Sign-Based Construction Grammar

{ Charles Fillmore
U.C. Berkeley

Paul Kay
U.C. Berkeley

Laura Michaelis
U. Colorado, Boulder

Ivan A. Sag
Stanford University }

June 22, 2007

Chapter 3

Signs and Constructions

(DRAFT of June 21, 2007)

3.1 Introduction

We 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). We understand 'sign' in a sense close to that of Saussure (1916). However, whereas the Saussaurian sign simply relates 'form' and 'meaning' (*signifiant*, *signifié*), we divide up Saussure's dichotomy of sign properties into features of phonology, (morphological) form, syntax, semantics, and context.

We model signs as **feature structures**. Feature structures are of two basic kinds:

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

The set of atoms includes, for analytic convenience, an infinite set of indices. The interesting feature structures are the functions. Each kind of functional feature structure maps elements of a certain domain – some proper subset of the set of features into an appropriate range. Since feature structures map some features onto atoms, while other features are mapped onto other functions, a general characterization of this kind of feature structure would be that it is a function mapping features to feature structures.

As in most constraint-based grammatical frameworks, SBCG makes a strict distinction between **model objects** and **descriptions** of those objects. As in HPSG

(Pollard and Sag 1994; Ginzburg and Sag 2000), 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 GPSG (Gazdar et al. 1985), constructs are in essence local trees that are licensed by some construction of the grammar (e.g. a grammar rule, schema or 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 list of signs, possibly empty. (Values of features can be either feature structures or lists of feature structures). Signs and constructs, we have noted, are feature structures – they are part of the language model. Constructions are descriptions that license classes of linguistic objects – they are part of the grammar (the description of the language model).

The linguistic objects we propose 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 *t* provides on its own.

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

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

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 we call a **Combinatoric Construction**. Each Combinatoric Construction specifies general properties associated with a class of constructs. Put differently, each such 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 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 (the **Constructicon**) selects the signs that constitute the language.

A second kind of construction that will be introduced in this chapter 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 the type *lexeme* or the type *word*):

(2) $lex \Rightarrow [\ \ldots\ ]$

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 chapter will provide further details about the various kinds of signs and constructs on the model side and types and constructions on the description side.

3.2 Feature Structures

We assume that grammatical objects of all kinds (including signs, case values, parts of speech, and constructions) are modeled as **feature structures**. We make the further assumption that feature structures are either atoms like *pl*(*ural*), *acc*(*usative*), $+$, etc.), indices, or else functions from features to feature structures.[1] 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:[2]

(3)
$$\begin{bmatrix} \text{CONTINUANT} & - \\ \text{VOICED} & - \\ \text{ANTERIOR} & + \\ \text{CORONAL} & - \\ \text{SONORANT} & - \\ \text{CONSONANTAL} & + \\ \text{VOCALIC} & - \end{bmatrix}$$

[1] Carpenter 1992. SWB 03.

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

Similarly, the fundamental tenet of 'X-Bar Theory'[3] is that familiar atomic categories like NP or VP are to be reanalyzed as functions, e.g. as in (4):[4]

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

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 $\bar{\bar{X}}$ (Chomsky 1974). 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 1970). The use of functions to model linguistic objects is thus nothing out of the ordinary, though notational idiosyncrasy and the failure of mainstream generative grammarians to make their modeling assumptions explicit 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) and the extensive subsequent work in GPSG, LFG, and HPSG,[5] we use functions to model all grammatical objects. Grammatical categories, for example, are here 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.

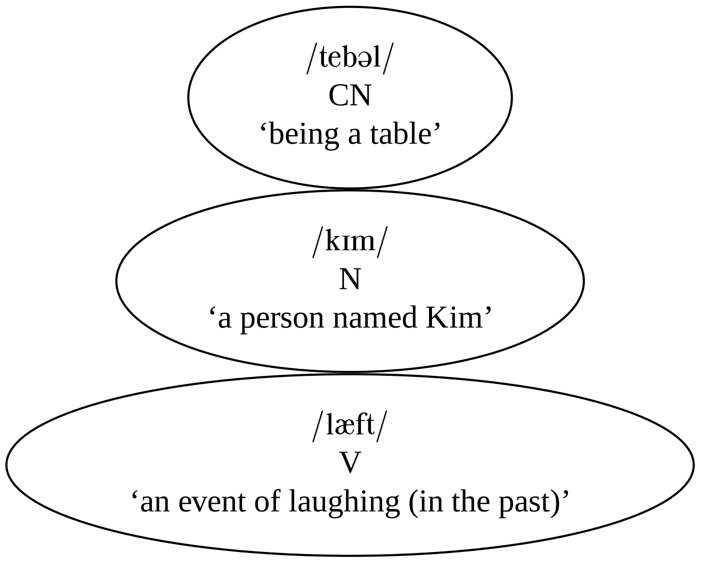
[3]Ref Chomsky 1974, Jackendoff 1977, Pullum 1985: Assuming some version of X-bar theory. In William H. Eilfort, Paul D. Kroeber, and Karen L. Peterson, eds., CLS 21 Part I: Papers from the General Session at the Twenty-First Regional Meeting, 323-353. Chicago Linguistic Society, Chicago IL. The X-bar theory of phrase structure. [Andràs Kornai and Geoffrey K. Pullum.] Language 66, 24-50. 1990.

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

[5]Bresnan et al. 1982. GKPS, P&S 87, 94, Carpenter 92, Dalrymple et al. 1995, Richter 2004.

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):

(5) a. b. c.



Signs, which we take 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:[6]

(6) a.
$$\begin{bmatrix} \text{PHONOLOGY} & /\text{kɪm}/ \\ \text{SYNTAX} & \text{NP} \\ \text{SEMANTICS} & \text{‘a person named Kim’} \end{bmatrix}$$

b.
$$\begin{bmatrix} \text{PHONOLOGY} & /\text{læft}/ \\ \text{SYNTAX} & \text{V}[\textit{fin}] \\ \text{SEMANTICS} & \text{‘a laughing event situated} \\ & \text{prior to the time of utterance’} \end{bmatrix}$$

[6]The semantics here is informal. The reader may think of the descriptions in single quotes as Saussure did – as concepts, i.e. psychological objects.

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

(7) a.

PHONOLOGY	/ɛvri lɪngwɪst/
SYNTAX	NP
SEMANTICS	'the set of properties linguists hold in common'

b.

PHONOLOGY	/pæt læft/
SYNTAX	S[*fin*]
SEMANTICS	'the proposition that there was a laughing event situated prior to the time of utterance where someone named Pat did the laughing'

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 almost every diagram in this monograph employs 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 structure 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 other feature structures that shouldn't satisfy our theory but do. That is, our grammar must neither overgenerate, by providing descriptions of feature structures that do not represent expressions of

English, nor undergenerate, by failing to provide descriptions of feature structures that are models of expressions of English.

Our feature structures have one more property that isn't part of the standard presentation of the basic theory of functions – 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, as well. 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 feature structure type and this type assignment guarantees 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 types are inter-related in 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 other type. In an example discussed in chapter 1, a word *has* might inherit from a type inherited by all third singular verb forms, a type inherited by all present tense verb forms, a type inherited by all verbs with the morphological form *have*, and so on. In SBCG the type hierarchy takes over the inheritance functions that **constructional inheritance** did in some earlier traditions of construction grammar (e.g Fillmore and Kay 1995, Fillmore 1999, Kay and Fillmore 1999, Kay 2002a, Kay 2002b, Koenig 1999). 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:

Add refs to Michaelis, Lambrecht etc?

(8) a. **Has he left?**

b. **Am I tired!**

c. Never **will I harm you**.

d. What **did Maisie 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. There is no inheritance between constructions in SBCG.[7]

The type hierarchy, including the defining specifications of each type, is defined by a grammar's **signature**,[8] 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.

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, as explained in section 3.4 below.

3.3 Signs

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

3.3.1 PHONOLOGY and FORM

We will have little very little to say here about morphology, and nothing at all about phonology, but we fully intend for phonological and morphological entities

[7]The generalizations and expressive economy expressed through inheritance of constructions in earlier constructional approaches are in SBCG expressed by constraint inheritance defined by the hierarchy of construct types.

[8]A 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.

to be part of our linguistic signs. Throughout this monograph, we will display the sequences of (morphological) objects that our analyses associate with the signs we discuss, leaving it to a largely autonomous set of constraints to characterize the relation between the phonological and morphological aspects of signs.

We thus assume two distinct sign-level features: PHONOLOGY (PHON) and FORM:

(9) a. The value of the feature PHON is a phonological phrase (ϕ-*phr*); we assume that these are modeled as feature structures of a particular type.

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 we leave open the precise characterization of formatives, though we will assume that they include **lemmas**, and **affixes**. Our morphological functions will take as input both a formative and a list of lexeme identifiers (see the discussion of the feature LEXICAL-ID below), allowing us to accommodate morphological operations that make such distinctions as the following:

(10) a. *lie/lay/lain* 'rest, recline' vs. *lie/lied/lied* 'tell falsehoods'

b. *can/could* 'be able to' vs. *can/canned* 'to 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, we will simplify our presentation of signs by subsituting conventional orthography for lists of formatives. We return to a discussion of related issues in Chapter 5.

3.3.2 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 3.3.3 below) and as the first member of its ARG-ST list.[9] Other nominal elements on an ARG-ST list are objects.

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:[10]

(11) $\langle$ NP , NP , PP $\rangle$

[9]We 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.

[10]Some abbreviations:

$$\text{NP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} & \textit{noun} \\ \text{VAL} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \qquad \text{PP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} & \textit{prep} \\ \text{VAL} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

$\langle\ \rangle$ = the empty list.

$$\text{CP} = \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} & \textit{comp} \\ \text{VAL} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

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.[11] 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).

3.3.3 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.

CATEGORY

The values of the feature CATEGORY are complex grammatical categories, treated here as feature structures of type *category* (*cat*).[12] The various subtypes of *cat* will specify values for appropriate features. For example, the signature of the grammar of English we assume here includes the following information:

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

b. CASE is used to distinguish the cases of nouns; the possible values of CASE (in English) are *nominative* (*nom*) and *accusative* (*acc*).[13]

[11]This 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).

[12]Note 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*.

[13]Note that genitive nominal expressions are not treated 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 are treated more fully in Chapter 8.

c. VFORM (VF) is used to specify the morphosyntactic category of a verb; the possible values of VF are *finite* (*fin*), *base*, *present-particple* (*prp*), *past-particple* (*psp*), and *passive-particple* (*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 (13), but none like the ones pictured in (14):[14]

(13) a. $\begin{bmatrix}\begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{nom} \\ \ldots & \end{bmatrix}\end{bmatrix}$ b. $\begin{bmatrix}\begin{bmatrix} \textit{verb} & \\ \text{VF} & \textit{fin} \\ \text{AUX} & + \\ \ldots & \end{bmatrix}\end{bmatrix}$ c. $\begin{bmatrix}\begin{bmatrix} \textit{verb} & \\ \text{VF} & \textit{prp} \\ \text{AUX} & - \\ \ldots & \end{bmatrix}\end{bmatrix}$

(14) a. $*\begin{bmatrix}\begin{bmatrix} \textit{noun} & \\ \text{VF} & \textit{fin} \\ \ldots & \end{bmatrix}\end{bmatrix}$ b. $*\begin{bmatrix}\begin{bmatrix} \textit{verb} & \\ \text{AUX} & - \\ \text{CASE} & \textit{nom} \\ \ldots & \end{bmatrix}\end{bmatrix}$ c. $*\begin{bmatrix}\begin{bmatrix} \textit{noun} & \\ \text{AUX} & - \\ \text{VF} & \textit{prp} \\ \ldots & \end{bmatrix}\end{bmatrix}$

We should make clear that we use 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 when we mean to illustrate a particular feature structure, rather than a functional description, we use double outermost brackets, as in (13)–(14).

