Sign-Based Construction Grammar

Editors:
Hans C. Boas & Ivan A. Sag

July 10, 2011
Contents

Introduction vii
   IVAN A. SAG AND HANS C. BOAS

1 Making the Case for Construction Grammar 1
   LAURA A. MICHAELIS

2 Sign-Based Construction Grammar:
   An informal synopsis 39
   IVAN A. SAG

3 The Distribution of that-Clauses in English: An SBCG
   Account 171
   GERT WEBELHUTH

4 Discontinuous Dependencies and Complex Determiners 197
   PAUL KAY & IVAN A. SAG

5 Reconstructing Syntax: Construction Grammar and the
   Comparative Method 227
   JÓHANNA BARDDAL & THÓRHALLUR EYTHÓRSSON

6 The FrameNet Constructicon 283
   CHARLES J. FILLMORE, RUSSELL R. LEE-GOLDMAN, AND
   RUSSELL RHODES
2

Sign-Based Construction Grammar: An informal synopsis

IVAN A. SAG

2.1 Introduction

This chapter\(^1\) is intended as an introduction to some of the central notions of Sign-Based Construction Grammar (SBCG).\(^2\) For a more general discussion of SBCG, including an entirely informal, high-level summary of the framework, its historical development, its commitment to psycholinguistic motivation, and its relation to other approaches to grammar (including other constructional approaches), the reader is referred to Sag and Boas this volume.

\(^1\) Above all, I’d like to thank Paul Kay for hours of discussion and useful feedback at every stage in the seemingly endless process of writing this chapter. In addition, I’m grateful to Chuck Fillmore and Laura Michaelis for many useful discussions of both content and exposition and for comments on an earlier draft, which was also improved by the comments of Hans Boas, Adele Goldberg and Adam Przepiórkowski. In addition, discussions with the following people have been quite useful: Farrell Ackerman, Emily Bender, Rui Chaves, Dan Flickinger, Mark Gawron, Andreas Kathol, Beth Levin, Bob Levine, Stefan Müller, Frank Van Eynde, Tom Wasow, and Gert Webelhuth. None of the views expressed here should be attributed to any of these individuals; nor should they be held responsible for any errors that might be found herein.

\(^2\) SBCG owes a considerable debt to the implementational work carried out within the LinGO and Delphin consortia, whose grammar development efforts have proceeded in parallel with the evolution of SBCG. See Copestake 2001, Flickinger 2000, and the online resources available at http://lingo.stanford.edu/ and http://www.delph-in.net/.
2.1.1 Preliminaries

SBGG is a framework blending ideas developed over a quarter century of research in Head-Driven Phrase Structure Grammar (HPSG) with those presented within the tradition of Berkeley Construction Grammar (BCG) over roughly the same period. The goal is to expand the empirical coverage of HPSG, while at the same time putting BCG on a firmer theoretical footing.

To readers steeped in HPSG theory, SBCG will no doubt seem like a minor variant of constructional HPSG (as developed in Sag 1997, Ginzburg and Sag 2000, and elsewhere), with the principal innovation being the introduction of the distinction between signs and constructs. There is a certain truth to this, but at the same time, it is my sincere hope that construction grammarians of all stripes will find that SBCG is recognizable as a formalized version of BCG, with a few straightforward notational (and only minimal notional) adjustments. What is gained from the increased analytic precision of SBCG is plain: clearer empirical prediction and falsifiability, enhanced comparability of analyses across languages, and a general theoretical clarity. Certain differences between SBCG and BCG or other versions of Construction Grammar (CG) will be noted where relevant.

Like Pollard and Sag (1987, 1994), Fillmore et al. (1988), Fillmore and Kay (1996), and Kay and Fillmore (1999), this study will take English as its focus. However, the goal of our enterprise, like that of other researchers in HPSG and BCG, is to provide a basis for the description of all human languages. Construction-based grammar has an advantage in this endeavor, as it concerns itself with words, generalizations about lexical classes, and the patterns according to which phrases are constructed. Every human language arguably has these components; hence there is no need to transform a language into a mold that ill suits it in order to provide a typologically realistic theory of grammar – one that also meets the objections to the Chomskyan conception of Universal Grammar raised, for example, by Evans and Levinson (2009). It is clear that construction-based grammar has deep roots in structuralist linguistics. My goal here is to convince the reader that a properly formalized theory of construction-based grammar can satisfy the demands of modern linguistic theory, as well, e.g. those outlined already by Lees (1957: 376):

1. freedom from contradiction,
2. maximal cohesion with other branches of knowledge,
3. maximal validity in coverage of known data, and
4. maximal elegance of statement.

---

2.1.2 Preview

Let us take a language to be an infinite set of signs and assume that the job of a grammarian is to provide a systematic account of those signs and their properties, including how they function in language processing and language use. The notion of ‘sign’ of course comes from Saussure (1916). However, while the Saussaurian sign is an association of sound (signifiant) and meaning (signifié), the signs of SBCG embody more components. These include at least phonological structure, (morphological) form, syntactic category, semantics, and contextual factors, including information structure.

Signs, like all linguistic entities in SBCG, are modeled as feature structures (FSs), which are of two basic kinds:

- atoms\(^5\) (e.g. accusative, +, bumble-bee, . . .),
- functions (as explained below).

A functional FS maps each feature in its domain (some proper subset of the set of features) to an appropriate value (atom or function).\(^6\) In general then, functional FSs map features to feature structures. The particular features and values that are presented here are given in the appendix.

As in LFG, BCG, and HPSG, SBCG makes a strict distinction between entities in the language model (model objects for short) and descriptions of those objects.\(^7\) As in HPSG, the most important model objects are signs, (the formal representations of actual words and phrases, including sentences). Each Lexical sign or fixed phrasal expression is licensed by a listeme (a ‘listed’ description of a word or phrase).\(^8\) Another, distinct kind of model ob-

\(^5\)The set of atoms includes, for analytic convenience, an infinite set of indices.

\(^6\)A function can be defined simply as a set of ordered pairs, where the first member of the pair is a member of the function’s domain (a set) and the second is a member of its range (also a set). The only further condition that has to be met is uniqueness, i.e. the condition that for any a, b and c, if b and c are distinct, then a function cannot contain both ⟨a, b⟩ and ⟨a, c⟩. A total function is a set of such pairs that contains a pair ⟨a, x⟩ for each member a of the domain set. Finally note that all finite FSs must ‘bottom out’ in atoms. That is, the most deeply embedded functions within a FS must map to an atom or else those functions will by definition not be the most deeply embedded.

\(^7\)Though the distinction between model and model description may seem unfamiliar to many linguists, drawing such a distinction reflects the standard practice of research in most scientific disciplines. Models allow the theorist to abstract away from irrelevant properties of the phenomena under investigation, while the descriptions themselves constitute the theory of the modeling objects. By placing the modeling objects in correspondence with the real-world phenomena, the descriptions become a theory of those phenomena – more precisely, a theory of the properties of those objects that are being modeled. In the present case, for example, we want to model the observable phonetic and semantic properties of signs. Other entities that we introduce into our models, e.g. syntactic categories, case values, contextual conditions, are posited because we believe they are necessary to explain other properties of signs deemed of interest.

\(^8\)The term ‘listeme’ is first proposed by Di Sciullo and Williams 1987 as a generalization of the notion ‘lexical entry’ to include multiword expressions of various kinds. Multiword expres-
ject in SBCG is the construct. As in Generalized Phrase Structure Grammar (GPSG – see Gazdar et al. 1985), constructs are local trees that are licensed by a particular kind of construction: a combinatoric construction. As we will see, a construct can be naturally accommodated within a FS-system as a functional FS that specifies values for the MOTHER (MTR) feature and the DAUGHTERS (DTRS) feature. The value of MTR is a sign and the value of DTRS is a nonempty list of signs.

Signs and constructs, as already noted, are FSs – they are part of the language model. Listemes and constructions are descriptions that license classes of linguistic objects (signs or constructs) – they are part of the grammar (the description of the language model).

The linguistic objects in SBCG here are classified in terms of a system of types, which are organized into a lattice-like structure that reflects a linguistically motivated classification. Thus polar-interrogative-clause is a maximal type (a type without subtypes) that is instantiated by clausal constructs like (1).

\[(1) \{[Will] [Sandy] [be there]?\}\]

But in order to be well-formed according to our theory, this construct must also satisfy the constraints the grammar imposes on all the supertypes of polar-interrogative-clause. These constraints take the form of further constructions – those which define the particular properties of the supertypes auxiliary-initial-construct, headed-construct, interrogative-clause, core-clause, and clause.

An SBCG also contains a grammar signature. It is here that the details of the type hierarchy are laid out, along with a characterization of the general properties of each type of feature structure. The latter specification associates each type of functional feature structure with a domain (a set of features) and assigns a unique type in the function’s range to each feature of the domain, as illustrated in (2):

\[(2) \text{type}_0 : \begin{bmatrix} \text{FEATURE}_1 \text{type}_1 \\ \vdots \\ \text{FEATURE}_n \text{type}_n \end{bmatrix}\]

This is to be interpreted as: The grammar recognizes FSs of type_0, which are functions whose domain includes FEATURE_1 \ldots FEATURE_n and which map

---

9The informal notation in (1), used for abbreviating constructs, is due to Chuck Fillmore. The entire construct is enclosed in curly braces, with each daughter in square brackets. Annotations are also possible, as indicated by the F’s in (i):

\[i) \{F[D_1], \ldots, [F D_n]\}\]

See Fillmore et al. this volume.
each FEATURE; to a FS of type \textit{type;} as indicated.

Against this background of possible FSs, a particular SBCG of a given language specifies which particular family of FSs exists in that language – those that are licensed by a listeme or a construction.\textsuperscript{10} The grammar signature assumed in this chapter is summarized in the appendix.

### 2.2 Feature Structures

As already noted, grammatical objects of all kinds (including signs, case values, parts of speech, and constructions) are modeled as FSs, either atoms or else functions from features to FSs.\textsuperscript{11} This is a simple, but powerful way of modeling linguistic objects, one that is already familiar from early work in generative phonology, where speech segments are often analyzed in this way. For example the following function characterizes the phone \textit{[t]} in the feature system of Chomsky and Halle (1968):

\[
\begin{array}{c}
\text{CONTINUANT} - \\
\text{VOICED} - \\
\text{ANTERIOR} + \\
\text{CORONAL} + \\
\text{SONORANT} - \\
\text{CONSONANTAL} + \\
\text{VOCALIC} -
\end{array}
\]

Similarly, a fundamental tenet of ‘X Theory’\textsuperscript{12} is that familiar atomic categories like NP or VP are to be reanalyzed as functions, e.g. as in (4):\textsuperscript{13}

\[
\begin{array}{c}
\text{NOUN} + \\
\text{VERB} - \\
\text{BAR} 2
\end{array}
\]

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. \textit{[+ CORONAL]} or \textit{+[V]} or to use other notations, e.g. \textit{X}\textsuperscript{1} (Harris 1946) or \textit{X} (Chomsky 1970). Yet it is clear that the analytic intent of such notions is accurately rendered by 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

\textsuperscript{10} Though the distinction between the signature’s type declarations and the constructions of the construction is sometimes more one of convenience, there are computer implementations of type systems where the type declarations (but not the type constraints) are used in a ‘top-down’ manner during processing.

\textsuperscript{11} Carpenter 1992. See also Sag et al. 2003.


\textsuperscript{13} These are the distinctive features of the category NP in the analysis proposed in Gazdar et al. 1985.
Chomsky and Halle 1968 or that of Chomsky 1974). The use of functions to model linguistic objects is thus nothing out of the ordinary, though lack of formalization and idiosyncratic or abstruse notation (especially in the case of generative-transformational syntactic theories) often obscures this fact.

Building on the more explicit ideas pioneered by computational linguistic work of the late 1970s, e.g. Martin Kay’s Functional Unification Grammar (M. Kay 1979), and the extensive subsequent work in GPSG, LFG, and HPSG, every grammatical object used here is modeled by a function that maps each member of a set of features to one of its possible values, as specified in (2) above. Grammatical categories, for example, are analyzed as complexes of various properties represented as feature-value pairs: nouns include specifications for the features \textsc{Case}, \textsc{Number}, and \textsc{Gender}; verbs are specified in terms of the feature \textsc{Verb-form} (VF) for their inflection class (as \{VF finite\}, \{VF present-participle\}, etc.) and will have a ‘+’ or ‘−’ value for the feature \textsc{Auxiliary} (Aux). This approach takes advantage of the power of functions to model complex linguistic entities, unlike the phonological and \textit{X} illustrations given above, where the values of the features are all atomic. The value of a feature may be an atom or it may be a function (another complex \textit{FS}). This allows for recursive embedding of feature structures within \textit{FSs}, analogous to the embedding of functions that is now standard practice in formalized approaches to semantic analysis.

Signs are no exception. Saussure regarded a sign as an ‘associative bond’ between a sound concept and a semantic concept. Adding in syntactic information, we arrive at representations like (5), presented in the Saussurean style (CN stands for common noun; N for proper noun; V for verb):

As already noted, signs are modeled as functions that specify a phonological and morphological structure, a meaning, contextual connections, and relevant syntactic information (including traditional syntactic category and combinatoric potential). These functions are described in terms of attribute-value matrices, i.e. diagrams like the following: 15

The informal semantics in (6) and (7) is a temporary expedient, and will be replaced by a more precise representation in section 2.3.4 below.

15The informal semantics in (6) and (7) is a temporary expedient, and will be replaced by a more precise representation in section 2.3.4 below.
The non-atomic FSs used to model linguistic objects are total functions. That is, once an appropriate feature domain is established for a particular type of feature structure, every FS of that type assigns an appropriate, fully determinate value to every feature in that domain. The value assigned to any feature must also be a feature structure, i.e. either an atom or a function that in turn assigns a value to every feature in its appropriate domain. A FS is thus always ‘complete’ in a simple, intuitive sense: every feature in a function’s domain is assigned a value in the appropriate range.

It is important to notice that although FSs themselves are complete (in that these are total functions, which map every feature in the appropriate domain to a value), FS descriptions are in practice always partial. For this reason, partial descriptions run rampant throughout the chapters in this book. Listemes are formulated as partial FS descriptions (typically being true of (or ‘satisfied by’) a large class of FSs); combinatoric constructions are also quite sparse, when compared with the feature structures that instantiate them. Underlying all our concerns will be the set of FSs that is specified by the grammar we develop. If some aspect of our grammar goes awry, we should be able to learn why by isolating certain (complete) FSs that should satisfy the constraints of our theory, but do not. Alternatively, we should be able to find some FS that incorrectly satisfies our theory. In particular, an SBCG must not license signs that fail to model something in the target language (an unacceptable sentence, for example). In addition, it must not fail to license a sign that is needed to serve as a model of a sentence that the grammarian decides is part of the target language.

FSs have one more property that isn’t part of the basic theory of functions (which I will assume only cursory familiarity with): FSs are organized in terms of linguistic types. A type is a classification associated with a set of FSs that have certain stated properties in common. One benefit derived from assigning FSs to types is that we can then better organize the properties that classes of grammatical objects have and simplify their description in the process. Intuitively, it makes no sense (in English, anyway) to ask what case a verb has or whether a noun is an auxiliary – certain grammatical featu-
ral distinctions are appropriate only for certain kinds of grammatical objects. This intuition is given formal expression in terms of the types that particular FSS instantiate. Each FSS instantiates a particular maximal type and the feature appropriateness conditions together determine which subset of features is appropriate for FSS of that type, ruling out verbal categories that specify CASE values, nominal categories that specify VF values, and so forth.

The space of types is hierarchically structured. In fact, the types are interrelated in terms of a multiple inheritance hierarchy. If a type B is a subtype of another type A, then FSS of type B must satisfy all constraints that the grammar imposes on objects of type A, as well as the grammatical constraints imposed on type B. In this case, the FSS of type B form a subset of the FSS of type A. This situation is informally characterized, following the terminology of BCG, by saying that ‘type B inherits from type A’. In a multiple inheritance hierarchy, a type can inherit from more than one immediate supertype. That is, type hierarchies behave like the nonlinguistic example in Figure 7, where an instance of each maximal type is given below it. Multiple-inheritance hierarchies are useful for analyzing cross-classifying properties of a set of objects, whether they are literary works, words, or constructs.

In SBCG, the more general notion of ‘type hierarchy’ takes over the inheritance functions that constructional inheritance performed in some earlier traditions of CxG. For example, Fillmore (1999) treats the various kinds of auxiliary-initial clause in terms of constructional inheritance from the supertype he calls ‘subject-auxiliary inversion’ (SAI). Some of these SAI constructions are illustrated in (8):

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

In the SBCG analog of Fillmore’s analysis, each of the bracketed sequences in (8) instantiates a type of construct that is a subtype of the more general type

---

47
(e.g. The Odyssey)  (e.g. Beowulf)  (e.g. Ode to a Nightingale)

**FIGURE 1** A Multiple-Inheritance Hierarchy

*auxiliary-initial-construct (aux-initial-cxt).* The Aux-Initial Construction (a combinatoric construction) places general constraints on instances of the type *aux-initial-cxt*, as sketched in (9):

\[
\begin{align*}
\text{An aux-initial-cxt must satisfy:} & \\
& \begin{bmatrix} \text{MTR} & S[...] \\ DTS & \langle V[\text{AUX }+] \ldots \rangle \end{bmatrix}
\end{align*}
\]

A more specific construction, i.e. a construction that characterizes a subtype of *aux-initial-cxt*, need specify only the properties that are specific to that subtype. In the case of aux-initial clauses, the subconstructions in question specify primarily semantic information, but also syntactic constraints about independent clause status or modifier status. For example, the Inverted Wish Construction (of which (8e) is an instance) defines the characteristic properties of feature structures instantiating the maximal type *inverted-wish-construct (inv-wish-cxt).* It specifies the appropriate semantics and the constraint that a FS of this type must be an independent clause. Inverted wish constructs will of course also exhibit the general properties of (obey the general constraints on) aux-initial constructs. This is accomplished simply by specifying the type hierarchy so that *inv-wish-cxt* is a subtype of *aux-initial-
2.3 Signs

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

2.3.1 PHONOLOGY and FORM

Little will be said here about morphology, and nothing at all about phonology, but it is fully intended that phonological and morphological entities be part of linguistic signs. I assume that a largely autonomous set of constraints characterize the relation between the phonological and morphological aspects of signs. There are thus two distinct sign-level features: PHONOLOGY (PHON) and FORM:

(10) a. The value of the feature PHON is a phonological structure i.e. a FS of type phonological-object.
   b. The value of the feature FORM is of type morphological-object (morph-obj); these are the elements that will be phonologically realized within the sign’s PHON value.

The precise characterization of both these structures is left open here, though for convenience PHON values are treated as lists of segments (represented in phonemic brackets) and FORM values as lists whose members include stems and more complex entities built up from stems.

In order to deal with inflectional realization, morphological functions must make reference to stem identity. And because stems can exhibit idiosyncrasy like that shown in (11), stems must be individuated so as to allow homophonous elements to exhibit distinct inflectional patterns:

(11) a. lie/laylain ‘rest, recline’ vs. lied ‘tell falsehoods’
   b. can/could ‘be able to’ vs. canned ‘put into cans’
   c. fly/flew (basic sense) vs. flied (various derived senses)
   d. sell/sold vs. celled
   e. write/wrote/written vs. right/righted/righted

To this end, the FORM value of a sign will be represented as a list of conventional orthographic representations augmented by indices to distinguish homophonous stems with divergent inflectional realizations (e.g. lie₁ vs. lie₂). Morphological functions of the sort assumed here provide a basis for dealing with data like (11), as well as a precise way of expressing ‘elsewhere’

---

conditions in morphological realization and a means for avoiding unwarranted analysis via morpheme sequences. I assume that the FORM feature will prove dispensible, once the advantages of constraint-based morphology and phonology are further developed and more widely recognized. In the meantime, however, FORM provides a convenient shorthand for illustrating the combinatoric structure of both words and phrases.

2.3.2 ARGUMENT-STRUCTURE

The basic purpose of the ARGUMENT-STRUCTURE (ARG-ST) feature is to encode the combinatoric potential of a lexical sign by listing its potential syntactico-semantic arguments. 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, for example, the first NP of an ARG-ST is the subject, the second NP (of a transitive verb’s ARG-ST) is its direct object, etc. This ‘rank-based’ encoding of grammatical relations, as shown by Keenan and Comrie and other researchers in relation-based syntactic theory, is independently motivated by the cross-linguistic patterns that have been observed for such phenomena as relative clause accessibility (which NP in a clause can be ‘relativized’), reflexive binding (which NP in a clause can bind a reflexive), and agreement (which NP in a clause the verb can mark agreement with). In a language like English, a verb’s subject is identified as the first member of its ARG-ST list, which is also its ‘external argument’ (see the discussion of the feature XARG in section 2.3.3 below).

Variable polyadicity of a given lexeme, e.g. active vs. passive vs. middle, causative vs. inchoative, or oblique-recipient vs. ditransitive, involves differences in the ARG-ST list. These differences can arise in two distinct ways in SBCG: by derivational construction (e.g. familiar analyses of passivization or causativization) or by lexical underspecification (as in certain analyses of locative (‘spray/load’) alternations).

Some examples of lexical classes associated with particular ARG-ST lists are the following:²¹

(12)  (NP)  die, laugh, … sintrans-v-lxm
     (NP, NP) hit, like, … trans-v-lxm
     (NP, NP, NP) give, tell, … ditrans-v-lxm
     (NP, NP, PP) put, place, … loc-trans-v-lxm

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 im-

²⁰That is, they provide a way of avoiding what Hockett (1987) has termed ‘The Great Agglutinative Fraud’ (For discussion, see Blevins 2008). For a different approach to elsewhere conditions, one more in line with Stump’s (2001) adaptation of ‘Pāṇini’s Principle’, see Bonami and Samvelian submitted.

