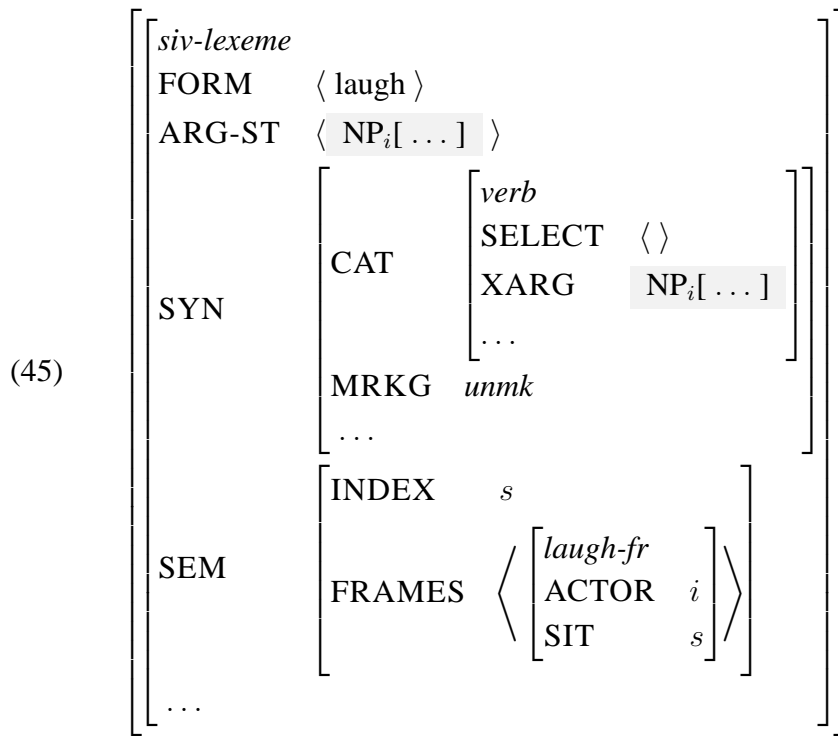
For various reasons, the class of lexical items that satisfy any lexical entry is infinite. This is true even in the case of the lexical entry for *Kim* given in (42) above, because there are infinitely many indices that could serve as the value of the feature INDEX in (43), corresponding to the fact that there is this no principled limit to the number of people who might be named *Kim*. Various of the features specified within the CNTXT value (e.g. SPEAKER, ADDRESSEE) are also index-valued, further expanding the space of feature structures. However, all feature structures that differ only in this way are equivalent for grammatical purposes; the only grammatically significant distinction among these functions is the value for the feature CASE.

In other circumstances, a given construction or lexical entry will license infinitely many feature structures that differ from one another in grammatically significant ways. One place where this arises has to do with the feature structures licensed by verbal, adjectival, and prepositional lexical entries that specify a nonempty ARG-ST (and VAL) list. For example, the lexical entry for the lexeme *laugh* is sketched in (44) (*siv-lexeme* abbreviates *strict-intransitive-verb-lexeme*):

$$(44) \left[\begin{array}{l} \textit{siv-lexeme} \\ \text{FORM} \quad \langle \textit{laugh} \rangle \\ \text{SEM} \quad \left[\begin{array}{l} \text{INDEX} \quad s \\ \text{FRAMES} \quad \left\langle \left[\begin{array}{l} \textit{laugh-fr} \\ \text{SIT} \end{array} \right] \right\rangle \quad s \end{array} \right] \end{array} \right]$$

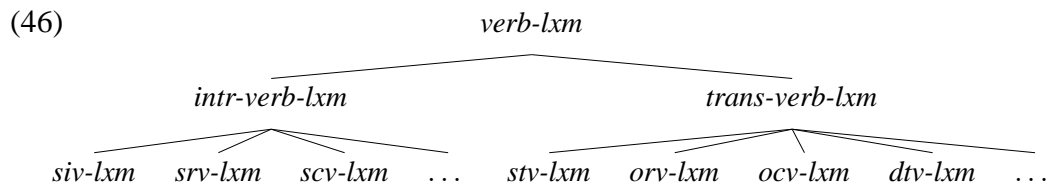
This lexical entry interacts with various constraints to license infinitely many feature structures like (45):

⁴¹For further discussion of the psychological plausibility of constraint-based grammars, see Jackendoff 1997, 2002, Culicover 1999, and Culicover and Jackendoff 2005.



The key thing to see here is that the NP_i[...] on the ARG-ST list (and the identical feature structure which is the XARG value; see below) must be fully specified in (45), even though neither the lexical entry in (44) nor any of the constraints that affect (45) places any restriction on what this NP feature structure should have as its FORM or SEM value, for instance.⁴² This is as it should be, since this NP feature structure will be identified with the subject of *laugh* and there are infinitely many NP signs that could perform that function, corresponding to infinitely many sentences of the form: ‘NP *laugh/laughed/laughs*’.

As noted earlier, the semantic and ARG-ST properties of lexeme classes are organized by the hierarchy of lexeme types, i.e. the subtypes of the type *lexeme*. This method is illustrated by the partial lexeme hierarchy in (46):⁴³



⁴²Hence, unlike certain other approaches to construction-based grammar, SBCG does not use lexical entries to impose ‘selectional restrictions’ (in the sense of Chomsky 1965). This follows the sage advice of McCawley (1971: 219), who recommends that someone who utters a sentence like *My toothbrush is alive and trying to kill me* be referred to a psychiatric clinic, rather than a remedial English course.

⁴³The following abbreviations are used: *strict-intransitive-verb-lexeme* (*siv-lxm*, e.g. *die*), *subject-raising-verb-lexeme* (*srv-lxm*, e.g. *seem* (to VP)), *subject-control-verb-lexeme* (*scv-lxm*, e.g. *try* (to VP)), *strict-transitive-verb-lexeme* (*stv-lxm*, e.g. *devour*), *object-raising-verb-lexeme* (*orv-lxm*, e.g. *believe* (NP to VP)), *object-control-verb-lexeme* (*ocv-lxm*, e.g. *persuade* (NP to VP)), *ditransitive-verb-lexeme* (*dtv-lxm*, e.g. *hand* (NP NP)).

I assume that all verbal lexemes obey the constraint in (47) (a lexical class construction), which says that a verbal lexeme is unmarked and that its first argument is its external argument.

$$(47) \quad \textit{verb-lxm} \Rightarrow \left[\begin{array}{l} \text{ARG-ST} \quad \langle X, \dots \rangle \\ \text{SYN} \quad \left[\begin{array}{l} \text{CAT} \quad \left[\begin{array}{l} \textit{verb} \\ \text{XARG} \quad X \end{array} \right] \\ \text{MRKG} \quad \textit{unmk} \end{array} \right] \end{array} \right]$$

But lexical class constructions are stated at diverse levels, so as to affect, for example, all lexemes, all verbal lexemes, all intransitive verb lexemes, or all lexemes of a particular maximal (leaf) type. A given lexeme must obey the constraints specified in the lexical entry that license it, but it must also obey the constraints that affect all the types that it instantiates. A ditransitive lexeme, for example, must obey whatever constraints affect *dtv-lexeme*, *trans-verb-lexeme*, *verb-lexeme*, and *lexeme*. A given lexical entry can thus be streamlined so as to include minimal stipulation, leaving it to the theory of lexical classes (embodied in the lexical class constructions) to determine which lexical properties are compatible with it.⁴⁴

This approach allows for underspecification of a kind that permits a given lexical entry to license lexemes of more than one lexical type. This appears to be the right way to analyze a variety of lexical problems, including the analysis of ‘*spray/load*’ alternations. That is, a verb like *spray* has one lexical entry that is compatible with two distinct lexical class constructions.

For example, suppose there are two lexeme types called *spray-load1-lxm* and *spray-load2-lxm* (both subtypes of *trans-verb-lxm*) that are constrained as follows:⁴⁵

$$(48) \quad \text{a. } \textit{spray-load1-lxm} \Rightarrow \left[\begin{array}{l} \text{ARG-ST} \quad \langle \text{NP}_x, \text{NP}_y, \text{PP}[\textit{dir}]_z \rangle \\ \text{SEM} \quad \left[\begin{array}{l} \text{INDEX} \quad s \\ \text{FRAMES} \quad \left\langle \begin{array}{l} \textit{spr-ld-fr} \\ \text{SIT} \quad s \\ \text{ACTOR} \quad x \\ \text{UNDGR} \quad y \\ \text{GOAL} \quad z \end{array} \right\rangle \end{array} \right] \end{array} \right]$$

⁴⁴This kind of simplification is typical of object-oriented analyses of complex data.

⁴⁵Here I abbreviate as follows:

$\text{PP}[\textit{dir}] = \text{PP} \ \& \ \left[\text{CAT|LID} \ \textit{dir-fr} \right] \quad \text{PP}[\textit{with}] = \text{PP} \ \& \ \left[\text{CAT|LID} \ \textit{with-fr} \right]$

The type *dir-fr* (*directional-frame*) is a supertype of the lexeme-specific frames that are associated with directional prepositions.

$$\text{b. } \textit{spray-load2-lxm} \Rightarrow \left[\begin{array}{l} \text{ARG-ST} \quad \langle \text{NP}_x, \text{NP}_z, \text{PP}[\textit{with}]_y \rangle \\ \\ \text{SEM} \quad \left[\begin{array}{l} \text{INDEX} \quad s \\ \\ \text{FRAMES} \quad \left\langle \begin{array}{l} \textit{spr-ld-fr} \\ \text{SIT} \\ \text{ACTOR} \\ \text{UNDGR} \\ \text{GOAL} \end{array} \right\rangle \begin{array}{l} s \\ x \\ y \\ z \end{array} \right. \end{array} \right. \end{array} \right]$$

Here *spr-ld-fr* designates a supertype of the frames associated with ‘*spray-load*’ verbs and ACTOR, UNDERGOER (UNDGR), and GOAL are the role features assumed to be common to a large class of action frames. Thus, simply by adding a lexical entry like (49) to the lexicon (which is compatible with both lexeme types in (48)), both kinds of lexical items are licensed:

$$(49) \quad \left[\begin{array}{l} \text{FORM} \quad \langle \textit{spray} \rangle \\ \text{SEM} \quad \left[\text{FRAMES} \quad \langle [\textit{spray-fr}] \rangle \right] \end{array} \right]$$

The proposal just sketched assumes that *spray-load* alternations involve no semantic difference, but minor modifications could be made that would accommodate different assumptions, as long as the two meanings for each verb are systematically related.

4.1.2 Morphological Functions

The next two sections discuss inflectional and derivational constructions. A key part of such word-building constructions are the **morphological functions**. As part of that discussion, as an example of an inflectional construction, let us consider the Preterite Construction. This construction builds the preterite form of a verbal *lexeme*. That is, constructs licensed by the Preterite Construction have a DTRS list containing exactly one feature structure of type *lexeme* and a mother of type *word*. The mother must include the appropriate FORM value and also the additional semantic bits corresponding to the meaning of the preterite word. The form of the mother is the image of the form of the daughter under the morphological function $\mathbf{F}_{\textit{pret}}$.

Morphological functions allow us to model ‘elsewhere’ phenomena in SBCG morphology without changing the overall logic of the architecture, as well as to deal with other problems posed by various kinds of irregularity. The FORM value of a given lexeme is a singleton list containing the lexical formative associated with that lexeme.⁴⁶

A member of the domain of a morphological function is an ordered pair giving the member of the lexeme’s FORM list and its LID value; the range (codomain) consists of the set of forms, including those constructed via affixation. Both FORM and LID values must be consulted

⁴⁶The discussion here simplifies in various respects, ignoring the possibility that multiple stems are associated a single lexical formative, for instance. Morphological functions will effect stem alternations as necessary. However, it is not clear that more than one stem is necessary for any English lexeme, although multiple stems for different tenses, cases, etc. are commonplace in languages generally. For an interesting discussion and a precise, compatible proposal for treating multiple stems in French morphology, see Bonami and Boyé 2006.

because in some cases it is the FORM value that determines the inflected form (e.g. *have* \Rightarrow *had* for all of the distinct lexemes *have*) and sometimes the LID value (*lie* 'recline' \Rightarrow *lay*, *lie* 'prevaricate' \Rightarrow *lied*). \mathbf{F}_{pret} might be defined along the following lines:

(50)

$\langle x, y \rangle$	$\mathbf{F}_{pret}(x, y)$
$\langle be, y \rangle$	undefined
$\langle have, y \rangle$	<i>had</i>
$\langle lie, recline-fr \rangle$	<i>lay</i>
$\langle dream, y \rangle$	<i>dreamt</i>
$\langle swim, y \rangle$	<i>swam</i>
$\langle buy, y \rangle$	<i>bought</i>
$\langle keep, y \rangle$	<i>kep+ed</i>
...	
Otherwise	
$\langle x, y \rangle$	$\langle x+ed \rangle$

Special constructions will be needed to specify the preterite forms of *be* (*was* for non-second person singular and *were* for 2nd person or plural). All lexemes whose FORM value is $\langle have \rangle$ will have the same irregular preterite form, i.e. *had*. *lie* 'recline' will be inflected as *lay* by the third line of the function and *lie* 'prevaricate' will be inflected as *lie+ed* by the 'Otherwise' clause. As shown here in simplified terms, \mathbf{F}_{pret} will only produce the irregular *dreamt* preterite form of *dream*. A special construction would be posited to get the *dream+ed* form. This formulation is consistent with the view that both forms are memorized. On the alternate view, that the regular forms are produced by the regular process and only the exceptional forms of the doublets are memorized, *dream* would be omitted from the exception list at the beginning of \mathbf{F}_{pret} . Then *dream+ed* would be licensed by the 'Otherwise' clause of \mathbf{F}_{pret} and a separate construction would be posited for *dreamt*. *Swam* and *bought* are unproblematic irregular forms without doublets. The inflection of $\langle keep, y \rangle$ as *kep+ed* is intended to illustrate the ability of morphological functions to deal with stem changes, although it is arguable that this inflection, and analogous ones in English preterites, should simply be viewed as suppletive.⁴⁷

4.1.3 Inflectional Constructions

In addition to the general constraints on lexical constructs, inflectional constructs have more specific properties that are specified as part of the grammar signature ($nelist(\tau)$ stands for a nonempty list, each of whose members is a feature structure of (some subtype of) type τ):

⁴⁷Non-past-tense uses of the preterite morphological form, such as counterfactual conditional protases (*If I had my way,...*) could in principle be licensed by a separate inflectional construction that also avails itself of \mathbf{F}_{pret} . Alternatively, one might pursue a semantics-changing post-inflectional 'pumping' construction, whose mother and unique daughter do not differ in FORM. In either case, special arrangements must be made to license and distinguish in meaning both, for example, counterfactual *I had...* from its past tense analogue *I had...* I will not resolve these issues here.

$$(51) \quad \text{infl-}cxt: \left[\begin{array}{ll} \text{MTR} & \text{word} \\ \text{DTRS} & \text{nelist}(\text{lexeme}) \end{array} \right]$$

(The mother of an inflectional construct is of type *word*; the daughters must be lexemes.)