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 the feature CASE. But any

[14][[Doubled brackets]] are 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

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.

The fact that we model linguistic entities as total functions that can be underspecified by a linguistic description has further utility. If there is more than one model corresponding to a given sentence description, this means that a sequence of formatives or a phonological structure is ambiguous. For example, the descriptions in (15) describe more than one feature structure, and hence predict the appropriate ambiguities:

(15) a. $\begin{bmatrix}\text{FORM} & \langle\ \text{I}\ ,\ \text{forgot}\ ,\ \text{how}\ ,\ \text{good}\ ,\ \text{beer}\ ,\ \text{tastes}\ \rangle\end{bmatrix}$

([[*how good*] [*beer tastes*]] vs. [*how* [[*good beer*] *tastes*]])

b. $\begin{bmatrix}\text{FORM} & \langle\ \text{flying}\ ,\ \text{planes}\ ,\ \text{can}\ ,\ \text{be}\ ,\ \text{dangerous}\ \rangle\end{bmatrix}$

(*flying* is an adjective modifying *planes* or else a gerund whose object is *planes*)

c. $\begin{bmatrix}\text{PHON} & \begin{bmatrix}\phi\text{-phrase} \\ \text{/aylbiliːvɨnyuː/}\end{bmatrix}\end{bmatrix}$

(*I'll believe in you* vs. *I'll be leavin' you*)

There are three other CAT features that need to be introduced:[15]

(16) 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.).

[15]The 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 we have been concerned with streamlining, e.g. by eliminating the features HEAD and LOCAL. Similarly, the fact that we treat SELECT as list-valued is to provide more uniformity in the treatment of constraints than Pollard and Sag were able to achieve.

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 a possibly empty list of signs. 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 list of the NP is empty, otherwise).

c. LEXICAL-ID (LID) is used to individuate lexical items; the value of LID is a possibly empty list of frames specifying the meaning of a lexeme, e.g. $\langle$ *book-frame* $\rangle$, { }, ...

We will discuss SELECT and XARG in more detail in section 3.4 below; LID is discussed in Chapter 5.

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.

In general, a lexeme's VAL list, whose members we refer to as the lexeme's **valents**, is exactly the same as its ARG-ST list, as long as no covert 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 (its subject), has the following singleton VAL list:

(17) $\langle$ NP $\rangle$

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

(18) $\langle\ \rangle$

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'.[16] Clauses, NPs,

[16]This way of looking at things, which has its origin in the argument cancellation of Categorial Grammar (Ajdukiewicz 1935; see also http://en.wikipedia.org/wiki/Categorial_grammar) involves a 'bottom-up' procedural metaphor where one starts with a verb and builds successively larger phrases of which that verb is the lexical head. It is important to recognize, however, that a SBCG, like a Context-Free Phrase Structure Grammar, is a set of static constraints defining well-formed local tree configurations.

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 until Chapter 10. Of direct relevance, however, is the phenomenon of **null instantiation**, which arises when a lexeme undergoes one of the derivational constructions discussed in Chapter 6.[17]

MARKING

The feature MARKING (MRKG), introduced by Pollard and Sag (1994) and refined in crucial ways by Van Eynde,[18] 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, we will assume various other MRKG values, such as those in (19):

(19)	*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 lead a double life as markers, as in the case of *than*, *as*, and *of*.[19]

[17]In many varieties of the Romance languages, clitic pronouns have been reanalyzed as inflectional affixes (see Philip Miller et Paola Monachesi 2003, Les pronoms clitiques dans les langues romanes" . In Godard, Danile. (d), Langues Romanes, problmes de la phrase simple. Paris: Editions du CNRS, pp. 67-123.) 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.

[18]See 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. We leave open the possibility of modifying our theory of MARKING to incorporate further of Van Eynde's insights.

[19]Add refs: Hankamer - CLS 73 on two thans in English. Ref Bonami et al. on French. Anne Abeillé, Olivier Bonami, Danièle Godard et Jesse Tseng. 2006. The syntax of French à and de: an HPSG analysis. Dans P. Saint-Dizier (ed.), Syntax and semantics of prepositions, pp. 147–162. Dordrecht: Springer.

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 3.4 below. The marking value of a head is passed up to its mother via constraints on the Head-Complement Construction. See also Chapter 8, where MRKG values are discussed in more detail.

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 (20):

(20)
$$\begin{bmatrix} \text{CAT} & \textit{verb} \\ \text{VAL} & \langle\ \rangle \end{bmatrix} \text{(S)}$$

$$\begin{bmatrix} \text{CAT} & \textit{verb} \\ \text{VAL} & \langle \text{ NP } \rangle \end{bmatrix} \text{(VP)}$$

$$\begin{bmatrix} \text{CAT} & \textit{noun} \\ \text{VAL} & \langle\ \rangle \\ \text{MKG} & \textit{det} \end{bmatrix} \text{(NP)}$$

But in addition, it is possible to describe very fine-grained categories, e.g. (21), which is the category of any determined NP whose head noun is the lexical item *advantage*:

(21)
$$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{LID} & \langle \textit{ advantage-frame } \rangle \end{bmatrix} \\ \text{VAL} & \langle\ \rangle \\ \text{MKG} & \textit{det} \end{bmatrix}$$

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, Kay and O’Connor 1988, Kay and Fillmore 1999). Mainstream generative grammar has failed to treat, indeed, has given the appearance of being uninterested in this problem (Chomsky 1995).

3.3.4 SEMANTICS

Chapter 4 is devoted to a more detailed presentation of the semantic framework employed in this monograph. 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:

(22) 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, clause).[20]

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.[21] 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 (23):[22]

(23)
$$\begin{bmatrix} \textit{sem-obj} & \\ \text{INDEX} & s \\ \text{FRAMES} & \left\langle \begin{bmatrix} \textit{laugh-fr} & \\ \text{ACTOR} & i \\ \text{SIT} & s \end{bmatrix} \right\rangle \end{bmatrix}$$

[20]Event nominalizations may also require treatment in terms of situational indices.

[21]Following the approach to semantic relations explored first by Pollard and Sag (1994, sec. 8.??) and further developed by Davis (2001). Linking by Types in the Hierarchical Lexicon Anthony R. Davis. OTHER REFS? FRAMENET REFS?

[22]This corresponds to a Davidsonian event variable, which is, roughly, a referential index that points to the entire scene described by an elementary predication.

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 (24):

(24)
$$\left[\begin{array}{ll} \textit{sem-obj} & \\ \text{INDEX} & i \\ \text{FRAMES} & \left\langle \left[\begin{array}{ll} \textit{name-fr} & \\ \text{NAME} & \text{Pat} \\ \text{NAMED} & i \end{array}\right] \right\rangle \end{array}\right]$$

Further abbreviatory conventions for semantic values will be introduced in the next chapter.

3.3.5 CONTEXT

We 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),[23] 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 (25):

(25)
$$\left[\begin{array}{ll} \textit{context} & \\ \text{C-INDS} & \left[\begin{array}{ll} \text{SPEAKER} & \textit{index} \\ \text{ADDRESSEE} & \textit{index} \\ \text{UTT-LOC} & \textit{index} \\ \ldots & \end{array}\right] \\ \text{BCKGRND} & \textit{set}(\textit{proposition}) \end{array}\right]$$

The various contextual indices specify contextual elements that ground an account of indexical and deictic expressions formulated in the style of Kaplan's seminal

[23]See also Georgia Green 1997. The structure of CONTEXT: The representation of pragmatic restrictions in HPSG. Proceedings of the 5th annual meeting of the Formal Linguistics Society of the Midwest; Georgia M. Green ”The nature of pragmatic information.” Grammatical interfaces in HPSG, edited by Ronnie Cann, Claire Grover, and Philip Miller. Stanford, CSLI. (2000). Ginzburg and Sag 2000; Jonathan Ginzburg and Robin Cooper. 2004 ”Clarification, Ellipsis, and the Nature of Contextual Updates” Linguistics and Philosophy 27(3): 297-365. OTHER REFS?

work (1989). The propositions specified in the BACKGROUND 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 mother sign is constructed from a sequence of daughter signs, its BCKGRND must include all elements in the BACKGRND 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 Lambrecht and Michaelis 1996. Engdahl and Vallduvi propose an analysis of ‘information packaging’ formulated in terms of the CONTEXT values structured as shown in (26):

(26)
$$\begin{bmatrix} \textit{context} & \\ \text{INFO-STRUCTURE} & \begin{bmatrix} \text{FOCUS} & \ldots \\ \text{GROUND} & \begin{bmatrix} \text{LINK} & \ldots \\ \text{TAIL} & \ldots \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

We will not explore such elaborations here.[24]

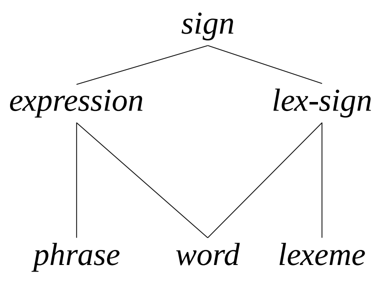
3.3.6 Signs: A Synthesis

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

[24]Nor 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 (FKO, Kay (EVEN), Israel XXX, Schwenter 1999 [Schwenter, Scott A. 1999. Two types of scalar particles: Evidence from Spanish. In J. Gutierrez-Rexach & F. Martnez-Gil (eds.) Advances in Hispanic linguistics, 546-561. Somerville, MA: Cascadilla Pres], Schwenter and Vasishth [BLS ca. 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) [Words and the Grammar of Context. Stanford: CSLI].

(27)
$$\begin{bmatrix} \textit{sign} & \\ \text{PHONOLOGY} & \textit{phonological-phrase} \\ \text{FORM} & \textit{list}(\textit{formative}) \\ \text{SYNTAX} & \textit{syn-obj} \\ \text{SEMANTICS} & \textit{sem-obj} \\ \text{CONTEXT} & \textit{context} \end{bmatrix}$$

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 (28):

(28)


We are now in a position to illustrate in more detail what the various signs introduced earlier in this chapter 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*:[25,26]

[25]A further abbreviation: $\text{NP}_i = \text{NP} \ \& \ \begin{bmatrix}\text{SEM}|\text{INDEX} \ \ i\end{bmatrix}$

[26]The PHON values here are schematic, simply indicating a phonological phrase (ϕ-*phr*) includes a given phoneme sequence. The phonological phrase could be modeled in terms of tree structure. Add refs.

(29) a.
$$\left[\left[\begin{array}{ll}
\textit{word} & \\
\text{PHON} & \left[\begin{array}{l}\phi\textit{-phr}\\ /\text{pæt}/\end{array}\right] \\
\text{FORM} & \langle\ \text{Pat}\ \rangle \\
\text{ARG-ST} & \langle\ \rangle \\
\text{SYN} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\textit{noun} & \\ \text{SELECT} & \langle\ \rangle \\ \text{XARG} & \langle\ \rangle \\ \ldots & \end{array}\right] \\
\text{VAL} & \langle\ \rangle \\
\text{MRKG} & \textit{det}
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{INDEX} & i \\
\text{FRAMES} & \left\langle \left[\begin{array}{ll}\textit{name-fr} & \\ \text{NAME} & \text{Pat} \\ \text{NAMED} & i\end{array}\right] \right\rangle
\end{array}\right] \\
\text{CNTXT} & \ldots
\end{array}\right]\right]$$

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

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

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 (or show a more ramified **derivation tree**). Our model of a phrase or sentence is a single sign. This sign contains all the relevant information. Trees come into the picture only as record of a kind of fictive history according to which the sign is ‘constructed’ – fictive because the system is declarative, all constraints applying simultaneously.