²¹Some abbreviations:
posed on particular arguments. Only lexical signs (lexemes or words) specify a value for ARG-ST.

ARG-ST lists are also the locus of constraints on coindexation (‘binding theory’). For example, a reflexive or reciprocal that is a member of an ARG-ST list must be coindexed with a sign preceding it on that list, if there is one (Principle A); personal pronominals must not be coindexed with any preceding element (Principle B).

2.3.3 SYNTAX
The value of the feature SYNTAX is a FS of type syntax-object (syn-obj). Functions of this type specify values for the features CATEGORY, VALENCE and MARKING, which I will discuss in turn.

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

(13) a. The immediate subtypes of the type category are: verbal and non-verbal. The subtypes of verbal are verb and complementizer and those of nonverbal are adverb, adjective (adj), and nominal. And the subtypes of nominal are noun and preposition (prep), yielding the hierarchy of CAT values shown in Figure 2.

\begin{align*}
\text{NP} &= \left[\text{sign} \begin{array}{c}
\text{CAT} \\
\text{noun}
\end{array} \langle \rangle \right] \\
\text{PP} &= \left[\text{sign} \begin{array}{c}
\text{CAT} \\
\text{prep}
\end{array} \langle \rangle \right] \\
\text{CP} &= \left[\text{sign} \begin{array}{c}
\text{CAT} \\
\text{comp}
\end{array} \langle \rangle \right]
\end{align*}

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. Although this tradition improves upon theories of binding based on constituent structure (including all theories based on the notion of ‘c-command’), it may still be overly reliant on grammatical structure, as argued by Runner and Kaiser (2005).

There are three further features to be included in the domain of feature structures of this type: GAP, WH, and REL. These are discussed in section 2.10.

Note that ‘CATEGORY’ denotes a feature and ‘category’ denotes a type. Features are represented in small capitals and types in lower case italics. The FSs of type category play a role similar to that of \(X\) categories, as extended by work in GPSG. For discussion, see Sag 2010b.
b. **CASE** is appropriate only for FSs of type *noun* (in English). The possible values of **CASE** (in English) are *nominative* (*nom*) and *accusative* (*acc*).\(^{25}\)

c. **VERB-FORM** (VF), appropriate only for FSs of type *verbal*, is used to specify the appropriate inflectional category of a verb or complementizer. The possible values of VF are *finite* (*fin*), *infinitive* (*inf*), *base*, *present-participle* (*prp*), *past-participle* (*psp*), and *passive-participle* (*pas*).

d. **AUXILIARY** (AUX) is used to specify whether a verb appears in one of the syntactic environments restricted to auxiliary verbs (e.g. sentential negation, inversion, contraction, or VP-Ellipsis; see section 2.9). The value of AUX is an atom of type *boolean*, the name often used to refer to a truth value, i.e. + (true) or − (false).

e. **INVERTED** (INV) is used to specify whether a verb is in clause-initial position; the values of INV are also of type *boolean*.

f. **INDEPENDENT-CLAUSE** (IC) is used to distinguish independent clauses (and the signs that project them) from their dependent clause counterparts. The values of IC, which is appropriate only for FSs of type *verbal*, are again of type *boolean*.

This partial signature countenances complex grammatical categories like those shown in (14), but none like the ones pictured in (15):

\(^{25}\)Note that genitive nominal expressions are not distinguished in terms of **CASE**. This is because case is a property of head nouns and the Modern English ’s is a phrasal clitic that appears in final position of a genitive NP, rather than as an inflection on the head noun:

(i) [[The man on the radio’s] voice] . . .

(ii)*[[The man’s on the radio] voice] . . .
FIGURE 2 Hierarchy of CATEGORY Values
It is worth repeating that attribute-value matrices (AVMs) are being used here to formulate FS descriptions. This contrasts with the objects that are being described, which are models of particular linguistic entities, i.e., functions of the appropriate kind. When a particular FS, i.e., a linguistic model, is being illustrated, rather than a description of a family of FSS, the AVM is displayed inside a box, as in (14)–(15).

As already noted, listemes are quite minimal, typically specifying just a form, a lexeme type and a meaning. But the set of possible FSS licensed by any given listeme is potentially vast, circumscribed only by the constraints of the grammar signature, which require that each appropriate feature have a value of an appropriate type. To take a simple example, the listeme licensing the proper noun Dale says nothing about the value of the feature CASE. But any given FS licensed by this listeme has a determinate value for CASE – one that is contextually resolved – in Dale likes you, it is resolved as nom; in You like Dale, it is resolved as acc. Similarly, a verbal listeme does not normally specify the FORM value of any of the signs on its ARG-ST list. However, since there is no upper bound on the length of a subject, object, or complement, there are infinitely many signs that could play each of these roles for any given verb, i.e., infinitely many distinct feature structures licensed by the verbal listeme, each with a distinctive PHON, FORM, SEM, or CONTEXT value.

Although the functional entities in our linguistic models are always total functions, essentially all of the business of grammar is conducted in terms of partial (or underspecified) descriptions of classes of these entities. Indeed, if a given grammar allows two distinct signs to have the same PHON specification, then it is predicting that this phonological structure exhibits a linguistic ambiguity. For example, the descriptions in (16) characterize more than one FS, and hence underspecify the indicated ambiguities:

(16) a. \[
\begin{array}{l}
\text{PHON} /\text{ayf}\text{æ}g\text{æt hawg}\text{a}d\text{b}\text{ir }\text{t}e\text{a}t\text{s}/ \\
\text{FORM} \langle I, \text{forgot, how, good, beer, tastes} \rangle \\
\end{array}
\]

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

b. \[
\begin{array}{l}
\text{PHON} /\text{viz}\text{æ}^{\text{t}i}\text{n}\text{r}\text{el}^{\text{at}i}\text{vz ke}\text{ni}\text{b}\text{i}^{\text{b}o}\text{r}\text{n}/ \\
\text{FORM} \langle visiting, relatives, can, be, boring \rangle \\
\end{array}
\]

(visiting is an adjective modifying relatives or else a gerund whose direct object is relatives)

c. \[
\begin{array}{l}
\text{PHON} /\text{ayl}^{\text{b}i}^{\text{li}^{\text{v}}\text{anyu}}/ \\
\end{array}
\]

(I’ll believe in you vs. I’ll be leavin’ you)

\[26\text{A caveat: a boxed AVM diagram will typically not mention all of the feature specifications included in a given feature structure.}\]
We will of course evaluate grammars in terms of their ability to model such ambiguities successfully.

The hierarchically organized CAT values just presented defines a pattern of natural classes allowing, for instance, a concise account of the constructional variation in the category of the filler phrase in various kinds of English filler-gap constructions (Sag 2010a). Similarly, this inventory of CAT features has been integrated into a system that provides a reasonably well-worked account of ‘main clause phenomena’, including construction-specific variation regarding the possibility of (or requirement for) auxiliary ‘inversion’ (Ginzburg and Sag 2000). In addition, this same feature inventory plays a critical role in the treatment of the English auxiliary system presented in Sag to appear (see section 2.9 below).

There are three other CAT features that must now be introduced:27

(17) a. SELECT is used to let an expression select what it can modify or combine with as a ‘marker’. The value of SELECT is either the distinguished atom none (in the case of expressions that are neither modifiers nor specifiers) or else a sign. If an expression’s SELECT value is a sign, then it is either a modifier (e.g. adjective, adverb) or else a marker (e.g. determiner) and its SELECT value imposes constraints on the element that it modifies or marks.

b. EXTERNAL-ARGUMENT (XARG) is used to specify the argument of an argument-taking expression that is visible from outside its local domain (i.e. from outside the phrase it projects). The value of XARG is either a sign or none.28 The external argument of a clause is its subject; an NP’s external argument is its prenominal genitive NP, if there is one (the XARG value of the NP is none, otherwise).

c. LEXICAL-IDENTIFIER (LID) is used to individuate lexical items semantically; the value of LID is a semantic frame that canonically specifies the (fine-grained) meaning of a lexeme, e.g. a FS of type book-frame (book-fr).29

---

27 Here we follow Van Eynde (1998), who builds directly on Allegranza 1998b, in eliminating Pollard and Sag’s (1994) features MOD and SPEC and SPR in favor of the single feature SELECT. The values of SELECT indicate properties of the phrasal head that are selected by a given modifier or specifier. See also Van Eynde 2006, 2007 and Allegranza 2007. The fundamental insights of the SELECT analysis presented here are indeed those of Van Eynde and Allegranza, 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 has been streamlined here, e.g. by eliminating the features HEAD and LOCAL.

28 Sag and Pollard (1991), who first introduced this feature, assumed its value to be an index, an assumption preserved in ongoing computational grammar development work using the English Resource Grammar and the Grammar Matrix. See also Copestake et al. 2005.

29 Exceptions to this are considered below (sec. 2.7.1) in the discussion of idioms.
The features select and xarg are discussed in more detail in section 2.8 below.

VALENCE

The basic function of the feature valence (val) is to specify which of an expression’s syntactic-semantic arguments it has yet to combine with syntactically. val is thus closely related to the feature ARG-ST. While the ARG-ST list specifies all of a word’s potential arguments, including those that could be ‘extracted’ in a filler-gap construction, those that could remain unexpressed, and those that could in some languages be realized morphologically instead of syntactically, the val list includes just the subset of these that are relevant to that word’s local syntactic combinatorics. I will refer to these arguments of a head as its valents.

In the simplest case, where no covert or nonlocal argument realization takes place, the ARG-ST value of a word is identical to its val list. That is, the grammar requires that a word’s val list is the same as its ARG-ST list, except that all covert expressions (see below) are removed.30 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 valents (i.e. its subject), is specified as follows:

(18) [val ⟨NP⟩]

Similarly, the clause My dad persuaded me to go, which contains all the verb’s valents, is specified as in (19):

(19) [val ⟨⟩]

The lexical head of the clause is the verb, and the phrases that it projects gradually ‘saturate’ the verb’s valence by ‘removing elements’ from the valence list.31 Clauses, NPs, pronouns, and proper names have an empty val list because they are already saturated, i.e. they need not – indeed they cannot – combine with subjects or complements. A VP or a predicative phrase of some other category has a singleton val list, reflecting the fact that it already contains (i.e. its head daughter has ‘already combined with’) all relevant com-

30In Sag et al. 2003, this is accomplished by a single grammatical constraint – the Argument Realization Principle.

31This way of looking at things, which has its origin in the argument cancellation of Categorial Grammar (Ajdakiewicz 1935; see also http://en.wikipedia.org/wiki/Categorial_grammar), involves a ‘bottom-up’ procedural metaphor where one starts with a predicador and builds successively larger phrases of which that predicador is the lexical head. It is important to recognize, however, that the constructions of an SBCG, like the rules of a Context-Free Phrase Structure (CFG) Grammar or a Categorial Grammar, are static constraints defining well-formed local tree configurations. The importance of this fundamental design property has been discussed by many over the years, including Kaplan and Bresnan 1982, Sag et al. 1986, Fenstad et al. 1987, Jackendoff 1997, 2002, Pullum and Scholz 2005, and Sag and Wasow 2011.
lements, but it still has the potential to combine with a subject.

Discrepancies between a word’s ARG-ST list and its VAL list can arise in several ways. One of these is **nonlocal realization**, discussed in section 2.10 below, where one of the word’s valents appears in a dislocated syntactic position. Another is the phenomenon of **null instantiation** (Fillmore 1986), which arises when a lexical sign undergoes a derivational construction whose syntactic consequence is that an argument (and element of the ARG-ST list) fails to appear on its VAL list. Finally, there is **morphological realization**. For example, in many varieties of the Romance languages, so-called ‘clitic’ pronouns have been shown to require reanalysis as inflectional affixes (see Miller and Monachesi 2003). This is a third kind of noncanonical argument expression, where a verb bearing pronominal affixes has a VAL list that omits specific elements of its ARG-ST list.32

**MARKING**

The feature **MARKING (MRKG)**, introduced by Pollard and Sag (1994) and refined in crucial ways by Van Eynde, is used to distinguish expressions like *than Kim read and the books* from their respective ‘unmarked’ counterparts *Kim read* and *books*. 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 (20). Some prepositions also lead a life as markers, as in the case of *than* and *as* (Hankamer 1973) and certain uses of *of*.34

(20)  
then compared phrases, e.g. *than we read*

as equated phrases, e.g. *as I could*

of some *of*-phrases, e.g. *of mine*

det ‘determined’ nominal signs (see below)

a a subtype of *det*, e.g. *a book*

def definite nominal signs, i.e. *the table, Prince, we*

An element that specifies a MRKG value other than *unmk* is informally 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 specified as [MRKG def]. The MRKG value of a marker is passed up to its mother via a constraint on the Head-Functor Construction introduced in section 2.8 below.

32For a detailed analysis of this phenomenon in French, broadly compatible with the framework developed here, see Miller and Sag 1997.

33See Van Eynde 2003, 2004, 2006. Van Eynde’s MRKG values are more complex than those assumed here, in large part because of his need to analyze complex morphological and agreement patterns absent in English. I leave open the possibility of modifying the theory of MRKG to incorporate further of Van Eynde’s insights.

34See Abeillé et al. 2006 for an analogous distinction in French.
The MRKG value of a head is passed up to its mother via constraints on certain other constructs, e.g. by the Head-Complement Construction.

2.3.4 SEMANTICS

A central thesis of CxG is that constructions can bear meaning. But there has been a disagreement in the CxG literature about whether or not ‘constructions must have meaning’. This debate has centered around the Aux-Initial (or ‘Subject-Auxiliary Inversion’) Construction discussed earlier. Goldberg (2006, ch. 8) has argued that there is a semantic commonality to the aux-initial clauses, be they interrogatives, exclamatives, wishes, etc. Fillmore (1999) argues that this is not the case and that constructions need not involve semantics.

Fillmore’s conclusion seems inevitable, even if Goldberg’s position turns out to be correct about aux-initial constructs. For example, there is a general type headed-construct that is constrained by the Headed Construction, which includes the constraints normally discussed under the rubric of the ‘Head Feature Principle’ (see sec. 2.8 below). This is a higher level construction with respect to the various headed constructions in just the same way that the Aux-Initial Construction generalizes over the various aux-initial construct types discussed by Fillmore and Goldberg. Yet it seems quite unlikely that there is some general meaning common to aux-initial, subject-predicate, head-modifier, head-marker, head-complement, and other kinds of headed constructs. Note further that such a meaning would have to be restricted to the headed constructs, i.e. absent from nonheaded constructs (otherwise, it would be formulated at some level in the type hierarchy higher than headed-construct). Surely there are some generalizations in grammar not grounded in meaning and some of these seem plausibly viewed as constructions. For further discussion of these issues, see Boas 2008a, Borsley and Newmeyer 2009, Goldberg 2009, and the references cited therein.

In sum, it is not necessary for a construction to bear meaning in SBCG. All that is at issue is whether or not a given class of signs or constructs is individuated in terms of semantic information. Although it is typical for both lexical class constructions and maximal combinatoric constructions (see sec. 2.5 below) to make reference to semantic properties, there is no reason to expect this always to be the case. Section 2.5 also includes a discussion of compositionality in SBCG, which arises from a single principle governing semantically canonical constructs.

SBCG is committed to the inclusion of a precise theory of meaning that can be used to describe the semantics of linguistic expressions in general, and the semantics of constructions in particular. In this regard, it should be noted that SBCG is in principle compatible with any explicit approach to semantic analysis. Most work in BCG has assumed some version of ‘Frame Semantics’ (Fillmore 1982, 1985, Fillmore and Baker 2010), while work in construction-
based HPSG (e.g. Ginzburg and Sag 2000) has usually embraced some version of Situation Semantics.\(^{35}\) In addition, Sag (2010a) presents a comprehensive discussion of English filler-gap constructions couched in terms of an SBCG that embraces a conventional, Montague-style possible-worlds semantics.

Here we will utilize a version of Frame Semantics, blended together with the basic approach provided by Minimal Recursion Semantics.\(^{36}\) Following the insights of recent and ongoing work in computational semantics, the representations we use are not hierarchically structured. Rather, the value of the feature \(\text{SEM}\) will include a list of labeled predications (here taken to be frames), accompanied by a set of constraints that limits the way these labels can be connected. This ‘flat’ conception of semantics simulates embedding by identifying the value of a feature in one frame with the label that identifies another frame. Thus to represent the meaning of a sentence like \textit{Lee says that Garfield ate Mighty Mouse}, we will write (21a), instead of (21b) or the like:

\begin{align*}
\text{(21) a.} & \quad \begin{array}{l}
\text{LABEL} \quad l \\
\text{SPEAKER} \quad \text{Lee} \\
\text{MESSAGE} \quad l' \\
\end{array} \quad \textbf{eating-fr} \\
& \quad \begin{array}{l}
\text{INGESTOR} \quad \text{Garfield} \\
\text{INGESTIBLE} \quad \text{Mighty-Mouse} \\
\end{array}
\end{align*}

Each semantic frame is assigned to a \textit{FS} type, with the types organized in a multiple inheritance hierarchy so that shared common properties can be assigned to common supertypes: \textit{saying-fr} is a subtype of \textit{statement-fr}; \textit{eating-fr} is a subtype of \textit{ingestion-fr}. Instances of a given (sub)type must specify values for all the features declared for that type and those inherited from its supertypes.


\(^{36}\)Wherever possible, I will borrow frames and features from FrameNet’s website (http://framenet.icsi.berkeley.edu/). The FrameNet lexicon, the practical implementation of Fillmore’s (1982) Frame Semantics, is a lexicographic database employing semantic frames as its main structuring device (Fillmore et al. 2003, Fillmore and Baker 2010).

For a general introduction to Minimal Recursion Semantics (MRS), see Copestake et al. 2005. Blackburn and Bos 2005 provides a useful introduction to the related framework of ‘Hole Semantics’, as well as a general discussion of issues in computational semantics.
types. Multiple inheritance is possible for frames, as it is for all types. The frame hierarchy furnishes many lexical entailments. For example, the fact that glimpse-fr is a subtype of see-fr, which in turn is a subtype of perceive-fr plays a crucial role in explaining the inference from She glimpsed it to She perceived it.

Our semantic discussion will be couched in terms of the following three features of the semantic objects serving as values of the feature SEM:

(22) a. INDEX is used to identify the referent of an expression. Its value is an index, functioning essentially as a variable assigned to an individual in the case of an NP (Situational indices, corresponding to VPs or Ss are discussed below).

b. LTOP (LOCAL-TOP) takes a label (of a frame) as its argument. This label is a pointer to the sign’s fully resolved semantics. It indicates the ‘top’ frame in the semantics of a sentence viewed as a rooted tree of frames in which the hierarchical relation is embedding (see below).

c. The feature FRAMES is used to specify the list of predications that together determine the meaning of a sign. The value of FRAMES is a (possibly empty) list of frames.

In order to treat modification, as well as quantification over situations or events, it is useful for frames to be able to make reference to the situations they are used to describe. To this end, we will make use of a further feature, SITUATION (SIT), whose value is a situational index. This corresponds to a Davidsonian event variable, which is, roughly, a referential index denoting the situation described by an elementary predication. Thus we have semantic objects like (23):

(23)

\[
\begin{align*}
\text{sem-obj} & \\
\text{INDEX} & \quad s \\
\text{FRAMES} & \\
\text{eating-fr} & \\
\text{LABEL} & \quad l \\
\text{SIT} & \quad s \\
\text{INGESTOR} & \quad i \\
\text{INGESTIBLE} & \quad j
\end{align*}
\]

The world of frames must encompass elements appropriate for the analysis of all parts of speech. I will assume that the type of the frame on the FRAMES list of a common noun like book, for example, can resolve either to bookpo-fr, corresponding to its ‘physical object’ sense, or to bookto-fr, which represents its ‘textual object’ sense. The book listeme, however, will be specified as in
(24), where book-fr is an immediate supertype of both these types (cn-lxm
stands for common-noun-lexeme):

\[
\begin{array}{c}
\text{FORM} \langle \text{book} \rangle \\
\text{INDEX} i \\
\text{SEM} \\
\text{FRAMES} \\
\text{LABEL} l_0 \\
\text{ENTITY} i
\end{array}
\]

Polysemy of this sort is systematic in English and is appropriately ana-
lyzed via lexical underspecification of this sort. In any model of the noun
book, the frame’s type must be either bookpo-fr or bookto-fr, since the type
of all feature structures must be maximally resolved. In addition, note that
bookpo-fr will also be a subtype of physical-object-fr and bookto-fr of textual-
object-fr, allowing more general linguistic properties of physical objects and
textual objects to be expressed in terms of constraints operating at a higher
level.

Note further that the lexical underspecification proposed here is indepen-
dently motivated on psycholinguistic grounds. Frazier and Raynor (1990)
found that examples like (25a) and (25b) are both read without any ‘garden
path’ effect:

(25) a. Apparently, the book didn’t sell, after having so many pages torn.
    b. Apparently, the book didn’t sell, after taking so long to write.

The following context in (25a) resolves the interpretation of book to its phys-
cical object sense, while in (25b), book is resolved to its textual object inter-
pretation. The fact that subjects effortlessly resolved the interpretation either
way suggests, as Frazier and Raynor argue, that the initial processing of pol-
ysemous words like book involves an underspecified semantic representation,
like the one sketched in (24). In particular contexts, of course, semantic and
pragmatic factors may force an early resolution of such ambiguities. The point
here is rather that such resolution is not forced.

This result contrasts with Frazier and Raynor’s findings with respect to
words like band, fan, bug, date, pitcher, and club – words which exhibit two
unrelated meanings. Here subjects committed to one interpretation initially,
which could cause a garden-path effect given inconsistent subsequent mate-
rial. This contrast can be explained by assuming that each of these words cor-
responds to multiple listemes, rather than to a single, underspecified listeme.
Hence the grammar provides no basis for an underspecified representation
constructed at an intermediate stage of processing.