This treatment embodies the traditional intuition that inflectional constructions are resources for building words from lexemes (Remember that affixes are not constituents in SBCG's essentially realizational morphology).⁴⁸

An inflected word like *laughed* is modeled via feature structures of the sort sketched in Figure 1. Because this is a well-formed construct, the feature structure in (52), which is the mother of the construct shown in 51, is constructionally licensed:

$$(52) \quad \left[\begin{array}{l} \text{word} \\ \text{FORM} \quad \langle \text{laugh+ed} \rangle \\ \text{ARG-ST} \quad \langle \text{NP}_i[\text{nom} \dots] \rangle \\ \text{SYN} \quad \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{verb} & \\ \text{VF} & \text{fin} \\ \text{SELECT} & \langle \rangle \\ \text{XARG} & \text{NP}_i[\text{nom}\dots] \\ \text{LID} & \text{laugh-fr} \\ \dots & \end{array} \right] \\ \text{VAL} & \langle \text{NP}_i[\text{nom} \dots] \rangle \\ \text{MRKG} & \text{unmk} \\ \text{IND} & s \end{array} \right] \\ \text{SEM} \quad \left[\begin{array}{l} \text{FRAMES} \quad \left\langle \left[\begin{array}{ll} \text{exist-fr} & \\ \text{BV} & s \end{array} \right], \left[\begin{array}{ll} \text{laugh-fr} & \\ \text{ACTOR} & i \\ \text{SIT} & s \end{array} \right], \left[\begin{array}{ll} \text{past-fr} & \\ \text{ARG} & s \end{array} \right] \right\rangle \end{array} \right] \\ \dots \end{array} \right]$$

Several observations are in order here. First, the SYN values of the mother and the daughter in Figure 1 are identical. Second, I have provided a Davidsonian analysis of the preterite that introduces a tense frame taking the situation of the lexeme's frame as its argument and an existential quantifier binding the situation variable.⁴⁹ This ignores many interesting issues in the semantics of tense in English; but however the semantic analysis of past tense is articulated, the past tense semantics will be absent from the lexeme daughter in Figure 1, but

⁴⁸There is usually, if not always, a single daughter in an inflectional construct. For convenience, I here ignore languages with layered inflection. See Orgun 1996, Miller and Sag 1997 and Malouf 2000 for relevant discussion.

⁴⁹This analysis is easily refined in terms of generalized quantification.

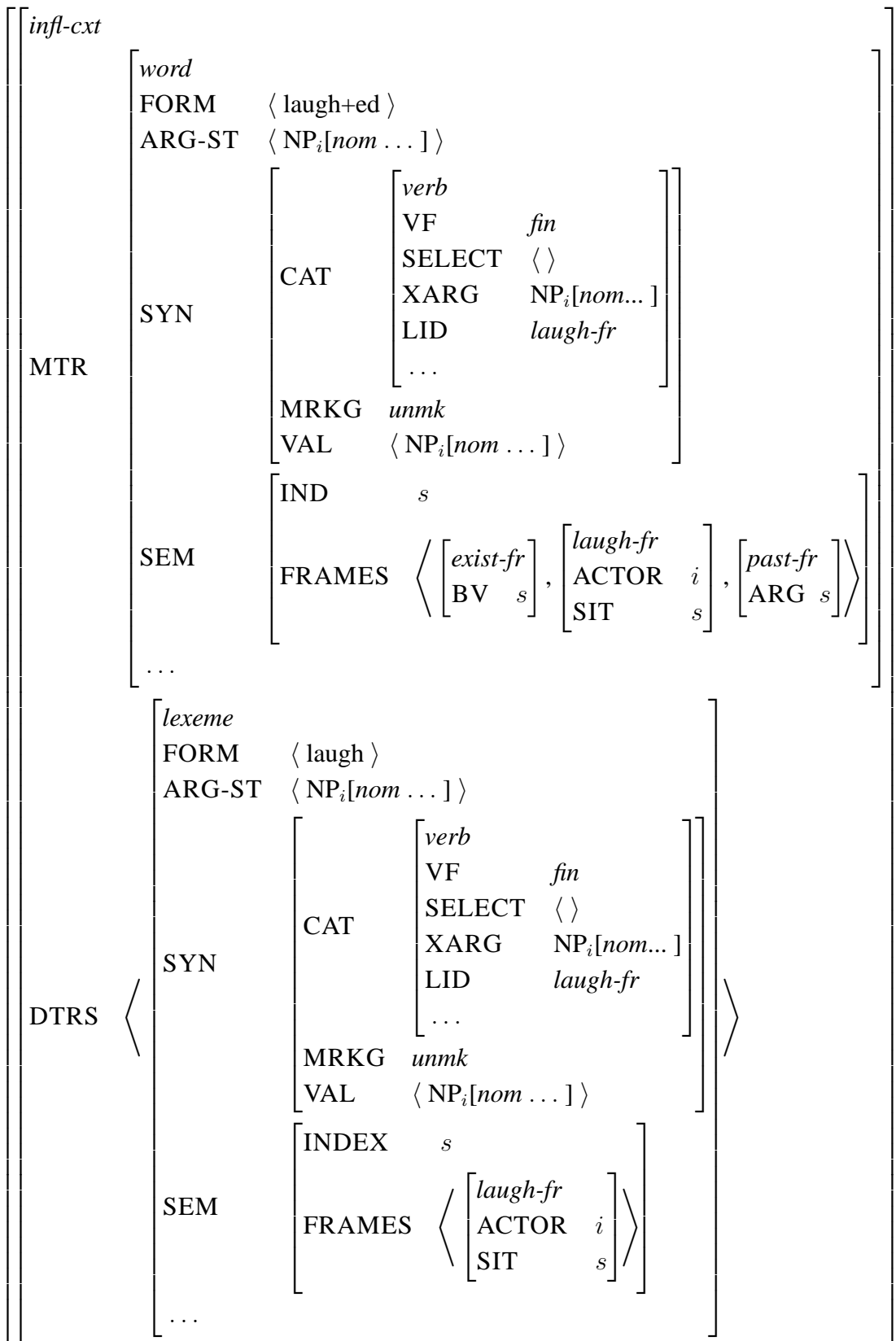


Figure 1: A *laughed* Construct

present in the semantics of the preterite word. Third, preterite constructs like the one in Figure 1 belong to the type *infl-cxt*, and hence must obey all constraints affecting feature structures of that type. Fourth, in the feature structure illustrated here the VAL list is identical to the ARG-ST list. This is not the only possibility. A word’s VAL list is shorter than its ARG-ST list whenever an argument corresponds to a gap in a filler-gap construction, is realized as a pronominal affix (e.g. in French ‘clitic’ pronouns, which are affixal in nature (Miller 1992, Miller and Sag 1997), or else participates in one of the various kinds of ‘null instantiation’ (Fillmore 1986). Similarly, the SELECT value is here realized as the empty list, reflecting the fact that the clause this verb will project (a non-relative clause) cannot function as a modifier. Finally, the information encoded in Figure 1 is exactly the same as what is presented in a more familiar diagram, the unary (non-branching) local tree in Figure 2. Because of their familiarity, I will use trees whenever possible to illustrate feature structures of type *construct*.

A preliminary version of the Preterite Construction is given in (53):⁵⁰

(53) **Preterite Construction (preliminary version):**

$$\begin{array}{l}
 \text{preterite-cxt} \Rightarrow \\
 \left[\begin{array}{l}
 \text{MTR} \left[\begin{array}{l}
 \text{FORM} \quad \langle \mathbf{F}_{\text{pret}}(X, Z) \rangle \\
 \text{ARG-ST} \quad L_1 \ \& \ \langle \text{NP}[\text{nom}], \dots \rangle \\
 \text{SYN} \quad Y \ \& \ \left[\text{CAT} \begin{array}{l} \text{VF} \quad \textit{fn} \\ \text{LID} \quad Z \end{array} \right] \\
 \text{SEM} \left[\begin{array}{l} \text{IND} \quad s \\ \text{FRAMES} \quad L_2 \oplus \left\langle \left[\begin{array}{l} \exists\text{-FR} \\ \text{BV} \quad s \end{array} \right], \left[\begin{array}{l} \textit{past-fr} \\ \text{ARG} \quad s \end{array} \right] \right\rangle \end{array} \right]
 \end{array} \right] \\
 \text{DTRS} \left\langle \begin{array}{l} \left[\begin{array}{l} \text{FORM} \quad \langle X \rangle \\ \text{ARG-ST} \quad L_1 \\ \text{SYN} \quad Y \\ \text{SEM} \left[\begin{array}{l} \text{IND} \quad s \\ \text{FRAMES} \quad L_2 \end{array} \right] \end{array} \right] \end{array} \right\rangle
 \end{array} \right]
 \end{array}
 \end{array}$$

One way of paraphrasing (53) is as follows: Given a verbal lexeme⁵¹ one can construct a verbal word meeting the following four conditions:

⁵⁰Variables such as X , X_1 , and Y range over feature structures in the constraints and constructions that are formulated here. L -variables range over lists of feature structures. An ‘&’ indicates that the immediately following constraint must be satisfied in addition to the immediately preceding constraint. A variable followed by ‘&’ and a description in essence indicates a restriction that values of the variable must satisfy. The semantic constraints illustrated here and throughout can be simplified by adopting the Principle of Compositionality (Sag et al. 2003), specifically the version proposed in Copestake et al. 2005, which provides a general way of incorporating constructional meaning into the FRAMES list of the mother sign.

⁵¹It must be a verb because its CAT value must be compatible with a VF specification, and the grammar signature ensures that VF is appropriate only for feature structures of type *verb*.

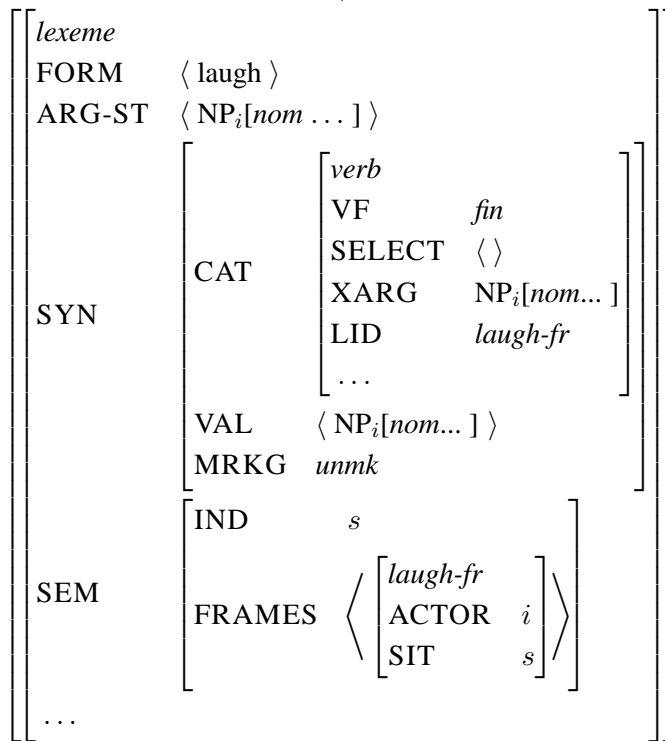
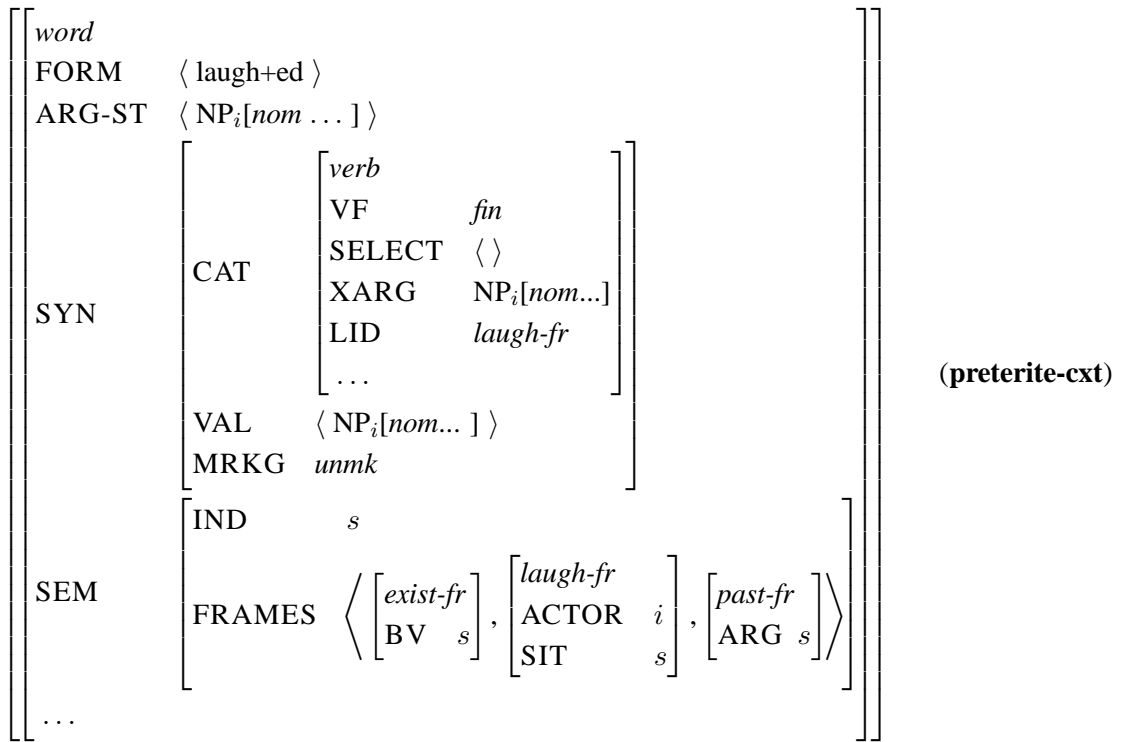


Figure 2: A *laughed* Construct in Tree Notation

- (54) a. the word's VF value is *finite*, as is the VF value of the lexeme.⁵²
- b. the word's FORM value is related to that of the lexeme (and its LID) via the morphological function \mathbf{F}_{pret} ,⁵³
- c. the word's SYN and ARG-ST values are identified with those of the lexeme daughter, thus requiring that everything in the lexical entry that licensed the lexeme be consistent with the constraints introduced by this construction, e.g. that the subject valent's CASE value be nominative, and
- d. the word's FRAMES list adds a 'pastness' frame and an existential quantifier to the lexeme's FRAMES list, identifying the argument of this frame with the situation specified in the lexeme's original frame and with the variable bound by the existential quantifier.⁵⁴

Notice that it would be redundant for the construction in (53) to explicitly stipulate that the MTR value be of type *word* or that the daughter must be of type *lexeme*. Because the constructs licensed by the Preterite Construction are all instances of the type *infl-cxt*, they must obey all constraints on feature structures of this type imposed by the grammar signature. As we have already seen (51), requires that the MTR value of all inflectional constructs be of type *word* and that all daughters of inflectional constructs be of type *lexeme*. Hence this is true of all preterite constructs as well, by the process of constraint inheritance (see section 2 above). It also follows that the 'output' of this construction (a word) can never 'undergo' the construction again (since the daughter of any inflectional construction must be of type *lexeme*).