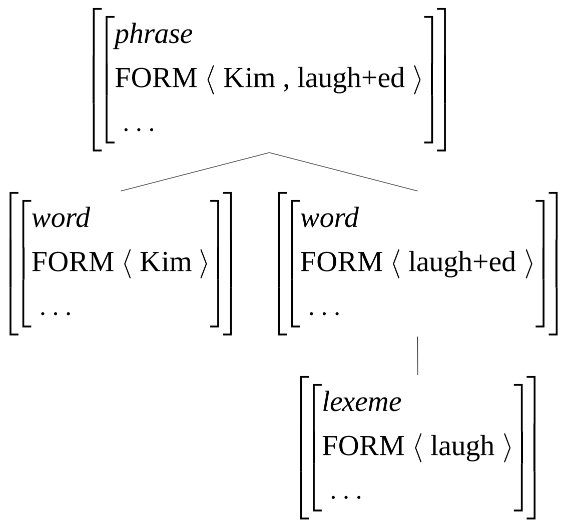
(30)

$$\left[\begin{bmatrix}\begin{array}{ll}
\textit{phrase} & \\
\text{PHON} & \begin{bmatrix}\phi\textit{-phr}\\ \text{/pætlæft/}\end{bmatrix}\\
\text{FORM} & \langle\ \text{Pat, laugh+ed}\ \rangle\\
\text{SYN} & \begin{bmatrix}\begin{array}{ll}
\text{CAT} & \begin{bmatrix}\begin{array}{ll}
\textit{verb} & \\
\text{VF} & \textit{fin}\\
\text{SELECT} & \langle\ \rangle\\
\text{XARG} & \langle\ \text{NP}[\textit{nom}...\ ]_i\ \rangle\\
\ldots &
\end{array}\end{bmatrix}\\
\text{VAL} & \langle\ \rangle\\
\text{MRKG} & \textit{unmk}
\end{array}\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\begin{array}{ll}
\text{INDEX} & s\\
\text{FRAMES} & \left\langle \begin{bmatrix}\begin{array}{ll}\textit{name-fr} & \\ \text{NAME} & \text{Pat}\\ \text{NAMED} & i\end{array}\end{bmatrix}, \begin{bmatrix}\begin{array}{ll}\textit{exist-fr} & \\ \text{BV} & s\end{array}\end{bmatrix}, \right.\\
 & \left. \begin{bmatrix}\begin{array}{ll}\textit{laugh-fr} & \\ \text{ACTOR} & i\\ \text{SIT} & s\end{array}\end{bmatrix}, \begin{bmatrix}\begin{array}{ll}\textit{past-fr} & \\ \text{ARG} & s\end{array}\end{bmatrix} \right\rangle
\end{array}\end{bmatrix}\\
\ldots &
\end{array}\end{bmatrix}\right]$$

We will return to various syntactic and semantic details presupposed here as we present the theory in which these signs are embedded.[27] We repeat at the risk of tedium 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 (which is representable as a derivation tree) involves a **construct** that consists of a mother and its daughters, as illustrated in (31):

[27]Note 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.

(31)



A constraint that defines the grammatically distinctive properties of a class of constructs is called a **combinatoric construction**.

3.4 Constructions

As emphasized above, a grammar signature defines a space of well-formed feature structures (functions) by placing general constraints on the domain and range of each type of feature structure. This space includes an infinite set of signs. But only a subset of the signs consistent with the constraints of a grammar signature are well-formed signs of the language in question (although this subset is also infinite). For example, we want to ensure that (32) is not a sign of English:

(32)
$$\left[\left[\begin{array}{ll}
\textit{phrase} & \\
\text{PHON} & \left[\begin{array}{l}\phi\textit{-phr}\\ /\text{pættoː}/\end{array}\right]\\
\text{FORM} & \langle\ \text{Kim, the}\ \rangle\\
\text{SYN} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\textit{verb} & \\ \text{VF} & \textit{fin}\\ \text{SELECT} & \langle\ \rangle \\ \text{XARG} & \langle\ \text{NP}[\textit{nom}...\ ]_j\ \rangle\\ \ldots & \end{array}\right]\\
\text{VAL} & \langle\ \rangle\\
\text{MRKG} & \textit{that}
\end{array}\right]\\
\text{SEM} & \left[\begin{array}{ll}\text{INDEX} & s\\ \text{FRAMES} & \left\langle \left[\begin{array}{ll}\textit{name-fr} & \\ \text{NAME} & \text{Bo}\\ \text{NAMED} & i\end{array}\right], \left[\begin{array}{ll}\textit{sneeze-fr} & \\ \text{ACTOR} & i\\ \text{SIT} & s\end{array}\right], \left[\begin{array}{ll}\textit{past-fr} & \\ \text{ARG} & s\end{array}\right]\right\rangle\end{array}\right]\\
\ldots &
\end{array}\right]\right]$$

Given what we have said so far, (32) 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 (32) are numerous:

(33) a. This is a finite clause whose FORM value is ⟨ Kim , the ⟩, yet *Kim the* is not a well-formed English clause..

b. The FORM value ⟨ Kim , the ⟩ has been phonologically realized as /pættoː/.

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. We rule out unwanted signs like (32) by introducing an inventory of **constructions** – a **constructicon** – and a general principle requiring well-formed signs (other than lexemes or unanalyzable words) to be licensed by some construction. We will follow tradition and distinguish the constructicon 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, p. 4) and much previous work in Construction Grammar. Rather, a construction in SBCG is a constraint that licenses a particular class of feature structures by specifying certain properties that they must have. In particular, combinatoric constructions specify classes of constructs, which are feature structures containing 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 that we now describe. It should also be noted at this point that signs and constructs are not the only objects in an SBCG model of a language.

Signs in SBCG are licensed by a general grammatical principle which we formulate as follows:

(34) **The Sign Principle:**
Every sign must be lexically or constructionally licensed.

- A sign is lexically licensed only if it satisfies some entry in the lexicon.
- A sign is constructionally licensed only if it is the mother of some construct.

This principle presupposes that in order for a feature structure to be well-formed, it must satisfy all the constraints 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.

3.4.1 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 Type Constructions, and partly by one of a set of Inflectional Constructions that allow words to be built from lexemes.[28] 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 constraint that the CASE value of its subject (its first, and only, ARG-ST member) must be *nom*, is determined by the preterite construction, one of a handful of inflectional constructions that survive in the grammar of Present-Day English.

We fit derivational and inflectional constructions uniformly into a two-level mode, one that is articulated in terms of a mother and its daughter(s).[29] 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. We will refer to a mother-daughter configuration of this sort as a **construct**.

In order to express constructional generalizations in a systematic way, it is useful, indicated above, to model constructs as feature structures of the form sketched in (35):

(35)
$$\begin{bmatrix} \text{MTR} & \textit{sign} \\ \text{DTRS} & \textit{nelist}(\textit{sign}) \end{bmatrix}$$

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 constructions. These licensed constructs in turn give rise to the set of well-formed signs, as per the Sign Principle in (34) above.[30]

[28]Our approach to morphology here is realizational, perhaps closest in spirit to the approach developed by Stump (refs) and others (refs). Lexical affixes are not signs; rather affixation (as well as more complex morphological processes) are as dictated by the morphological functions associated with specific lexical constructions.

[29]The approach to lexical constructions adopted here is based on ideas developed originally (to the best of our knowledge) by Copestake 1992. See also Sag et al. 2003, ch. 16, Sag to appear, Koenig 1999, and OTHER REFS??

[30]ADJUST THIS FOOTNOTE: The terms ‘construct’ and ‘construction’ thus have a meaning here that is distinct from the way these terms have been used in the Berkeley Construction Grammar (BCG) tradition exemplified by Fillmore, Kay and O’Connor 1988, Fillmore and Kay

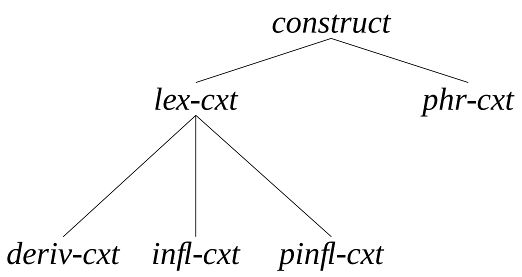
What then is a construction? In SBCG, a construction is a constraint defining the properties that are common to all instances of a given feature structure type. That is, a construction is a constraint of the form shown in (36), where each τ is a type name and D is a description that all instances of τ must satisfy:

(36) $\tau \;\Rightarrow\; D$

Each 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*).[31] This hierarchy of construct types is sketched in (37):

(37)

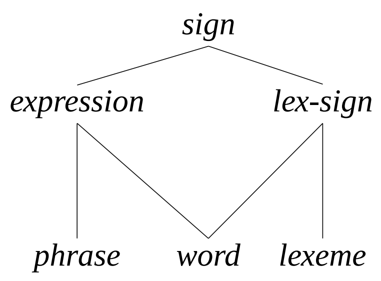


The lexical signs (but not the phrases) repeated below are function as the mothers and daughters of the lexical constructs depicted above in (28):

1996, Kay and Fillmore 1999, Kay 2002a[tags], Michaelis and Lambrecht (000), Michaelis (000), others??? In the BCG tradition, a construct was any fleshed-out 'feature structure tree' (See Kay 2002b[inf. sketch]) – of any degree of configurational complexity, including single node feature structure trees. In the present approach, feature structure trees have been recast as feature structures specifing values for the features MTR (a sign) and DTRS (a list of signs). In addition, the notion 'construct' is restricted to the intuitive equivalent of a fully determinate local tree. 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.

[31] Add refs. Runner and Aronovich.

(38)



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

(39) $\textit{lex-cxt}\text{: } \begin{bmatrix} \text{DTRS} & \textit{list}(\textit{lex-sign}) \end{bmatrix}$

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

Lexical Entries

Let us begin with lexical entries. As in most feature-based theories of grammar, a lexical entry (le) is specified as a constraint relating a form, a syntactic category, and a 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, possibly, words.

A lexical entry like (40) licenses a feature structure like (41) (*pn-wd* stands for *proper-noun-wd*.):[32,33]

[32]We assume here that the predications of proper names are part of their semantic content. Alternate views are possible; see, for example, Pollard and Sag (1987, 1994), who treat these predications as part of the set of background conditions specified within CNTXT values.

[33]It is unnecessary to indicate ". . ." in the lexical entry (40) because this information is already contained in the reference to the *lex-item* construct. The reader should also bear in mind throughout that whenever a type is indicated in a diagram all the constraints impinging on the supertypes of that type are inherited, and so are not explicitly shown.