Frame Semantics must also accommodate determiners and NPs, whose
semantic status as generalized quantifiers is now firmly established. 37 This can be accomplished by positioning a general type in the frame space that has all generalized quantifier frames as a subtype. Generalized quantifier frames specify values for the features RESTRICTION (RESTR) and SCOPE, as well as BOUND-VARIABLE (BV), and of course LABEL. The frames most-fr and few-fr are maximal in this part of the frame hierarchy, while intermediate types may be assumed to provide a locus for stating constraints that define such classes as monotone-increasing or anti-additive quantifiers, which have been discussed at length in the considerable literature on generalized quantifiers.

The NP every book will thus be modelled by a sign that includes all the information shown in (26):

Note that in (26) the label of book-fr ($l_2$) has been identified with the RESTR argument of every-fr, which intuitively conveys the information that the range of the quantification is restricted to books. Neither the SCOPE value of the generalized quantifier ($l_3$) nor its LABEL value ($l_1$) are identified with the label of any other frame. This indicates that the quantifier’s scope has not yet been assigned.

To further illustrate how the semantics works, consider example (27):

(27) Some student knows every answer.

This sentence is ambiguous – either ‘one student knows all the answers’, or else the weaker ‘there is no answer unknown to every student’. This is usually represented as a scope ambiguity, where quantifiers are here represented as (quantifier variable, restriction) (restriction is a sentential formula), as illustrated in (28):

(28) a. (some $i$, student($i$))(every $j$, answer($j$))(know($i,j$))

37See, for example, Keenan and Westerståhl 1997 and Keenan 2002 (but also Hobbs 1983, 1996, for an interesting alternative).
b. \( (\text{every } j, \text{answer}(j))(\text{some } i, \text{student}(i))(\text{know}(i,j)) \)

A slightly different, but equivalent, way of representing these meanings, in terms of \((\text{quantifier}, \text{variable}, \text{restriction}, \text{scope})\) formulas \((\text{restriction} \text{ and } \text{scope} \text{ are both formulas})\), is shown in (29):

(29)  
   a. \((\text{some } i, \text{student}(i), (\text{every } j, \text{answer}(j))(\text{know}(i,j)))\)
   b. \((\text{every } j, \text{answer}(j), (\text{some } i, \text{student}(i))(\text{know}(i,j)))\)

It is quantificational representations of this kind that are provided by our feature structures of type \textit{quantifier-frame}.

Notice that the formulas in (29), indeed all standard logic formulas, may equivalently be expressed as trees. For example, (29a) and (29b) could be represented as (30a) and (30b), respectively:

(30)  
   a. 
      \[ \begin{array}{c}
           \text{S}_1 \\
           \text{some } i \quad \text{S}_2 \\
           \text{student } i \quad \text{S}_3 \\
           \text{every } j \quad \text{S}_4 \\
           \text{answer } j \quad \text{S}_5 \\
           \text{know } i \quad \text{j}
        \end{array} \]
   
   b. 
      \[ \begin{array}{c}
           \text{S}_1 \\
           \text{every } j \quad \text{S}_4 \\
           \text{answer } j \quad \text{S}_5 \\
           \text{some } i \quad \text{S}_2 \\
           \text{student } i \quad \text{S}_3 \\
           \text{know } i \quad \text{j}
        \end{array} \]

Trees of this kind correspond to resolved MRS structures, where all scope is explicitly indicated. For example, (31) is equivalent to (30a) and (32) to
Note further that the local top (the value of \textsc{LTop}) is the label of the node at the top of the semantic tree, which corresponds to the quantifier with the widest scope in the formula. This is a ‘local’ top because the entire clause (\textit{some student knows each answer}) might be embedded within a larger semantic structure, whose semantics would have its own local top. The \textsc{LTop} is thus a useful bookkeeping device that lets us specify a sign’s semantics (the top of

\begin{itemize}
\item Resolved MRS structures must meet the further condition that every occurrence of a variable must appear within the scope of the quantifier that binds it. For convenience and clarity, I am ignoring the feature \textsc{sit} in this discussion.
\end{itemize}
its semantic tree) without committing to a particular interpretation in the case of scope ambiguity.

In a simple clause (i.e. one without verbal modifiers), the LTOP of the verb will be identified with that of the VP and the S that it projects. Since the clause may contain quantifiers, e.g. in examples like those just considered, the verb’s listeme (more precisely, the listeme of its lexeme) cannot identify the frame on its FRAMES list with its LTOP value. However, that listeme should impose the restriction that in the resolved MRS, the LTOP is either identical to the label of the verb’s frame or else ‘higher’ in the structure, where some quantifier embeds the verb’s frame in its SCOPE (and some other quantifier might embed that quantifier in its SCOPE, and so forth). Such a restriction is the same as requiring that the verb’s LTOP be identical to the simple clause’s LTOP ‘modulo quantification’. Whenever we need to impose such a condition, i.e. that $l_0$ is either equal to $l_1$ or higher than $l_1$, separated from it only by quantifiers, we will informally write $l_0 \geq l_1$.

Scope underspecification is the heart and soul of MRS, which was originally developed for computational purposes. Since scope resolution is a difficult and currently unsolved research problem, the advantage of scope underspecification in computational work, e.g. in machine translation, is significant. However, as pointed out by Sanford and Sturt (2002), it may be useful in psycholinguistics, as well. For example, certain experimental results, e.g. those of Tunstall (1998) about the processing of two-quantifier sentences, might be explained by assuming that initial sentence processing sometimes involves a scope-neutral representation. Tunstall showed that subjects found dialogues like (33) and (34) to be equally (and fully) natural:

(33) Kelly showed every photo to a critic last month.
    The critic was from a major gallery.

(34) Kelly showed every photo to a critic last month.
    The critics were from a major gallery.

The continuation in (33) imposes an ‘A > Every’ interpretation on the preceding two-quantifier sentence, while (34) resolves it to an ‘Every > A’ interpretation. The fact that both continuations are equally natural, Sanford and Sturt argue, can be simply explained by assuming that the initial process-

---

39This is not true in more complex sentences. For example, scopal modifiers in sentences like (i) introduce their own LTOP value:
(i) Some student frequently knows every answer.
Here the clause’s LTOP is inherited from the adverb, which embeds the verb’s LTOP as an argument. For further discussion, see Copestake et al. 2005.

40That research was conducted in the context of the LINQO Project at Stanford’s Center for the Study of Language and Information (CSLI), which has participated in a number of computational projects involving SBCG-style grammars.
ing of the two-quantifier sentence involves a scope-neutral representation that is unproblematically refined (monotonically) into either interpretation. Thus MRS, which allows partially resolved, as well as fully scope-neutral, representations, may prove useful in terms of the broader goal of embedding SBCG within a realistic model of language use.

Our grammar associates some student knows each answer with an unresolved MRS like the one shown in (35):^{41}

\[
\begin{align*}
\text{(35)} & \\
& \begin{array}{c}
\text{sem-obj} \\
\text{LTOP} & l_{0 \geq 5} \\
\text{FRAMES} & \begin{cases}
\text{some-fr} \\
\quad \begin{array}{c}
\text{LABEL} & l_1 \\
\text{BV} & i \\
\text{RESTR} & l_2 \\
\text{SCOPE} & l_6
\end{array}, \\
\quad \begin{array}{c}
\text{student-fr} \\
\quad \begin{array}{c}
\text{LABEL} & l_2 \\
\text{ENTITY} & i
\end{array}
\end{array}
\end{cases} \\
\quad \begin{array}{c}
\text{every-fr} \\
\quad \begin{array}{c}
\text{LABEL} & l_3 \\
\text{BV} & i \\
\text{RESTR} & l_4 \\
\text{SCOPE} & l_7
\end{array}, \\
\quad \begin{array}{c}
\text{answer-fr} \\
\quad \begin{array}{c}
\text{LABEL} & l_4 \\
\text{ENTITY} & j
\end{array}
\end{array}
\end{array}
\end{cases}
\end{array}
\end{align*}
\]

This description leaves open which scoping is intended, by not identifying the value of LTOP with the any frame’s LABEL value. However, by imposing the requirement that an MRS formula must be scope-resolved (and hence representable as a tree), we ensure that a description like (35) has two distinct models, depending on whether the resolution proceeds by identifying the labels in (36a) or those in (36b):

\[
\begin{align*}
\text{(36)} & \\
a. & l_0 = l_1, l_3 = l_6, \text{ and } l_5 = l_7 \\
b. & l_0 = l_3, l_1 = l_7, \text{ and } l_5 = l_6
\end{align*}
\]

The former gives us (31) and the latter, (32), thus accounting for the ambiguity of our two-quantifier sentence. Note finally that these are the only possible ways of resolving the scope of (35) into a semantic tree that contains all of the frames shown in (35).

^{41}Note that the treatment of quantifier restrictions presented here is in fact a simplification. In MRS, a quantifier that appears inside an NP may be scoped inside the restriction of that NP’s quantifier, as in examples like (i):

\[
\begin{align*}
\text{(i)} & \\
& \text{Each solution to a problem was posted on the internet.}
\end{align*}
\]
2.3.5 CONTEXT

I will have relatively little to say about CONTEXT (CNTXT) values here. A basic approach to the structure of CNTXT values (FSs of type context) is developed by Pollard and Sag (1994). This is based on such features as BACKGROUND (BCKGRND) and CONTEXTUAL-INDICES (C-INDS), where the latter specifies values for such features as SPEAKER (SPKR), ADDRESSEE (ADDR), and UTTERANCE-LOCATION (UTT-LOC). The context-objects in such a theory look like (37):

(37) \[
\begin{bmatrix}
\text{context} \\
\text{C-INDS} \\
\text{BCKGRND}
\end{bmatrix}
\begin{bmatrix}
\text{SPKR index} \\
\text{ADDR index} \\
\text{UTT-LOC index} \\
\cdots
\end{bmatrix}
\text{list(proposition)}
\]

The various contextual indices specify contextual elements that underly an account of indexical and deictic expressions formulated in the style of David Kaplan’s (1989) seminal work. The propositions specified in the BCKGRND value correspond to the set of utterance felicity conditions, which may in principle be contributed by any expression within the sentence being uttered. That is, the BCKGRND of a construct’s mother sign must include the elements in the BCKGRND lists of all the daughter signs.

This brings us to the analysis of proper names, which will be treated here, building on Pollard and Sag (1994), in terms of a background condition (a ‘presupposition’) that the individual being referred to by a particular proper name in fact exists (or exists in some more specific sense involving relevance to the current conversation) and is ‘so named’. The listeme licensing a proper noun like Pat will thus include information like that sketched in (38):

---


43I am of course skirting around a myriad of subtle issues about the meaning and multiple uses of names. This essentially Fregean analysis is also close to those suggested by Recanati (1997), Geurts (1997), and the variant adapted to ‘naming’ uses by Matushansky (2008). Clearly, this is not the only analysis available within SBCG. However, by making context ever-present in the derivation of signs, SBCG is compatible with a wide range of context-dependent analyses of interpretational phenomena.
It is also possible to augment these assumptions about contextual features, incorporating, for example, features like TOPIC and/or FOCUS, as in Michaelis and Lambrecht 1996. Engdahl and Vallduvi (1994, 1996) propose an analysis of ‘information packaging’ formulated in terms of the CONTEXT values structured as shown in (39):

(39) \[
\begin{align*}
\text{context} & \quad \text{INFO-STRUCTURE} \\
& \quad \quad \text{FOCUS} \quad \text{\ldots} \\
& \quad \quad \text{GROUND} \quad \text{LINK} \quad \text{\ldots} \\
& \quad \quad \text{TAl} \quad \text{\ldots} \\
\end{align*}
\]

I will not explore such elaborations here.\(^\text{44}\)

Finally, it is worth observing that there has been considerable interesting work embedding contextual structures of the sort illustrated in this section within larger models of dialogue. Ginzburg (to appear)’s theory of dialogue, which crucially employs such context-objects (associated with diverse sentence types in Ginzburg and Sag 2000), expands the inventory of relevant features to include DIALOGUEGAMEBOARD (DGB), which allows him to draw a distinction between ‘public’ and ‘private’ information relevant to di-

\(^{44}\)Nor will I explore the ways in which a set of background propositions may be structured by relations such as unilateral entailment, as in a scalar model (Fillmore et al. 1988, P. Kay 1990, Israel 1996, Schwenter 1999, Schwenter and Vaisbth 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 P. Kay 1997.
alogue. Other features, e.g. QUESTION UNDER DISCUSSION (QUD), FACTS, and MOVES provide part of a bridge between the contextual requirements of particular sentences and the general requirements of the theory of dialogue; this gives an account of such notions as utterance coherence, conversational moves, updating and downdating, and conversational repair.\footnote{Building on some of Ginzburg’s ideas, but introducing interesting modifications of the theory of DGB, Marandin and colleagues approach the problems of focus and intonational meaning (Beyssade and Marandin 2007), evidential meaning (Marandin 2008), and the meaning/force of negation (Godard and Marandin 2006).}

### 2.3.6 Signs: A Synthesis

Signs are analyzed as FSs that specify values for the five features PHON, FORM, SYN, SEM, and CNTXT, whose values have now all been introduced. Accordingly, the grammar signature contains the following type declaration:

\[
\text{sign} : \begin{cases}
\text{PHON} & \text{phon-obj} \\
\text{FORM} & \text{morph-obj} \\
\text{SYN} & \text{syn-obj} \\
\text{SEM} & \text{sem-obj} \\
\text{CNTXT} & \text{context-obj}
\end{cases}
\]

The immediate subtypes of sign are lexical-sign (lex-sign) and expression. Lexical signs include ARG-ST specifications in virtue of the type declaration in (41):

\[
\text{lex-sign} : [\text{ARG-ST list(expression)}]
\]

The immediate subtypes of expression are covert-expr(ession) and overt-expr(ession), the latter of which has the two immediate subtypes word and phrase. There are two immediate subtypes of lexical-sign – 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 a daughter in a phrasal construct, discussed in the next section) and that words share a property with lexemes that phrases lack (that is, having an ARG-ST list). These cross-cutting properties of words are analyzed by treating FSs of this type as both a kind of (overt) expression and a kind of lexical sign. The resulting multiple-inheritance hierarchy is sketched in (42):
We are now in a position to illustrate in more detail what the various signs introduced earlier will look like in SBCG. (Recall that boxed AVMs indicate a function – i.e. an item of the language model, rather than a function description, which is an item of the grammar (a constraint typically satisfied by a family of FSS)). The diagrams in (43) display the relevant properties of the FSS corresponding to the words Pat and laughed, and the lexeme laugh:46

Further abbreviations:

NP\_i = NP & [SEM\_INDEX \_i] \quad NP[\_nom] = NP & [SYN\_CAT\_CASE \_nom] 
S\_SOURCE = SOUND\_SOURCE

Recall that a feature structure is a total function on the appropriate domain but that a diagram illustrating such a function may not include every feature-value pair that it contains.
(43) 

(a)  

word  
PHON /pet/  
FORM (Pat)  
ARG-ST ()  
SYN  
  CAT [noun]  
  SELECT none  
  XARG none  
 VAL ()  
 MRKG det  
 SEM  
  IND i  
 FRAMES ()  
 CNTXT  
 BCKGRND  
  [some-fr]  
  LABEL l₁  
  BV j  
  RESTR l₂  
  SCOPE l₀  
  [naming-fr]  
  LABEL l₂  
  ENTITY j  
  NAME (Pat)  
  [equal-fr]  
  LABEL l₀  
  ARG₁ j  
  ARG₂ i  

(b)  

word  
PHON /lɛf-d/  
FORM (laughed)  
ARG-ST (NP[nom]₁)  
SYN  
  CAT [verb]  
  VF fin  
  SELECT none  
  XARG NP[nom]₁  
 MRKG unmk  
 VAL (NP[nom]₁)  
 IND s  
 LTOP l₀  
 SEM  
 FRAMES  
  [some-fr]  
  LABEL l₀  
  BV s  
  RESTR l₁  
  SCOPE l₂  
  [laughing-fr]  
  LABEL l₂  
  SIT s  
  S-SOURCE i  
  [past-fr]  
  LABEL l₁  
  ARG s  

July 10, 2011
Note that the FS in (43c) must have a determinate value for each of its features, including (at the appropriate internal level) VF. So the value *psp* illustrated here represents an arbitrary choice – any value for VF would satisfy the requirements imposed by the *laugh* listeme. Each such choice point thus gives rise to a family of well-formed FSs licensed by that listeme. Finally, observe that each of these FSs instantiates a maximal type.

The sign corresponding to the (sentential) phrase *Pat laughed* is shown in Figure 3. It should be noted that the diagram does not represent a construct, but rather a single sign. Indeed, the model that SBCG provides for each expression of a language is a sign, even though the ‘construction’ of that sign may involve a construct with that sign as mother and one or more signs as daughters.

2.4 Well-Formed Feature Structures

As already explained, the grammar signature of an SBCG defines a space of well-formed FSs by placing general constraints on the domain and range of each type of (functional) FS. The minimal conditions that must be satisfied in order for a FS to be well formed are shown in (44):

Note also that a resolved semantics for this sign must have the quantifier as its ‘top’ operator.

In this case, the quantifier’s label is identified with the LTOP value.
FIGURE 3 The Phrase Pat laughed
A feature structure $F$ is well-formed just in case:

1. $F$ instantiates a maximal type $\tau$,
2. $F$ is either an atom or a total function whose domain and range are specified for $\tau$ by the grammar signature, and
3. $F$ satisfies all constraints that the grammar imposes on $F$s of type $\tau$ and $\tau$’s supertypes.

However, if the only constraints imposed on feature structures were the general domain/range specifications of the grammar signature, then the grammar would license many $F$s that are not well-formed. To see this, consider the $F$ in Figure 4, which is not a sign of English. Given what has been said so far, Figure 4 illustrates a well-formed $F$ of type $\text{sign}$. Each feature ($\text{PHON}$, $\text{FORM}$, $\text{SYN}$, $\text{SEM}$, and $\text{CNTXT}$) has a value of an appropriate type and each of these values is also a feature structure conforming to the grammar signature. However, even though this is the case, there are numerous problems with the $F$ in Figure 4, including the following:

(45) a. This is a finite clause whose $\text{FORM}$ value is $\langle \text{Kim, the} \rangle$, yet $\text{Kim the}
is not a well-formed English expression.

b. The FORM value ⟨Kim, the⟩ cannot be phonologically realized as /pæt tu/.

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 in English by uttering /pæt tu/.

Clearly, the inventory of types and the feature declarations that are specified in the grammar’s signature are insufficient for doing the business of grammar.

Unwanted signs like the one in Figure 4 are ruled out by the inventory of listemes and constructions that are part of the SBCG of English. Each construction is a constraint defining the properties that are common to all instances of a given FS type. That is, a construction is a conditional constraint of the form shown in (46), where τ (the antecedent) is a type name and D (the consequent) is a FS description:

\[(46) \quad \tau \Rightarrow D\]

(All FSs of type τ must satisfy D.)

Once the grammar is appropriately populated with its ‘lexicon’ (a set of listemes) and its ‘constructicon’ – a set of (lexical-class and combinatoric) constructions, then the definition of well-formed FS can be augmented to include the following grammatical principle:  

\[(47) \quad \text{The Sign Principle:}\]

Every sign must be listemically or constructionally licensed, where:

a. a sign is listemically licensed only if it satisfies some listeme, and

b. a sign is constructionally licensed only if it is the mother of some well-formed construct.

The Sign Principle works in tandem with the lexicon, the constructicon, and the well-formedness conditions in (44) above. That is, it specifies a further condition that must be satisfied by FSs that are of type sign: lexemes, words, and phrases. The goal of the next two sections is to lay out some of the particulars of this relatively simple formulation of grammatical theory.

---

48 Note that my terminology here differs somewhat from that assumed by most researchers in BCG, who assume that a constructicon includes both lexical entries and constructions. Despite this difference, there is a far more important common intent, namely that of revealing the basic commonality of listemes and constructions. In SBCG, constructions are constraints on classes of signs or constructs and listemes are likewise constraints on signs. That is, because of the Sign Principle, the listemes in the lexicon together provide a disjunctive constraint on lexical signs (other than those derived by combinatoric construction). In fact, it would be possible to modify SBCG to allow a BCG-style constructicon simply by making lexical entries constructs whose DTRS list is empty. In the present context, however, any such modification seems to introduce an otherwise unnecessary level of complexity and hence will be avoided.
2.5 Constructs and Constructions

The Construction Grammar community usually defines construction informally as ‘any conventionalized pairing of form and meaning’.49

This conception of construction is preserved, but slightly refined in SBCG. A listeme is a constraint stated directly on the form-meaning correspondence of a particular sign.50 Lexical-class constructions (see below), which define the properties of classes of lexemes and words, may also include direct constraints on this correspondence. But since a SBCG is a recursive system, the constructions that define the ways that signs can combine to build larger signs (combinatoric constructions) specify form-meaning correspondences indirectly – by stating the relation between a (construct’s) mother’s form and meaning and the those of its daughters.

SBCG models constructs as FSs of the form specified in (48).51

\[(48) \quad \text{construct} : [\text{MTR sign}] \quad \text{DTRS nelist(sign)}\]

The MOTHER (MTR) feature is used to place constraints on the set of signs that are licensed by a given construct. The feature DAUGHTERS (DTRS) specifies information about the one or more signs that contribute to the analysis of a construct’s mother; the value of DTRS is a nonempty list of signs. The inventory of constructions determines which constructs are well-formed and this inventory of constructs in turn licenses a set of signs, as per the Sign Principle in (47).

The term ‘construct’ thus has a meaning here that is somewhat different from the way it has been used in the tradition of BCG. In BCG, a construct was sometimes regarded as a fleshed-out ‘FS tree’ of any degree of configurational complexity (See P. Kay 2002b). Even a single node FS tree was a construct in this conception of BCG, as well as a non-local tree like (49), referred to here as an analysis tree:

---

49 Goldberg (2006: 5) offers a slightly different statement of this definition:

Any linguistic pattern is recognized as a construction as long as some aspect of its form or function is not strictly predictable from its component parts or from other constructions recognized to exist. In addition, patterns are stored as constructions even if they are fully predictable as long as they occur with sufficient frequency.