As our analysis of English morphology scales up, this construction can be further simplified. Since all finite forms in English require nominative subjects, as shown in (55),

- (55) a. She/*her walked home.
- b. They/*them walk home after work.
- c. I suggested that he/*him walk home.
- d. I/*me am walking home today.

we will surely want to posit a subtype of *infl-cxt* (and supertype of *preterite-cxt*) to express this generalization. Let us call this new type *finite-cxt* and posit the following type constraint:

⁵²Recall that a lexeme's VF value is in general unconstrained, and hence free to be equated with that of the mother in this construct. This treatment allows a clean analysis of modal verbs that only have finite forms. Since their lexical entries say [SYN|CAT|VFORM *fin*], they fail to license any lexemes that could give rise to nonfinite words. Note in addition that, since such elements are equated, the choice of whether to write the relevant constraints in terms of the mother (as in the case of VF and LID in (54)) or in terms of the daughter is entirely arbitrary.

⁵³Recall that these morphological entities are distinct from (and more 'abstract' than) the phonological entities that realize them. See the discussion in section 3.1 above.

⁵⁴Our semantic analysis of the preterite, as represented in (53) is simplified in various ways, as previously noted.

$$(56) \quad \textit{finite-cxt} \Rightarrow \left[\begin{array}{l} \textit{infl-cxt} \\ \text{MTR} \left[\begin{array}{l} \text{ARG-ST} \quad \langle \text{NP}[\textit{nom}], \dots \rangle \\ \text{SYN} \quad \left[\text{CAT} \quad \left[\text{VF} \quad \textit{fin} \right] \right] \end{array} \right] \end{array} \right]$$

Once this ‘finite words assign nominative case to their first argument’ constraint is inherited, rather than being stipulated as part of the Preterite Construction, the latter (and all its sister constructions) can be simplified further:

(57) **Preterite Construction (final version):**

$$\textit{preterite-cxt} \Rightarrow \left[\begin{array}{l} \text{MTR} \left[\begin{array}{l} \text{FORM} \quad \langle \mathbf{F}_{\textit{pret}}(X, Z) \rangle \\ \text{ARG-ST} \quad L_1 \\ \text{SYN} \quad Y \ \& \ \left[\text{CAT} \quad \left[\begin{array}{l} \text{VF} \quad \textit{fin} \\ \text{LID} \quad Z \end{array} \right] \right] \\ \text{SEM} \left[\text{FRAMES} \ L_2 \oplus \left\langle \left[\begin{array}{l} \exists\text{-FR} \\ \text{BV} \ s \end{array} \right], \left[\begin{array}{l} \textit{past-fr} \\ \text{ARG} \ s \end{array} \right] \right\rangle \right] \end{array} \right] \\ \text{DTRS} \left\langle \left[\begin{array}{l} \text{FORM} \quad \langle X \rangle \\ \text{ARG-ST} \quad L_1 \\ \text{SYN} \quad Y \\ \text{SEM} \left[\begin{array}{l} \text{IND} \ s \\ \text{FRAMES} \ L_2 \end{array} \right] \end{array} \right] \right\rangle \end{array} \right]$$

The simplification achieved here may appear slight, but it contributes to the overall goal of type-based constraint inheritance, which is to eliminate unmotivated redundancy from grammar.

4.1.4 Derivational Constructions

Derivational constructions are structured as shown in (58):

$$(58) \quad \textit{deriv-cxt}: \left[\begin{array}{l} \text{MTR} \quad \textit{lexeme} \\ \text{DTRS} \quad \textit{nelist}(\textit{lex-sign}) \end{array} \right]$$

(The mother of a derivational construct is of type *lexeme*; the daughters of a derivational construct are lexical signs (words or lexemes).)

Derivational constructions thus allow new lexemes to be built from one or more lexical signs. For example, there is an *un*-prefixation construction, sketched in (59), which allows *un*-verb lexemes to be derived from a specifiable class of verb lexemes:

(59) *Un-Verb Construction*:

$$\text{unverb-cxt} \Rightarrow \left[\begin{array}{c} \text{MTR} \\ \text{DTRS} \end{array} \left[\begin{array}{l} \left[\begin{array}{l} \text{FORM} \quad \langle \mathbf{F}_{un}(X) \rangle \\ \text{ARG-ST} \quad L_1 \\ \text{SYN} \quad Y \\ \text{SEM} \quad [\text{FRAMES} \quad L_2 \oplus \dots] \end{array} \right] \\ \left\langle \begin{array}{l} \text{stv-lxm} \\ \text{FORM} \quad \langle X \rangle \\ \text{ARG-ST} \quad L_1 \\ \text{SYN} \quad Y \\ \text{SEM} \quad [\text{FRAMES} \quad L_2] \end{array} \right\rangle \end{array} \right]$$

I have assumed here that only strict-transitive verbal lexemes (lexemes of type *stv-lxm*) can give rise to *un-verb* lexemes. However, the restrictions on this construction are partly semantic, as *uncrush* and *unlift* seem impossible precisely because the events denoted by *crush* and *lift* are ‘irreversible’. As usual, there is room for disagreement about whether the relevant constraints that should be placed on (59) are syntactic, semantic, or some combination of the two.

Since inflectional constructs are required to have a daughter of type *lexeme*, a natural relation exists between the two types of construction: derivational constructions feed inflectional constructions. That is, a derived lexeme, one that is the mother of a construct licensed by some derivational construction, can then serve as the daughter of a construct licensed by an inflectional construction, as illustrated in Figure 3, where the two constructs are conflated, with the shared lexeme serving simultaneously as mother of the derivational construct and daughter of the inflectional construct. Derivational constructions can also feed other derivational constructions and inflectional constructions can sometimes feed derivational constructions; an example is the case of nominal compounds in which the modifying noun is inflected for plural: e.g. *grants secretary*.⁵⁵

All of the following phenomena are assumed to be analyzed via derivational constructions:

- (60) a. passivization, which feeds overt inflectional constructions in many languages and word-formation processes in English (see Bresnan 2001),
- b. agentive noun formation,
- c. denominal verb formation (See Clark and Clark 1979),
- d. various kinds of nominalization,
- e. the *Un-Adjective Construction*, and

⁵⁵There is thus no strong theory of ‘level ordering’ (Kiparsky 1982) assumed here, though it would be easy enough to incorporate one simply by tightening the condition on the value of DTRS in (58) above. See note 58.

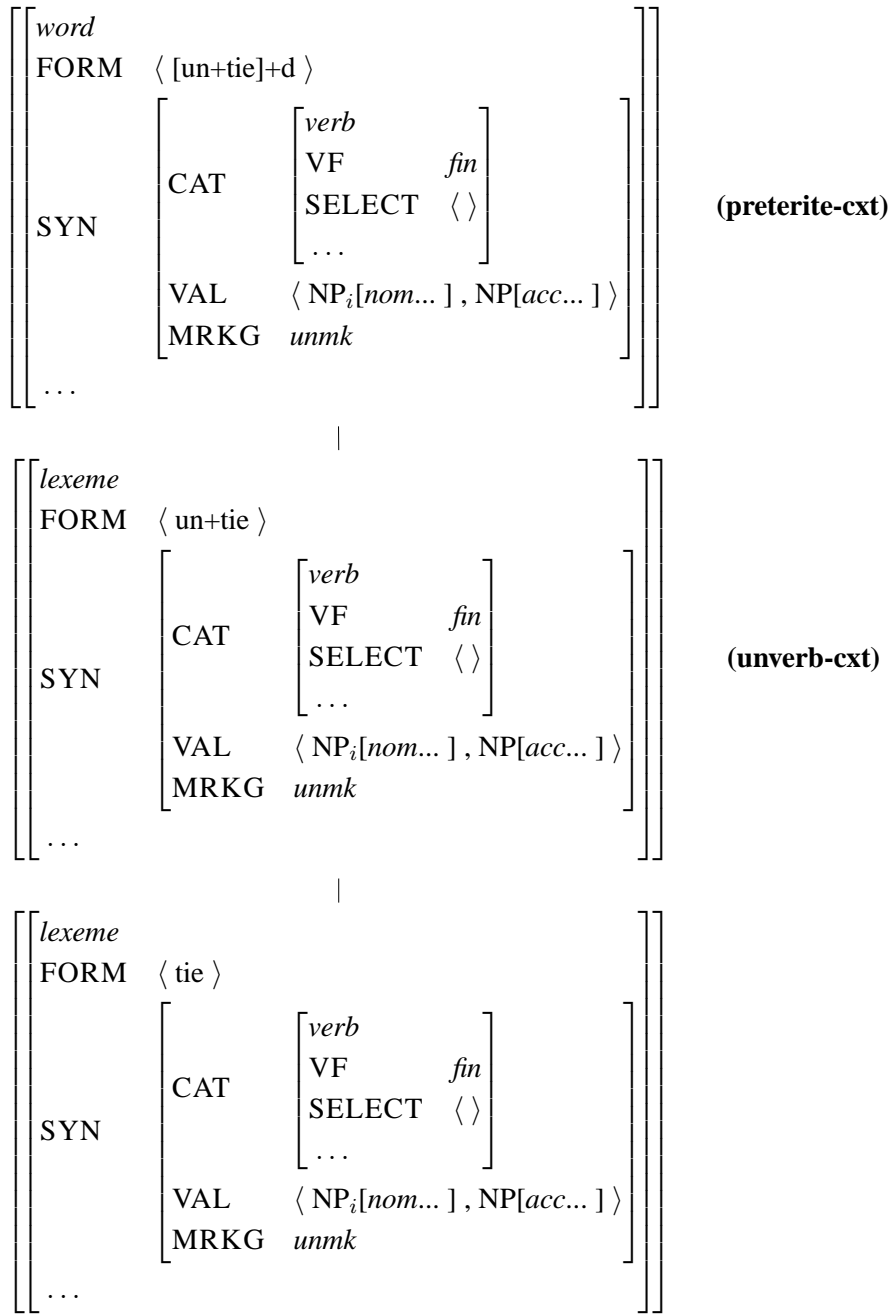


Figure 3: *Un-Verb Construction Feeding Preterite Construction*

f. the *-Able* Adjective Construction.

An example of a binary derivational construction is English noun-noun compounding. By specifying the DTRS value of a *deriv-cxt* to be a list of *lex-signs*, members of compounds are allowed to be inflected words, as well as lexemes.

The general compounding construction, which appeals to a contextually salient (but otherwise arbitrary) property to relate the interpretations of two nouns, accounts for compounds like the following:⁵⁶

- (61) a. pumpkin bus: ‘a bus that was used in some previous excursion to a pumpkin patch familiar to the relevant interlocutors’ (Downing 1977)
- b. Jaeger potato: ‘potato of the kind that the speaker once used for something when spending an evening with someone named Jaeger’
- c. Beatles fan, women friends, people smugglers, pubs inspector, munitions dump

Examples (61a) and (61b) illustrate attested innovative compounds. The examples in (61c) are also attested and exemplify some of the subtypes of noun-noun compounds exhibiting internal inflection.⁵⁷

It is also possible to incorporate proposals like that of Copestake and Lascarides (1997) (formalizing the basic intuition of Levi (1978)), who posit a number of more specific constructions specifying patterns that fit particular classes of nouns together in conventionalized ways. I won’t explore that possibility here.⁵⁸

4.1.5 Postinflectional Constructions

Postinflectional constructs are structured as follows:

$$(62) \quad \textit{pinfl-cxt}: \left[\begin{array}{ll} \text{MTR} & \textit{word} \\ \text{DTRS} & \textit{list(word)} \end{array} \right]$$

(The mother and daughters of a postinflectional construct are of type *word*.)

Postinflectional constructions thus allow for words to be derived from other words. Sag et al. (2003) introduce this category as a way of incorporating a number of proposals that have been made (by Warner (1993), Bouma and van Noord (1994), Kim and Sag (2002), Kim 2000 and

⁵⁶See Kay and Zimmer 1978, Downing 1977, Copestake and Lascarides 1997, and Levi 1978.

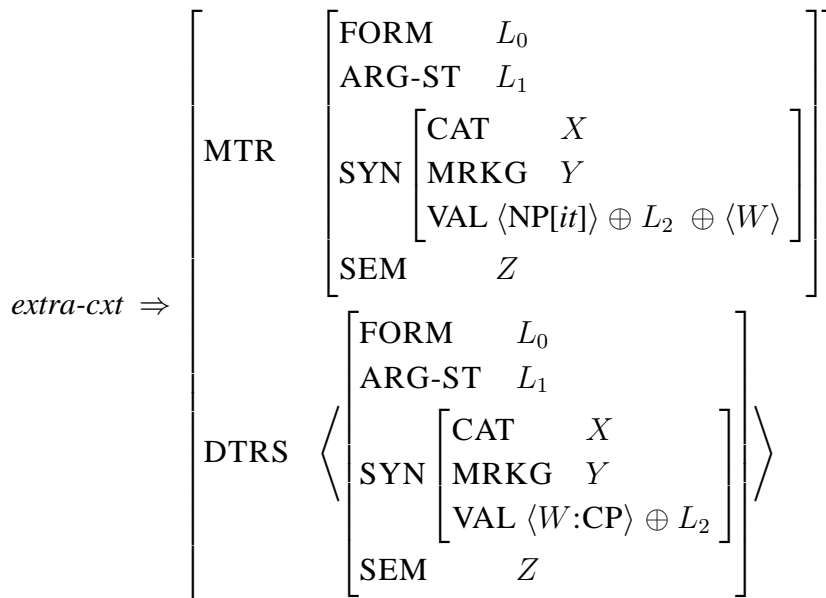
⁵⁷See Bauer and Reynaud 2001.

⁵⁸As noted, the first member of most noun-noun compounds is a lexeme (*computer screen, pumpkin bus*, etc.), but in others it is a word: *algorithms course, sales tax*, etc. Pinker (1999) and Kiparsky (1982) give reasons that that the first member of a nominal compound cannot be a word. Bauer and Reynaud 2001, in a corpus study, discuss the circumstances under which it is likely to be one. For further critical discussion and experimental evidence that there are multiple factors determining well-formedness of plural nouns in nominal compounds (rather than a simple grammatical constraint blocking any such combinations), see Harrison 2001 and Ramscar 20???

others) in terms of lexical rules that create adverb-selecting auxiliary verbs (e.g. *did (not)*, *will (not)*), as well as *not*-contracted words (e.g. *didn't*, *couldn't*) and related elements.

Various other lexical regularities can be analyzed in terms of a postinflectional construction. For example, Sag et al. (2003) present a postinflectional analysis of *it*-extraposition. This can be recast in the present framework in the manner sketched in (63):

(63) *It*-Extraposition Construction:



This construction licenses words that take *it*-subjects and sentential complements on the basis of the existence of phonologically, and presumably semantically, indistinguishable counterparts that take sentential subjects. The words licensed by this construction are fully equipped to project head-complement phrases of the sort discussed below, with the extraposed clause appearing as a complement.⁵⁹ Verbs or adjectives whose cooccurring clauses occur only in extraposed position, e.g. the instances of (non-raising) *seem* and *appear* illustrated in (64), are simply listed in the lexicon with the same ARG-ST value as signs that are constructed via (63):

- (64) a. It seems that you've been called for jury duty.
 b.*That you've been called for jury duty seems.