(40)
$$\left[\begin{array}{ll} \textit{pn-wd} & \\ \text{FORM} & \langle \text{ Kim } \rangle \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & i \\ \text{FRAMES} & \left\langle \left[\begin{array}{ll} \textit{name-fr} & \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{array}\right] \right\rangle \end{array}\right] \end{array}\right]$$

(41)
$$\left[\left[\begin{array}{ll} \textit{pn-wd} & \\ \text{FORM} & \langle \text{ Kim } \rangle \\ \text{ARG-ST} & \langle\ \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \textit{noun} & \\ \text{SELECT} & \langle\ \rangle \\ \text{CASE} & \textit{nom} \\ \text{XARG} & \langle\ \rangle \\ \ldots & \end{array}\right] \\ \text{VAL} & \langle\ \rangle \\ \text{MRKG} & \textit{det} \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & i \\ \text{FRAMES} & \left\langle \left[\begin{array}{ll} \textit{name-fr} & \\ \text{NAME} & \text{Kim} \\ \text{NAMED} & i \end{array}\right] \right\rangle \end{array}\right] \\ \ldots & \end{array}\right]\right]$$

Notice that there must be a value for CASE in (41) – (either *nom* or *acc*), since (41) is a total function (as are all the functions within it), not a constraint on functions. Consequently, a diagram just like (41) except for having *nom* replace by *acc* would have served just as well to illustrate a construct satisfying the lexical entry in (40).

The grammar contains various type constraints that impose further conditions on feature structure well-formedness. Because the feature structure in (41) is of type *pn-wd*, for example, it must obey the type constraint (a lexical class construction) sketched in (42):

(42)
$$
\textit{pn-wd} \Rightarrow \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \textit{noun} & \\ \text{SELECT} & \langle\,\rangle \\ \text{XARG} & \langle\,\rangle \end{bmatrix} \\ \text{VAL} & \langle\,\rangle \\ \text{MRKG} & \textit{det} \end{bmatrix} \end{bmatrix}
$$

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 either out of nothing (in the case of lexical entries) or else out of one or more distinct signs (in the case of all other constructions). The constructions thus form a recursive system much like a Context-Free Grammar. Crucially, 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 make excellent psycholinguistic sense, as argued by Sag and Wasow (in press).

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 (40) above, because there are infinitely many indices that could serve as the value of the feature INDEX in (41). 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 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 (43) (*siv-lexeme* abbreviates *strict-intransitive-verb-lexeme*.):

(43)
$$
\begin{bmatrix} \textit{siv-lexeme} & \\ \text{FORM} & \langle\,\text{laugh}\,\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{FRAMES} & \left\langle \begin{bmatrix} \textit{laugh-fr} & \\ \text{SIT} & s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}
$$

This lexical entry interacts with various constraints to license infinitely many feature structures like (44):

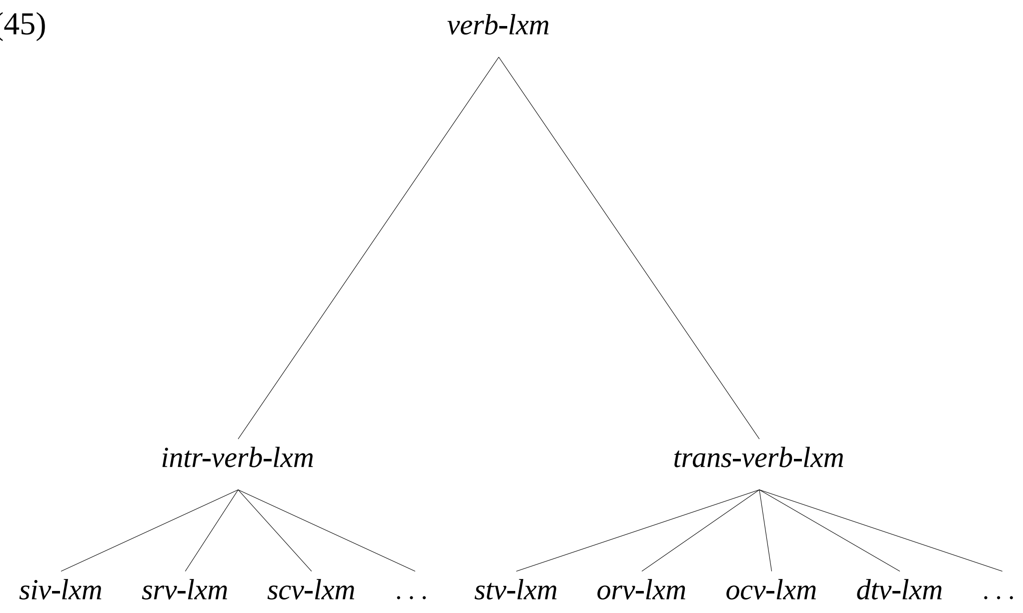
(44)
$$\left[\begin{array}{ll}
\textit{siv-lexeme} & \\
\text{FORM} & \langle \text{ laugh } \rangle \\
\text{ARG-ST} & \langle \text{ NP}_i[\,\ldots\,] \rangle \\
\text{SYN} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \textit{verb} & \\ \text{SELECT} & \langle\ \rangle \\ \text{XARG} & \langle \text{ NP}_i[\,\ldots\,] \rangle \\ \ldots & \end{array}\right] \\ \text{MRKG} & \textit{unmk} \\ \ldots & \end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll} \text{INDEX} & s \\ \text{FRAMES} & \left\langle \left[\begin{array}{ll} \textit{laugh-fr} & \\ \text{ACTOR} & i \\ \text{SIT} & s \end{array}\right] \right\rangle \end{array}\right] \\
\ldots &
\end{array}\right]$$

The key thing to see here is that the NP$_i$[. . .] on the ARG-ST list (and the identical feature structure on the XARG list; see below) must be fully specified in (44), even though neither the lexical entry in (43) nor any of the constraints that affect (44) places any restriction on what this NP feature structure should have as its FORM or SEM value, for instance. 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 (45):[34]

[34]We abbreviate as follows: *strict-intransitive-verb-lexeme* (*siv-lxm*, e.g. *die*), *subject-raising-verb-lexeme* (*srv-lxm*, e.g. *seem*), *subject-control-verb-lexeme* (*scv-lxm*, e.g. *try*), *strict-transitive-verb-lexeme* (*stv-lxm*, e.g. *devour*), *object-raising-verb-lexeme* (*orv-lxm*, e.g. *believe*), *object-control-verb-lexeme* (*ocv-lxm*, e.g. *persuade*), *ditransitive-verb-lexeme* (*dtv-lxm*, e.g. *hand*).

(45)



We assume all verbal lexemes obey the constraint in (46) (a lexical class construction), which says that a verb's first argument is its external argument and that verbs are unmarked.[35]

(46)

$$\textit{verb-lxm} \Rightarrow \begin{bmatrix} \text{ARG-ST} & \langle X, \ldots \rangle \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \textit{verb} \\ \text{XARG} & \langle X \rangle \end{bmatrix} \\ \text{MRKG} & \textit{unmk} \end{bmatrix} \end{bmatrix}$$

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

[35]Variables such as X, X_1, and Y range over feature structures in the constraints and constructions that are formulated here. Σ-variables and L-variables range over sets of feature structures and lists of feature structures, respectively. A colon indicates that the immediately following constraint must be satisfied by all values of the immediately preceding variable, i.e. it introduces a restriction on a variable.

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.[36]

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

For examples, 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:[37]

(47) a.

$$\textit{spray-load1-lxm} \Rightarrow \begin{bmatrix} \text{ARG-ST} & \langle\, \text{NP}_x\,, \text{NP}_y, \text{PP}[\textit{dir}]_z \,\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{FRAMES} & \left\langle \begin{bmatrix} \textit{sl-fr} & \\ \text{SIT} & s \\ \text{ARG1} & x \\ \text{ARG2} & y \\ \text{ARG3} & z \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

b.

$$\textit{spray-load2-lxm} \Rightarrow \begin{bmatrix} \text{ARG-ST} & \langle\, \text{NP}_x\,, \text{NP}_z, \text{PP}[\textit{with}]_y \,\rangle \\ \text{SEM} & \begin{bmatrix} \text{INDEX} & s \\ \text{FRAMES} & \left\langle \begin{bmatrix} \textit{sl-fr} & \\ \text{SIT} & s \\ \text{ARG1} & x \\ \text{ARG2} & y \\ \text{ARG3} & z \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Here *sl-fr* designates a supertype of the frames associated with '*spray-load*' verbs and ARG1, ARG2, ARG3 are the role features that we assume are common to all

[36]This kind of simplification is typical of object-oriented analyses of complex data.

[37]Here we abbreviate as follows:

$\text{PP}[\textit{dir}] = \text{PP}\ \&\ \begin{bmatrix}\text{CAT}|\text{LID}\ \ \textit{dir-fr}\end{bmatrix} \qquad \text{PP}[\textit{with}] = \text{PP}\ \&\ \begin{bmatrix}\text{CAT}|\text{LID}\ \ \textit{with-fr}\end{bmatrix}$

We assume that *dir-fr* (*directional-frame*) is a supertype of those frames that are lexically associated with directional prepositions.

such frames. Thus, simply by adding a lexical entry like (48) to the lexicon (which is compatible with both lexeme types in (47), we allow for both kinds of lexical items to be licensed:

(48)
$$\begin{bmatrix} \text{FORM} & \langle\ \text{spray}\ \rangle \\ \text{SEM} & \begin{bmatrix} \text{FRAMES} & \langle\ [\textit{spray-fr}]\ \rangle \end{bmatrix} \end{bmatrix}$$

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

Morphological Functions

In the next two sections we discuss inflectional and derivational constructions. A key part of such word-building constructions are the **morphological functions**. As part of that discussion we present, as an example of an inflectional construction, 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 $\mathbf{F}_{pret}$.

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. We take the FORM value of a given lexeme (of type *form*) to be a singleton list containing the inflectional stem associated with that lexeme.[38]

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 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

[38]Our discussion here simplifies in various respects, ignoring the possibility of multiple stems, 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. ADD REFS. Bonami, etc.

have) and sometimes the LID value (*lie* 'recline' $\Rightarrow$ *lay*, *lie* 'prevaricate' $\Rightarrow$ *lied*). $\mathbf{F}_{pret}$ might be defined along the following lines:

(49)

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

Special constructions will be needed to specify the preterite forms of BE for various person and number pairings. All lexemes with FORM *have* will be inflected the same for preterite as *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 for illustrative purposes, $\mathbf{F}_{pret}$ 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 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, 'keep'*$\rangle$ 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. [39]

[39]Non-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 semantically bleaching post-inflectional 'pumping' construction, whose mother and unique daughter do not differ in FORM. In either case, special arrangements must be made to distinguish, for example, counterfactual *I were...* from its past tense analogue *I was....* We will not resolve these issues here.