50 ‘Form’ here is interpreted broadly to include syntactic, as well as morphological information.

51 For any type $\tau$, $\text{nelist}(\tau)$ stands for a nonempty list, each of whose members is a feature structure of type $\tau$. (Note that if $\tau$ is nonmaximal, then each feature structure will belong to some maximal subtype of $\tau$.)
In SBCG, an expression – a word, phrase or sentence – is modeled by a sign. Although such signs are usually licensed by reference to a well-formed construct, i.e. a local tree, an analysis tree like (49) (which is the rough equivalent in the present architecture of an FS tree in BCG) has no ‘official’ status in SBCG.

Such diagrams are of course useful for illustrating the recursive effect of grammatical constructions or for demonstrating that a given sign is licensed by the grammar, but they are not part of the language model, nor are they part of the grammar. A consequence of this is that there is no natural way to formulate a linguistic constraint that makes reference to global properties of analysis trees. For example, notions like ‘c-command’ have no role to play in SBCG.

From the BCG perspective then, the major theoretical changes embodied in SBCG are: (1) the use of signs and constructs as language models, rather than FS trees and (2) the imposition of locality on constructions, limiting their purview to local trees.

The immediate subtypes of construct are lexical-construct (lexical-cxt) and phrasal-construct (phrasal-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-

---

52 This is a pleasing consequence, given that c-command fails to provide the appropriate basis for the statement of binding conditions. In fact, ‘Principle C’ effects are better analyzed in discourse/pragmatic terms, while Principle A/B effects can be given a grammatical account in terms of purely local constraints like those advocated by Pollard and Sag (1992, 1994) and Manning and Sag (1998). It should be noted, however, that a minor modification of SBCG would redefine the licensed objects as analysis trees, rather than local trees, as was done, for instance, in GPSG (Gazdar et al. 1985).
construct (postinfl-ext). This hierarchy of construct types is sketched in (50):

(50)

```
    construct
     /     \\  
    lexical-cxt phrasal-cxt
     /     \\  
  deriv-cxt  infl-cxt postinfl-cxt ...
     ...
```

For each type of construct, the constructicon will contain a combinatoric construction – a conditional constraint – with that type as the antecedent of the conditional. In this way, there is a one-to-one relation between constructions and the types that name the class of FSSs that they characterize.

2.6 Licensing Words

The properties of a word – a verb, say – are partly determined by a listeme (a lexeme-description in the constructicon), partly by lexical-class constructions, and partly by derivational and/or inflectional constructions. Following traditional terminology, derivational constructions define ways that lexemes can be formed from other lexemes and inflectional constructions define the patterns by which words can be constructed from lexemes. The morphological stem, semantic frame, and valence list of laughed, for example, are specified in the listeme licensing the lexeme laugh (see below), but its inflected form and the added constraints – that (i) its VF value must be fin and (ii) the CASE value of its subject (its first, and only, ARG-ST member) must be nom –

---

53 My approach to morphology here is realizational (Matthews 1991, Anderson 1992), perhaps closest in spirit to the approach developed by Stump (2001) and related work. Lexical affixes are not signs; rather affixation effects (as well as more complex morphological processes) are determined by the morphological functions associated with specific lexical constructions. I have not addressed the issues that distinguish this kind of approach from others that have been developed. In particular, there are constructional approaches to morphology that have challenged a number of the premises that I have simply assumed here, e.g. that of Booij 2010. I believe that insights of Booij’s approach and others like it can also be preserved within SBCG. However, the issues involved, some of which involve the matter of constructional locality (see sec. 2.8.1 below), are beyond the scope of the present chapter.
are determined by the preterite construction, one of a handful of inflectional constructions that remain in the grammar of Present-Day English.

Derivational and inflectional constructions fit uniformly into a two-level mode, one that is articulated in terms of a mother and its daughter(s). 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, which licenses a construct with the word as mother and the lexeme as daughter.

2.6.1 Listemes and Lexical Class Constructions

Let us begin with listemes. As in most feature-based theories of grammar, a lexical entry is specified as a constraint relating form, syntactic category, and meaning. In SBCG, listemes contain varying degrees of information about all aspects of lexemes. A listeme is thus usually a description of a set of FSs of type *lexeme* or *word*. Since the value of the FORM feature is a list, the generalization from lexical entry to listeme is quite natural in SBCG.

Listemes are typically quite spartan. This is because an SBCG grammar contains type constraints that play a significant role in FS well-formedness. These are the *lexical-class constructions* mentioned above. The following lexical-class construction, for example, defines once and for all the properties that distinguish proper nouns from other lexical classes (*pn-lxm* stands for *proper-noun-lexeme*):

---


55 L variables range over lists of feature structures. These seem necessary in (51) to allow for arbitrarily long proper names, e.g. John Jacob Jingleheimer Smith, or definite descriptions that function as proper names, e.g. The Lord God Almighty. The notation ‘↑type’ is used to indicate the immediate supertype(s) of the type being characterized by a given construction. This is purely for the reader’s convenience, as the signature’s type hierarchy provides a complete specification of the hierarchical relations among types.
Once constraints like this are in place, a proper name’s listeme can be reduced to a specification of (1) a FORM value, (2) the relevant lexical type and (3) any exceptional properties. For example, a listeme like (52) licenses a FS like the one in Figure 5:

\[
\text{FORM} \{ L \}
\]

\[
\text{FLOW} \{ \text{noun} \}
\]

\[
\text{SELECT} \{ \text{none} \}
\]

\[
\text{XARG} \{ \text{none} \}
\]

\[
\text{VAL} \{ \}
\]

\[
\text{MRKG} \{ \text{def} \}
\]

\[
\text{IND} \{ \text{i} \}
\]

\[
\text{FRAMES} \{ \}
\]

\[
\text{BCKGRND} \{ \text{some-fr} \}
\]

\[
\text{BV} \{ j \}
\]

\[
\text{REST} \{ l_2 \}
\]

\[
\text{SCOPE} \{ l_0 \}
\]

\[
\text{equal-fr} \{ \}
\]

\[
\text{LABEL} \{ l_0 \}
\]

\[
\text{ARG1} \{ j \}
\]

\[
\text{ARG2} \{ i \}
\]

\[
\text{naming-fr} \{ \text{LABEL} \{ l_2 \}, \text{ENTITY} \{ j \}, \text{NAME} \{ L \} \}
\]

The FS in Figure 5 must obey the type constraint sketched in (51) because it is of type \text{pn-lxm}.

But this lexeme is not yet ready to combine with a VP to form a subject-predicate clause. Only overt expressions, signs of type overt-expr, may participate in phrasal constructions, as we will see. Hence a lexeme like the one in Figure 5 must give rise to a corresponding word, which is accomplished via an inflectional construction. In many other languages, this constructional ‘pumping’ of a lexeme to a phonologically identical word is of course replaced by constructions that add inflectional morphology to nouns, adjectives, or other types of lexeme.

The basic intuition behind the theoretical and terminological innovations
FIGURE 5 A Model of the Proper Noun Kim
presented here (which distinguish SBCG from earlier work in Construction Grammar), is that constructions define the patterns that organize lexical classes and the patterns for building words and phrases from other signs or sequences of signs. The constructions thus form a recursive system for generating signs. Crucially, a combinatoric construction – like a rule of a Context-Free Grammar – is a static constraint that licenses a particular kind of mother-daughter configuration (i.e. a construct). An SBCG grammar, since it contains no other structure-licensing devices, provides a declarative and order-independent characterization of sign well-formedness.

For various reasons, the class of lexical items that satisfy any listeme is infinite. Strictly speaking, this is true even in the case of the proper noun’s listeme in (52) above, because there are infinitely many indices that could serve as the value of the feature INDEX in Figure 5, corresponding to the fact that there is no principled limit to the number of people who might be named Kim. However, all feature structures that differ only in this way are equivalent for grammatical purposes; the only grammatically significant distinction among these functions is the value for the feature CASE.

In other circumstances, a given construction or listeme will license infinitely many FSSs that are substantively different from one another. This arises, for example, whenever a listeme specifies a nonempty ARG-ST list, as illustrated in (53):

\[
\begin{align*}
\text{FORM} & \quad \langle \text{laugh} \rangle \\
\text{SEM} & \quad \text{FRAMES} \quad \langle \text{laughing-fr} \rangle
\end{align*}
\]

This listeme interacts with the lexical-class constructions defining the properties of FSSs of type \text{sintrans-v-lxm} and those of its supertypes to license infinitely many FSSs like the one in Figure 6. It is important to see here that the NP, included in the ARG-ST list (and the identical FS which is the XARG value; see below) must be fully specified in Figure 6, even though neither the lexical entry in (53) nor any of the constraints that affect Figure 6 places any further restrictions on this FS.\footnote{Unlike some other approaches to construction-based grammar, SBCG does not use lexical entries to impose ‘selectional restrictions’ (in the sense of Chomsky 1965). Selectional phenomena, following McCawley’s (1971, 219) sage advice, are primarily nonlinguistic in nature.} This is as it should be, since this NP will be 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’.\footnote{To call attention to the fact that some grammatical constraint has identified two pieces of a feature structure, I label them with the same ‘tag’ – a boxed integer, e.g. \boxed{1}, or (in the case of a list) a boxed letter, e.g. \boxed{L}.}
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 (54):\textsuperscript{58}

\textsuperscript{58}Some further abbreviations:
intrans-v-lxm = intransitive-verb-lexeme
scontrol-v-lxm = subject-control-verb-lexeme
ocrcontrol-v-lxm = object-control-verb-lexeme
aux-v-lxm = auxiliary-verb-lexeme
sraising-v-lxm = subject-raising-verb-lexeme
oraising-v-lxm = object-raising-verb-lexeme
main-v-lxm = main-verb-lexeme
A given lexeme must obey the constraints specified in the listeme that licenses it; in addition it must obey all the relevant lexical class constructions (those characterizing every type that the lexeme instantiates). Hence all verbal lexemes must obey the lexical class construction in (55), which requires that a verbal lexeme be unmarked and that the first member of its ARG-ST list be its external argument:

(55) **Verb Lexeme Construction** ([lexeme]):

\[
\text{verb-lxm} \Rightarrow \begin{cases} 
\text{ARG-ST} & \langle X, \ldots \rangle \\
\text{SYN} & \begin{cases} 
\text{CAT} & \begin{cases} 
\text{verb} & \text{SELECT} & \text{none} \\
\text{XARG} & X 
\end{cases} \\
\text{MRKG} & \text{unmk}
\end{cases}
\end{cases}
\]

But lexical class constructions are stated at diverse levels, so as to affect, for example, all lexemes, all verb lexemes (as in (55)), all main-verb lexemes (as in (56)), all intransitive verb lexemes, or all instances of a particular max-
imal type of verb lexeme.

(56) **Main Verb Lexeme Construction** (\|lexeme): 

\[
\text{main-v-lex} \Rightarrow \left[ \begin{array}{c}
\text{SYN} \\
\text{CAT} \\
\text{INV} \\
\text{\ldots} \\
\text{IND} \quad s \\
\text{FRAMES} \quad \langle [\text{SIT} \; s] \rangle
\end{array} \right]
\]

A ditransitive lexeme, for example, must simultaneously satisfy the lexical class constructions that characterize the types \(\text{ditrans-verb-lex}\), \(\text{trans-verb-lex}\), \(\text{main-v-lex}\), \(\text{verb-lex}\), and \(\text{lexeme}\). A given verbal listeme can thus be streamlined, leaving it to the theory of lexical classes, embodied in the lexical class constructions, to determine which lexical properties are compatible with it.

In this approach, an underspecified listeme may sometimes be compatible with more than one maximal subtype of \(\text{lexeme}\). This provides one way of analyzing argument structure variations, as we will see when we turn to a discussion of verb alternations in section 2.7 below.

**2.6.2 Morphological Functions**

The next two sections discuss inflectional and derivational constructions. A key part of such word-building constructions are the **morphological functions**, which determine the morphological shape of a given lexeme or word. In a well-worked out theory of morphology like that of Stump 2001, there are both morpholexical rules and paradigm realization functions. The former map lexemes to lexemes and the latter map lexemes to the shape they exhibit in a particular paradigm slot, associated with an appropriate feature bundle. These entities in Stump’s theory play roughly the same role as derivational and inflectional constructions in SBCG.\(^{59}\) Thus when I speak here of ‘morphological functions’, I am talking more narrowly about the relation between the forms of two lexemes or the relation between the form of a lexeme and the form of a word that realizes that lexeme. The constructions in which the morphological functions are embedded will do some of the work that the corresponding entities in Stump’s theory are intended to do.

Let us first consider the preterite forms of verbal lexemes. Constructs licensed by the Preterite Construction – FSS of type \(\text{preterite-construct} (\text{preter-}\)

\(^{59}\)For a more detailed consideration of how to integrate Paradigm Function Morphology and constraint-based frameworks like HPSG/SBCG, see Bonami and Samvelian submitted.
ite-cxt) – have a DTRS list containing exactly one FS of type lexeme and a mother of type word. The mother must include an 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 daughter’s form under the morphological function $F_{pret}$; the semantics of the mother situates the lexeme’s situation argument in past time.

Morphological functions allow us to model ‘elsewhere’ phenomena in SBCG morphology. In addition, they provide a way of dealing with other problems posed by various kinds of irregularity. The FORM value of a given lexeme is a singleton list containing the lexical formative associated with that lexeme, which we will assume is a stem.\footnote{For some languages, morphological functions must effect stem alternations as necessary (e.g. French va-lal- ‘go’; Sanskrit gam-gacch- ‘go’). However, it is arguably the case that no English lexeme requires a multiple stem analysis. For an interesting discussion and a precise, compatible proposal for treating multiple stems in French morphology, see Bonami and Boyé 2006.}

The domain of an inflectional realization function is the set of stems and its range is the set of inflected forms, including those constructed via affixation. Note that the stems must be distinguished in some fashion in order to license inflectional variation among homophonous stems. For example, while the preterite form of have is had for all of the semantically distinct have lexemes, the preterite form of lie is lay if the lexeme’s semantics is ‘recline’, and lied if it is ‘prevaricate’. We will therefore follow Stump (2001) in positing distinct stems $lie_1$ and $lie_2$, each specified as the FORM value of the appropriate listeme.

$F_{pret}$ can be defined along the following lines:

\[
\begin{array}{|l|l|}
\hline
\text{Stem} & \text{$F_{pret}(\text{Stem})$} \\
\hline
be & \text{undefined} \\
have & \text{had} \\
lie_1 & \text{lay} \\
swim & \text{swam} \\
buy & \text{bought} \\
keep & \text{kept} \\
\ldots & \\
otherwise & \text{x-ed} \\
\hline
\end{array}
\]

We will need special specifications for the preterite forms of be (was for non-second person singular and were for 2nd person or plural). Note that this is properly analyzed in terms of constructions, as there are multiple be-listemes.
(in order to accommodate different uses of be, including a large number of multiword expressions) and each of these has all the same preterite forms. Similarly, all have-listemes show the same irregular preterite realizations, i.e. had, irrespective of meaning.

The lexeme lie_1 ‘recline’ is realized as lay by the third line of the function definition in (57) and lie_2 ‘prevaricate’ is realized as lied by the ‘otherwise’ clause. Swam and bought are unproblematic irregular forms without doublets. 61

2.6.3 Inflectional Constructions

Lexical constructs are constrained by the following type declaration, specified in the grammar’s signature:

\[(58)\]
\[
\text{lex-ext: } \left[ \text{dtrs list(lex-sign)} \right]
\]

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

In addition to the general constraints on lexical constructs, inflectional constructs have more specific properties that are also specified as part of the grammar signature:

\[(59)\]
\[
\text{inf-ext: } \left[ \text{mtr word dtrs list(lexeme)} \right]
\]

(The mother of an inflectional construct is of type word; the daughters must be lexemes. 62)

This treatment embodies the traditional intuition that inflectional constructions are resources for building words from lexemes (Recall that affixes are not signs).

An inflected word like laughed is modeled as a FS of type word, built in accordance with the Preterite Construction, sketched in (60):

---

61 Non-past-tense uses of the preterite morphological form, such as the antecedent clause of a counterfactual conditional (If I had my way,...) could in principle be licensed by a separate inflectional construction that also avails itself of F\text{pre}\_1. Alternatively, one might explore a semantics-changing, unary post-inflectional construction whose mother and daughter do not differ in FORM. In either case, special arrangements must be made to license and distinguish these two entities (counterfactual had and its past time homophone). I will not resolve these matters here.

62 There is usually, if not always, a single daughter in an inflectional construct. The requirement that the daughters be lexemes may not be universal: in languages with more ‘layered’ morphology, it is possible that the daughter is not always a lexeme. Indeed, such languages could provide motivation for a finer distinction among types of morphological construction than the standard derivational/inflectional dichotomy that I have assumed here. See Orgun 1996, Miller and Sag 1997, and Malouf 2000 for some relevant discussion.
This construction requires explanation. First, the variables $X$, $Y$, and $Z$ range over feature structures, while $L$-variables range over lists of feature structures, as before. The notation $[\text{FEATURE } X: D]$ indicates that the value of FEATURE must satisfy the description $D$ and that it is being tagged as $X$ for the purposes of identifying the value of FEATURE with some feature structure elsewhere in the FS being described. Thus the mother’s SYN value must satisfy the constraint specified after the colon – its VF value must be finite. The daughter’s SYN value (since it is also specified as $Y$) must be just like the SYN value of the mother. In addition, the sole member of mother’s FORM list must be the image under $F_{pret}$ of the daughter’s FORM value, $X$. Also, the mother’s SEM value must differ from that of the daughter in the way indicated in (60). Note finally that the LTOP value of the mother is identified with that of the daughter and the existential quantifier introduced on the mother’s FRAMES list is bounded by that LTOP.

One way of paraphrasing (60) – albeit more procedurally – is as follows: Given a verbal lexeme, one can construct a verbal word meeting the following four conditions:

(61) a. the word’s VF value is finite, as is the VF value of the lexeme,

b. the word’s FORM value is related to that of the lexeme via the morphological function $F_{pret}$.
c. the word’s SYN and ARG-ST values are identified with those of the lexeme daughter, thus requiring that everything in the listeme that licensed the lexeme be consistent with the constraints introduced by this construction, e.g. requiring the subject’s CASE value be nominative, and

d. the word’s FRAMES list includes that of the lexeme, but it also includes an existential quantifier binding a situation index (variable) restricted to past time. This index is also identified with the situation index specified in the lexeme’s frame. The quantifier’s scope is bounded by the verb’s LTOP.63

This construction therefore licenses constructs of the sort sketched in Figure 7; and because this is a well-formed construct, the FS in (62) (the mother of the construct in Figure 7) is constructionally licensed.

63This correctly allows for scope ambiguities in examples like Every man laughed: ‘Every man laughed at some time in the past’ or ‘at some time in the past, every man laughed’:

(i) l0: every(i, l1, l2), l1: man(i), l2: some(s, l3, l4), l3: past(s), l4: laugh(s, i)

‘For every man i, there is a past situation where i was laughing’

(ii) l2: some(s, l3, l0), l3: past(s), l0: every(i, l1, l4), l1: man(i), l4: laugh(s, i)

‘There is a past situation where every man was laughing’

For details, see Copestake et al. 2005.


**Figure 7** A FS of Type `preterite-construct`
Notice that it would be redundant for the construction in (60) to explicitly require that the MTR value be of type word or that the daughter be of type lexeme. Because the constructs licensed by the Preterite Construction are all instances of the type preterite-cxt, they must obey all constraints the grammar imposes on FSS of that type and its supertypes, including infl-cxt. As we have already seen, (59) requires that the MTR value of all inflectional constructs be of type word and that all daughters of inflectional constructs be of type lexeme. Hence this is true of all preterite constructs as well, by the process of constraint inheritance (see section 2.2 above). It also follows that the ‘output’ of this construction (a word) can never ‘undergo’ the construction again (since the daughter of any inflectional construction must be of type lexeme). I make the further assumption that in all English inflectional constructions, the mother and daughter share ARG-ST and CNXT values. Hence the construct in Figure 7 obeys these constraints, as well. Finally, the information encoded in Figure 7 is identical to what is represented in a more familiar diagram – the unary local tree in Figure 8. Because of their familiarity, I will use trees
whenever possible to illustrate constructs.

Finally, as a trivial example of an inflectional construction, let us consider the Zero Inflection Construction, which ‘promotes’ uninflectable lexemes to word status. This construction takes an uninflectable lexeme as daughter, licensing mothers that are words, but otherwise identical to the daughter:

(63) **Zero Inflection Construction** ($\uparrow$infl-cxt):

$$
\text{zero-infl-cxt} \Rightarrow \begin{bmatrix}
\text{MTR} & X \!: \text{word} \\
\text{DTRS} & \langle X : \text{invariant-lxm} \rangle
\end{bmatrix}
$$

This analysis assumes that $\text{invariant-lxm}$ is a supertype of $\text{pn-lxm}$ and the other lexeme types we posit for the analysis of uninflectable lexemes, e.g. prepositions, adjectives, adverbs, and conjunctions.