Finally, it should be noted that it is sometimes difficult to discern the differing consequences of a postinflectional analysis and a derivational one. Sometimes the issue is decided by the feeding relations between the construction in question and other derivational constructions. For example, a word licensed by a postinflectional construction cannot usually serve as the daughter of a derivational construct because most derivational constructions require a

⁵⁹For a more comprehensive analysis of English *it*-extraposition, where object-extraposition is also included and where extraposed clauses are treated as non-complement dependents, see Kim and Sag 2005.

daughter of type *lexeme*. Hence, treating a given alternation via a postinflectional construction insures that the result cannot feed most derivational constructions.

4.2 Phrasal Constructions

Phrasal (syntactic) constructs work in the same way as lexical constructs, except that they construct phrases from expressions:

$$(65) \quad \textit{phr-cxt}: \left[\begin{array}{ll} \text{MTR} & \textit{phrase} \\ \text{DTRS} & \textit{list(expression)} \end{array} \right]$$

(The mother of a phrasal construct must be a phrase and the daughters must be expressions, i.e. words or phrases.)

4.2.1 The Subject-Predicate Construction

Simple declarative clauses are licensed by the Subject-Predicate Construction, sketched in (66):

(66) **Subject-Predicate Construction** (preliminary version):

$$\textit{subpred-cxt} \Rightarrow \left[\begin{array}{ll} \text{MTR} & \left[\text{SYN} \left[\begin{array}{l} \text{MRKG } M \\ \text{VAL } \langle \rangle \end{array} \right] \right] \\ \text{DTRS} & \left\langle X, \left[\text{SYN} \left[\begin{array}{l} \text{CAT } [\text{VF } \textit{fin}] \\ \text{MRKG } M:\textit{unmk} \\ \text{VAL } \langle X \rangle \end{array} \right] \right] \right\rangle \end{array} \right]$$

This construction says that a [VAL $\langle \rangle$] phrase can be built from two daughters, as long as the second is a finite (and hence verbal) sign that selects for the first via the VAL feature. Independent principles (that is, linear precedence constraints) require that the FORM value of the mother be the result of adding the members of the second daughter's FORM value to the FORM value of the first daughter.⁶⁰ Similarly, the Principle of Compositionality (Sag, Wasow and Bender 2003, Copestake et al. 2005) requires that the FRAMES list of the two daughters be merged to form the mother's FRAMES list. With the interaction of these principles, (66) licenses phrasal constructs like the one in Figure 4. Notice that the mother of this construct is just the phrasal sign illustrated in (32) above.

⁶⁰For convenience, I will henceforth omit discussion of linear ordering, assuming that the order of elements on the DTRS list determines the order of elements on the mother's FORM list. This is a gross simplification of a complex set of issues that have motivated ID-LP format (the separation of constructions and the principles that order their daughters) and 'Linearization Theory', the augmentation of sign-based grammar to allow interleaving of daughters as an account of word order freedom. On ID-LP grammars, see Gazdar and Pullum 1981, Gazdar et al. 1985, and Pollard and Sag 1987, among others. On Linearization Theory, see Reape 1994, Mueller 1995, Donohue and Sag 1999, 1999, Kathol 2000, 2002, 2004, and Daniels and Meurers 2004.

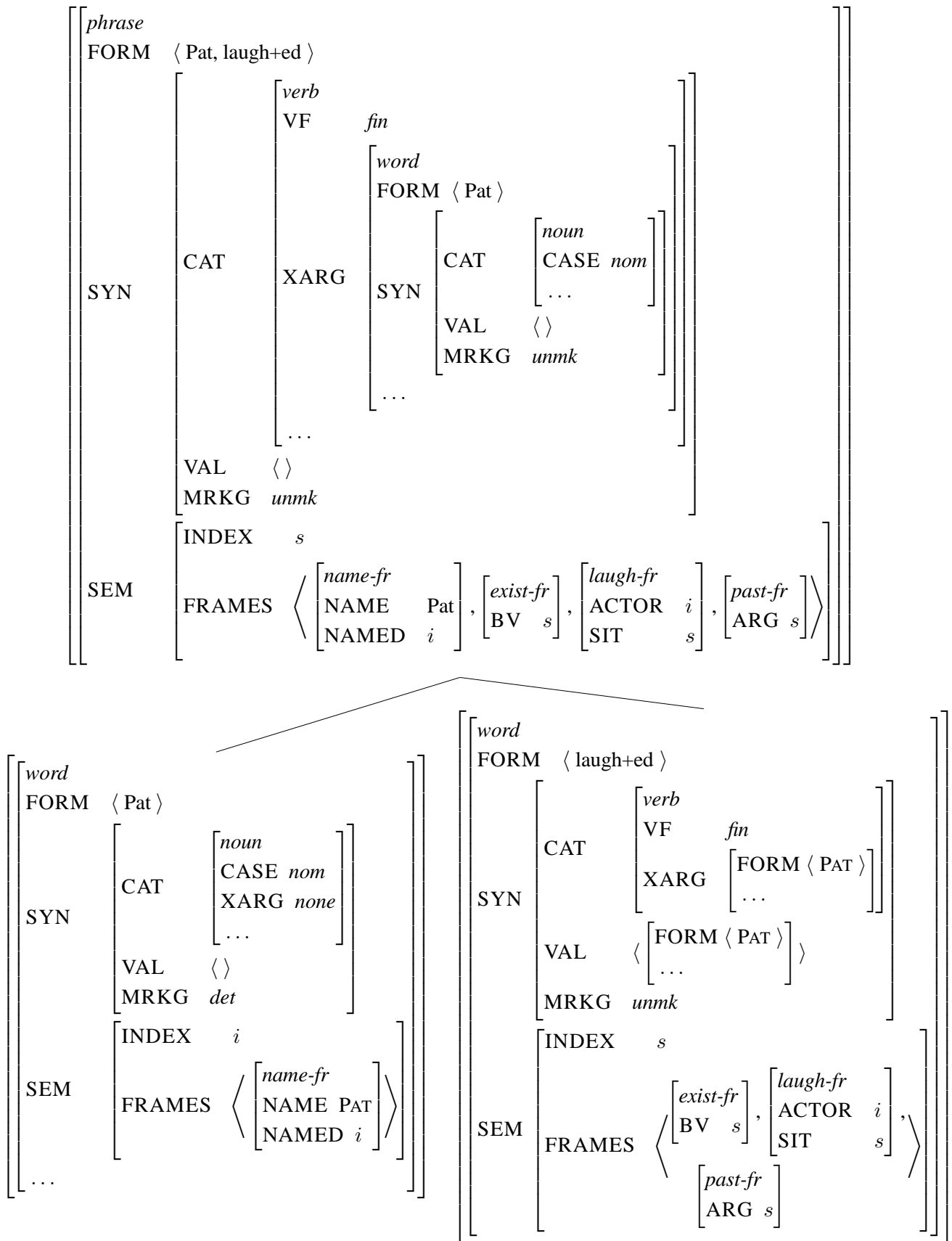


Figure 4: A Subject-Predicate Construct

Although our theory of constructions is strictly localist, i.e. our constructions – like rules in a context-free grammar – can only make reference to a mother and its daughters, we can nevertheless accommodate grammatical dependencies that are non-local. In particular, we build on work in the GPSG/HPSG tradition that has used feature specifications to locally encode information about long-distance dependencies. Just as the featural representation of a category like ‘NP’ encodes the head word within whose category is of type *noun*, other feature specifications can encode key grammatical information about an element that is present in (or absent from) a phrase. For example, the VF value of a verbal phrase (VP or S) encodes a morphosyntactic property of the head word within that phrase. Similarly, the feature GAP⁶¹ is used to encode the absence of an ‘extracted’ element (or, as linguists often put it: the ‘presence of a gap’) within a given phrase. By developing a theory of such feature specifications and the principles that govern their distribution throughout constructs, we are *ipso facto* developing a theory of what nonlocal information can be lexically selected at a higher level of structure.

As has been recognized at least since Chomsky (1965),⁶² lexical restrictions are circumscribed, i.e. they are localized in a fashion that must be made precise. Behind the search for the precise characterization of the relevant notion of locality of selection is the clear intuition that no language has, for example, a verb that requires a clausal complement that must contain an overt subject that is feminine, or singular, etc. Early accounts of locality excluded subjects, but since idiosyncratic case assignment in numerous languages (perhaps most famously in Icelandic⁶³) clearly involves the subjects of verbs, the most likely first approximation of the relevant locality domain is: a lexical element’s grammatical dependents. We may formulate the relevant hypothesis as in (67):

(67) Selectional Locality

For purposes of category selection (subcategorization), (nonanaphoric) agreement, semantic role assignment, and case assignment, a lexical head has access only to those elements that it is in a grammatical relation with (subject of, complement of, etc.). In SBCG, this amounts to a restriction that the only non-local elements that can be selected are those whose grammatical information is encoded by constraints placed on elements of the ARG-ST list of the predicator that is doing the selecting at the higher level.

Our various features and the particular choices made about the nature of their values, taken together with general constraints on how information is percolated as phrasal signs are constructed, constitute a precise formulation of the basic idea embodied in (67). In particular, the information lexically specified on an element’s ARG-ST and SELECT lists constrain the nature of the elements it combines with, providing access to grammatical dependents, but not to elements within them. And by adding a specific feature like XARG to systematically propagate certain information about elements embedded within dependents, we in effect localize certain nonlocal information, making it available to an element selecting that dependent.

⁶¹This feature name is due to Sag et al. (2003). For more detailed, explicit analyses that employ the feature name ‘SLASH’, see Gazdar 1981, Gazdar et al. 1985, Pollard and Sag 1987, 1994, Bouma et al. 2001, and Levine and Hukari 2006.

⁶²See also Kajita 1968 and Sag to appear b.

⁶³See for example Thráinsson 1979 and Andrews 1982, 1990.

4.2.2 External Arguments

There are two more features and two more constructions that we must discuss. Our head complement construction, which is used to build VPs, APs, PPs, and common noun phrases (CNPs), makes use of the feature XARG, which was introduced briefly in section 3.3.1 above. This section reviews some of the basic motivation for this feature.

As a number of researchers have recently shown, there are phenomena in diverse languages whose analysis requires that a verb selecting a sentential complement, be able to place constraints on the subject within that complement. It is interesting to examine some of the specific seemingly nonlocal phenomena that have led to such conclusions and the proposals that they have given rise to in various languages.

Bender and Flickinger (1999) analyze agreement in English tag questions by allowing the subject's agreement information to percolate up to the level of the clause. When a clause is combined with the tag of a tag question, this agreement information is then identified with that of the pronoun in the tag. This induces the familiar tag question agreement pattern illustrated in (68):

$$(68) \quad [\text{They left,}] \text{ didn't } \left\{ \begin{array}{l} \text{they} \\ *(s)\text{he} \\ *we \\ *you \\ *I \end{array} \right\} ?$$

The problem here is not selectional locality, but rather the related issue of constructional locality, about which we may formulate the following hypothesis:

(69) **Constructional Locality** (Context-Freeness⁶⁴):

Constructions license mother-daughter configurations without reference to embedding or embedded contexts.

Notice that Constructional Locality is an immediate consequence of the feature geometry assumed in SBCG, which, unlike earlier work in HPSG, draws a fundamental distinction between signs and constructs. Constructional Locality does not preclude an account of non-local dependencies in grammar, it simply requires that they be locally encoded in signs in such a way that information about such a dependency can be accessed locally at higher levels of a derivation.

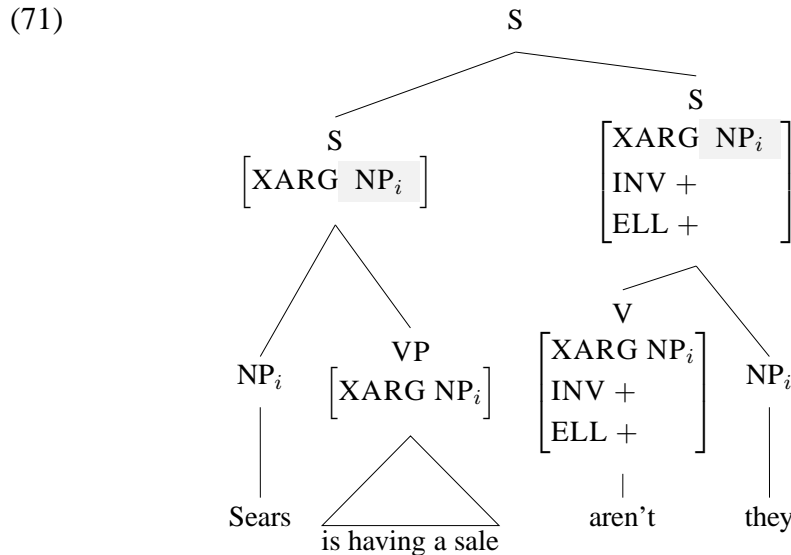
Bender and Flickinger assume that the agreement between the two subjects is syntactic, and hence that the two verbs and the two subjects in any tag question must all agree. This view, however, is inconsistent with well known data like (70):

- (70) a. Sears is having a sale, aren't they?
 b. At least one of us is sure to win, aren't we?

⁶⁴This is not to claim that the stringset of an SBCG must be a context-free language.

c. The crowd is getting agitated, aren't they?

Following Oehrle (1987) and Culicover (1992), Kay (2002a) argues that the agreement between the two subjects here is semantic in nature, whereas the agreement between each verb and its subject is syntactic in nature. Notice, however, that in any analysis positing a structure for tags along the lines shown in (71), the agreement relation between the two subjects is non-local, i.e. it involves agreement between two elements that are not sisters:



By positing an analysis wherein a clausal sign includes an XARG value reflecting the agreement properties of the clause's subject, we make it possible to treat the agreement in tag questions locally, i.e. via a constraint requiring the relevant identity (coindexing) between the XARG value of the main clause and the pronominal XARG value of the tag clause (the NPs that are shaded in (71)).

Many researchers have pointed out phenomena in diverse languages that motivate the propagation of external argument information of the sort just illustrated. One of these is English 'copy raising' (Rogers 1974, Potsdam and Runner 2001, Asudeh 2002), illustrated in (72):⁶⁵

- (72) a. There looks like $\left\{ \begin{array}{l} \text{there's going to be a storm} \\ \text{*it's going to rain} \\ \text{*Kim's going to win} \end{array} \right\}.$
- b. ?Kim looks like $\left\{ \begin{array}{l} \text{there's going to be a storm} \\ \text{it's going to rain} \\ \text{Pat's going to win} \end{array} \right\}.$

⁶⁵The judgments here are nuanced, as *looks (like)* is systematically ambiguous between a copy raising verb and a dyadic predicator where the subject is assigned a semantic role. For this reason, the examples in (72b) are possible on the latter reading, though the acceptability is slightly degraded.