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*(*T*) stands for a nonempty list, each of whose members is a feature structure of (some subtype of) type *T*.):

(50) $$\textit{infl-cxt}: \begin{bmatrix} \text{MTR} & \textit{word} \\ \text{DTRS} & \textit{nelist}(\textit{lexeme}) \end{bmatrix}$$

(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.[40]

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 (51) is constructionally licensed:

[40]There is usually, if not always, a single daughter in an inflectional construct. For convenience, we here ignore languages with layered inflection.

$$
\left[\begin{array}{ll}
\textit{infl-cxt} & \\
\text{MTR} & \left[\begin{array}{ll}
\textit{word} & \\
\text{FORM} & \langle \text{ laugh+ed } \rangle \\
\text{ARG-ST} & \langle \text{ NP}_i[\textit{nom} \ldots] \rangle \\
\text{SYN} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\textit{verb} & \\
\text{VF} & \textit{fin} \\
\text{SELECT} & \langle\ \rangle \\
\text{XARG} & \langle \text{ NP}_i[\textit{nom...}] \rangle \\
\ldots &
\end{array}\right] \\
\text{MRKG} & \textit{unmk} \\
\text{VAL} & \langle \text{ NP}_i[\textit{nom} \ldots] \rangle
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{IND} & s \\
\text{FRAMES} & \left\langle \left[\begin{array}{ll} \textit{exist-fr} & \\ \text{BV} & s \end{array}\right], \left[\begin{array}{ll} \textit{laugh-fr} & \\ \text{ACTOR} & i \\ \text{SIT} & s \end{array}\right], \left[\begin{array}{ll} \textit{past-fr} & \\ \text{ARG} & s \end{array}\right] \right\rangle
\end{array}\right] \\
\ldots &
\end{array}\right] \\
\text{DTRS} & \left\langle \left[\begin{array}{ll}
\textit{lexeme} & \\
\text{FORM} & \langle \text{ laugh } \rangle \\
\text{ARG-ST} & \langle \text{ NP}_i[\textit{nom} \ldots] \rangle \\
\text{SYN} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\textit{verb} & \\
\text{VF} & \textit{fin} \\
\text{SELECT} & \langle\ \rangle \\
\text{XARG} & \langle \text{ NP}_i[\textit{nom...}] \rangle \\
\ldots &
\end{array}\right] \\
\text{MRKG} & \textit{unmk} \\
\text{VAL} & \langle \text{ NP}_i[\textit{nom} \ldots] \rangle
\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}
\text{INDEX} & s \\
\text{FRAMES} & \left\langle \left[\begin{array}{ll} \textit{laugh-fr} & \\ \text{ACTOR} & i \\ \text{SIT} & s \end{array}\right] \right\rangle
\end{array}\right] \\
\ldots &
\end{array}\right] \right\rangle
\end{array}\right]
$$

Figure 3.1: A *laughed* Construct

(51)
$$
\left[\begin{bmatrix} \textit{word} & \\ \text{FORM} & \langle \text{laughed} \rangle \\ \text{ARG-ST} & \left\langle \text{NP}_i \begin{bmatrix} \textit{nom} \\ \ldots \end{bmatrix} \right\rangle \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \textit{verb} & \\ \text{VF} & \textit{fin} \\ \text{SELECT} & \langle\ \rangle \\ \text{XARG} & \langle \text{NP}_i[\textit{nom}\ldots\,] \rangle \end{bmatrix} \\ \text{VAL} & \left\langle \text{NP}_i \begin{bmatrix}\textit{nom}\\ \ldots\end{bmatrix}\right\rangle \\ \text{MRKG} & \textit{unmk}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{IND} & s \\ \text{FRAMES} & \left\langle \begin{bmatrix}\textit{exist-fr} & \\ \text{BV} & s\end{bmatrix}, \begin{bmatrix}\textit{laugh-fr} & \\ \text{ACTOR} & i \\ \text{SIT} & s\end{bmatrix}, \begin{bmatrix}\textit{past-fr} & \\ \text{ARG} & s\end{bmatrix}\right\rangle \end{bmatrix} \\ \ldots & \end{bmatrix}\right]
$$

Several observations are in order here. First, the SYN values of the mother and the daughter in Figure 1 are identical. Second, we 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.[41] 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 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

[41]In the next chapter, we will refine this analysis in terms of restricted (generalized) quantification.

a filler-gap construction, is realized as a pronominal affix (e.g. in French 'clitic' pronouns, which are affixal in nature REFS), 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, we will use trees whenever possible to illustrate feature structures of type *construct*.

The Preterite Construction can now be formulated as follows:[42]

(52) **Preterite Construction (preliminary formulation):**

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

One way of paraphrasing (52) is as follows: Given a verbal lexeme[43] one can construct a verbal word meeting the following four conditions:

(53) a. the word's VF value is *finite*

[42]See note 35 above. Recall that a variable followed a colon and a description indicates a restriction that values of the variable must satisfy.

[43]It 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*.

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

Figure 3.2: A *laughed* Construct in Tree Notation

b. the word's FORM value is related to that of the lexeme via the morphological function $\mathbf{F}_{pret}$,[44]

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.[45]

Notice that it would be redundant for the construction in (52) 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 (50), 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 3.2 above).

Moreover, as we scale up our analysis of English morphology, this construction can be further simplified. Since all finite forms in English require nominative subjects, as shown in (54),

(54) 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:

[44] Recall that these morphological entities are distinct from (and more 'abstract' than) the phonological entities that realize them. See the discussion in section 3.3.1 above.

[45] Our semantic analysis of the preterite, as represented in (52) is simplified in various ways.

(55)
$$\textit{finite-cxt} \Rightarrow \begin{bmatrix} \textit{infl-cxt} & \\ \text{MTR} & \begin{bmatrix} \text{ARG-ST} & \langle\ \text{NP}[\textit{nom}]\,,\ \ldots\ \rangle \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{VF} & \textit{fin} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

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:

(56) **Preterite Construction (final formulation):**

$$\textit{preterite-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \text{FORM} & \langle\ \mathbf{F}_{pret}(X)\ \rangle \\ \text{ARG-ST} & L_1 \\ \text{SYN} & Y \\ \text{SEM} & \begin{bmatrix} \text{FRAMES}\ L_2 \oplus \left\langle \begin{bmatrix} \exists\text{-fr} \\ \text{BV}\ s \end{bmatrix}, \begin{bmatrix} \textit{past-fr} \\ \text{ARG}\ s \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \left\langle \begin{bmatrix} \text{FORM} & \langle\ X\ \rangle \\ \text{ARG-ST} & L_1 \\ \text{SYN} & Y \\ \text{SEM} & \begin{bmatrix} \text{IND}\ s \\ \text{FRAMES}\ L_2 \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

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.

Derivational Constructions

Derivational constructions are structured as shown in (57):

(57)
$$\textit{deriv-cxt}\text{:} \begin{bmatrix} \text{MTR} & \textit{lexeme} \\ \text{DTRS} & \textit{nelist}(\textit{lex-sign}) \end{bmatrix}$$

(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, we assume that there is an *un*-prefixation construction, sketched in (58), which allows *un*-verb lexemes to be derived from a specifiable class of verb lexemes:

(58) ***Un*-Verb Construction:**

$$\textit{unverb-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \text{FORM} & \langle\, \mathbf{F}_{un}(X)\, \rangle \\ \text{ARG-ST} & L_1 \\ \text{SYN} & Y \\ \text{SEM} & [\text{FRAMES } L_2 \oplus \ \ldots] \end{bmatrix} \\ \text{DTRS} & \left\langle \begin{bmatrix} \textit{stv-lxm} & \\ \text{FORM} & \langle\, X\, \rangle \\ \text{ARG-ST} & L_1 \\ \text{SYN} & Y \\ \text{SEM} & \begin{bmatrix} \text{FRAMES } L_2 \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

We have assumed here that only strict-transitive verbal lexemes (lexemes of type *stv-lxm*) can give rise to *un*-verb lexemes. However, there is room for disagreement about whether the relevant constraints that should be placed on (58) 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*.

Derivational constructions, which we will have more to say about in Chapter 5, include, among others:

(59) a. passivization, which feeds overt inflectional constructions in many languages and word-formation processes in English (see Bresnan 2001 PAGE?),

$$\left[\begin{array}{ll} \textit{word} & \\ \text{FORM} & \langle \text{[un+tie]+d} \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \textit{verb} & \\ \text{VF} & \textit{fin} \\ \text{SELECT} & \langle\ \rangle \\ \ldots & \end{array}\right] \\ \text{VAL} & \langle \text{NP}_i[\textit{nom}\text{...]}, \text{NP}[\textit{acc}\text{...]} \rangle \\ \text{MRKG} & \textit{unmk} \end{array}\right] \\ \ldots & \end{array}\right] \quad \textbf{(preterite-cxt)}$$

$$|$$

$$\left[\begin{array}{ll} \textit{lexeme} & \\ \text{FORM} & \langle \text{un+tie} \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \textit{verb} & \\ \text{VF} & \textit{fin} \\ \text{SELECT} & \langle\ \rangle \\ \ldots & \end{array}\right] \\ \text{VAL} & \langle \text{NP}_i[\textit{nom}\text{...]}, \text{NP}[\textit{acc}\text{...]} \rangle \\ \text{MRKG} & \textit{unmk} \end{array}\right] \\ \ldots & \end{array}\right] \quad \textbf{(unverb-cxt)}$$

$$|$$

$$\left[\begin{array}{ll} \textit{lexeme} & \\ \text{FORM} & \langle \text{tie} \rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \textit{verb} & \\ \text{VF} & \textit{fin} \\ \text{SELECT} & \langle\ \rangle \\ \ldots & \end{array}\right] \\ \text{VAL} & \langle \text{NP}_i[\textit{nom}\text{...]}, \text{NP}[\textit{acc}\text{...]} \rangle \\ \text{MRKG} & \textit{unmk} \end{array}\right] \\ \ldots & \end{array}\right]$$

Figure 3.3: *Un*-Verb Construction Feeding Preterite Construction

b. agentive noun formation,

c. denominal verb formation (ref. Clark and Clark),

d. various kinds of nominalization,

e. the *Un*-Adjective Construction, and

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*, we allow members of compounds 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:[46]

(60) a. pumpkin bus: 'a bus that was used in some previous excursion to a pumpkin patch familiar to the relevant interlocutors'

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

Examples (60a) and (60b) illustrate attested innovative compounds. Examples c, also attested, exemplify some of the subtypes of noun-noun compounds exhibiting internal inflection.[47]

It is also possible to incorporate proposals like that of Copestake and Lascarides (1997), which posits a number of more specific constructions specifying patterns that fit particular classes of nouns together in conventionalized ways. We will not examine that possibility here.[48]

[46]Ref. Kay and Zimmer 1976. Downing 1977. Copestake, A. and A. Lascarides [1997] Integrating Symbolic and Statistical Representations: The Lexicon Pragmatics Interface, Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL97), Madrid, July 7th–12th 1997, pp136–143., Levi, J.H. (1978) *The Syntax and Semantics of English Nminals. New York: Academic Press.*

[47]Cf. Bauer and Reynaud (2001) [= Bauer, Laurie and Antoinette Reynaud. A corpus-based study of compounding in English. *Journal of English Linguistics* 29, 101-123].

[48]As we have 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

Postinflectional Constructions

Postinflectional constructs are structured as follows:

(61) $$\textit{pinfl-cxt}: \begin{bmatrix} \text{MTR} & \textit{word} \\ \text{DTRS} & \textit{list}(\textit{word}) \end{bmatrix}$$

(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), Kim and Sag (2002), Kim (2000) and 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 (62):

(62) **Extraposition Construction:**

(REF), Kiparsky (REF) give reasons that that the first member of a nominal compound cannot be a word. Bauer and Reynaud, in a corpus study, discuss the circumstances under which it is likely to be one. Ramscar (REF) discusses many further such examples.

$$\textit{extra-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \text{FORM} & L_0 \\ \text{ARG-ST} & L_1 \\ \text{SYN} & \begin{bmatrix} \text{CAT} & X \\ \text{MRKG} & Y \\ \text{VAL} & \langle \text{NP}[\textit{it}]\rangle \oplus L_2 \oplus \langle W\rangle \end{bmatrix} \\ \text{SEM} & Z \end{bmatrix} \\ \text{DTRS} & \left\langle \begin{bmatrix} \text{FORM} & L_0 \\ \text{ARG-ST} & L_1 \\ \text{SYN} & \begin{bmatrix} \text{CAT} & X \\ \text{MRKG} & Y \\ \text{VAL} & \langle W\text{:CP}\rangle \oplus L_2 \end{bmatrix} \\ \text{SEM} & Z \end{bmatrix} \right\rangle \end{bmatrix}$$

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.[49] Verbs or adjectives whose cooccurring clauses occur only in extraposed position, e.g. the instances of (non-raising) *seem* and *appear* illustrated in (63), are simply listed in the constructicon with the same ARG-ST value as elements that are constructed by (62):

(63) 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 daughter of type *lexeme*. Hence,

[49]For a more comprehensive analysis of English extraposition, where extraposed clauses are treated as non-complement dependents, see Kim and Sag (2005, in press).

treating a given alternation via a postinflectional construction insures that the result cannot feed most derivational constructions.