### 2.6.4 Derivational Constructions

Derivational constructions are structured as shown in (64):

(64) **deriv-cxt**: $\begin{bmatrix}
\text{MTR} & \text{lexeme} \\
\text{DTRS} & \text{list(lex-sign)}
\end{bmatrix}$

(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, there is an un-prefixation construction, sketched in (65), which allows un-verb lexemes to be derived from a specifiable class of verb lexemes:\footnote{The morphological function $F_{\text{un}}$ is utilized by more than one construction (see below).}

\[\text{un}-\text{prefixation construction}\]

\[\uparrow\text{deriv-cxt}:\]

$$
\begin{bmatrix}
\text{MTR} & \text{lexeme} \\
\text{DTRS} & \text{list(lex-sign)}
\end{bmatrix}
$$

(65) **Example**:

```
\begin{align*}
\text{MTR} & \text{un} \\
\text{DTRS} & \text{list(verb)}
\end{align*}
```
FIGURE 8 A FS of Type preterite-construct in Tree Notation
The formulation in (65) presupposes that only strict-transitive verbal lexemes (lexemes of type $strans-v-lxm$) can give rise to $un$-verb lexemes. However, the restrictions on this construction are partly semantic. For example, the impossibility of $uncrush$ and $unlift$ may be attributed to the fact that the events denoted by $crush$ and $lift$ are ‘irreversible’. Moreover, there are various intransitive verbs that give rise to $un$-verbs, e.g. $roll \mapsto unroll$, $congeal \mapsto uncongeal$, via what may be the same derivational construction. Likewise, there are nonce occurrences of other intransitive verbs with $un$- (e.g. $unstink$, $unwaffle$, $unburn$, $unshine$). Hence, there is room for disagreement about what constraints should be placed on (65) and whether their nature is 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 the analysis tree shown in Figure 10. Derivational constructions can also feed other derivational constructions and inflectional constructions can sometimes feed derivational constructions, e.g. in noun-noun compounds such as $grants\ secretary$ or $weapons\ specialist$, where the modifying noun bears plural inflection.$^{65}$

Sag and Boas (this volume) summarize a few of the arguments that have been made against constructional analyses based on lexical underspecifica-

---

$^{65}$Thus on empirical grounds, it seems prudent not to build any strong theory of ‘level ordering’ (Kiparsky 1982) into grammatical theory, though it would be easy enough to incorporate this idea into SBCG, simply by tightening the condition on the value of $DTRS$ in (64) above. See note 68.
FIGURE 9 The Un-Verb Construction Feeding the Able-Adjective Construction
tion and constructional inheritance. Further evidence that more structure is required in order to make sense of construct interaction, as also noted by Müller (2006), comes from the ambiguity of expressions like unlockable. In addition to the analysis sketched in Figure 10, there is another, shown in Figure 11, where the Able-Adjective Construction creates a derived lexeme lock-able from lock. This adjective may then serve as the daughter of a construct licensed by the Un-Adjective Construction, resulting in another adjective, as shown. The meanings of the two words are structured in accordance with this difference in analysis as well:

(66) a. able(un(lock))  [Figure 10]  
    b. un(able(lock))  [Figure 11]  

Thus, simply allowing the Able-Adjective Construction to unify with either the Un-Adjective Construction or the Un-Verb Construction fails to account for the observed interpretational difference. In fact, it is hard to see how any variant of the ‘construction unification’ approach is going to provide an account of facts such as these.

In addition to the constructions just discussed, all of the following phenomena are reasonably assumed to be analyzed via derivational constructions: passivization (which feeds overt inflectional constructions in many languages) and word-formation processes like adjectivalization in English (see Bresnan 2001), denominal verb formation (porch noun → porch verb; see Clark and Clark 1979), agentive noun formation (drive → driver), and various other kinds of nominalization. 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 lexical signs, members of compounds are permitted to be inflected words, as well as lexemes, subject to the particular constraints of individual derivational constructions.

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.66

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

Examples (67a) and (67b) illustrate attested innovative compounds. The examples in (67c) are also attested and exemplify some of the subtypes of noun-

FIGURE 10 The Un-Verb Construction Feeding the Able-Adjective Construction
FIGURE 11 The Able-Adjective Construction Feeding the Un-Adjective Construction
noun compounds exhibiting internal inflection.\textsuperscript{67}

It is also possible to incorporate proposals like that of Copestake and Lascarides (1997) (extending the basic intuition of Levi 1978), who posit a number of more specific constructions that specify patterns fitting particular classes of nouns together in conventionalized ways. Such explorations are beyond the scope of this chapter.\textsuperscript{58}

2.6.5 Postinflectional Constructions

Postinflectional constructs are structured as follows:

\[(\text{postinflect-context}) = \begin{bmatrix}
MTR & \text{word} \\
DTRs & \text{list(word)}
\end{bmatrix}\]

(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, e.g. by Warner (1993b), Bouma and van Noord (1994), Kim and Sag (2002), Kim 2000 and others, in terms of lexical rules that create adverb-selecting auxiliary verbs (e.g. will (combining with not)), as well as not-contracted words (e.g. didn't, couldn't) and related elements.

The analysis of finite negation, following this line of research, involves imparting to finite forms of auxiliary verbs the ability to take a negative adverbial complement, e.g. not or never. The general construction that accomplishes this, shown in (69), involves the postlexical construct type \textit{negative-aux-construct} (neg-aux-cxt):

\begin{itemize}
  \item \textsuperscript{67}See Bauer and Reynaud 2001.
  \item \textsuperscript{58}As noted, the first member of most noun-noun compounds is a lexeme (\textit{computer screen}, \textit{pumpkin bus}, etc.), but in certain types of compound, the first element is a word: \textit{algorithms course}, \textit{sales tax}, etc. Pinker (1999) and Kiparsky (1982) argue that the first member of a nominal compound cannot be a word. Bauer and Reynaud (2001), in a detailed corpus study, discuss the circumstances under which it is likely to be one. For further critical discussion and experimental evidence that there are multiple factors determining the well-formedness of plural nouns in nominal compounds (rather than a grammatical constraint blocking any such combination), see Harrison 2001 and Haskell et al. 2003.
\end{itemize}
(69) **Negative Auxiliary Construction (↑post-infl-cxt):**

\[
\begin{align*}
\text{MTR} & \quad \text{FORM} & W \\
\text{SYN} & \quad X & \begin{cases}
\text{CAT} & Y \! [\text{AUX} \! -] \\
\text{VAL} & \langle \text{Adv}[-\text{neg}] \rangle \oplus L
\end{cases} \\
\text{SEM} & \quad Z \\
\text{DTR} & \quad \text{FORM} & W \\
\text{SYN} & \quad X : \begin{cases}
\text{CAT} & Y \! [\text{AUX} \! +] \\
\text{VA F} & \text{fin} \\
\text{VAL} & L
\end{cases} \\
\text{SEM} & \quad Z
\end{align*}
\]

Here the notation ‘\[F_1 \ X \! [F_2 \ D]\]’ means that $F_1$’s value must be identical to the feature structure tagged as $X$ elsewhere in the description, except with respect to the value of the feature $F_2$, whose value must satisfy the description $D$.\(^9\) In (69), this means that the mother’s SYN value must be identical to that of the daughter, except with respect to the features CAT and VAL. In addition, the mother’s CAT and VAL values must differ from those of the daughter, as indicated in (69) and illustrated in Figure 12. Nonauxiliary verb lexemes are instances of the type main-v-lxm (see (56) above) and are always lexically specified as [AUX −]. And because (69) requires the daughter to be [AUX +], words derived from nonauxiliary lexemes can never acquire the potential to select a negative adverbial as complement. Once an auxiliary verb acquires the adverb-augmented valence, it is specified as [AUX −], and hence is ineligible to be the daughter of another negative-auxiliary construct, thus ruling out double finite negation. The auxiliary verb illustrated as the mother in Figure 12 can combine with its two complements in accordance with the Predicative Head-Complement Construction (discussed in section 2.8.3 below) to

\(^9\)It is sometimes useful to restrict feature identity so as to simply exclude values of particular features. Thus ‘$X \! [F]$’ is equivalent to ‘$X \! [F \ [\]]$’, where no constraint is imposed on the value of $F$. More generally, (i) means that the values of the features $A$ and $B$ are identical except for the values of the features $F_1, \ldots , F_n$, whose values are free to differ as they may. The difference may also be one of types, as in (ii):

\[(i) \begin{bmatrix} A & X & [F_1, \ldots , F_n] \\ B & X \end{bmatrix} \quad (ii) \begin{bmatrix} A & X & \text{type}_1 \\ B & X : \text{type}_2 \end{bmatrix}\]
license verb phrases like the one in (70):\textsuperscript{70}
\begin{equation}
(70) \quad \text{Kim} \{ V^p \ [\text{will} \ [\text{not} \ [\text{sign the letter}]]] \}
\end{equation}

For discussion of semantic aspects of this construction, see Sag to appear.

\section*{2.7 Some Expressions of Interest}

\subsection*{2.7.1 Multiword Expressions}

Multiword expressions (MWEs) are a diverse lot and do not have a unified analysis, contrary to what has often been assumed in the generative literature. Consider first the inert MWEs that Bauer (1983) refers to as ‘fixed expressions’, including: by and large, in short, kingdom come, every which way, ad hoc, jack of all trades, Palo Alto, Newcastle Upon Tyne, etc. These ‘words with spaces’ can be accommodated simply by positing listemes which specify the appropriate lexical type and whose FORM value is a non-singleton list. Since these expressions are fully lexicalized (*in shorter, *in very short), their treatment need be no more complicated than this.\textsuperscript{71}

Semantically decomposable idioms (Nunberg et al. 1994, Sag et al. 2002) present a greater challenge. These are MWEs where the idiomatic meaning is distributed throughout the meaning of the subexpressions, including spill the beans, keep tabs on, pull strings, and many others. As Nunberg et al. observe, only semantically decomposable idioms are syntactically flexible (in English) and only semantically decomposable idioms allow internal quantification and modification:

\begin{equation}
(71) \quad \text{Syntactic Flexibility:}
\begin{align*}
a. & \quad \text{Strings had been pulled to get Sandy the job.} \\
b. & \quad \text{It was the close tabs they kept on our parents that upset us most.}
\end{align*}
\end{equation}

\begin{equation}
(71) \quad \text{Internal Quantifiability:}
\begin{align*}
c. & \quad \text{The FBI kept closer tabs on Kim than they kept on Sandy.} \\
d. & \quad \text{They took more advantage of the situation than they should have.}
\end{align*}
\end{equation}

\begin{equation}
(71) \quad \text{Internal Modifiability:}
\begin{align*}
e. & \quad \text{Many Californians jumped on the bandwagon that Perot had started.} \\
f. & \quad \text{She spilled the beans that cost them the contract.}
\end{align*}
\end{equation}

\textsuperscript{70}Finally, it should be noted that it can be 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, treating a given alternation via a postinflectional construction ensures that the result cannot feed most derivational constructions.

\textsuperscript{71}For further discussion, see Sag et al. 2002 and Kay and Sag ms.
FIGURE 12 Constructing a Negation-Selecting Auxiliary Verb
These observations lead directly to the view that MWEs in this class have internal combinatorics — and this is inconsistent with the simple treatment of fixed expressions just discussed. Instead, in all the cases illustrated in (71), there must be a listeme for the verbal head and separate listemes for other appropriate parts of the dependents that these heads combine with, in accordance with independently motivated grammatical constructions. For example, we might treat the MWE pull strings via the following two listemes:\(^{72}\)

\[
\begin{array}{l}
\text{FORM} \quad \langle \text{pull} \rangle \\
\text{ARG-ST} \quad \left[\left[\text{SYN} \left[\text{NP} \left[\right]\right]\right], \left[\text{SEM} \left[\text{IND} \ i\right]\right]\right]\right] \\
\text{SYN} \quad [\text{CAT} \ [\text{LID} \ X]] \\
\text{SEM} \quad [\text{FRAMES} \left[\left[\text{pulling}_{\text{strings-fr}}\right] \left[\text{AGENT} \ i\right] \left[\text{ENTITY} \ j\right]\right]\right]
\end{array}
\]

\[
\begin{array}{l}
\text{FORM} \quad \langle \text{strings} \rangle \\
\text{ARG-ST} \quad \langle \rangle \\
\text{SYN} \quad [\text{CAT} \ [\text{LID} \ X]] \\
\text{SEM} \quad [\text{IND} \ s] \\
\text{FRAMES} \left[\left[\text{i-strings-fr}\right] \left[\text{ENTITY} \ i\right]\right]\right]
\end{array}
\]

The listeme in (72) licenses a strict-transitive verb lexeme (\textit{trans-v-lxm}; see (54) above) whose second \textsc{arg-st} member (the direct object NP) must be specified as [\text{LID i-strings-fr}]. The only NPs compatible with this specification are those headed by the idiomatic noun \textit{strings}, whose listeme is shown in (73). In this way, the idiomatic interpretation is decomposed into an idiomatic pulling frame (\textit{pulling\_strings-fr}) involving two participants, an \textsc{agent} and an entity, where the latter has idiomatic strings (\textit{i-strings-fr}) predicated of it.

The listeme in (72) licenses FSSs that will give rise to constructs licensing inflected forms, including those in (74):

(74) a. They’re pulling strings to get you the job.
   b. We have pulled strings more than once.

\(^{72}\)Note that NP[ ] abbreviates a syntax object, in contrast to NP or NP\(_i\), which abbreviates a sign.
c. We pulled strings to get invited.
d. He pulls strings whenever he can.

Moreover, this mode of analysis, which makes crucial use of independent constructions of the language, allows morphological and syntactic flexibility, as well as internal modification and quantification. The Passive Construction (a derivational construction), for example, will give rise to constructs that license the passive analogue of (72), and this in turn will allow for examples like Strings had been pulled to get Sandy the job. Finally, notice that this approach correlates the possibility of flexibility and modifiability with the decomposability of the meaning and the lexicalization of the analysis in an intuitively satisfying manner – the more ‘decomposable’ the meaning of an MWE is, the more likely it is for speakers to analyze it in terms of the general principles for composing complex expressions.

This analysis, developed in more detail in Kay and Sag ms, specifies that the frames are classified as idiomatic frames (of type i-frame) or canonical frames (of type c-frame), with idiomatic predicators (e.g. pull\textsuperscript{strings-fr}, spill\textsuperscript{beans-fr}) being classified as c-frames. The i-frame analysis is motivated by the basic fact that an idiomatic argument (e.g. strings in its idiomatic sense) can only appear in construction with the right governor (e.g. pull in its appropriate idiomatic sense). Hence, the reason why the examples in (75) only allow a nonidiomatic interpretation, and are therefore hard to contextualize, is that the listemes for the verbs in these sentences select arguments whose LID value must be of type c-frame:

(75) a. Leslie found the strings that got Pat the job.
     b. We resented their tabs.
     c. The beans impressed us.

The motivation for classifying idiomatic predicators as c-frames is that, in spite of their idiomatic meanings, they project phrases (typically VPs or Ss) that freely appear in nonidiomatic environments. That is, their distribution shows none of the restrictions that idiomatic arguments must obey, as illustrated in (76):

(76) a. I think [Kim spilled the beans],
     b. They tried to [pull strings to get Lee the job],
     c. [With [my kids [keeping tabs on the stock market]]], I can finally think of retiring.
     d. [Taking care of homeless animals] is rewarding.

In this analysis, both idiomatic arguments and idiomatic predicators correspond to the idiomatic meanings of the parts of an MWE: pull\textsuperscript{strings-fr} might be glossed as ‘exert’ and i-strings-fr as ‘influence’. Because the idiomatic meaning is distributed over the parts in this way, it is possible to modify or
quantify these parts using the very same analysis that is responsible for the modification and quantification of nonidiomatic expressions.

MWES exhibit varying degrees of morphological or syntactic flexibility. For example, *kick the bucket* ‘die’, which Bauer classifies as a ‘semi-fixed expression’, allows inflectional variants of kick (*kicked, kicking, kicks,…*), but otherwise exhibits none of the flexibility and modifiability just illustrated for decomposable idioms. Unlike the examples in (71), all of the following examples allow only nonidiomatic interpretations:

(77) a. The bucket had been kicked many times in that community.
    b. It was the bucket(s) that they had kicked that upset us most.
    c. Europeans will kick more buckets this year than last.
    d. Many Californians were kicking the bucket that the Georgetown kool aid made available to them.

*Kick the bucket* can be analyzed in terms of the following two listemes:

\[
(78) \begin{align*}
&\text{FORM} = \langle \text{kick} \rangle \\
&\text{SYN} = [\text{CAT} [\text{VF} \sim \text{pas}]] \\
&\text{ARG-ST} = \langle \text{SYN NP[ ] } \rangle, \langle \text{SYN CAT MKG the } \rangle \\
&\text{SEM} = \langle \text{FRAMES } \langle \text{death-fr PROTAGONIST } i \rangle \rangle \\
\end{align*}
\]

\[
(79) \begin{align*}
&\text{FORM} = \langle \text{bucket} \rangle \\
&\text{SYN} = [\text{CAT} [\text{LID i-bucket-fr}]] \\
&\text{SEM} = \langle \text{IND none } \rangle \\
&\text{FRAMES } \langle \rangle \\
\end{align*}
\]

Although lexemes licensed by (79) have *i-bucket-fr* as their LID value, they have no frame on their FRAMES list and an IND value of *none*. Hence the *bucket* contributes nothing to the semantic composition of the sentence and provides nothing for a modifier to modify or for a quantifier to restrict, predicting the absence of idiomatic readings for (77b-d). The failure of idiomatic *kick* to passivize (cf. (77a)) is accounted for by the constraint in (78) requiring that the VF value not be *pas*, but this could be replaced by a less stipulative account, should one be properly motivated. Note that although the idiomatic
bucket provides no semantic argument for an internal modifier, the idiomatic bucket may nonetheless be modified by metalinguistic elements, which do not make reference specifically to the common noun’s meaning or index. Thus we find contrasts like the following, discussed more fully in Kay and Sag ms:

(80) a.*Kim kicked the awful bucket.
     b.*They kicked the bucket that they knew was inevitable.
     c. They kicked the proverbial bucket.

LID is not the only feature relevant to the analysis of MWES; XARG has a significant role to play, as well. There are many English idioms that require referential and agreement identity between a possessor within an NP and some other argument of the idiom, or which assign a semantic role to the embedded possessor. Some of these are illustrated in (81)–(82):

(81) a. He lost [his/*her$_j$ marbles].
     b. They kept [their/*our$_j$ cool].

(82) a. That$_i$ made [her$_j$ hair] stand on end.
     b. That$_i$ tickled [your$_j$ fancy].

As noted in section 2.3.3 above, the presence of a prenominal genitive within an NP is encoded via a nonempty value for the feature XARG. If an object NP includes information about its prenominal genitive in its XARG value, then the listeme of a verb like lose (in its ‘lose your cool’ sense) can be formulated as in (83):

```
(83) [strans-v-lxm
     FORM ⟨ lose ⟩
     ARG-ST \[ NP$_i$, \[ XARG NP$_j$, [pron] \[
        \[ LID i-cool-fr \[
        \[ IND j \[
     SEM \[ FRAMES \[ losing$cool-fr \[
       \[ AGENT i \[
       \[ ENTITY j \[
]
```

This specification requires both that the object NP contain a prenominal pronominal genitive NP and that that pronoun be coindexed with the subject of lose (blocking *He lost your cool and the like).

I am assuming that NPs like your cool are built via the same Genitive Nominal Construction that is used for such NPs generally. This construction
requires that the mother’s XARG value be identified with the prenominal gen-
itive NP, as shown in (84):

![Diagram](image)

Thus, because only certain verbs, e.g. *keep*, *lose* and *blow* (in their relevant
idiomatic senses) select a direct object whose LID is *i-cool-fr*, these are the
only lexical governors that can cooccur with NPs like this (with *cool* in its
relevant idiomatic sense). the possessor within the *cool* NP and the subject
of governor are always coindexed. Various semantic treatments are possible.
The lexical entry in (83) assumes that *lose* is dyadic, with the direct object
NP forming a second semantic argument (the ENTITY argument).

The phenomena just discussed are outside the analytic scope of the ver-
sion of HPSG developed by Pollard and Sag (1987, 1994). As argued in Sag
2007, 2010b, these data (and others discussed in section 2.8.1 below) provide
critical motivation for the analysis of verbal and nominal signs in terms of
nonempty XARG specifications. Finally, note that XARG values, unlike VAL
lists, do not ‘shrink’ in a bottom-up progression from head daughter to mother
within an analysis tree. That is, no elements are ‘cancelled off’ an XARG list –
the information about the external argument is locally visible at the top of
the phrasal domain projected by the lexical head because XARG is a CAT fea-
ture and hence is regulated by the Head Feature Principle introduced as (110)
below.

### 2.7.2 Dative Alternations

As of June, 2010, FrameNet, which I use as the basis of the semantics pre-
sented here (see footnote 36), posits a single lexical entry to account for both
of the following subcategorizations of *give* (in its ‘caused possession’ sense):

(85) a. Sandy gave Bo the beer. [Ditransitive]
    b. Sandy gave the beer to Bo. [To-transitive]
This analysis is easily accommodated in SBCG by positing a single listeme like (86):

\[
\begin{align*}
\text{trans-verb-lxm} \\
\text{FORM} \langle \text{give} \rangle \\
\text{SEM} \left\{ \text{FRAMES} \langle \text{giving-fr} \rangle \right\}
\end{align*}
\]

This listeme licenses lexemes that are compatible with either of the following lexical-class constructions (\textit{giving}^+-fr names the supertype classifying the meanings of all alternating \textit{give}-type verbs):

\[
\begin{align*}
\text{(87) a. } \text{ditrans-lxm} \Rightarrow \\
\text{ARG-ST} \langle \text{NP}_x, \text{NP}_z, \text{NP}_y \rangle \\
\text{SEM} \text{ FRAMES} \langle \text{giving}^+-fr \rangle \\
& \quad \langle \text{DONOR} \ x, \ \text{THEME} \ y, \ \text{RECIPIENT} \ z \rangle
\end{align*}
\]

\[
\begin{align*}
\text{(87) b. } \text{to-trans-lxm} \Rightarrow \\
\text{ARG-ST} \langle \text{NP}_x, \ \text{NP}_y, \ \text{PP}_z [\text{to}] \rangle \\
\text{SEM} \text{ FRAMES} \langle \text{giving}^+-fr \rangle \\
& \quad \langle \text{DONOR} \ x, \ \text{THEME} \ y, \ \text{RECIPIENT} \ z \rangle
\end{align*}
\]

The result is that the two sentences in (85) are assigned the same basic semantics.