Assuming, following Pollard and Sag (1994) that there are three subtypes of the type *index – ref* (*referential-index*), *it* (*expletive-it-index*), and *there* (*expletive-there-index*) – contrasts like these can be treated simply by associating the relevant *look* lexeme with the ARG-ST list in (73):⁶⁶

(73) [ARG-ST ⟨ NP_i, PRT[*like*], S[XARG ⟨ NP_i[*pro*]] ⟩]

Also relevant are controlled pronominal subjects in Serbo-Croatian (Zec 1987), Halkomelem Salish (Gerdtz and Hukari 2001) and other languages, where control verbs also include the ARG-ST specification in (73). The problems of raising across Polish prepositions (Przepiórkowski 1999, Dickinson 2004) and complementizer agreement in Eastern Dutch dialects (Höhle 1997) are similar, and submit to similar analysis.

Finally, there are many English idioms that require referential and agreement identity between a subject and a possessor within an object NP, or which assign a semantic role to the object's possessor. These are illustrated in (74)–(75):

(74) a. He_i lost [his_i/*her_j marbles].

b. They_i kept [their_i/*our_j cool].

(75) a. That_i made [her_j hair] stand on end.

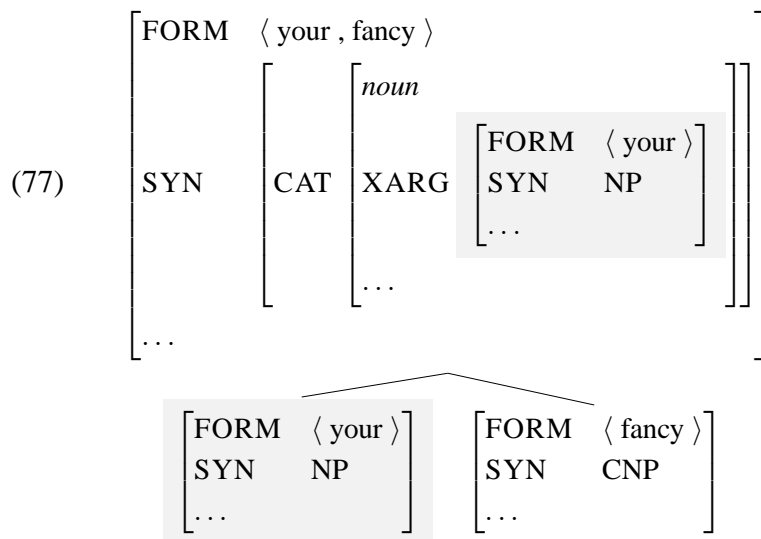
b. That_i tickled [your_j fancy].

If an object NP includes information about its (prenominal) possessor in its XARG value, then an idiomatic verb like *lose* can be specified as in (76):

(76) [ARG-ST ⟨ NP_i, NP[XARG ⟨ NP_i[*pro*]] ⟩]

And, similarly, a verb like *tickle* can assign a semantic role to its object's possessor. In both cases, all that is required is that the NP's XARG value be identified with the NP's possessor, as shown in (77):

⁶⁶NP_i[*pro*] indicates a pronominal noun phrase. An alternate analysis would treat *like* as a marker that combines with an infinitival VP.



All of the phenomena just enumerated are beyond the analytic scope of the version of HPSG developed by Pollard and Sag (1987, 1994). As argued in Sag to appear b, these data provide motivation for XARG specifications as part of the CAT value of sentential and NP signs. Note that XARG values, unlike VAL lists, do not shrink as larger phrases are constructed. That is, elements are not ‘cancelled off’ an XARG list. This proposal is an extension of earlier proposals that have been made within HPSG,⁶⁷ synthesizing them into SBCG.

As noted above [(47)], we assume that the XARG value of a verbal lexeme is also the first member of the verb’s ARG-ST list and its VAL list, as shown in the following lexeme licensed by the lexical entry for *love*:

⁶⁷Pollard’s (1994) ERG feature is an early proposal of a noncancelling feature coding a dependency relation, based on unpublished ideas of Andreas Kathol’s. Kiss (1995) introduced a feature for the subject of German verbal clauses and called it SUBJECT; this is the feature used by Meurers (1999) and others. However, since we also use this in the analysis of NPs, in order to make possessor NPs available for external selection, the more neutral name XARG is preserved.

(78)

FORM	< love >																
SYN	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">CAT</td> <td style="padding-left: 10px;"> <table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">XARG</td> <td style="padding-left: 10px;">NP_i[...]</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">SELECT</td> <td style="padding-left: 10px;">< ></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">...</td> <td></td> </tr> </table> </td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">MRKG</td> <td style="padding-left: 10px;"><i>unmk</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">VAL</td> <td style="padding-left: 10px;">< NP_i[...] , NP_j[...] ></td> </tr> </table>	CAT	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">XARG</td> <td style="padding-left: 10px;">NP_i[...]</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">SELECT</td> <td style="padding-left: 10px;">< ></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">...</td> <td></td> </tr> </table>	XARG	NP _i [...]	SELECT	< >	...		MRKG	<i>unmk</i>	VAL	< NP _i [...] , NP _j [...] >				
CAT	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">XARG</td> <td style="padding-left: 10px;">NP_i[...]</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">SELECT</td> <td style="padding-left: 10px;">< ></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">...</td> <td></td> </tr> </table>	XARG	NP _i [...]	SELECT	< >	...											
XARG	NP _i [...]																
SELECT	< >																
...																	
MRKG	<i>unmk</i>																
VAL	< NP _i [...] , NP _j [...] >																
ARG-ST	< NP _i [...] , NP _j [...] >																
SEM	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">INDEX</td> <td style="padding-left: 10px;"><i>s</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">FRAMES</td> <td style="padding-left: 10px;"> <table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;"><i>love-fr</i></td> <td style="padding-left: 10px;"></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">ACTOR</td> <td style="padding-left: 10px;"><i>i</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">UNDGR</td> <td style="padding-left: 10px;"><i>j</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">SIT</td> <td style="padding-left: 10px;"><i>s</i></td> </tr> </table> </td> </tr> </table>	INDEX	<i>s</i>	FRAMES	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;"><i>love-fr</i></td> <td style="padding-left: 10px;"></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">ACTOR</td> <td style="padding-left: 10px;"><i>i</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">UNDGR</td> <td style="padding-left: 10px;"><i>j</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">SIT</td> <td style="padding-left: 10px;"><i>s</i></td> </tr> </table>		<i>love-fr</i>			ACTOR	<i>i</i>		UNDGR	<i>j</i>		SIT	<i>s</i>
INDEX	<i>s</i>																
FRAMES	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;"><i>love-fr</i></td> <td style="padding-left: 10px;"></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">ACTOR</td> <td style="padding-left: 10px;"><i>i</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">UNDGR</td> <td style="padding-left: 10px;"><i>j</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"></td> <td style="padding-left: 10px;">SIT</td> <td style="padding-left: 10px;"><i>s</i></td> </tr> </table>		<i>love-fr</i>			ACTOR	<i>i</i>		UNDGR	<i>j</i>		SIT	<i>s</i>				
	<i>love-fr</i>																
	ACTOR	<i>i</i>															
	UNDGR	<i>j</i>															
	SIT	<i>s</i>															
...																	

By contrast, semantically inert (‘case-marking’) prepositions like *to* or *of* have no external argument, and are specified as shown in (79):

(79)

FORM	< of >											
SYN	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">CAT</td> <td style="padding-left: 10px;"> <table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"><i>prep</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">XARG</td> <td style="padding-left: 10px;"><i>none</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">SELECT</td> <td style="padding-left: 10px;">< ></td> </tr> </table> </td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">VAL</td> <td style="padding-left: 10px;">< NP[...] ></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">MRKG</td> <td style="padding-left: 10px;"><i>unmk</i></td> </tr> </table>	CAT	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"><i>prep</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">XARG</td> <td style="padding-left: 10px;"><i>none</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">SELECT</td> <td style="padding-left: 10px;">< ></td> </tr> </table>	<i>prep</i>	XARG	<i>none</i>	SELECT	< >	VAL	< NP[...] >	MRKG	<i>unmk</i>
CAT	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 10px;"><i>prep</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">XARG</td> <td style="padding-left: 10px;"><i>none</i></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 10px;">SELECT</td> <td style="padding-left: 10px;">< ></td> </tr> </table>	<i>prep</i>	XARG	<i>none</i>	SELECT	< >						
<i>prep</i>												
XARG	<i>none</i>											
SELECT	< >											
VAL	< NP[...] >											
MRKG	<i>unmk</i>											
ARG-ST	< NP[...] >											
SEM	[FRAMES < >]											
...												

4.2.3 A Head-Complement Construction

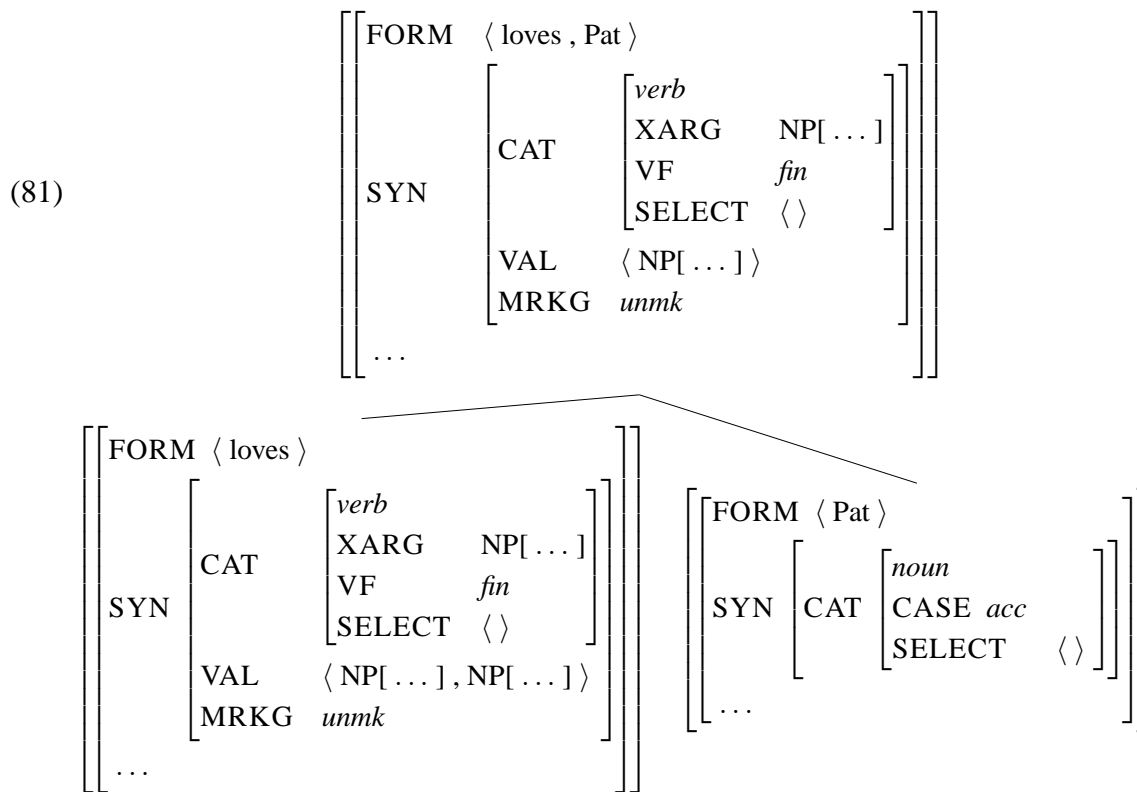
With these lexical contrasts in place, we can now analyze the relevant properties of VPs and predicational APs and PPs. I will assume the head-complement construction sketched in (80):⁶⁸

⁶⁸Here, we assume a distinct head-complement construction for English ‘case-marking’ prepositions, which have no subject or external argument. These elements combine with their object valent to form a valence-saturated ([VAL < >]) phrase.

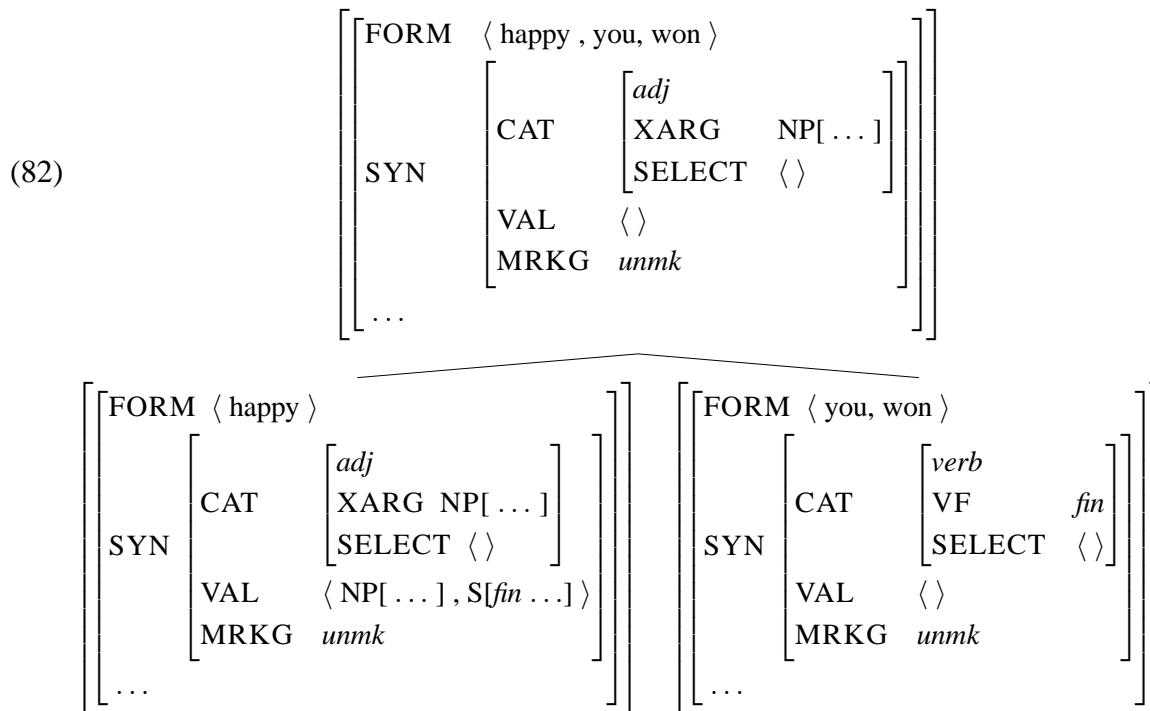
(80) **Predicational Head-Complement Construction** (preliminary version):

$$pred-hd-comp-cxt \Rightarrow \left[\begin{array}{l} \text{MTR} \left[\text{SYN} \left[\begin{array}{l} \text{VAL} \langle Y \rangle \\ \text{MRKG} X: unmk \end{array} \right] \right] \\ \text{DTRS} \left\langle \left[\begin{array}{l} \text{word} \\ \text{SYN} \left[\begin{array}{l} \text{VAL} \langle Y \rangle \oplus L \\ \text{MRKG} X \end{array} \right] \end{array} \right] \right\rangle \oplus L:nelist \end{array} \right]$$

What (80) says is that a predicational head-complement construct must involve a mother whose MARKING value matches that of the first daughter (the head daughter). The first daughter, in addition, must be unmarked and followed by all of the valents that it selects, except for the first (the subject). The mother's VAL value is the singleton list containing the head daughter's first valent. A grammar that includes this construction licenses constructs like the following:⁶⁹



⁶⁹Here, we omit SEMANTICS as well as PHONOLOGY and CONTEXT.