3.4.2 Phrasal Constructions

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

(64)
$$\textit{phr-cxt}\text{:}\begin{bmatrix}\text{MTR} & \textit{phrase}\\ \text{DTRS} & \textit{list}(\textit{expression})\end{bmatrix}$$

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

The Subject-Predicate Construction

Simple declarative clauses are licensed by the Subject-Predicate Construction, sketched in (65):

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

$$\textit{sp-cxt} \Rightarrow \begin{bmatrix}\text{MTR} & \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{CAT}\begin{bmatrix}\text{MRKG} & X_1\end{bmatrix}\\ \text{VAL}\ \langle\ \rangle\end{bmatrix}\end{bmatrix}\\ \\ \text{DTRS} & \left\langle X_2\ ,\ \begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{CAT}\begin{bmatrix}\text{VF} & \textit{fin}\end{bmatrix}\\ \text{MRKG} \ \ X_1\text{:}\textit{unmk}\\ \text{VAL}\ \langle\ X_2\ \rangle\end{bmatrix}\end{bmatrix}\right\rangle\end{bmatrix}$$

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.[50]

[50]For convenience, we 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

Similarly, the Principle of Compositionality introduced in the next chapter requires that the FRAMES list of the two daughters be merged to form the mother's FRAMES list. With the interaction of these principles, (65) licenses phrasal constructs like the one in Figure 4. Notice that the mother of this construct is just the phrasal sign illustrated in (30) above.

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 fact there is a head word within whose category is 'noun', other feature specifications can encode key grammatical information about an element 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 GAP[51] 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. As we develop a theory of such feature specifications and the principles that govern their distribution throughout constructs, we will be developing a general theory of what nonlocal information can be lexically selected at a higher level of structure or referenced by a construction higher in a phrasal derivation.

As has been recognized at least since Chomsky (1965),[52] lexical restrictions are circumscibed, 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 Icelandic[53]) clearly involves the subjects of verbs, the most likely first approximation of the

(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 1983, Gazdar et al. 1985, Sag 1987 and ???. On Linearization Theory, see Reape 1994, Mueller 1995, 1999, 2002, 2004, Kathol 2000, Donohue and Sag to appear, other refs?.

[51]Gazdar 1981, Sag 1982, Gazdar et al 1985, Pollard and Sag 1987, 1994, Sag et al. 2003. Explain history - SLASH to GAP).

[52]See also Kajita 1968 and Sag to appear.

[53]See Thraınsson 1975, Andrews 1982, 19??

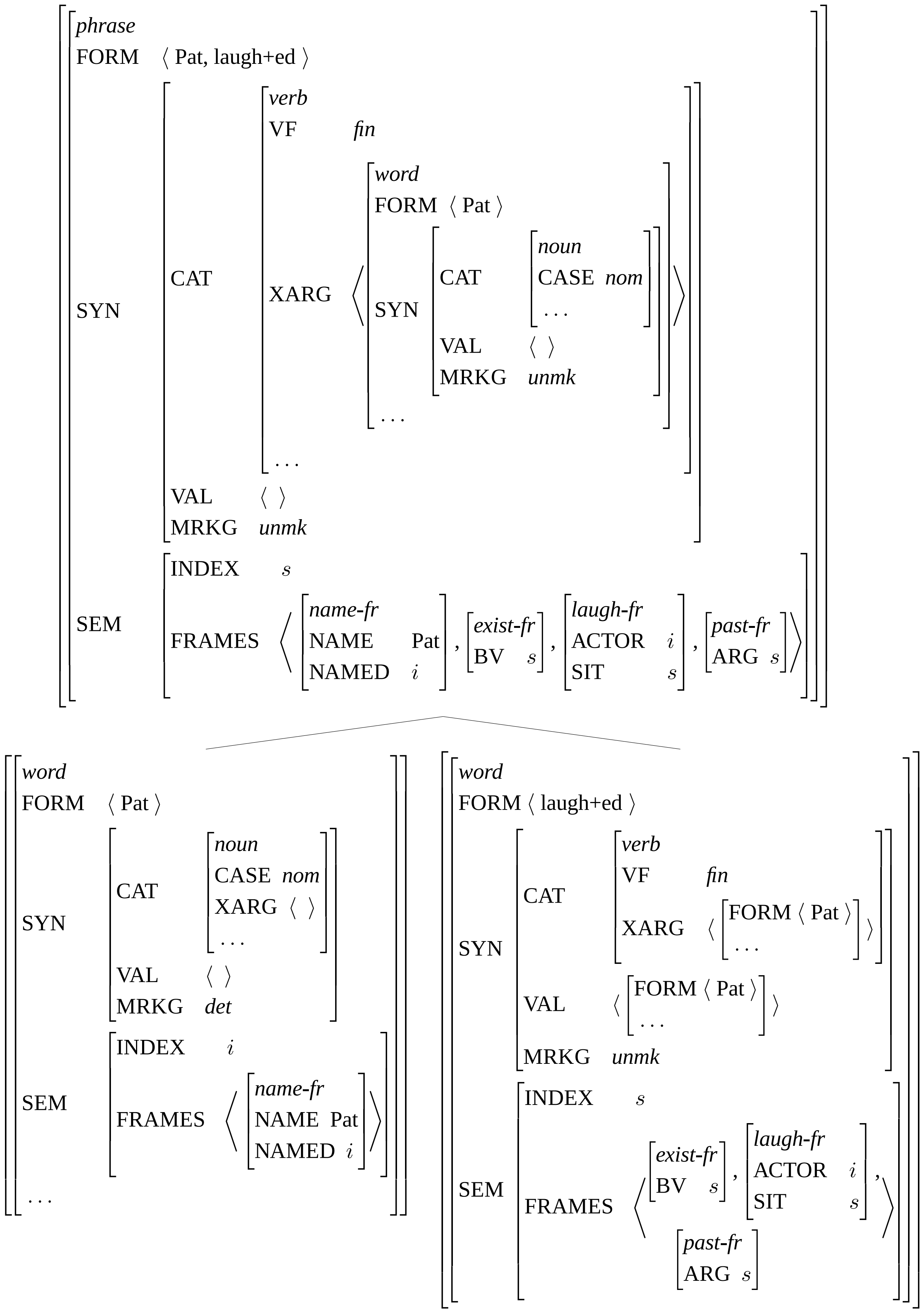


Figure 3.4: A Subject-Predicate Construct

relevant locality domain is: a lexical element's grammatical dependents. We may formulate the relevant hypothesis as in (66):

(66) **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.).

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 (66). 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.

External Arguments

Before proceeding to semantic matters, 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 3.3.3 above. In this section, we review 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 the pronoun in the tag. This induces the familiar tag question agreement pattern illustrated in (67):

(67)

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

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

(68) **Constructional Locality** (Context-Freeness[54]):

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 (69):

(69) a. Sears is having a sale, aren't they?

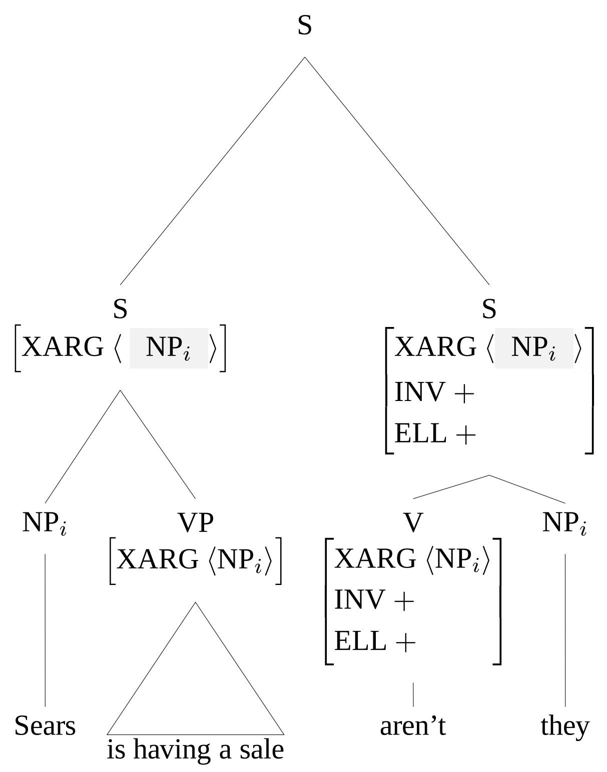
b. At least one of us is sure to win, aren't we?

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

Following Oehrle (1987) and Culicover (1992), Kay (2002) 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 (70), the agreement relation between the two subjects is non-local, i.e. it involves agreement between two elements that are not sisters:

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

(70)



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 (70)). We develop this analysis further in Chapter 9.

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 to appear), illustrated in (71):[55]

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

[55]Ash Asudeh. 2002. Richard III. In Mary Andronis, Erin Debenport, Anne Pycha and Keiko Yoshimura (eds.), CLS 38: The main session. Chicago, IL: Chicago Linguistic Society. Presented April 26, 2002

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 *looks like* construction with the ARG-ST list in (72):[56]

(72) $\left[\text{ARG-ST} \left\langle \text{NP}_i, \text{S[XARG} \langle \text{NP}_i[\textit{pro}] \rangle] \right\rangle \right]$

Also relevant are controlled pronominal subjects in Serbo-Croatian (Zec 1987), Halkomelem Salish (Gerdts and Hukari 2000) and other languages, where control verbs also include the ARG-ST specification in (72). The problems of raising across Polish prepositions (Przepiórkowski 1999, Dickinson 2004),[57] and complementizer agreement in Eastern Dutch dialects (Höhle 1997) are similar, and submit to similar analysis.

Finally, as discussed further in Chapter 5, 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 (73)–(74):

(73) a. He$_i$ lost [his$_i$/*her$_j$ marbles].

b. They$_i$ kept [their$_i$/*our$_j$ cool].

(74) 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 (75):

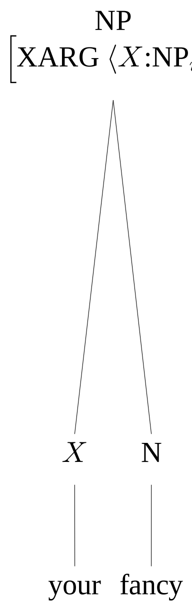
(75) $\left[\text{ARG-ST} \left\langle \text{NP}_i, \text{NP[XARG} \langle \text{NP}_i[\textit{pro}] \rangle] \right\rangle \right]$

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 member be identified with the NP's possessor, as sketched in (76):

[56]'NP$_i$[*pro*]' indicates a pronominal noun phrase.

[57]Polish Numeral Phrases and Predicative Modification Markus Dickinson May 5, 2004. Unpublished, Georgetown University.

(76)



All of the phenomena just enumerated provide motivation for XARG specifications as part of the CAT value of sentential and NP signs. Note that XARG lists, unlike VAL lists, do not shrink as larger phrases are constructed. That is, elements are not ‘cancelled off’ the XARG list. This proposal is an extension of earlier proposals that have been made within HPSG,[58] synthesizing them into SBCG.