But meaning has played a critical role in the analysis of dative alternations. In a tradition going back at least to Green 1974, Oehlere 1975, Jackendoff 1983 and Pinker 1989, researchers have attempted to associate semantic differences of various kinds with the ditransitive and \textit{to}-transitive variants of dative alternations. A particularly influential proposal involves distinguishing the ‘caused possession’ semantics (associated with the ditransitive) from the ‘caused motion’ semantics (associated with the \textit{to}-transitive). Such an analysis is also easily implemented within SBCG by organizing the type hierarchy so as to distinguish these two meaning types, perhaps as in (88):\textsuperscript{73}

\textsuperscript{73}\textit{giving-cp-fr} abbreviates \textit{giving-caused-possession-frame}; \textit{giving-cm-fr} abbreviates \textit{giving-caused-motion-frame}; etc.
With this type hierarchy in place, the listeme for *give* can remain as formulated in (86) above, but the lexical class constructions will need to be revised along the lines of (89) in order to resolve the semantics of the ditransitive construction to ‘caused possession’ and that of the *to*-transitive to ‘caused motion’ (AGENT replaces FrameNet’s DONOR feature):

\[
\begin{align*}
(89) \quad & \text{a. ditrans-lxm} \Rightarrow \\
& \left[\begin{array}{l}
\text{ARG-ST} \\
\text{SEM} \\
\text{FRAMES}
\end{array}\right] \begin{cases}
\text{caused-poss-fr} \\
\text{AGENT} \\
\text{y}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
(89) \quad & \text{b. to-trans-lxm} \Rightarrow \\
& \left[\begin{array}{l}
\text{ARG-ST} \\
\text{SEM} \\
\text{FRAMES}
\end{array}\right] \begin{cases}
\text{caused-motion-fr} \\
\text{AGENT} \\
\text{y}
\end{cases}
\end{align*}
\]

The constructions will thus interact with the listemes to associate the appropriate valence patterns with the appropriate meanings.

However, this analysis assumes that verbs like *give* are ambiguous between a ‘caused possession’ and a ‘caused motion’ interpretation, a pervasive claim that has been challenged by Rappaport Hovav and Levin (2008) (henceforth RH&L; see also Levin 2008). RH&L argue that *give*-type verbs have only a ‘caused possession’ interpretation, while other classes, e.g. the *throw*-type and *send*-type verbs, allow both ‘caused motion’ and ‘caused possession’ interpretations (though in different ways). Their arguments for this are based on certain contrasts involving the possibility of extraction with *where* (*Where did you kick/throw/*give/*hand the ball*?), the observation that *give*-type verbs lack a motion semantics in any of their uses, and further contrasts involving metaphorical interpretations of ‘caused possession’ verbs.
One way of implementing the RH&L analysis is via the frame hierarchy sketched in (90):

(90)

caused-poss-fr giving\textsuperscript{\textdagger} -fr \ldots throwing\textsuperscript{\textdagger} -fr caused-motion-fr

giving-fr handing-fr \ldots throwing-fr \ldots

throwing-cp-fr throwing-cm-fr

The familiar semantic conditions that RH&L associate with ‘caused possession’ and ‘caused motion’, namely those in (91), can be recast as entailments (or meaning postulates) associated with the relevant frame types, in the manner outlined in Davis and Koenig 2000 and Koenig and Davis 2001:

(91)  
a. In any situation framed by a \textit{caused-poss-fr}, where \textsc{x} is the A\textsc{gent}, \textsc{y} is the R\textsc{ecipient} and \textsc{z} is the T\textsc{heme}, \textsc{x}’s action causes it to be the case that \textsc{y} has \textsc{z}.

b. In any situation framed by a \textit{caused-motion-fr}, where \textsc{x} is the A\textsc{gent}, \textsc{y} is the T\textsc{heme}, and \textsc{z} is the P\textsc{ath}, \textsc{x}’s action causes it to be the case that \textsc{y} goes along \textsc{z}.

This analysis involves three lexical class constructions, as indicated in (92):

(92)  
a. \textit{ditrans-lxm} ⇒

\[
\begin{align*}
\text{ARG-ST} & \quad \langle \text{NP}_x, \text{NP}_z, \text{NP}_y \rangle \\
\text{SEM} & \quad \text{FRAMES} \quad \langle \text{caused-poss-fr} \rangle \\
& \quad \langle \text{AGENT} \quad x \quad \text{THEME} \quad y \quad \text{RECIPIENT} \quad z \rangle
\end{align*}
\]
Here PP\[\text{to}\] abbreviates a PP headed by the bleached (or ‘case-marking’) preposition \text{to} (RH&L refer to this as an ‘allative’ use), while PP[\text{dir}] designates the class of PPs headed by a directional preposition, a class that includes the homophonous directional preposition \text{to}. Note that the \text{PATH} argument is a situation and that the unexpressed subject (\text{pro}) of the directional PP is identified with the verb’s \text{THEME} argument, treating caused-motion verbs as an instance of object-control.

The verb classes at play in this analysis, according to RH&L, are sketched in (93), with sample listemes shown in (94):

(93) a. **give-type verbs:** give, hand, lend, loan, rent, sell, . . . ; includes ‘verbs of future having’: allocate, allow, bequeath, forward, grant, offer, promise, . . .

b. **send-type verbs:** mail, send, ship, . . .

c. **throw-type verbs:** fling, flip, kick, lob, slap, shoot, throw, toss, . . .

(94)  
\[
\begin{align*}
\text{trans-verb-lxm} & \Rightarrow \\
\text{FORM} & \langle \text{give} \rangle \\
\text{SEM} & \text{FRAMES} \{ \text{[giving-fr]} \} \\
\end{align*}
\]

\[
\begin{align*}
\text{trans-verb-lxm} & \Rightarrow \\
\text{FORM} & \langle \text{send} \rangle \\
\text{SEM} & \text{FRAMES} \{ \text{[sending-fr]} \} \\
\end{align*}
\]
In sum, various analyses of the dative alternation that have been sketched in the literature find a natural home in SBCG.

2.7.3 Locative Alternations

The range of analyses available for locative alternations like (95)–(96) are similar to those just examined for the dative alternation:

(95) a. They sprayed the wall with paint.
   b. They sprayed paint on the wall.

(96) a. We loaded the truck with hay.
   b. We loaded hay onto the truck.

And analogous to the treatment in (87) above for the FrameNet analysis of dative alternations, it is possible to specify a single listeme for each spray/load verb in such a way that it is compatible with two lexical class constructions, each of which assigns the same basic meaning, modulo argument role assignment:

(97) \[
\begin{array}{c}
\text{trans-verb-lxm} \\
\text{FORM } \langle \text{spray} \rangle \\
\text{SEM } \left[ \text{FRAMES } \langle \text{spray-fr} \rangle \right]
\end{array}
\]

Under these assumptions, the alternations in (95)–(96) arise because both spray-fr and load-fr are instances of the type loc-motion-fr, and hence a listeme like (97) would be compatible with two lexical class constructions, resulting in identical semantics for pairs like (95a-b).

But investigations into locative alternations, like studies of the dative alternation, have focused on semantic differences between the two relevant valence patterns, as well as differences in the semantic interactions between these patterns and particular verbs. Various analyses since Pinker 1989 have attempted to accommodate such semantic differences. For example, Iwata (2005, 2008) proposes to distinguish two broad classes of meanings that he associates with subcategorization frames as shown in (98):

(98) a. \( \langle \text{NP, NP, PP} \rangle \) ‘X acts upon Y, thereby causing Y to go Z’
b. \( \langle \text{NP}, \text{NP}, \text{PP} \rangle \with \)

‘X acts upon Y, by exerting force over the surface\(^{74}\) of Y with Z’

Iwata distinguishes a verb’s ‘lexical’ meaning from the ‘phrasal’ meaning that results from the conceptualization of that meaning that is compatible with the choice of subcategorization frame. Of course, the latter kind of meaning can be regarded as lexemic, rather than phrasal, in SBCG, as in the various analyses of dative variants we examined in the previous section. This is because a lexeme can make reference to the relevant phrasal information (the particular complements) on its ARG-ST list.

Despite this difference, the basic intuition of proposals like Iwata’s – that there are two alternate conceptualizations of a given lexical meaning, either of which can be associated with a given listeme – can be formulated straightforwardly. For example, the type hierarchy can be organized as shown in (99):

(99)

\[
\begin{array}{c}
\text{loc-motion-fr} \quad \text{spray-fr} \quad \ldots \\
\text{load-fr} \quad \text{loc-with-fr} \\
\text{spray-lm-fr} \quad \text{spray-lw-fr} \quad \text{load-lm-fr} \quad \text{load-lw-fr}
\end{array}
\]

This partial frame hierarchy works in tandem with lexical class constructions like the following:

(100)  a. **Transitive Locative Construction:** (\(\uparrow\text{trans-verb-}l_{xm}\)):

\[
\text{trans-loc-l}_{xm} \Rightarrow \\
\begin{bmatrix}
\text{ARG-ST} \langle \text{NP}_x, \text{NP}_y, \text{PP}[\text{dir}], \text{PP} \rangle \\
\text{SEM} \text{FRAMES} \left[ \text{AGENT } x, \text{THEME } y, \text{PATH } s \right]
\end{bmatrix}
\]

b. **Applicative Construction:** (\(\uparrow\text{trans-verb-}l_{xm}\)):

---

\(^{74}\)Iwata’s imposes a further requirement that the force be exerted horizontally over the surface of Y. This is questionable, or at least in need of perspectival modification, since it is clear that one can speak of spraying a wall, for example, even if all the movement involved is vertical and the event fails to be ‘over’ (horizontally overlapping; vertically superior to) the wall.
On this proposal, like the analyses of dative alternations reviewed earlier, the classifying types (in this case, loc-motion-fr and loc-with-fr) are associated with the appropriate entailments that Iwata posits to characterize his phrasal meanings. The listemes required (one per verb) are just as before, involving semantic and lexemic underspecification like that shown in (97) above. Nonalternating verbs of various kinds (see Levin 1993) are accommodated by more specific listemes that are compatible with only one maximal type of verb lexeme.

Alternative analyses of locative alternations are possible, each with its own strengths and weaknesses. Note that there is ample room here for other proposals involving a more fine-grained semantic analysis (of both dative and locative alternations), as urged by numerous researchers (see, for example, Boas 2005, 2008b, 2010). These may be executed in terms of lexical subclasses, semantically tuned lexical-class constructions, and exceptional lexical specifications. Here my aim is simply to show that SBCG provides a comfortable environment in which precisely formulated analyses of the kind that have been proposed in the literature can be developed and better evaluated.

2.7.4 Extended Valence Constructions

Construction Grammarians have spent considerable time examining extended valence patterns like those illustrated in (101):

(101) a. Pat sneezed the napkin off the table. [Caused Motion]  
b. They drank themselves silly. ['Fake' Reflexive]  
c. Chris lied his way into the meeting. [X’s Way]  

Researchers have been influenced by different aspects of these examples. For example, Goldberg (1995, 200) is struck by the enormous variety of verbs that can cooccur with X’s way, while Boas (2003) argues that both the resultative and caused-motion constructions lack productivity. For present purposes, I will follow Goldberg in assuming that all the phenomena in (101) are appropriately analyzed in terms of grammatical constructions.

---

75P. Kay (to appear) is similarly impressed by the lack of productivity shown by the caused motion phenomenon, proposing to treat it as a 'pattern of coinage', rather than a grammatical construction.
Further examples containing the idiomatic phrase X’s way are given in (102):

(102) a. The hunters made their way into the forest.
    b. Bo ate his way across Europe.
    c. But he consummately ad-libbed his way through a largely secret press meeting. (OUP; cited by Goldberg 1995)
    d. The players will maul their way up the middle of the field. (OUP; cited by Goldberg 1995)

The support verb make occurs with X’s way more frequently than any other and involves a separate listeme whose instantiations give rise to sentences like (102a). This listeme remains separate from the construction I will propose.

Sentences containing X’s way typically situate a verb in an extended valence pattern of the form: verb + the way + path-naming (directional) expression. The meaning of the sentence, according to Levin and Rappoport (1988) and Jackendoff (1990), can be characterized in terms of motion (along the path) with the verb naming an activity that is either the means of the motion or some ‘coextensive action or manner’. Thus (102c-d) might be paraphrased as in (103) or (104), if we interpret the relation between the two relevant events as one of means:

(103) a. He consummately made his way through a largely secret press meeting by means of ad-libbing.
    b. The players made their way up the middle of the field by means of mauling.

The ‘coextensive action or manner’ interpretation associated with X’s way sentences is more naturally illustrated by examples like the following:

(104) a. She whistled her way out of the room.
      ‘She exited the room while whistling’
    b. You drank your way through college.
      ‘You went through college, drinking all the way’

The basic properties of this class of sentences can be analyzed by positing a derivational construction like the one in (105), where ME/MA abbreviates the cover term MEANS/MANNER, intended to allow both of the interpretations just illustrated.76

76On the interpretation of ‘Y ![…]’, see footnote 69.
What (105) says is that an intransitive or transitive verbal lexeme (the daughter) may give rise to another lexeme (the mother) whose ARG-ST list includes the daughter’s subject and whose FRAMES list include that of the daughter. The elements added to the mother’s ARG-ST list are an NP headed by the idiommatic noun way and a directional PP. The element added to the daughter’s FRAMES list is a going-fr whose THEME is identified with the index of the subject NP, whose PATH is identified with the PP’s situational index ($s_2$), and whose MANNER is the situational index of the daughter ($s_1$). In addition, the XARG of the way NP must be an overt pronominal coindexed with the subject NP and the unexpressed subject of the directional PP. This construction thus gives rise to constructs like the one shown in Figure 13.

Given that constructs like the one in Figure 13 are well-formed, it follows from the Sign Principle that the mother in such a construct is licensed. Such a lexeme gives rise to words that can combine with the appropriate nonsubject valents (by the Head-Complement Construction discussed in section 2.8 below) to build a VP like (106):

\[(106) \{ VP \ [ad-libbed] \ [his way] \ [through a largely secret press meeting] \} \]
FIGURE 13 A Construct Instantiating the Verb-Way Construction
And this can combine with an adverb and a subject NP to form sentences like (102c) above.\(^7\)

The three constructions illustrated in (101) form a family, i.e. they exhibit a ‘family resemblance’. For example, they all involve a unary construct whose mother’s valence extends that of the daughter. The constructions differ from one another in terms of what the valence extensions can be and what constructional meaning is involved (though these meanings also resemble one another). It is thus natural to analyze the three phenomena in terms of constructions characterizing three types of constructs that have a common supertype, just as Goldberg (1995) and P. Kay (2005) propose for the ‘Argument Structure Constructions’. This approach allows each individual construction to be simplified slightly, leaving the supertype construction to capture all properties of the family at large. I will not explore the details of this analysis here. Note, however, that if the Passive Construction allows only daughters that belong to a subtype of \textit{trans-v-lxm}, then the individual extended valence constructions can specify whether the mother belongs to \textit{derived-intrans-v-lxm} or \textit{derived-trans-v-lxm}. This allows an account of the differential ability of these derived lexemes to passivize, as illustrated in (107):

(107) a. The napkin was sneezed off the table (by Pat). \hfill \text{[Caused Motion]}
    b.\,*The men were drunk silly (by themselves). \hfill \text{[‘Fake’ Reflexive]}
    c.\,*Themselves were drunk silly (by the men). \hfill \text{[‘Fake’ Reflexive]}
    d.\,*His way was lied into the meeting (by Chris). \hfill \text{[X’s Way]}

This is not the only possible analysis of the contrasts in (107). The extended valence constructions in fact present numerous issues of syntactic and semantic interest whose resolution is beyond the scope of the present study, but which I believe can be profitably addressed within the basic approach of SBCG.

2.8 Licensing Phrases

Phrasal (syntactic) constructs work in the same way as lexical constructs, except that they empower the grammar to build phrases from overt expressions (words or other phrases), as shown in the following declaration for type \textit{phr-ext}:

\[
(108) \quad \text{phr-ext: } \begin{bmatrix} \text{MTR} & \text{phrase} \\ \text{DTRS} & \text{list(overt-exp)} \end{bmatrix}
\]

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

\(^7\)For an SBCG treatment of the analog of this construction in Dutch, see Poss 2010.
An important subtype of phr-cxt is headed-construct (hd-cxt), which introduces the further feature HEAD-DAUGHTER (HD-DTR):

(109) \[ \text{hd-cxt: } [\text{HD-DTR overt-expr}] \]
(Headed constructs have a head daughter, which is an overt expression i.e. a word or phrase.)

And an important constraint associated with headed constructs is the Head Feature Principle (HFP), which requires the mother’s syntactic category to match that of its head daughter. This constraint, which is the ‘X’ of X Theory, is stated succinctly in (110):

(110) **Headed Construction (Head Feature Principle) (↑phr-cxt):**
\[
\text{hd-cxt} \Rightarrow [\text{MTR} [\begin{array}{c}
\text{SYN} \begin{array}{c}
\text{CAT} X
\end{array}
\end{array}]]
\text{HD-DTR} [\begin{array}{c}
\text{SYN} \begin{array}{c}
\text{CAT} X
\end{array}
\end{array}]
\]

The HFP is a general constraint with significant consequences for the headed structures of a language.

### 2.8.1 The Subject-Predicate Construction

Simple declarative clauses are licensed by the Subject-Predicate Construction, sketched in (111) (*subj-head-cl* is an immediate subtype of *hd-cxt*).\(^{78}\)

(111) **Subject-Predicate Construction (↑subj-hd-cxt):**
\[
\text{subj-pred-cl} \Rightarrow [\begin{array}{c}
\text{MTR} [\begin{array}{c}
\text{SYN} \begin{array}{c}
\text{VAL} \langle \rangle
\end{array}
\end{array}]
\end{array}]
\text{DTRS } X, Z: [\begin{array}{c}
\text{SYN} \begin{array}{c}
\text{CAT}
\end{array}
\begin{array}{c}
\text{INV} -
\end{array}
\text{AUX} -
\text{MARK unmk}
\text{VAL} \langle X \rangle
\end{array}]
\text{HD-DTR} Z
\]

This construction says that two daughters can combine as long as the second is a finite (and hence verbal) sign that selects the first via the VAL feature. The mother of a subject-predicate clause will be specified as [VF fin], [INV −], [AUX −], and [MRKG unmk] because the head (second) daughter is. This is because the SYN values of mother and head daughter are identified, except for the VAL value, and also (redundantly) by the HFP. The [INV −] specification prevents VPs headed by [INV +] verbs from heading up subject

\(^{78}\)In fact, this construction should be simplified by moving certain of its constraints to the Subject-Head Construction. I ignore this refinement here.
predicate constructs (hence *I aren’t included) and the [AUX −] specification prevents VPs headed by [AUX +] verbs from doing the same (hence *I dò listen to you). Finally, the mother of a subject-predicate construct, unlike its head daughter, must be [VAL ⟨ ⟩]. That is, the sign built by this construction is a finite sentence – a verbal projection that has ‘consumed’ all the valents of its lexical head.

The Subject-Predicate Construction says nothing about semantics. This is because I am assuming that there is a general principle – the Principle of Compositionality (Sag, Wasow and Bender 2003) – that imposes the requirement that the FRAMES list of the all the daughters in a given construct be merged to form the mother’s FRAMES list. This principle and (111) together give rise to constructs like the one in Figure 14, whose mother is just the phrasal sign illustrated in Figure 3 above.

2.8.2 Issues of Locality

Although SBCG 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 nonlocal. 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 phrase-internal head word whose category is of type noun, other feature specifications can encode key grammatical information about an element that is present in (or absent from) a phrase. For example, the VF value of a verbal phrase (VP or S) encodes a morphosyntactic property of the head word within that phrase. Similarly, the feature GAP is used to encode the absence of an ‘extracted’ element (or, as syntacticians often put it, the ‘presence of a gap’) within a given phrase. By developing a theory of such feature specifications and the principles that govern their distribution throughout constructs, we are ipso facto developing a theory of what nonlocal information can be lexically selected at a higher level of structure.

As has been recognized at least since Chomsky (1965), lexical restric-

---

79 Following Copestake et al. 2005, there is a further feature CONSTRUCTIONAL-CONTENT, whose value is a possibly empty list of frames that is also included in the mother’s FRAMES list. This treatment (see the appendix) systematically allows for constructional meaning.

80 Figure 14 also illustrates the effect of two further constraints: the first, applying to most headed constructs, identifies the mother’s LTOP with that of the head daughter; the second, applying to all constructs, requires the mother’s BCKGRND list to include all the frames of the daughters’ BCKGRND lists.


82 See also Kajita 1968 and Sag 2010b.
FIGURE 14 A Subject-Predicate Clause
tions are circumscribed, i.e. they are localized in a fashion that must be delimited by grammatical theory. Behind the search for the precise characterization of the relevant notion of selectional locality 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\textsuperscript{83}) clearly involves the subjects of verbs, the most likely first approximation of the relevant locality domain can be formulated as follows:

\begin{equation}
(112) \textbf{Selectional Locality}
\end{equation}

For purposes of category selection (subcategorization), case assignment, (constructional, not anaphoric) agreement, and semantic role assignment, a lexical head has access only to those elements that are on its ARG-ST or SELECT list, i.e. the elements that it is connected to via a grammatical relation (subject of, modifier of, etc.).

In SBCG, this amounts to a restriction that the only nonlocal elements that can be selected are those whose grammatical information is encoded by phrases that contain them. That is, a verb can require the noun that follows it to be accusative because accusativity is a property of the noun phrase projected from that noun. 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 (112).

This mode of analysis is implicit in \textit{X}-Theory, which since its inception has involved the percolation of category information from a lexical head to the maximal phrase it projects. Work in GPSG and HPSG extended the set of HEAD features to include features like CASE, VF, and AGR, and thus expanded the scope of \textit{X}-Theory to include the percolation of case, verb inflection, and agreement information through exactly the same domains. The HPSG/SBCG theory of VALENCE (SUBCAT), SELECT (MOD), and GAP (SLASH) extends the range of feature-based analysis to include the selection of valents by heads, the selection of modified elements by modifiers, and the cooccurrence of fillers and gaps.