4.2.4 Headed Constructs

These constructs illustrate another familiar property of headed constructions: that the category properties of the head daughter and its mother are shared. This sharing is guaranteed by a constraint often referred to as the **Head Feature Principle** (HFP), the essential ingredient of all work in X-Bar Theory. Our use of the HFP builds directly on Pollard and Sag's (1987, 1994) reformulation of the Head Feature Convention of Gazdar et al. 1985. In order to state the HFP, we need to be able to identify the head daughter in phrasal constructs. We accommodate this need by introducing *headed-construct* as a subtype of *phr-ctx*:

- (83) a. *headed-construct* (*hd-ctx*) is an immediate subtype of *phr-ctx*.
- b. HD-DTR is used to specify the head daughter of a headed construct; the value of HD-DTR is of type *expression*.

The two phrasal constructions that we have considered thus far – the Subject Predicate Construction and the Predicational Head-Complement Construction – both license headed constructs. In the former case, the second daughter (the VP) is the head daughter; in the latter case, the head is the first daughter (which is of type *word*). The final versions of these two constructions are given in (84):

(84) a. **Predicational Head-Complement Construction** (final version):

$$pred-hd-comp-cxt \Rightarrow \left[\begin{array}{l} \text{MTR} \quad \left[\text{SYN} \left[\begin{array}{l} \text{VAL} \langle X \rangle \\ \text{MRKG} \quad M: unmk \end{array} \right] \right] \\ \text{DTRS} \quad \langle H \rangle \oplus L_2:nelist \\ \text{HD-DTR} \quad H: \left[\begin{array}{l} \text{word} \\ \text{SYN} \left[\begin{array}{l} \text{VAL} \langle X \rangle \oplus L_2 \\ \text{MRKG} \quad M \end{array} \right] \end{array} \right] \end{array} \right]$$

b. **Subject-Predicate Construction** (final version):

$$subjpred-cxt \Rightarrow \left[\begin{array}{l} \text{MTR} \quad \left[\text{SYN} \left[\begin{array}{l} \text{VAL} \langle \rangle \\ \text{MRKG} \quad M \end{array} \right] \right] \\ \text{DTRS} \quad \langle X, H \rangle \\ \text{HD-DTR} \quad H: \left[\text{SYN} \left[\begin{array}{l} \text{CAT} [\text{VF} \quad \textit{fn}] \\ \text{VAL} \langle X \rangle \\ \text{MRKG} \quad M: unmk \end{array} \right] \right] \end{array} \right]$$

Recall that constraints on the type *phr-cxt* given in (65) above require the mother to be of type *phrase*, hence (since *hd-cxt* is a subtype of *phr-cxt*) constructs licensed by the constructions in (84) are so constrained without the need for further stipulation.

The Head Feature Principle can now be stated as a construction, i.e. as a type constraint restricting the well-formedness of headed constructs:

(85) **Head Feature Principle:**

$$hd-cxt \Rightarrow \left[\begin{array}{l} \text{MTR} \quad \left[\text{SYN} [\text{CAT} \quad \square] \right] \\ \text{HD-DTR} \quad \left[\text{SYN} [\text{CAT} \quad \square] \right] \end{array} \right]$$

(The category of a phrase and its head daughter are identical.)

Notice that MRKG and VAL are syntactic features, but are not part of the CAT value, and hence are not covered by the Head Feature Principle (HFP). The resulting analysis is illustrated by the derivation tree in Figure 5, where the effects of the Head Feature Principle are highlighted.

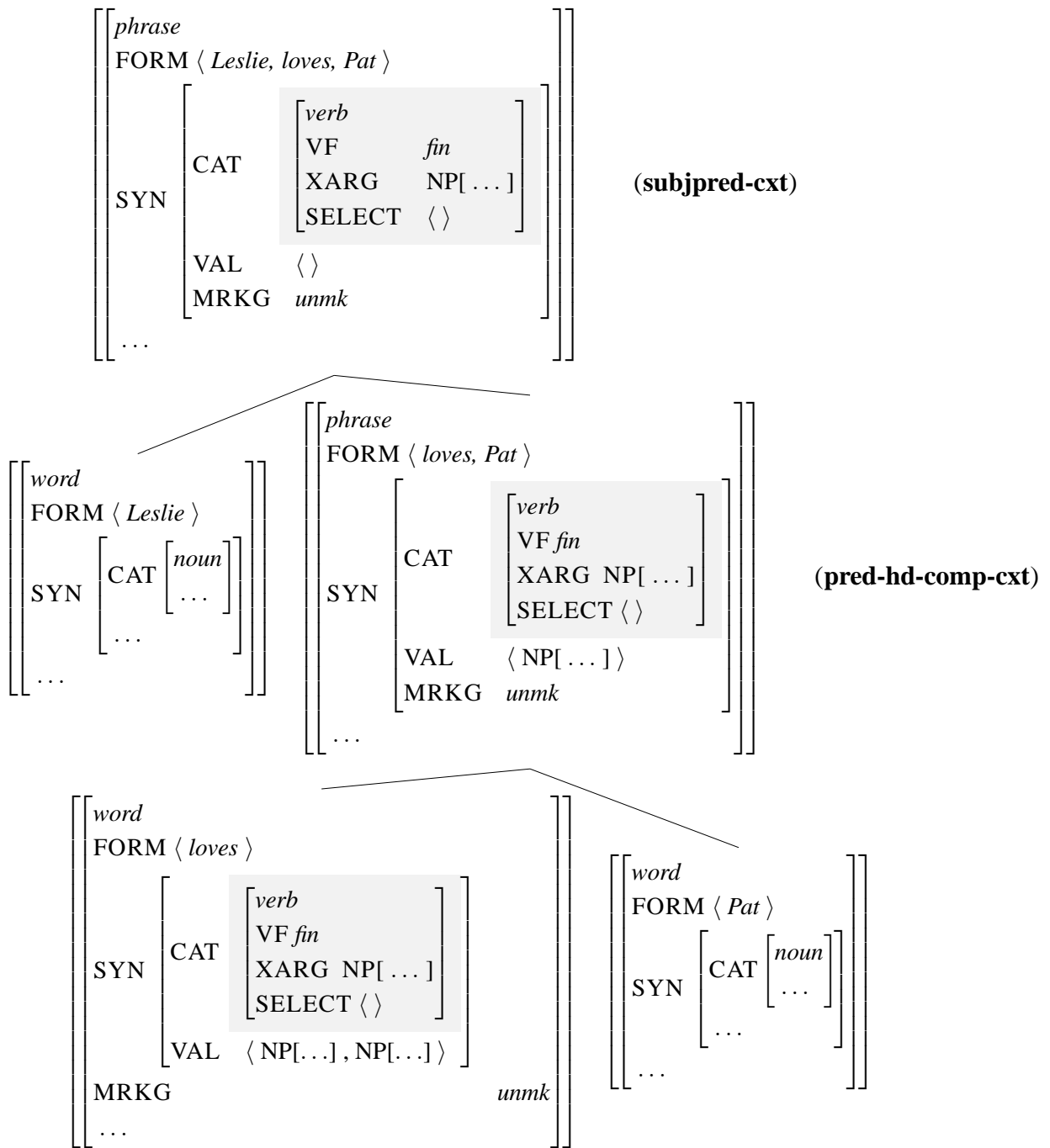


Figure 5: Derivation Tree for *Leslie loves Pat*

4.2.5 The Head-Functor Construction

We now turn to the Head-Functor Construction. We follow the essential insights of Van Eynde (2006a,b), who argues that significant generalizations in the grammar of nominals are missed by analyses based on so-called ‘functional categories’.⁷⁰ In their place, he offers a unified analysis of markers (including determiners) and modifiers in terms of a simple, direct combination of a ‘functor’ expression and the head that it selects, based on the SELECT feature, discussed in section 3.3.1 above.

All major categories specify values for SELECT in Van Eynde’s theory: nouns, adjectives, adverbs, prepositions, and verbs. For some of these, e.g. finite verbs, the value is $\langle \rangle$. Attributive adjectives, by contrast, select nominal heads; complementizers (whose category assignment Van Eynde does not discuss) select verbal heads, as illustrated in (86):

$$(86) \quad \left[\begin{array}{l} \text{FORM} \quad \langle \text{happy} \rangle \\ \text{SYN} \quad \left[\begin{array}{l} \text{CAT} \quad \left[\begin{array}{l} \text{SELECT} \quad \left[\text{SYN} \left[\begin{array}{l} \text{CAT} \quad \textit{noun} \\ \text{MRKG} \quad \textit{unmk} \end{array} \right] \right] \\ \text{MRKG} \quad \textit{unmk} \end{array} \right] \end{array} \right] \end{array} \right]$$

$$\left[\begin{array}{l} \text{FORM} \quad \langle \text{that} \rangle \\ \text{SYN} \quad \left[\begin{array}{l} \text{CAT} \quad \left[\begin{array}{l} \text{SELECT} \quad \left[\text{SYN} \left[\begin{array}{l} \text{CAT} \quad \textit{verb} \\ \text{VAL} \quad \langle \rangle \\ \text{MRKG} \quad \textit{unmk} \end{array} \right] \right] \\ \text{MRKG} \quad \textit{that} \end{array} \right] \end{array} \right] \end{array} \right]$$

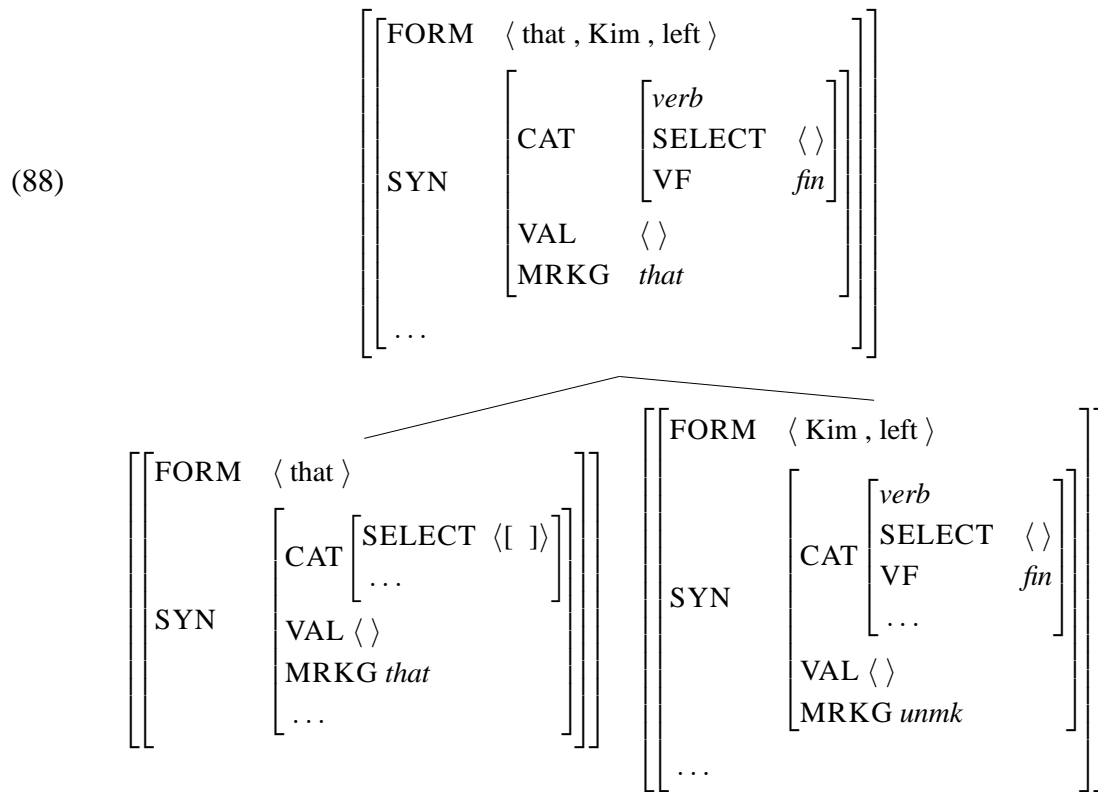
Given these lexical specifications, we can formulate the Head-Functor Construction as follows:

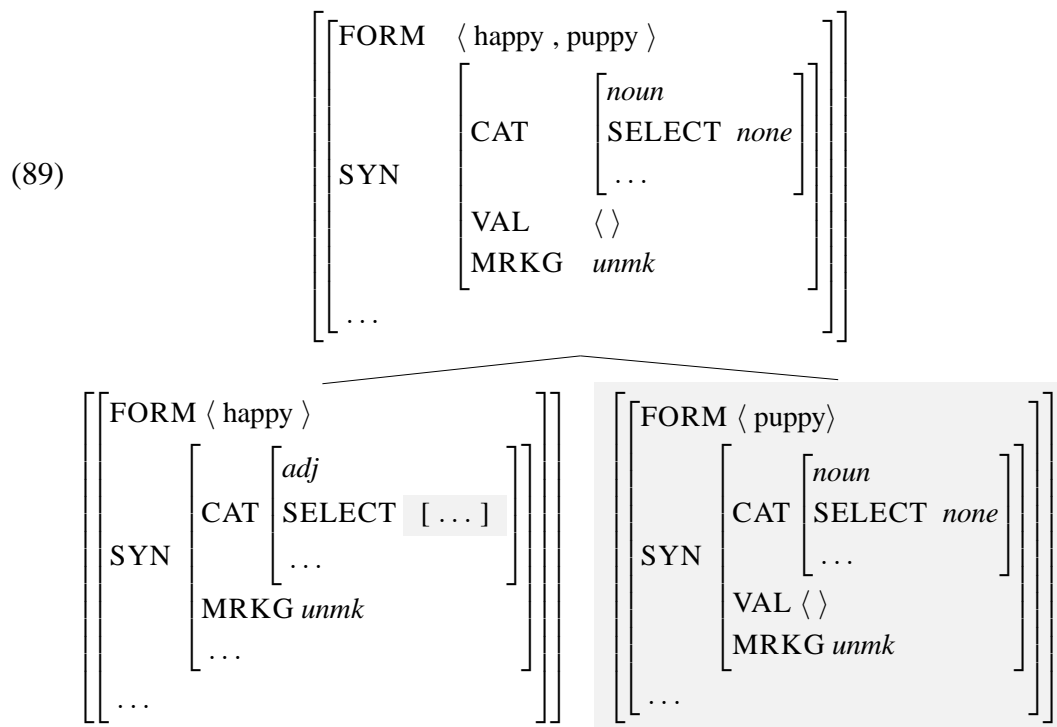
(87) **Head-Functor Construction:**

$$hd\text{-func}\text{-cxt} \Rightarrow \left[\begin{array}{l} \text{MTR} \quad \left[\text{SYN} \left[\begin{array}{l} \text{VAL} \quad L \\ \text{MRKG} \quad M \end{array} \right] \right] \\ \text{DTRS} \quad \left\langle \left[\text{SYN} \left[\begin{array}{l} \text{CAT} \quad [\text{SELECT } X] \\ \text{MRKG} \quad M \end{array} \right] \right] \right\rangle, X \\ \text{HD-DTR} \quad X: [\text{SYN} [\text{VAL } L]] \end{array} \right]$$

⁷⁰See also Hudson 2000, 2004

This construction allow us to construct both marked clauses and modified nominal phrases, as shown in (88)–(89), as well as ordinary NPs containing determiners. Expressions like **that for Kim to leave* and **happy the puppy* are blocked because the relevant lexical entries for *that* and *happy* ensure that they select unmarked elements.