As noted above [(46)], we assume that the XARG value of verbal lexemes is a singleton list whose member 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*:[59]

[58]Pollard’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 (1996) introduced a feature for the subject of German verbal clauses and called it SUBJECT; this is the feature used by Meurers (1999) and by Levine (ms.). However, since we also use this in our analysis of NPs to make possessor NPs available for external selection, we have adopted the more neutral term ‘EXTERNAL-ARGUMENT’, which was originally introduced in a similar context by Sag and Pollard (1991).

[59]The XARG value is defined as a list in order to simplify the formulation of the Head-Complement Construction, discussed in the next section.

(77)
$$\left[\begin{array}{l}
\left[\begin{array}{ll}
\textit{stv-lxm} & \\
\text{FORM} & \langle\ \text{love}\ \rangle \\
\text{SYN} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll} \text{XARG} & \langle\ \text{NP}[\ \ldots\ ]\ \rangle \\ \text{SELECT} & \langle\ \rangle \\ \ldots & \end{array}\right] \\
\text{MRKG} & \textit{unmk} \\
\text{VAL} & \langle\ \text{NP}_i[\ \ldots\ ]\ , \text{NP}_j[\ \ldots\ ]\ \rangle
\end{array}\right] \\
\text{ARG-ST} & \langle\ \text{NP}_i[\ \ldots\ ]\ , \text{NP}_j[\ \ldots\ ]\ \rangle \\
\text{SEM} & \left[\begin{array}{ll}
\text{INDEX} & s \\
\text{FRAMES} & \left\langle \left[\begin{array}{ll} \textit{love-fr} & \\ \text{ACTOR} & i \\ \text{UNDGR} & j \\ \text{SIT} & s \end{array}\right] \right\rangle
\end{array}\right]
\end{array}\right] \\
\ldots
\end{array}\right]$$

By contrast, semantically inert (‘case-marking’) prepositions like *to* or *of* have an empty XARG list:

(78)
$$\left[\begin{array}{ll}
\text{FORM} & \langle\ \text{of}\ \rangle \\
\text{SYN} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll} \textit{prep} & \\ \text{XARG} & \langle\ \rangle \\ \text{SELECT} & \langle\ \rangle \end{array}\right] \\
\text{VAL} & \langle\ \text{NP}[\ \ldots\ ]\ \rangle \\
\text{MRKG} & \textit{unmk}
\end{array}\right] \\
\text{ARG-ST} & \langle\ \text{NP}[\ \ldots\ ]\ \rangle \\
\text{SEM} & \left[\begin{array}{ll} \text{FRAMES} & \langle\ \rangle \end{array}\right] \\
\ldots &
\end{array}\right]$$

The Head-Complement Construction

With these lexical contrasts in place, we can now analyze the different properties of VP and PPs without proliferating head-complement constructions. We posit the general head-complement construction sketched in (79):

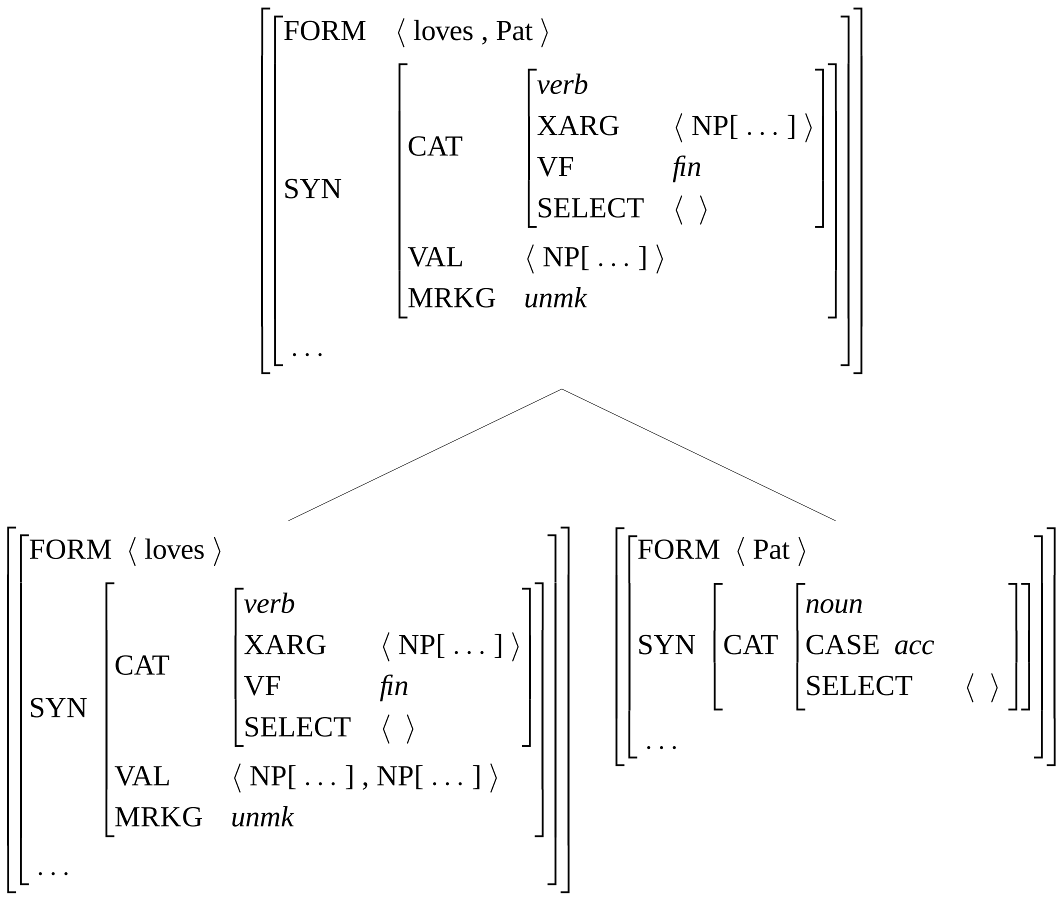
(79) **Head-Complement Construction** (preliminary version):

$$\textit{hd-comp-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \text{SYN} & \begin{bmatrix}\text{VAL} & L_1 \\ \text{MRKG} & X\end{bmatrix}\end{bmatrix} \\ \text{DTRS} & \left\langle \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{CAT} & [\text{XARG } L_1\ ] \\ \text{VAL} & L_1 \oplus L_2 \\ \text{MRKG} & X \end{bmatrix}\end{bmatrix}\right\rangle \oplus L_2{:}\textit{nelist}\end{bmatrix}$$

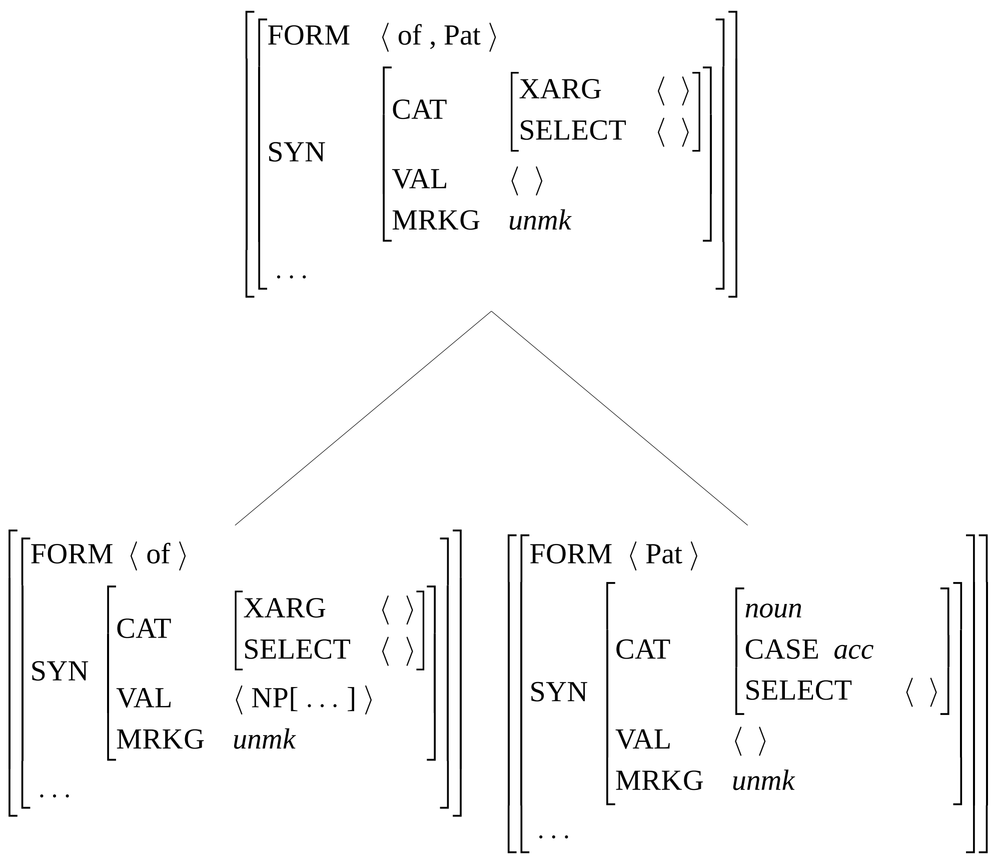
What (79) says is that a head-complement construct must involve a (phrasal) mother whose MARKING value matches that of the first daughter (the head daughter). The first daughter, in addition, must be followed by all of the valents that it selects, except for the XARG, if there is one. The mother's VAL value is the head daughter's XARG list, which will be singleton in the case of a verb, but empty when the head daughter is a case-marking preposition. A grammar that includes this construction licenses constructs like the following:[60]

[60]Here, we omit SEMANTICS as well as PHONOLOGY and CONTEXT.

(80)


FORM ⟨ loves , Pat ⟩
SYN
CAT
verb
XARG ⟨ NP[. . .] ⟩
VF *fin*
SELECT ⟨ ⟩
VAL ⟨ NP[. . .] ⟩
MRKG *unmk*
. . .
FORM ⟨ loves ⟩
SYN
CAT
verb
XARG ⟨ NP[. . .] ⟩
VF *fin*
SELECT ⟨ ⟩
VAL ⟨ NP[. . .] , NP[. . .] ⟩
MRKG *unmk*
. . .
FORM ⟨ Pat ⟩
SYN
CAT
noun
CASE *acc*
SELECT ⟨ ⟩
. . .

(81)



Headed Constructs

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

(82) a. *headed-construct* (*hd-cxt*) is an immediate subtype of *phr-cxt*.

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 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 (83):

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

$$
\textit{hd-comp-cxt} \Rightarrow \begin{bmatrix}
\text{MTR} & \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & L_1 \\ \text{MRKG} & X \end{bmatrix} \end{bmatrix} \\
\text{DTRS} & \langle\, Y \,\rangle \oplus L_2{:}\textit{nelist} \\
\text{HD-DTR} & Y{:}\begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{CAT} & [\text{XARG}\ \ L_1\,] \\ \text{VAL} & L_1 \oplus L_2 \\ \text{MRKG} & X \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

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

$$
\textit{sp-cxt} \Rightarrow \begin{bmatrix}
\text{MTR} & \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{MRKG} & X_1 \end{bmatrix} \\ \text{VAL}\ \langle\ \rangle \end{bmatrix} \end{bmatrix} \\
\text{DTRS} & \langle\, X_2\ ,\ X_3 \,\rangle \\
\text{HD-DTR} & X_3{:}\begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{VF} & \textit{fin} \\ \text{MRKG} & X_1{:}\textit{unmk} \end{bmatrix} \\ \text{VAL}\ \langle\, X_2 \,\rangle & \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Recall that constraints on the type *phr-cxt* given in (64) 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 (83) are so constrained without the need for further stipulation.