And by adding a specific feature like XARG to systematically propagate certain information about elements embedded within dependents, we have localized certain nonlocal information, making it available to an element higher in a structure that selects or imposes constraints on the subject NP within a clause. For example, Bender and Flickinger (1999) analyze agreement in English tag questions by allowing the subject’s agreement information to perco-

late up to the top of the clause via $X_{ARG}$. When a clause is combined with the tag, the $X_{ARG}$ values of the two daughters must be compatible. This induces the familiar tag question agreement pattern illustrated in (113):

(113) $\begin{cases} \text{they} \\ *\text{(s)he} \\ *\text{we} \\ *\text{you} \\ *\text{I} \end{cases}$

[They left,] didn’t

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

(114) **Constructional Locality:**

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 nonlocal dependencies in grammar, it simply requires that all such dependencies be locally encoded in signs in such a way that information about them can be accessed locally at higher levels of structure.

On the basis of data like (115), it is clear that the agreement between the two subjects here is semantic in nature, whereas the agreement between each verb and its subject is intuitively syntactic in nature.

(115) 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?

Notice, however, that in an analysis along the lines shown in Figure 15, the seemingly nonlocal agreement relation between the two subject NPs is localized. That is, by positing $X_{ARG}$ values reflecting the agreement properties of a clause’s subject, we make it possible to treat the agreement in tag questions via a constraint requiring the relevant semantic relation between the $X_{ARG}$ value of the main clause and the pronominal $X_{ARG}$ value of the tag clause (the NPs that are boxed in Figure 15).

There is further independent motivation for the feature $X_{ARG}$. A case in point is English ‘copy raising’ (Rogers 1974, Potsdam and Runner 2001, 84See Oehrle 1987, Culicover 1992, and P. Kay 2002a for discussion.

123
Asudeh 2002), illustrated in (116):\(^{85}\)

\[(116)\]

\[
\begin{align*}
\text{a. There looks like} & \quad \{ \text{there's going to be a storm} \} \\
& \quad \{ \text{*it's going to rain} \} \\
& \quad \{ \text{*Kim's going to win} \}
\end{align*}
\]

\[
\begin{align*}
\text{b. ?Kim looks like} & \quad \{ \text{there's going to be a storm} \} \\
& \quad \{ \text{it's going to rain} \} \\
& \quad \{ \text{Pat's going to win} \}
\end{align*}
\]

Assuming, following Pollard and Sag (1994) that there are three subtypes of the type \textit{index} – \textit{ref} (referential-index), \textit{it} and \textit{there} – contrasts like these can be treated simply by associating the relevant \textit{look} lexeme with the \textsc{arg-st} list in (117).\(^{86,87}\)

\(^{85}\)These judgments are nuanced, as \textit{looks (like)} is systematically ambiguous between a copy raising verb and a dyadic predicator where the subject is assigned a semantic role. For this reason, the examples in (116b) are possible on the latter reading, though the acceptability is slightly degraded.

\(^{86}\)An alternate analysis would treat \textit{like} as a marker that combines with a finite clause.

\(^{87}\)Also relevant are controlled pronominal subjects in Serbo-Croatian (Zec 1987), Halkomelem Salish (Gerðts and Hukari 2001) and other languages, where control verbs also include the \textsc{arg-st} specification in (116). The analytic problems of raising across Polish prepositions (Przepiórkowski 1999, Dickinson 2004) and complementizer agreement in Eastern Dutch dialects (Höhle 1997) are similar, and submit to similar analysis.
As noted above [(55)], the XARG value of a verbal lexeme is also the first member of the verb’s ARG-ST list (and of its VAL list). Hence (117) guarantees that the clausal complement of look will have a pronominal subject coindexed with the subject of look—a nonlocal dependency locally encoded.

2.8.3 The Head-Complement Constructions

With these lexical contrasts in place, we can now discuss the basic analysis of VPs, APs, PPs, and Ns. There are two basic patterns for complement realization in English and many other languages. The first, which is typical of predicative expressions of all categories and also of VPs, requires all complements except the subject to be realized within a head-complement construct, as illustrated in Figure 16. Note that the mother in such a construct retains the head daughter’s subject valent on its VAL list and as its XARG value.

The second pattern of complement realization is utilized by ‘case-marking’ prepositions of the sort we have already seen in section 2.7.2 above. According to this ‘saturating’ mode of realization, all the head’s valents are realized as sisters of the lexical head, as shown in Figure 17.

These two patterns can be analyzed in terms of the following two phrasal constructions:

(118) Predicational Head-Complement Construction (↑hd-ctx):

\[
\text{pred-hd-comp-ctx} \Rightarrow \left[ \text{MTR} \left[ \text{SYN} \ V \ [\text{VAL} \ Y] \right] \right.
\]

\[
\left[ \text{DTRS} \ (Z) \oplus L : \text{nelist} \right.
\]

\[
\left[ \text{HD-DTR} \ Z : \left[ \text{SYN} \ X : \left[ \text{CAT} \ [\text{XARG} \ Y] \right] \right. \right.
\]

\[
\left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \left. \ left
FIGURE 16 A Predicational Head-Complement Construct
FIGURE 17  A Saturational Head-Complement Construct
(19) **Saturational Head-Complement Construction (**\(hd\)-**cxt\):**

\[
sat-hd-comp-cxt \Rightarrow \begin{cases} 
  \text{MTR} & [\text{SYN } X ! [\text{VAL } (\ )] ] \\
  \text{DTRS} & \langle Z \rangle \oplus L : \text{nelist} \\
  \text{word} & \\
  \text{HD-DTR} & Z : \begin{cases} 
    \text{SYN } X : \begin{cases} 
      \text{CAT} \begin{cases} 
        \text{prep} \\
        \text{XARG none}
      \end{cases} \\
      \text{VAL } L
    \end{cases} \\
  \end{cases}
\end{cases}
\]

What (18) says is that a word that specifies an external argument may combine with all its valents except the first to form a phrase. (19) allows a preposition lacking an external argument to combine with all its valents. The mother’s **SYN** value in both constructions must match that of the first daughter (the head daughter), except for the **VAL** value. This analysis thus presupposes the existence of lexemes like (20) and (21).\(^{88}\)

(20)

\[
\begin{array}{c}
\text{FORM} \\
\langle \text{love} \rangle \\
\text{SYN} \\
\begin{cases} 
  \text{CAT} \\
  \text{MRKG} \\
  \text{VAL} \\
  \text{ARG-ST} \\
  \text{SEM}
\end{cases} \\
\begin{cases} 
  \text{verb} \\
  \text{SELECT none} \\
  \{ \text{NP}_i, \text{NP}_j \} \\
  \{ \text{IND } s \}
\end{cases}
\end{array}
\]

\(^{88}\)These constructs of course also obey further constraints on linear order that I will not discuss here. For convenience, I have omitted 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 simplification of a complex set of issues that have motivated ID-LP format (the separation of constructions and the principles that order their daughters) and ‘Linearization Theory’, the augmentation of sign-based grammar to allow interleaving of daughters as an account of word order freedom. On ID-LP grammars, see Gazdar and Pullum 1981, Gazdar et al. 1985, and Pollard and Sag 1987, among others. On Linearization Theory, see Reape 1994, Müller 1995, 1999, Donohue and Sag 1999, Kathol 2000, 2002, 2004, and Daniels and Meurers 2004.
2.8.4 The Head-Functor Construction

We now turn to the Head-Functor Construction. I follow the essential insights of Van Eynde (1998, 2006, 2007), who argues that significant generalizations in the grammar of nominals are missed by analyses based on so-called ‘functional categories’. In their place, he develops a unified analysis of markers (including determiners) and modifiers in terms of a simple, direct combination of a ‘functor’ expression and the head that it selects, based on the SELECT feature, discussed in section 2.3.3 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 none. Attributive adjectives, by contrast, select unmarked nominal heads and are themselves unmarked ([MKG unmk]). Determiners are similar, but have a specific MKG value, as illustrated in (122):

(122)  
```plaintext
<table>
<thead>
<tr>
<th>FORM</th>
<th>⟨ of ⟩</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT</td>
<td>[prep]</td>
</tr>
<tr>
<td>XARG</td>
<td>none</td>
</tr>
<tr>
<td>SELECT</td>
<td>none</td>
</tr>
<tr>
<td>VAL</td>
<td>⟨ X : NP ⟩</td>
</tr>
<tr>
<td>MRKG</td>
<td>unmk</td>
</tr>
<tr>
<td>ARG-ST</td>
<td>⟨ X ⟩</td>
</tr>
<tr>
<td>SEM</td>
<td>[FRAMES ⟨ ⟩ ]</td>
</tr>
</tbody>
</table>
```

See also Hudson 2000, 2004.
Given lexical specifications like these, we can formulate the Head-Functor Construction as follows:

(123) **Head-Functor Construction: (∣*hd-cxt*):**

\[
hd-func-cxt \Rightarrow \begin{cases}
\text{MTR} & [\text{SYN } X \mid \text{MRKG } M] \\
\text{DTRS} & \left[ \begin{array}{c}
\text{SYN} \\
\text{CAT}_{\text{SELECT } Y} \frac{\text{MRKG } M}{\text{MRKG } \text{unmk}}
\end{array} \right], Y : [\text{SYN } X]
\end{cases}
\]

This construction allows us to construct both modified and determined nominal phrases, as shown in (124)–(125). Expressions like *happy the puppy* are blocked because attributive adjectives (e.g. *happy*) select unmarked nominals.

---

90 The details in (122) need not all be specified in individual listemes. I am here conflating the effect of lexical class constructions and listemes for the sake of expedience.
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.

2.9 Auxiliaries

As is well known, English makes an important distinction between auxiliary and non-auxiliary phenomena. The data in (126) illustrate what have been referred to (see Quirk et al. 1985, Warner 1993a) as the NICE properties:

\[
\text{(126) The NICE Properties:}
\]

Negation (Finite): Lee will not eat apples / *Kim eats not apples.
Inversion: Has Lee eaten apples? / *Eats Lee apples?
Contraction of not: didn’t, shouldn’t / *eatn’t,...
Ellipsis (of VP): Kim isn’t kicking the ball, but Lee is _ / *but Lee likes _.

Following Sag (to appear), we add a fifth property of (finite) auxiliaries: their ability to perform a ‘reaffirmation’ function either by being prosodically focused or by combining with the particles too or so (in American varieties). These combinations are used to reaffirm the truth of a proposition that has just been denied:

\[
\text{(127) A: Kim won’t read it.}
\]
\[
\text{B: Kim WILL read it.}
\]

\[
\text{(128) A: Kim won’t read it.}
\]
The resulting NICER properties constitute the basic empirical domain that analyses of the English Auxiliary System (EAS) must address.

The basis of this analysis of the EAS is the contrast between the listemes licensing auxiliary and non-auxiliary verbal elements. Auxiliary verbs belong to the type aux-v-lxm (a subtype of sraising-v-lxm; see (54) above) and (with various lexical exceptions, as noted) are unspecified for both AUX and INV. Non-auxiliary verbs, by contrast, instantiate the type main-v-lxm and are required to be \[\text{AUX} - \] and \[\text{INV} - \], as was shown in (56) above. This allows listemes to be simplified, as illustrated in (129):

\[
\begin{align*}
\text{(129) a.} & \quad \text{aux-lxm} \\
& \{ \text{can} \} \\
& \text{FORM} \\
& \text{ARG-ST} \left[ X, \left[ \begin{array}{c} \text{VP} \\
\text{VF base} \\
\text{IND s} \end{array} \right] \right] \\
& \text{SYN} \left[ \text{CAT} \left[ \text{VF fin} \right] \right] \\
& \text{SEM} \left[ \text{FRAMES} \left[ \begin{array}{c} \text{can-fr} \\
\text{EVENT s} \end{array} \right] \right] \\
\text{b.} & \quad \text{nonaux-sraising-v-lxm} \\
& \{ \text{continue} \} \\
& \text{FORM} \\
& \text{ARG-ST} \left[ X, \left[ \begin{array}{c} \text{VP} \\
\text{IND s} \end{array} \right] \right] \\
& \text{SEM} \left[ \text{FRAMES} \left[ \begin{array}{c} \text{continuing-fr} \\
\text{EVENT s} \end{array} \right] \right]
\end{align*}
\]

This gives rise to lexeme contrasts like the following:
Four of the NICER properties (negation, contraction, ellipsis, and reaffirmation) are analyzed in terms of lexical constructions whose daughter must be specified as [\text{AUX} +], hence excluding all nonauxiliary verbs. The mother in all the constructs allowed by these constructions, however, must be [\text{AUX} -], as we saw in the analysis of finite negation in section 2.6.5 above. The resulting [\text{AUX} -] words then project [\text{AUX} -] VPs, in accordance with the Head Feature Principle and the Predicational Head-Complement
Construction presented in section 2.8.3 above. Since the Subject-Predicate Construction in section 2.8.1 above requires a head daughter that is [AUX −], this provides an analysis of sentences like those in (132):

(132) a. Lee can not eat apples.
    b. Lee can’t eat apples.
    c. Lee can’t __.
    d. Lee can so/too eat apples.
    e. Lee CAN eat apples.

In sentences like (133), by contrast, the listemically unspecified AUX value of *is* has simply been resolved to ‘−’:

(133) a. Kim can eat apples.
    b. Kim is eating an apple.

Since finite forms of the auxiliary verbs, but not those of nonauxiliary verbs, can be resolved to [INV +], it also follows that only auxiliary verbs will be able to function as the head daughter in the various aux-initial constructs, some of which were discussed in section 2.2 above (see also Fillmore et al. this volume). Sag (to appear) discusses both positive and negative exceptions to ‘inversion’, showing that specifying a listeme as [INV +] guarantees that the words licensed via that listeme appear only in aux-initial contexts (*I aren’t the right choice*), while a listemic [INV −] specification prevents an element from occurring in such contexts (*Better they do that*). This provides an analysis covering a range of data that has never been systematized by any transformational treatment, as far as I am aware.

Finally, Sag (to appear) provides an account of the exceptional auxiliary verb *do*, which has required considerable machinery within previous transformational analyses. The proposed treatment involves nothing more than a lexical exception: auxiliary *do* is lexically specified as [AUX +]. Because it is so specified, it can appear in any of the NICER environments, but it cannot appear in an environment requiring that its AUX value simply resolve to ‘−’. That is, it cannot appear in examples like (134):

(134)*Kim did eat apples.

The essential ingredient of this analysis of *do* is the reinterpretation of the feature AUX. In previous analyses, the specification [AUX +] meant ‘I am an auxiliary verb’, while in the present proposal, this specification means ‘I am in a special auxiliary context (i.e. one of the NICER contexts)’. The difference in interpretation allows a straightforward account of the exceptionality of auxiliary *do*. For a fuller discussion, including a treatment of the vexing problem of nonfinite auxiliary *do* in British English, e.g. (135), see Sag to appear.
For one thing, a postponement will be seen worldwide as a declaration that Britain is in crisis; tourism would suffer even more than it is doing already. *(The Guardian, 24/03/2001, cited in Miller 2002, q.v.)*

### 2.10 Filler-Gap Constructions

There are numerous ‘filler-gap’ constructions in English, including the following, all of which are surveyed in Sag 2010a (q.v.):

1. **Wh-Interrogative Clause:**
   a. {{How foolish} [is he]}
   b. I wonder {{how foolish} [he is]}.

2. **Wh-Exclamative Clause:**
   a. {{What a fool} [he is]}
   b. It’s amazing {{how odd} [they are]}.

3. **Topicalized Clause:**
   {{The bagels,} [I like]}.

4. **Wh-Relative Clause:**
   a. I met the person {{who} [they nominated]}.  
   b. I’m looking for a bank {{in which} [to place my trust]}

5. **The-Clause:**
   a. The more people I met, {{the happier} [I became]}.  
   b. {{The more people} [I met]}, the happier I became.

All five kinds of clause exhibit a filler-gap dependency between a clause-initial filler phrase and a gap located within the sentential head daughter. However there are a number of parameters of variation distinguishing these varieties of clause from one another, including the following:

6. **Parameters of Variation in FG Clauses:**
   a. Is there a distinguished *wh* element in the filler daughter, and if so, what kind?
   b. What are the possible syntactic categories of the filler daughter?
   c. What are the possible syntactic categories of the head daughter?
   d. Can the head daughter be inverted/finite? Must it be?
   e. What is the semantics and/or syntactic category of the mother?
   f. What is the semantics and/or syntactic category of the head daughter?
   g. Is the clause an island? Must it be an ‘independent clause’?
Following Gazdar (1981), the analysis of filler-gap dependencies naturally breaks down into three problems: (1) the binding environment, where the filler is introduced, (2) the filler-gap dependency path, and (3) the realization of the gap. Building on a long tradition, beginning with Gazdar’s pioneering work and including Pollard and Sag 1994, Bouma et al. 2001, and Levine and Hukari 2006, the presence of a gap is encoded in terms of a nonempty specification for the feature \texttt{GAP}. (e.g. \texttt{[GAP (NP)]}). By contrast, an expression containing no unbound gaps is specified as \texttt{[GAP ( )]}. Here I follow Ginzburg and Sag (2000), whose traceless analysis allows a lexical head to appear without a valent (subject, object, or other complement) and its \texttt{GAP} list contains an element corresponding to that valent. That is, a word’s \texttt{VAL} list is shorter than its \texttt{ARG-ST} list just in case the missing element is on the word’s \texttt{GAP} list. These \texttt{GAP} lists must also include elements that are on the \texttt{GAP} lists of the word’s valents, as shown in (142):

(142) a. No Gap (\textit{Bo likes Lou}):

\[
\begin{array}{c}
\text{FORM} \\
\text{ARG-ST} \\
\text{SYN}
\end{array} \begin{array}{c}
\{ \text{\texttt{likes}} \} \\
\{ \text{\texttt{SYN NP}} \}, \{ \text{\texttt{SYN NP}} \} \\
\{ \text{\texttt{VAL}} \}, \{ \text{\texttt{GAP}} \}
\end{array}
\]

b. Object Gap (\textit{that Bo likes} \_

\[
\begin{array}{c}
\text{FORM} \\
\text{ARG-ST} \\
\text{SYN}
\end{array} \begin{array}{c}
\{ \text{\texttt{likes}} \} \\
\{ \text{\texttt{SYN NP}} \}, \{ \text{\texttt{SYN NP}} \} \\
\{ \text{\texttt{VAL}} \} \\
\{ \text{\texttt{GAP}} \}
\end{array}
\]

c. Gap within Object (\textit{that Bo likes [your review of} \_

\[
\begin{array}{c}
\text{FORM} \\
\text{ARG-ST} \\
\text{SYN}
\end{array} \begin{array}{c}
\{ \text{\texttt{likes}} \} \\
\{ \text{\texttt{SYN NP}} \}, \{ \text{\texttt{SYN NP}} \} \\
\{ \text{\texttt{VAL}} \} \\
\{ \text{\texttt{GAP}} \}
\end{array}
\]
d. Gaps within Subject and Object

(\textit{that [proponents of \_] like [my discussion of \_]}):

\[
\begin{align*}
\text{FORM} & \quad \langle \text{\textit{likes}} \rangle \\
\text{ARG-ST} & \quad \langle \left[ \begin{array}{c}
\text{SYN} \\
\text{GAP} \\
\text{NP}
\end{array} \right], \left[ \begin{array}{c}
\text{SYN} \\
\text{GAP} \\
\text{NP}
\end{array} \right] \rangle \\
\text{SYN} & \quad \langle \left[ \begin{array}{c}
\text{VAL} \\
\text{GAP} \\
\text{NP}
\end{array} \right] \rangle
\end{align*}
\]

In (142a), neither of the verb’s valents contains a gap; hence the GAP value of both the subject and the object is the empty list, which in turn is registered as the verb’s GAP value. The entire object NP is identified with the verb’s GAP value in (142b), and in (142b) it is the GAP value of the object NP that is identified with the verb’s GAP value. Finally, in (142d), the GAP values of the subject and object arguments are merged and identified with the verb’s GAP value. This gives rise to (so-called) parasitic gaps, where two gaps are associated with a single filler. Note that in all cases shown in (142), the verb registers the information about what unbound gaps appear in its local syntactic context. This information is passed up to higher syntactic contexts by simple constraints. For example, in non-gap-binding constructs, a head daughter’s GAP list must be the same as its mother’s GAP list. Thus GAP specifications are inherited precisely as indicated in the structure shown in Figure 18.

There are three other features that play a role in the analysis of filler-gap constructions: \textit{WH}, \textit{REL}, and \textit{STORE}. Specifications for all of these features percolate up through the filler daughter to provide an account of the ‘pied-piping’ phenomenon. For example, in (143), the filler NP \textit{whose friend} has a nonempty WH value that percolated up from the interrogative \textit{wh}-word it contains:

\[(143) \quad \{[\text{Whose suggestion}] \ [\text{do you think Kim likes?}]\}\]

The percolation here obeys the same general constraints as with the feature GAP: when the daughters all have an empty specification for the feature in question, so does the mother; but when one of the daughters bears a nonempty specification (e.g. the GAP specification of each verb in Figure 18 or the WH specification of the word \textit{whose} in the phrase \textit{whose suggestion}, as shown in (144)), then the mother bears the same nonempty specification.\footnote{The expression \([x, \text{ pers-fr}(x)]\) designates a parameter, described more fully in Ginzburg and Sag 2000. A parameter is essentially a pair consisting of a variable and a restriction that must hold of its values. I use \(\text{pers-fr}(x)\) as a shorthand for \(\begin{array}{c}
\text{person-fr} \\
\text{ENTITY} \ x
\end{array}\).}
FIGURE 18 An Incomplete Derivation Showing ‘Percolation’ of GAP Specifications
Nonempty specifications for WH mark the presence of an interrogative or exclamative wh-word within the filler; nonempty REL-specifications play a similar role, marking the presence of a relative wh-word (\{whose mother\} [I like _]) or the-word (\{the more\} [you read]).