Note that in each of these constructs, the mother's SELECT specification is inherited from the head daughter, in accordance with the Head Feature Principle in (85) above.

5 Conclusion

This essay has tried to achieve a number of goals:

1. providing tools to overcome certain analytic inadequacies of the versions of HPSG developed in Pollard and Sag 1987, Pollard and Sag 1994, and Ginzburg and Sag 2000,
2. incorporating key insights from work conducted within the Construction Grammar community, and
3. modifying the framework of HPSG so as to incorporate a theory of selectional locality [(67) above] and constructional locality [(69) above].

I hope that the framework sketched here has made some progress on all three fronts.

Appendix

Hierarchy of Types	
Type	Immediate Subtypes
<i>feature-structure</i>	<i>atom, function</i>
<i>atom</i>	<i>case, vform, boolean, cat, mrk, formative, frame, index</i>
<i>formative</i>	<i>lex-form, aff</i>
<i>frame</i>	<i>actor-fr, undgr-fr, quant-fr, ...</i>
<i>actor-fr</i>	<i>actor-only-fr, act-undgr-fr, ...</i>
<i>undgr-fr</i>	<i>undgr-only-fr, act-undgr-fr, ...</i>
<i>act-undgr-fr</i>	<i>spr-ld-fr, love-fr, ...</i>
<i>quant-fr</i>	\exists -fr; ...
<i>function</i>	<i>sign, syn-obj, sem-obj, cat, cxt, frame, phon-phr</i>
<i>sign</i>	<i>lex-sign, expression</i>
<i>expression</i>	<i>phrase, word</i>
<i>lex-sign</i>	<i>lexeme or word</i>
<i>lexeme</i>	<i>verb-lxm, noun-lxm, ...</i>
<i>verb-lxm</i>	<i>intr-verb-lxm, tran-verb-lxm</i>
<i>intr-verb-lxm</i>	<i>siv-lxm, srv-lxm, scv-lxm, ...</i>
<i>tran-verb-lxm</i>	<i>stv-lxm, orv-lxm, ocv-lxm, dtv-lxm, spray-load1-lxm, spray-load2-lxm, ...</i>
<i>cxt</i>	<i>phr-cxt, lex-cxt</i>
<i>phr-cxt</i>	<i>hd-cxt, unhd-cxt</i>
<i>hd-cxt</i>	<i>pred-hd-comp-cxt, subjpred-cxt, hd-func-cxt, ...</i>
<i>lex-cxt</i>	<i>deriv-cxt, infl-cxt, pinfl-cxt</i>
<i>deriv-cxt</i>	<i>unverb-cxt, ...</i>
<i>infl-cxt</i>	<i>preterite-cxt, ...</i>
<i>pinfl-cxt</i>	<i>extra-cxt, ...</i>

Atomic Type Assignments	
Type	Instances
<i>case</i>	<i>nom, acc, obl</i>
<i>vform</i>	<i>fin, prp, psp, base, pass</i>
<i>boolean</i>	<i>+, -</i>
<i>cat</i>	<i>noun, verb, prep, adj, conj, ...</i>
<i>mrk</i>	<i>that, whether, than, det, ...</i>
<i>index</i>	<i>i, j, k, s, t, ...</i>

Type Declarations			
Type	Appropriate Features & Value Type		
<i>sign</i>	PHON	<i>phon-phr</i>	
	FORM	<i>list(formative)</i>	
	SYN	<i>syn-obj</i>	
	SEM	<i>sem-obj</i>	
	CNTXT	<i>context</i>	
<i>lex-sign</i>	<i>sign</i> &	ARG-ST	<i>list(sign)</i>
<i>syn-obj</i>	CAT	<i>cat</i>	
	VAL	<i>list(sign)</i>	
	MRKG	<i>mrk</i>	
<i>sem-obj</i>	IND	<i>index</i>	
	FRAMES	<i>list(frame)</i>	
<i>frame</i>	SIT	<i>index</i>	
<i>actor-fr</i>	<i>frame</i> &	ACTOR	<i>index</i>
<i>undgr-fr</i>	<i>frame</i> &	UNDGR	<i>index</i>
<i>spr-ld-fr</i>	<i>act-undgr-fr</i> &	GOAL	<i>index</i>
<i>quant-fr</i>	BV	<i>index</i>	
	...		
<i>cat</i>	XARG	<i>sign</i>	
	SELECT	<i>list(sign)</i>	
	LID	<i>frame</i>	
<i>noun</i>	<i>cat</i> &	CASE	<i>case</i>
<i>verb</i>	<i>cat</i> &	VF	<i>vform</i>
		AUX	<i>boolean</i>
<i>context</i>	C-INDS	<i>c-inds</i>	
	BCKGRND	<i>list(frame)</i>	
<i>cxt</i>	MTR	<i>sign</i>	
	DTRS	<i>list(sign)</i>	
<i>phr-cxt</i>	<i>cxt</i> &	MTR	<i>phrase</i>
		DTRS	<i>list(expression)</i>
<i>hd-cxt</i>	<i>phr-cxt</i> &	HD-DTR	<i>expression</i>
<i>lex-cxt</i>	<i>cxt</i> &	DTRS	<i>list(lex-sign)</i>

References

- Abeillé, A., Bonami, O., Godard, D. and Tseng, J. 2006. The Syntax of French *à* and *de*: an HPSG Analysis. In P. Saint-Dizier (ed.), *Syntax and Semantics of Prepositions*, pp 147–162, Dordrecht: Springer.
- Ajdukiewicz, K. 1935. Die syntaktische Konnexität. *Studia Philosophica* 1, 1–27.
- Anderson, S. 1992. *Amorphous Morphology*. Cambridge: Cambridge University Press.
- Andrews, A. 1982. The Representation of Case in Modern Icelandic. In J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*, pp 427–503, Cambridge, MA: The MIT Press.
- Andrews, A. 1990. Case Structures and Control in Modern Icelandic. In J. Maling and A. Zanen (eds.), *Modern Icelandic Syntax*, pp 187–234, New York: Academic Press.
- Asudeh, A. 2002. Richard III. In A. P. Mary Andronis, Erin Debenport and K. Y. a (eds.), *Proceedings of the 25th Regional Meetings of the Chicago Linguistics Society*, Chicago: Chicago Linguistics Society.
- Asudeh, A. and Klein, E. 2002. Shape Conditions and Phonological Context. In F. V. Eynde, L. Hellan and D. Beermann (eds.), *The Proceedings of the 8th International Conference on Head-Driven Phrase Structure Grammar*, pp 20–30, Stanford: CSLI Publications.
- Bauer, L. and Reynaud, A. 2001. A corpus-based study of compounding in English. *Journal of English Linguistics* 29, 101–123.
- Bender, E. and Flickinger, D. P. 1999. Peripheral Constructions and Core Phenomena: Agreement in Tag Questions. In G. Webelhuth, J.-P. Koenig and A. Kathol (eds.), *Lexical and Constructional Aspects of Linguistic Explanation*, pp 199–214, Stanford: CSLI Publications.
- Bird, S. and Klein, E. 1994. Phonological Analysis in Typed Feature Systems. *Computational Linguistics* 20(3), 455–491.
- Bonami, O. and Boyé, G. 2006. Deriving Inflectional Irregularity. In S. Müller (ed.), *The Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar*, pp 361–380, Stanford: CSLI Publications.
- Bouma, G., Malouf, R. and Sag, I. A. 2001. Satisfying Constraints on Extraction and Adjunction. *Natural Language and Linguistic Theory* 1(19), 1–65.
- Bouma, G. and van Noord, G. 1994. Constraint-based Categorical Grammar. In *32th Annual Meeting of the Association for Computational Linguistics*, New Mexico.
- Bresnan, J. 1982. The Passive in Lexical Theory. In J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*, pp 3–86, MIT Press.
- Bresnan, J. W. 2001. *Lexical-Functional Syntax*. Oxford: Basil Blackwell's.
- Carpenter, B. 1992. *The Logic of Typed Feature Structures with Applications to Unification-based Grammars, Logic Programming and Constraint Resolution*. New York: Cambridge University Press.
- Chomsky, N. 1955. The Logical Structure of Linguistic Theory, ms., Society of Fellows, Harvard University. Published in 1975 as *The Logical Structure of Linguistic Theory* by Plenum. Now available from the University of Chicago Press, Chicago, Illinois.
- Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton.
- Chomsky, N. 1965. *Aspects of the Theory of Syntax*. MIT Press.

- Chomsky, N. 1970. Remarks on Nominalization. In R. Jacobs and P. S. Rosenbaum (eds.), *Readings in English Transformational Grammar*, Waltham, Massachusetts: Ginn and Co.
- Chomsky, N. 1974. Questions of Form and Interpretation. Paper delivered at LSA Golden Anniversary Symposium, Amherst, Massachusetts, July 1974. Montreal Working Papers in Linguistics 3, December 1974. Reprinted in *The Scope of American Linguistics*, edited by R. Austerlitz. Lisse: Peter De Ridder Press, 1975; reprint. *Linguistic Analysis* 1:1 (1975): 75-109. Reprinted as "The Amherst Lectures." Université de Paris VII, Département de Recherches Linguistiques, February 1975.
- Chomsky, N. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.
- Chomsky, N. and Halle, M. 1968. *The Sound Pattern of English*. New York: Harper and Row.
- Clark, E. V. and Clark, H. H. 1979. When Nouns Surface as Verbs. *Language* 55, 776–811.
- Copestake, A. 1992. The Representation of Lexical Semantic Information. Cognitive Science Research Papers 280, University of Sussex.
- Copestake, A., Flickinger, D., Pollard, C. J. and Sag, I. A. 2005. Minimal Recursion Semantics: an Introduction. *Research on Language and Computation* 3(4), 281–332.
- Copestake, A. and Lascarides, A. 1997. Integrating Symbolic and Statistical Representations: the Lexicon Pragmatics Interface. In *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*, Madrid: Association for Computational Linguistics.
- Culicover, P. 1992. English Tag Questions in Universal Grammar. *Lingua* 88, 193–226.
- Culicover, P. 1999. *Syntactic Nuts: Hard Cases, Syntactic Theory and Language Acquisition*. Oxford: Oxford University Press.
- Culicover, P. and Jackendoff, R. 2005. *Simpler Syntax*. Oxford University Press.
- Dalrymple, M., Kaplan, R., Maxwell, J. and Zaenen, A. (eds.). 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford: CSLI Publications.
- Daniels, M. and Meurers, D. 2004. GIDL P: A Grammar Format for Linearization-Based HPSG. In S. Müller (ed.), *Proceedings of the HPSG-2004 Conference, Center for Computational Linguistics, Katholieke Universiteit Leuven*, pp 93–111, Stanford: CSLI Publications.
- Davis, A. R. 2001. *Linking by Types in the Hierarchical Lexicon*. CSLI Publications.
- de Saussure, F. 1916. *Course in General Linguistics*. by McGraw Hill, published in 1959.
- Dickinson, M. 2004. Polish Numeral Phrases and Predicative Modification. Unpublished manuscript, Georgetown University.
- Donohue, C. and Sag, I. A. 1999. Domains in Warlpiri. In *Sixth International Conference on HPSG—Abstracts. 04–06 August 1999*, pp 101–106, Edinburgh.
- Downing, P. 1977. On the Creation and Use of English Compound Nouns. *Language* 53, 810–842.
- Engdahl, E. and Vallduví, E. 1994. Information Packaging and Grammar Architecture: A Constraint-Based Approach. In E. Engdahl (ed.), *Integrating Information Structure into Constraint-Based and Categorical Approaches*, pp 39–79, Amsterdam: ILLC, dYANA-2 Report R.1.3.B.
- Engdahl, E. and Vallduví, E. 1996. Information Packaging in HPSG. In C. Grover and E. Vallduví (eds.), *Edinburgh Working Papers in Cognitive Science, Vol. 12: Studies in HPSG*,

- Chapter 1, pp 1–32, Scotland: Centre for Cognitive Science, University of Edinburgh.
- Fenstad, J., Halvorsen, P.-K., Langholm, T. and Benthem, J. V. 1987. *Situations, Language and Logic*. Dordrecht: Reidel.
- Fillmore, C. 1999. Inversion and constructional inheritance. In G. Webelhuth, J.-P. Koenig and A. Kathol (eds.), *Lexical and Constructional Aspects of Linguistic Explanation*, Chapter 21, pp 113–128, Stanford: CSLI Publications.
- Fillmore, C., Kay, P. and O’Connor, M. 1988. Regularity and Idiomaticity in Grammatical Constructions: The Case of *Let Alone*. *Language* 64, 501–538.
- Fillmore, C. J. 1985. Frames and the Semantics of Understanding. *Quaderni di Semantica* 6, 222–254.
- Fillmore, C. J. 1986. Pragmatically Controlled Zero Anaphora. In *Proceedings of the Twelfth Annual Meeting of the Berkeley Linguistics Society*, pp 95–107, BLS.
- Fillmore, C. J. and Kay, P. 1996. Construction Grammar Coursebook, unpublished manuscript, UC Berkeley.
- Gazdar, G. 1981. Unbounded Dependencies and Coordinate Structure. *Linguistic Inquiry* 12, 155–84.
- Gazdar, G., Klein, E., Pullum, G. K. and Sag, I. A. 1985. *Generalized Phrase Structure Grammar*. Oxford: Basil Blackwell’s and Cambridge, MA: Harvard University Press.
- Gazdar, G. and Pullum, G. K. 1981. Subcategorization, Constituent Order and the Notion of “Head”. In Moortgat, H. van der Hulst and Hoekstra (eds.), *The Scope of Lexical Rules*, pp 107–123, Dordrecht: Foris.
- Gerdts, D. and Hukari, T. 2001. A-Subjects and Control in Halkomelem. In D. Flickinger and A. Kathol (eds.), *The Proceedings of the 7th International Conference on Head-Driven Phrase Structure Grammar*, pp 100–123, Stanford: CSLI Publications.
- Ginzburg, J. and Cooper, R. 2004. Clarification, Ellipsis, and the Nature of Contextual Updates. *Linguistics and Philosophy* 27(3), 297–366.
- Ginzburg, J. and Sag, I. A. 2000. *Interrogative Investigations: the form, meaning, and use of English Interrogatives*. Stanford, California: CSLI Publications.
- Goldberg, A. 1995. *A Construction Grammar Approach to Argument Structure*. Chicago: University of Chicago Press.
- Green, G. M. 1996. The Structure of CONTEXT: The Representation of Pragmatic Restrictions in HPSG. In J. Yoon (ed.), *Studies in the Linguistic Sciences. Proceedings of the 5th annual meeting of the Formal Linguistics Society of the Midwest*.
- Green, G. M. 2000. The Nature of Pragmatic Information. In R. Cann, C. Grover and P. Miller (eds.), *Grammatical Interfaces in HPSG*, pp 113–138, Stanford: CSLI Publications.
- Haji-Abdolhosseini, M. 2003. A Constraint-Based Approach to Information Structure and Prosody Correspondence. In S. Müller (ed.), *Proceedings of the HPSG-2003 Conference, Michigan State University, East Lansing*, pp 143–162, Stanford: CSLI Publications.
- Hankamer, J. 1973. Why there are two *thans* in English. In C. Corum, T. Smith-Stark and A. Weiser (eds.), *Papers from the 25th Regional Meeting of the Chicago Linguistic Society*, pp 179–191, CLS, Chicago, Illinois: CLS.
- Harrison, A. J. 2001. Mice Traps: Plural Inflection in English. MSc. Unpublished MSc dissertation, School of Cognitive Science, University of Edinburgh.
- Höhle, T. N. 1997. Vorangestellte Verben und Komplementierer sind eine Natürliche Klasse.