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

(84) **Head Feature Principle:**

$$\textit{hd-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix}\text{SYN [CAT } \boxed{1}\text{]}\end{bmatrix} \\ \text{HD-DTR} & \begin{bmatrix}\text{SYN [CAT } \boxed{1}\text{]}\end{bmatrix} \end{bmatrix}$$

(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.

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 are missed by analyses based on so-called 'functional categories'.[61] In their place, he offers a unified analysis of determiners, markers 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 3.3.3 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 ⟨ ⟩. Attributive adjectives, by contrast, select nominal heads; complementizers (whose category assignment Van Eynde does not discuss) select verbal heads, as illustrated in (85):

(85)
$$\begin{bmatrix} \text{FORM} & \langle \text{ happy } \rangle \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \textit{adj} & \\ \text{SELECT} & \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} & \textit{noun} \\ \text{MRKG} & \textit{unmk} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{MRKG} & \textit{unmk} \end{bmatrix} \end{bmatrix}$$

[61] Other critiques. Newmeyer. Hudson. Who else?

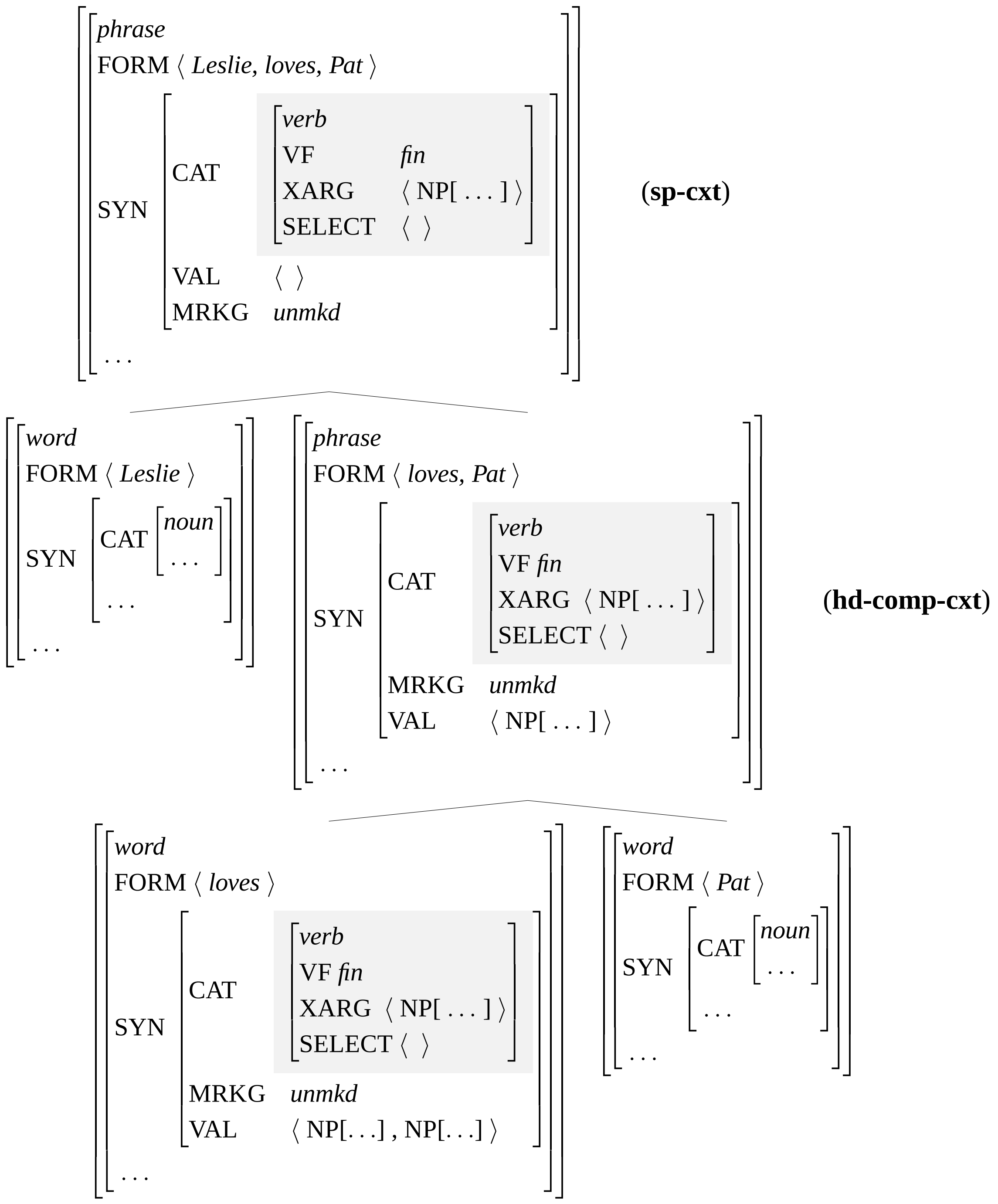


Figure 3.5: Derivation Tree for *Leslie loves Pat*

$$\begin{bmatrix} \text{FORM} & \langle \text{ that } \rangle \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{SELECT} & \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} & \textit{verb} \\ \text{VAL} & \langle\ \rangle \\ \text{MRKG} & \textit{unmk} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{MRKG} & \textit{that} \end{bmatrix} \end{bmatrix}$$

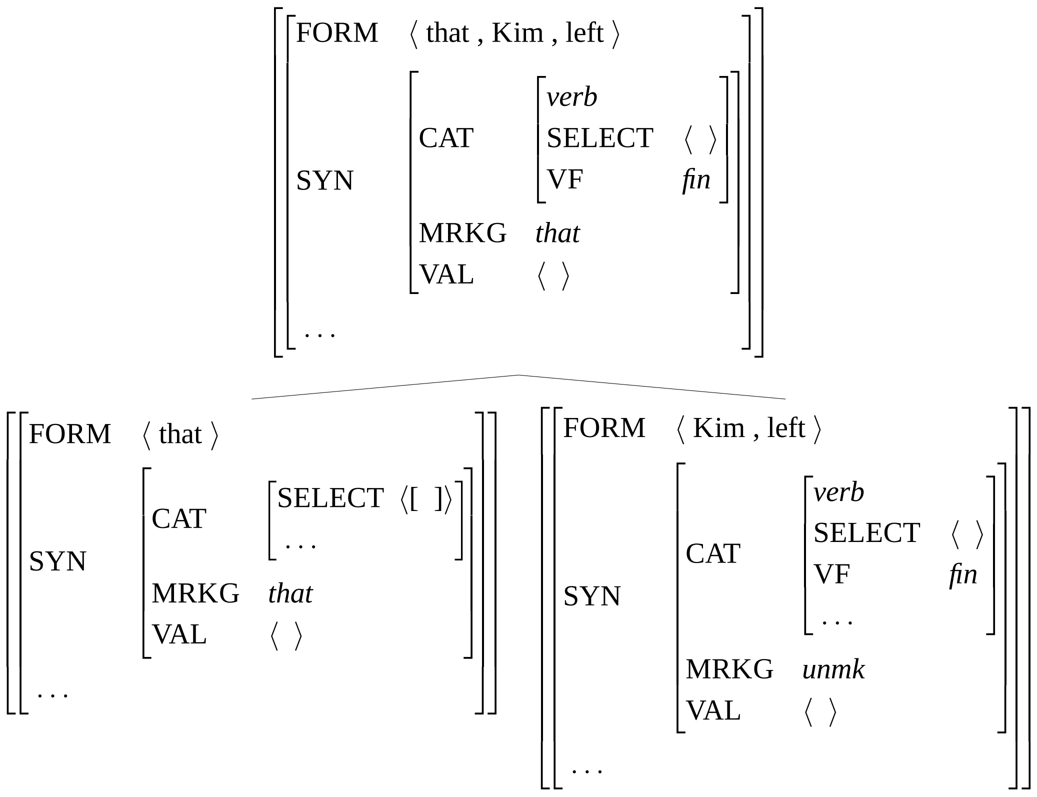
Given these lexical specifications, we can formulate the Head-Functor Construction as follows:

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

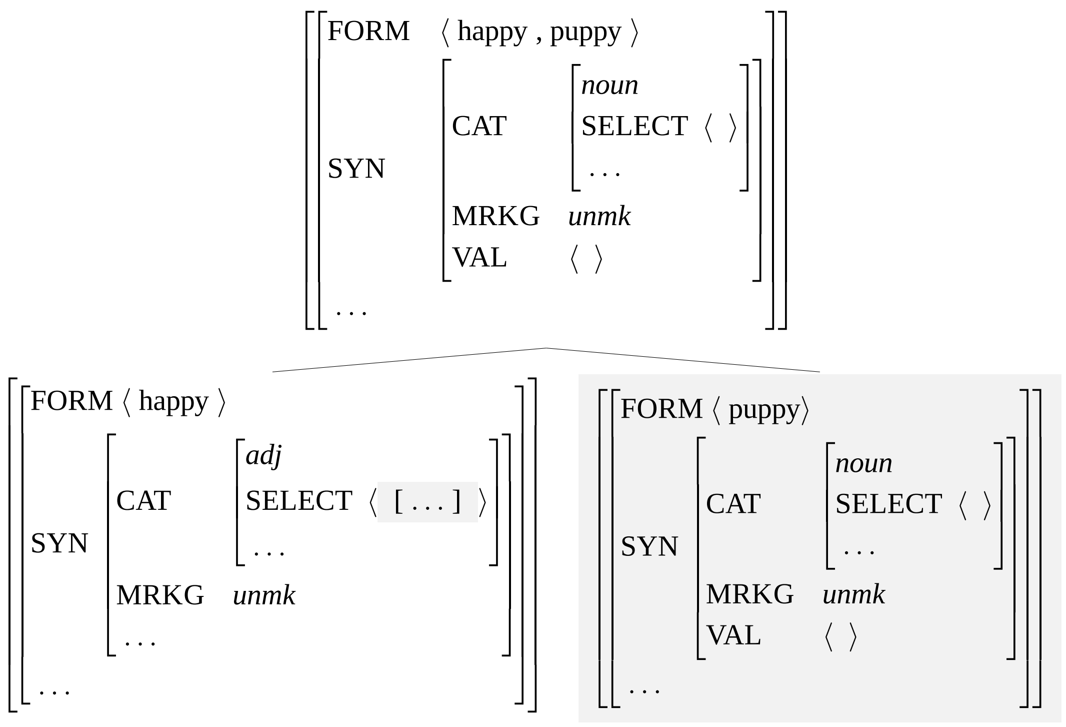
$$\textit{hd-func-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{VAL} & L_1 \\ \text{MRKG} & X_1 \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \left\langle \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} \ [\text{SELECT} \ X_2 \ ] \\ \text{MRKG} \ X_1 \end{bmatrix} \end{bmatrix}, X_2 \right\rangle \\ \text{HD-DTR} & X_2\text{:[SYN [VAL } L_1 \text{]]} \end{bmatrix}$$

This construction allow us to construct both marked clauses and modified nominal phrases, as shown in (87)–(88). 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.

(87)


FORM ⟨ that , Kim , left ⟩
SYN
CAT
verb
SELECT ⟨ ⟩
VF fin
MRKG that
VAL ⟨ ⟩
. . .
FORM ⟨ that ⟩
SYN
CAT
SELECT ⟨[ ]⟩
. . .
MRKG that
VAL ⟨ ⟩
. . .
FORM ⟨ Kim , left ⟩
SYN
CAT
verb
SELECT ⟨ ⟩
VF fin
. . .
MRKG unmk
VAL ⟨ ⟩
. . .

(88)



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 ((84) above).

3.5 Conclusion

In its time...