- In K. R. Durscheid, C. and M. Schwarz (eds.), *Sprache im Fokus: Festschrift für Heinz Vater zum 65. Geburtstag*, pp 107–120, Max Niemeyer Verlag.
- Hudson, R. 2000. Grammar without Functional Categories. In R. Borsley (ed.), *The Nature and Function of Syntactic Categories*, pp 7–35, Academic Press.
- Hudson, R. 2004. Are Determiners Heads? *Functions of Language* 11, 7–42.
- Israel, M. 1996. Scalar Implicature as Lexical Semantics. *Linguistics and Philosophy* 19, 619–666.
- Jackendoff, R. S. 1977. *X' Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT press.
- Jackendoff, R. S. 1997. *The Architecture of the Language Faculty*. Cambridge, MA: MIT Press.
- Jackendoff, R. S. 2002. *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press.
- Johnson, D. 1977. Remarks and Replies: On Keenan's Definition of "Subject of". *Linguistic Inquiry* 8(4), 673–692.
- Kajita, M. 1968. *A Generative-Transformational Study of Semi-Auxiliaries in Present-Day American English*. Tokyo: Sanseido.
- Kaplan, D. 1989. Demonstratives: An Essay on the Semantics, Logic, Metaphysics, and Epistemology of Demonstratives and Other Indexicals. In J. Almog, J. Perry and H. Wettstein (eds.), *Themes from Kaplan*, pp 481–614, Oxford: Oxford University Press.
- Kaplan, R. M. and Bresnan, J. 1982. Lexical-Functional Grammar: A Formal System for grammatical representation. In J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*, pp 173–281, MIT Press, reprinted in Mary Dalrymple, Ronald Kaplan, John Maxwell, and Annie Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*. Stanford: CSLI Publications. Pages 29–130.
- Kathol, A. 2000. *Linear Syntax*. New York, Oxford: Oxford University Press.
- Kay, M. 1979. Functional Grammar. In *Proceedings of the 5th Annual Meetings of the Berkeley Linguistics Society*, Berkeley: Berkeley Linguistic Society.
- Kay, P. 1990. EVEN. *Linguistics and Philosophy* 13, 59–111.
- Kay, P. 1997. *Words and the Grammar of Context*. Stanford, CA: CSLI Publications.
- Kay, P. 2002a. English Subjectless Tagged Sentences. *Language* 78, 453–481.
- Kay, P. 2002b. An Informal Sketch of the Formal Architecture of Construction Grammar. *Grammars* 5, 1–19.
- Kay, P. and Fillmore, C. 1999. Grammatical Constructions and Linguistic Generalizations: The *What's X Doing Y?* Construction. *Language* 75, 1–33.
- Kay, P. and Zimmer, K. 1978. On the Semantics of Compounds and Genitives in English. In S. Tsohatzidis (ed.), *Meanings and Prototypes*, pp 239–246, Routledge.
- Keenan, E. and Comrie, B. 1977. Noun Phrase Accessibility and Universal Grammar. *Linguistic Inquiry* 8(1), 63–99.
- Kim, J.-B. 2000. *The Grammar of Negation: A Constraint-Based Approach*. Stanford, CA: CSLI Publications.
- Kim, J.-B. and Sag, I. A. 2002. French and English Negation without Head-Movement. *Natural Language and Linguistic Theory*.
- Kim, J.-B. and Sag, I. A. 2005. English Object Extraposition: A Constraint-Based Approach. In S. Müller (ed.), *The Proceedings of the 12th International Conference on Head-Driven*

- Phrase Structure Grammar, Department of Informatics, University of Lisbon*, pp 192–212, Stanford: CSLI Publications.
- Kiparsky, P. 1982. From Cyclic Phonology to Lexical Phonology. In H. van der Hulst and N. Smith (eds.), *The Structure of Phonological Representations (Part 1)*, Dordrecht: Foris.
- Kiss, T. 1995. *Infinite Komplementation*. Tübingen: Max Niemeyer Verlag.
- Klein, E. 2000. Prosodic Constituency in HPSG. In R. Cann, C. Grover and P. Miller (eds.), *Grammatical Interfaces in HPSG*, pp 171–203, Stanford: CSLI Publications.
- Koenig, J.-P. 1999. *Lexical Relations*. Stanford, CA: CSLI Publications.
- Kornai, A. and Pullum, G. K. 1990. The X-bar Theory of Phrase Structure. *Language* 66, 24–50.
- Levi, J. H. 1978. *The Syntax and Semantics of English Nominals*. New York: Academic Press.
- Levine, R. D. and Hukari, T. E. 2006. *The Unity of Unbounded Dependency Constructions*. Stanford University: CSLI Publications.
- Malouf, R. 2000. A head-driven account of long-distance case assignment. In *Grammatical Interfaces in HPSG*, pp 201–214, CSLI Publications.
- Manning, C. D. 1996. *Ergativity: Argument Structure and Grammatical Relations*. Stanford: CSLI Publications.
- Manning, C. D. and Sag, I. A. 1998. Argument Structure, Valence, and Binding. *Nordic Journal of Linguistics* 21.
- Matthews, P. H. 1991. *Morphology*. Cambridge: Cambridge University Press, second edition.
- McCawley, J. D. 1971. Where do Noun Phrases Come From? In D. D. Steinberg and L. A. Jakobovits (eds.), *Semantics: An Interdisciplinary Reader in Philosophy, Linguistics and Psychology*, pp 217–231, New York: Cambridge University Press.
- Meurers, W. D. 1999. German Partial-VP Fronting Revisited. In G. Webelhuth, J.-P. Koenig and A. Kathol (eds.), *Lexical and Constructional Aspects of Linguistic Explanation*, Stanford: CSLI Publications, (revised version of the paper presented at the 3rd Int. Conference on HPSG, Marseille, May 1996).
- Michaelis, L. A. and Lambrecht, K. 1996. Toward a Construction-Based Model of Language Function: The Case of Nominal Extraposition. *Language* 72, 215–247.
- Miller, P. 1992. *Clitics and Constituents in Phrase Structure Grammar*. New York: Garland.
- Miller, P. and Monachesi, P. 2003. Les pronoms clitiques dans les langues romanes. In D. Godard (ed.), *Les langues romanes: Problèmes de la phrase simple*, pp 67–123, Paris: Editions du CNRS.
- Miller, P. H. and Sag, I. A. 1997. French Clitic Movement without Clitics or Movement. *Natural Language and Linguistic Theory* 15(3), 573–639.
- Müller, S. 1995. Scrambling in German – Extraction into the *Mittelfeld*. In *Proceedings of the Tenth Pacific Asia Conference on Language, Information and Computation*, City University of Hong Kong.
- Müller, S. 1999. *Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche*. Tübingen: Max Niemeyer Verlag.
- Müller, S. 2002. Blockaden und Deblockaden: Perfekt, Passiv und Modale Infinitive. In D. Reitter (ed.), *Proceedings of TaCoS 2002*, Potsdam.
- Müller, S. 2004. An Analysis of Depictive Secondary Predicates in German without Discontinuous Constituents. In S. Müller (ed.), *Proceedings of the HPSG-2004 Conference*, Cen-

- ter for Computational Linguistics, Katholieke Universiteit Leuven, pp 202–222, Stanford: CSLI Publications.
- Oehrle, R. T. 1987. Multidimensional Categorical Grammars and Linguistic Analysis. In E. Klein and J. van Benthem (eds.), *Categories, Polymorphism and Unification*, pp 231–260, Centre for Cognitive Science, University of Edinburgh.
- Orgun, C. O. 1996. *Sign-Based Morphology and Phonology*. Ph. D.thesis, University of California, Berkeley.
- Orgun, C. O. 2000. Sign-Based Morphology: A Declarative Theory of Phonology-Morphology Interleaving. In B. Hermans and M. van Oostendorp (eds.), *The Derivational Residue in Phonological Optimality Theory*, page 247 ff., J. Benjamin's.
- Pinker, S. 1999. *Words and Rules: The Ingredients of Language*. London: Weidenfeld and Nicolson.
- Pollard, C. J. 1994. Toward a Unified Account of Passive in German. In J. Nerbonne, K. Netter and C. J. Pollard (eds.), *German in Head-Driven Phrase Structure Grammar*, pp 273–296, Stanford University: CSLI Publications.
- Pollard, C. J. and Sag, I. A. 1987. *Information-based Syntax and Semantics, Vol. 1*. Stanford University: CSLI Publications [Distributed by University of Chicago Press].
- Pollard, C. J. and Sag, I. A. 1992. Anaphors in English and the Scope of Binding theory. *Linguistic Inquiry* 23, 261–303.
- Pollard, C. J. and Sag, I. A. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.
- Potsdam, E. and Runner, J. T. 2001. Richard Returns: Copy Raising and its Implications. In *Proceedings of The Thirty-Seventh Regional Meeting of the Chicago Linguistics Society*, pp 453–468.
- Przepiórkowski, A. 1999. *Case Assignment and the Complement-Adjunct Dichotomy: A Non-Configurational Constraint-Based Approach*. PhD dissertation, Universität Tübingen.
- Pullum, G. K. and Scholz, B. C. 2005. Contrasting Applications of Logic in Natural Language Syntactic Description. In L. V.-V. Petr Hájek and D. Westerstaahl (eds.), *Logic, Methodology and Philosophy of Science: Proceedings of the Twelfth International Congress*, pp 481–503, London: King's College Publications.
- Reape, M. 1994. Domain Union and Word Order Variation in German. In J. Nerbonne, K. Netter and C. J. Pollard (eds.), *German in Head-Driven Phrase Structure Grammar*, pp 151–197, Stanford University: CSLI Publications.
- Richter, F. 2004. *A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar*. Phil. dissertation (2000), Eberhard-Karls-Universität Tübingen.
- Rogers, A. 1974. A Transderivational Constraint on Richard? In *Papers from the Tenth Regional Meeting of the Chicago Linguistic Society*, pp 551–558, Chicago Linguistic Society.
- Runner, J. T. and Aranovich, R. 2003. Noun Incorporation and Rule Interaction in the Lexicon. In S. Müller (ed.), *Proceedings of the HPSG-2003 Conference, Michigan State University, East Lansing*, pp 359–379, Stanford: CSLI Publications.
- Sag, I. A. to appear a. English Filler-Gap Constructions, unpublished manuscript, Stanford University. Available on-line at <http://lingo.stanford.edu/sag/papers/xcons.pdf>.
- Sag, I. A. to appear b. Feature Geometry and Predictions of Locality, unpublished manuscript,

- Stanford University. Paper presented at Conference on Syntactic Features, King's College London – September, 2007.
- Sag, I. A., Kaplan, R. M., Karttunen, L., Kay, M., Pollard, C., Shieber, S. and Zaenen, A. 1986. Unification and Grammatical Theory. In *Proceedings of the Fifth West Coast Conference on Formal Linguistics*, pp 228–254, Stanford University: CSLI Publications/SLA.
- Sag, I. A. and Pollard, C. J. 1991. An Integrated Theory of Complement Control. *Language* 67, 63–113.
- Sag, I. A. and Wasow, T. to appear. Performance-Compatible Competence Grammar. In K. Börjars and R. Borsley (eds.), *New Models of Grammar*.
- Sag, I. A., Wasow, T. and Bender, E. M. 2003. *Syntactic Theory: A Formal Introduction*. Stanford: CSLI Publications, second edition.
- Schwenter, S. 1999. Two Types of Scalar Particles: Evidence from Spanish. In J. Gutierrez-Rexach and F. Martinez-Gil (eds.), *Advances in Hispanic Linguistics*, pp 546–561, Somerville, MA: Cascadilla Pres.
- Schwenter, S. A. and Vasisht, S. 2000. Absolute and Relative Scalar Particles in Spanish and Hindi. In *Proceedings of the 26th Annual Meeting of the Berkeley Linguistics Society*, pp 225–233, Berkeley, CA.
- Stump, G. T. 2001. *Inflectional Morphology*. Cambridge: Cambridge University Press.
- Thráinsson, H. 1979. *On Complementation in Icelandic*. New York: Garland Publishing.
- Van Eynde, F. 1998. Tense, Aspect and Negation. *Studies in Machine Translation and Natural Language Processing* 10, 209–280.
- Van Eynde, F. 2003. On the Notion 'Determiner'. In S. Müller (ed.), *Proceedings of the HPSG-2003 Conference, Michigan State University, East Lansing*, pp 391–396, Stanford: CSLI Publications.
- Van Eynde, F. 2004. Pied Piping is a Local Dependency. In S. Müller (ed.), *Proceedings of the HPSG-2004 Conference, Center for Computational Linguistics, Katholieke Universiteit Leuven*, pp 313–334, Stanford: CSLI Publications, [Available at <http://cslipublications.stanford.edu/HPSG/5/>].
- Van Eynde, F. 2006. NP-Internal Agreement and the Structure of the Noun Phrase. *Journal of Linguistics* 42(1), 139–186.
- Warner, A. R. 1993. The Grammar of English Auxiliaries: An account in HPSG. Research Paper YLLS/RP 1993-4, Department of Language and Linguistic Science, University of York.
- Zec, D. 1987. On Obligatory Control in Clausal Complements. In M. Iida, S. Wechsler and D. Zec (eds.), *Working Papers in Grammatical Theory and Discourse Structure*, pp 139–168, Stanford: CSLI Publications.