Though space limitations prevent me from spelling out all the details of this analysis here (see Sag 2010a), the essentials of the analysis of wh-constructions can be sketched. The common properties of the various filler-gap clauses enumerated earlier are in part expressed in terms of the common construct type filler-head-construct (filler-head-cxt), whose instances are constrained by the following (nonmaximal) construction:

\[ \text{Filler-Head Construction (↑headed-cxt):} \]

\[
\begin{aligned}
\text{filler-head-cxt} & \Rightarrow \\
MTR & \left[ \text{SYN } X_1 ! \left[ \text{GAP } L \right] \right] \\
DTRS & \left[ \text{SYN } X_2 ! \left[ \text{WH } \right] \left[ \text{REL } \right] \right] . H \\
& \left[ \text{SEM } \left[ \text{IND } \alpha \right] \right] \\
& \left[ \text{STORE } \Sigma \right]
\end{aligned}
\]

Filler-head constructs thus require exactly two daughters: a filler and a head daughter. The construction in (145) links the INDEX and STORE values of the filler and the filler’s SYN value (except values for the features WH and REL) to the corresponding values of the first element of the head daughter’s GAP list.

\[ ^{92}\text{Here and throughout, } \Sigma \text{ variables refer to sets of feature structures.} \]
This GAP element is in turn identified with a gap within the head daughter, in the manner just illustrated in Figure 18 and in (142). Any remaining elements on the head daughter’s GAP list (members of the list \( L \)) must become part of the GAP list of the mother, which allows unbound gaps to be ‘passed up’ to a higher binder in the case of sentences with overlapping filler-gap dependencies (e.g. Problems this difficult, I never know \{[who] [to talk to \_ about \_]\}). The syntactic category of the head daughter (and hence that of its mother) is required to be verbal, which must resolve (see Figure 2 above) to one of its two subtypes, i.e. to verb or complementizer. Accordingly, the head daughter of a filler-gap construction must always be a verbal projection (S or VP) or a CP.

The theory of filler-gap constructions is articulated in terms of the construct hierarchy shown in Figure 19. Each type shown in this figure is associated with a construction that specifies the defining properties of a given class of constructs – the class of topclausal clauses (top-cl), the class of finite wh-relative clauses (f-wh-rcl), infinitival wh-relative clauses (i-wh-rcl), etc. Thus, the Filler-Head Construction enforces only those constraints that apply to all filler-gap clauses.\(^{93}\)

One of the filler-gap constructions discussed in both Ginzburg and Sag 2000 and Sag 2010a is the Nonsubject Wh-Interrogative Construction, which I formulate here as (146) (The Interrogative Clause Construction is explained below):

\(^{93}\)The types interrogative-clause (int-cl), relative-clause (rel-cl), exclamative-cl (excl-cl), and declarative-cl (decl-cl) are motivated by the existence of general properties that are characteristic of each type of clause, e.g. properties that hold of wh- and non-wh-relatives alike.
FIGURE 19 Hierarchy of Filler-Gap Clauses (after Sag 2010a)
(146) **Nonsubject Wh-Interrogative Construction**

\[ (\text{↑filler-head-ctx} \& \text{↑interrogative-cl}) : \]

\[
\begin{align*}
&\text{MTR} \left[ \text{SEM} \left[ \text{FRAMES} \left[ \begin{array}{c}
\text{PARAMS} \\
\text{PROP} \\
\{\pi, \ldots\} \\
\end{array} \right] \oplus L \right] \right] \\
&\text{DTRS} \left[ \begin{array}{c}
\text{SYN} \\
\text{CAT} \\
\text{nonverbal} \\
\{\pi\} \\
\end{array} \right] \\
&\text{SYN} \left[ \begin{array}{c}
\text{CAT} \\
\text{inv} \\
\text{X} \\
\{\pi\} \\
\end{array} \right] \\
&\text{SYN} \left[ \begin{array}{c}
\text{VAL} \\
\langle \rangle \\
\end{array} \right] \\
&\text{SEM} \left[ \begin{array}{c}
\text{LTOP} \\
\text{L} \\
\end{array} \right] \\
&\text{FRAMES} \left[ L \right]
\end{align*}
\]

in constructs defined by (146), the head daughter (and hence, this is a headed construct, the mother) will include matching specifications for the features IC and INV. This ensures (for nonsubject wh-interrogatives) that an aux-initial head daughter is possible just in case the construct is an independent clause:

\[
(147) \quad \begin{align*}
a. & \quad \{[\text{Who}] [\text{will you visit } \_] \}? \\
b. & \quad *\{[\text{Who}] [\text{you will visit } \_] \}? \\
c. & \quad \text{They don’t know } \{[\text{who}] [\text{you will visit } \_] \}.
\end{align*}
\]

Moreover, (146) permits the range of filler constituents in wh-interrogatives to be quite broad – NP, PP, AP, and AdvP fillers are all possible:

\[
(148) \quad \begin{align*}
a. & \quad \{[\text{Whose suggestion}] [\text{do you like } \_] \}?
\end{align*}
\]

Finally, (147) specifies an appropriate meaning for a wh-interrogative clause, based on the semantics developed in Ginzburg and Sag 2000. The basic components are a proposition (PROP), here determined by the LTOP, the LOCAL-TOP of the head daughter, and an associated set of parameters that must include the parameters (the \(\pi\)s in the PARAMS value) of the interrogative wh-words within the filler daughter (e.g., the parameter \([x, \text{pers-fr(x)}]\) introduced
This general approach to the semantics of interrogatives is discussed in considerable detail in Ginzburg and Sag 2000.

In Figure 20, EXPRNCR abbreviates EXPERIENCER.

The specification ‘((x, pers-fr(x)))’ means that the indicated parameter is optional, i.e. that the set indicated in (149) may be either empty or singleton.

97 ‘−’ is a ‘contained’ set difference operation that removes elements from a set nonvacuously. That is, the result of ‘−’ is defined only if the elements to be removed are members of the set in question, i.e. if \( \Sigma_1 \) is a subset of \( \Sigma_2 \) in (150). Thus \( \{x, y\} - \{y\} = \{x\} \), but \( \{x, y\} - \{z\} \) is undefined.
FIGURE 20  A Construct Licensed by the Nonsubject Wh-Interrogative Construction
within an interrogative clause; that is, relative wh-words are barred from appearing in situ. Second, according to (146), the parameter $\pi$ in the filler daughter’s WH value must be included in the mother’s $\text{PARAMS}$ set. And because of (150), this parameter must also be included in the head daughter’s $\text{STORE}$ value, but absent from the mother’s, i.e. contained in $\Sigma_0 - \Sigma_1$. That is (thinking in terms of a ‘bottom-up’ derivation), $\pi$ and possibly some other parameters are retrieved from the head daughter’s $\text{STORE}$ value and the remaining parameters are passed up, becoming the mother’s $\text{STORE}$ value. This is a general property of interrogative clauses in both the analysis presented here and those of Ginzburg and Sag 2000 and Sag 2010a, where the inheritance of stored parameters must proceed as shown in Figure 21. The resulting analysis provides an account of the famous ‘Baker ambiguities’ like (151):

(151) Who remembers where we bought what?
The analysis just sketched, as shown in Ginzburg and Sag 2000, provides a comprehensive account of the essential facts of English interrogatives, as well as a treatment of numerous facts not generally discussed in the literature.

2.10.1 What’s X Doing Y?

The analysis of *wh*-interrogatives just sketched also provides a natural home for a treatment of the ‘What’s X Doing Y?’ (WXD Y) Construction, discussed by Kay and Fillmore (1999) and illustrated in (152):

(152) a. What are they doing being so polite to Bo and Pat?
   ‘Why are they being so polite to Bo and Pat?’
   b. What is your name doing in my book? (Kay and Fillmore 1999, 3)
   c. ‘How come your name is in my book?’

The semantic fact of particular importance in Kay and Fillmore’s discussion is the unexpected causal interpretation paraphrasable in terms of *why*, *how come* or *what is the reason that*, as indicated in (152).

The essential ingredients of WXD Y, according to Kay and Fillmore are the following:

(153) a. an interrogative filler *what* participating in a *wh*-interrogative construction,
   b. a form of the copula governing *doing*,
   c. a gap associated with the object of the progressive participle of the verb *do*,
   d. a predicative XP following *doing*, forming a constituent with it,
   e. the impossibility of negation, either of *be* or of *do*,
   f. a causal interrogative semantics, and
   g. a pragmatic attribution of incongruity of the proposition whose cause is being questioned.

These points are illustrated by the following examples:

(154) a. I wonder what the salesman will say this house is doing without a kitchen. (Kay and Fillmore 1999, 3)
   b.*What does your name keep doing in my book?
   c.*What will your name (be) do in my book?
   d. What is he doing? (lacks WXD Y semantics)
   e.*What aren’t they doing being so polite to Bo and Pat?
   f. [see glosses in (152)]
   g.#What is he doing drunk, which everyone knew he would be?
Example (154a) is of particular importance, for it shows that the scope of the causal operator is not necessarily the same as the clause following the what. That is, though the position of what demarcates the top of the interrogative clause, it is the embedded structure this house is doing without a kitchen whose causality is to be explained by the salesman. (154a) does not mean ‘I wonder why it is that the salesman will say that this house lacks a kitchen’.

WXDY finds a simple analysis within SBCG. Perhaps surprisingly, this analysis is purely lexical in nature. First, we posit a be-lexeme licensed by the following listeme:

\[
\begin{align*}
\text{FORM} & \quad \langle \text{be} \rangle \\
\text{ARG-ST} & \quad \langle X, \text{[LID i-doing-fr]} \rangle \\
\text{VAL} & \quad \langle X \rangle \\
\text{SYN} & \quad \langle \text{verb} \rangle \\
\text{CAT} & \quad \text{LID i-doing-fr} \\
\text{VF} & \quad \text{prp} \\
\text{VAL} & \quad \langle X, Z \rangle \\
\text{GAP} & \quad \langle Y : \text{[STORE \{[x, \text{thing-fr}(x)]\}] \rangle \\
\text{INDEX} & \quad s \\
\text{FRAMES} & \quad \langle \text{justification-fr} \rangle \\
\text{EXPLICANS} & \quad x \\
\text{EXPLICANDUM} & \quad l \\
\end{align*}
\]

This be selects a subject \((X)\) and a VP complement whose LID is the idiomatic i-doing-frame. Like a number of other be-lexemes, this is an auxiliary verb with subject-raising properties. And because its FRAMES list is empty, it makes no contribution to the semantics.

There is just one listeme, shown in (156), whose LID is i-doing-fr, and hence only one lexeme that gives rise to words that can head the VP complement of the be in (155):

\[
\begin{align*}
\text{FORM} & \quad \langle \text{do} \rangle \\
\text{ARG-ST} & \quad \langle X, Y, Z : \text{[LTOP l]} \rangle \\
\text{SYN} & \quad \langle \text{verb} \rangle \\
\text{CAT} & \quad \text{LID i-doing-fr} \\
\text{VF} & \quad \text{prp} \\
\text{VAL} & \quad \langle X, Z \rangle \\
\text{GAP} & \quad \langle Y : \text{[STORE \{[x, \text{thing-fr}(x)]\}] \rangle \\
\text{INDEX} & \quad s \\
\text{FRAMES} & \quad \langle \text{justification-fr} \rangle \\
\text{EXPLICANS} & \quad x \\
\text{EXPLICANDUM} & \quad l \\
\end{align*}
\]

\footnote{An earlier version of this analysis was developed together with Susanne Riehemann.}
Notice that because this listeme mentions the present-participle value of \(VF\), the only \(F\)s satisfying (156) will be specified as \([VF\, prp]\). Hence (156) will give rise to no words other than those whose FORM value is \(\langle do+ing\rangle\), as desired.

The ARG-ST list of the verb in (156) contains three elements: a subject \((X)\) an object \((Y)\) and a predicational phrase \((Z)\). The object argument, however, is absent from the VAL list and present on the verb’s GAP list. Moreover, this GAP element must be specified as \([\text{index } x]\) and \([\text{store } \{[x, \text{thing-fr}]\}]\).

Since the gap’s \(\text{store}\) and index values are identified with those of the filler daughter in all Filler-Gap clauses (see (145) above), the filler daughter will also be so specified. A consequence of this is that the filler daughter must simply be \(\text{what}\), since there is no other way that its INDEX can be identical to the index of the indicated parameter. The listeme for \(\text{what}\) is shown in (157):

\[
(157) \quad \begin{array}{c}
\text{form} \\
\text{syn} \\
\text{sem}
\end{array}
\begin{array}{c}
\langle \text{what} \rangle \\
\text{noun} \\
\text{select} \\
\text{none} \\
\text{store} \\
\Sigma \\
\text{wh} \\
\Sigma: \{[x, \text{thing-fr}(x)]\} \\
\text{rel} \\
\{\} \\
\text{index} \\
\Sigma:
\end{array}
\begin{array}{c}
x \\
\text{frames} \\
\langle \rangle \\
\text{frames} \\
\langle \rangle \\
\end{array}

Finally, note that by couching the semantics in terms of a justification-frame, we may be able to predict the pragmatic incongruity effect observed by Kay and Fillmore without further stipulation. Observe that the pragmatic effect of (158a,b) seem comparable.

(158) a. What is your name doing in my book?

b. What is the justification for your name being in my book?

In sum, the WXDY Construction is analyzed in terms of a \(be\) listeme that selects for a complement whose \(\text{lid}\) is \(i\text{-doing-fr}\). Because (156) is the only listeme mentioning \(i\text{-doing-fr}\), it is the only listeme that can license the word that heads the complement of this \(be\). But since the words licensed by (156) introduce a constraint requiring the GAP value’s first member to have the \(\text{store}\) and index properties of \(\text{what}\), these words must appear in a \(\text{wh}\)-interrogative clause. Moreover, the filler daughter at the top of this interrogative clause must be the word \(\text{what}\). The listeme in (156) also links things together to produce a semantics that asks about the justification for a certain proposition, where that proposition is composed of the subject of \(do\) (= the

---

99I thank Paul Kay for this suggestion.
subject of be) and its final complement. A construct illustrating WXDY, a FS of type ns-wh-int-cl, is shown in Figure 22.

2.11 Conclusion

In this chapter, I have attempted to present and explain the basic concepts of Sign-Based Construction Grammar without excessive formalization. At the same time, I’ve taken pains to illustrate how SBCG may be applied to a number of important grammatical problems that have been addressed in the literature – both the generative transformational literature and the now extensive bodies of work in both Construction Grammar and Head-Driven Phrase Structure Grammar. I believe that the approach sketched here will ‘scale up’ to provide consistent, comprehensive linguistic descriptions. In addition, I have tried to show that many ideas developed within the Construction Grammar tradition fit naturally within SBCG. Any treatise that undertakes such a goal is by necessity programmatic in nature, and mine is no exception. But the initial results seem promising. The other chapters in this volume, I hope, will convince the reader that there is further evidence that SBCG provides a natural framework in which to develop a construction-based theory of grammar that is both descriptively and theoretically satisfying.
FIGURE 22 A Construct Illustrating the WXDY Phenomenon
Appendix: English Grammar

A1: Grammar Signature: A Partial Type Hierarchy
A2: Grammar Signature: Some Type Declarations

\[
\begin{align*}
\text{sign} : & \quad \begin{cases}
\text{PHON} & \text{phon-obj} \\
\text{FORM} & \text{morph-obj} \\
\text{SYN} & \text{syn-obj} \\
\text{SEM} & \text{sem-obj} \\
\text{CNTXT} & \text{context-obj}
\end{cases} \\
\text{lex-sign} : & \quad [\text{ARG-ST} \text{list(expression)}] \\
\text{context-obj} : & \quad \begin{cases}
\text{C-INDS} & \text{contextual-index} \\
\text{BCKGRND} & \text{list(proposition)}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{sem-obj} : & \quad \begin{cases}
\text{LTOP} & \text{label} \\
\text{IND} & \text{index} \\
\text{FRAMES} & \text{list(frame)}
\end{cases} \\
\text{syn-obj} : & \quad \begin{cases}
\text{CAT} & \text{category} \\
\text{VAL} & \text{list(expression)} \\
\text{MRKG} & \text{mark} \\
\text{GAP} & \text{list(expression)} \\
\text{WH} & \text{set(expression)} \\
\text{REL} & \text{set(expression)}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{contextual-index} : & \quad \begin{cases}
\text{SPKR} & \text{index} \\
\text{ADDR} & \text{index} \\
\text{UTT-LOC} & \text{index}
\end{cases} \\
\text{construct} : & \quad \begin{cases}
\text{MTR} & \text{sign} \\
\text{DTRS} & \text{list(sign)} \\
\text{CTX-CONTENT} & \text{list(frame)}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{lex-cxt} : & \quad [\text{DTRS} \text{list(lex-sign)}] \\
\text{infl-cxt} : & \quad [\text{MTR} \text{word} \\
& \quad \text{DTRS} \text{list(lexeme)}] \\
\text{pinfl-cxt} : & \quad \begin{cases}
\text{MTR} & \text{word} \\
\text{DTRS} & \text{list(word)}
\end{cases} \\
\text{phr-cxt} : & \quad \begin{cases}
\text{MTR} & \text{phrase} \\
\text{DTRS} & \text{list(overt-expr)}
\end{cases} \\
\text{hd-cxt} : & \quad \begin{cases}
\text{SELECT} & \text{sign-or-none} \\
\text{XARG} & \text{sign-or-none} \\
\text{LID} & \text{frame-or-none}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{verbal} : & \quad \begin{cases}
\text{VF} & \text{vform} \\
\text{IC} & \text{boolean}
\end{cases} \\
\text{verb} : & \quad \begin{cases}
\text{AUX} & \text{boolean} \\
\text{INV} & \text{boolean}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{noun} : & \quad [\text{CASE} \text{case}]
\end{align*}
\]
A3: Some Lexical-Class Constructions

Proper Noun Construction (↑invariant-lxm):

\[
\text{pn-lxm} \Rightarrow \begin{cases}
\text{FORM} & L \\
\text{SYN} & \begin{cases}
\text{CAT} & \text{SELECT} \hspace{1em} \text{none} \\
\text{XARG} & \text{none}
\end{cases} \\
\text{VAL} & \langle \rangle \\
\text{MRKG} & \text{def} \\
\text{SEM} & \begin{cases}
\text{IND} & i \\
\text{FRAMES} & \langle \rangle 
\end{cases}
\end{cases}
\]

Verb Lexeme Construction (↑lexeme):

\[
\text{verb-lxm} \Rightarrow \begin{cases}
\text{ARG-ST} & (X, \ldots) \\
\text{SYN} & \begin{cases}
\text{CAT} & \text{SELECT} \hspace{1em} \text{none} \\
\text{XARG} & X \\
\text{MRKG} & \text{unmk}
\end{cases}
\end{cases}
\]
A4: Some Combinatoric Constructions:

Principle of Compositionality:

\[
construct \Rightarrow \left[ \begin{array}{c}
MTR [\text{SEM } [\text{FRAMES } L_0 \oplus \ldots \oplus L_n]] \\
\text{DTRS } ( [\text{SEM } [\text{FRAMES } L_1]], \ldots, [\text{SEM } [\text{FRAMES } L_n]] ) \\
\text{CTX-CONTENT } L_0
\end{array} \right]
\]

Preterite Construction (\(\langle \text{infl-cxt} \rangle\)):

\[
\text{preterite-cxt } \Rightarrow \left[ \begin{array}{c}
\text{MTR} \\
\text{SEM} \\
\text{FRAMES} \\
\text{DTRS}
\end{array} \right]
\]

Un-Verb Construction (\(\langle \text{deriv-cxt} \rangle\)):
Negative Auxiliary Construction (↑post-infl-ext):

\[\text{neg-aux-ext} \Rightarrow \]

Verb-Way Construction (↑deriv-cxt):

\[\text{verb-way-cxt} \Rightarrow \]
Headed Construction (Head Feature Principle) (↑phr-cxt):

\[ \text{hd-ct} \Rightarrow \begin{bmatrix} \text{MTR} & [\text{SYN} [\text{CAT } X ]] \\ \text{HD-DTR} & [\text{SYN} [\text{CAT } X ]] \end{bmatrix} \]

Subject-Predicate Construction (↑subj-head-cxt):

\[ \text{subj-pred-cl} \Rightarrow \begin{bmatrix} \text{MTR} & [\text{SYN} Y ! [\text{VAL } \langle \rangle ]] \\ \text{DTRS} & \left( X , Z : \begin{bmatrix} \text{CAT} \quad \text{INV} \quad \text{AUX} \quad \text{VF} \quad \text{fin} \\ \text{MRKG} \quad \text{unmk} \\ \text{VAL} \quad \langle X \rangle \end{bmatrix} \right) \end{bmatrix} \]

Predicational Head-Complement Construction (↑headed-cxt):

\[ \text{pred-hd-comp-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & [\text{SYN } X ! [\text{VAL } \langle Y \rangle ]] \\ \text{DTRS} & \langle Z \rangle \oplus L : \text{nestlist} \\ \text{HD-DTR} & Z : \begin{bmatrix} \text{word} \\ \text{SYN} X : \begin{bmatrix} \text{CAT} & [\text{XARG } Y] \\ \text{VAL} & \langle Y \rangle \oplus L \end{bmatrix} \end{bmatrix} \end{bmatrix} \]

Saturational Head-Complement Construction (↑headed-cxt):

\[ \text{sat-hd-comp-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & [\text{SYN } X ! [\text{VAL } \langle \rangle ]] \\ \text{DTRS} & \langle Z \rangle \oplus L : \text{nestlist} \\ \text{HD-DTR} & Z : \begin{bmatrix} \text{word} \\ \text{SYN} X : \begin{bmatrix} \text{CAT} & [\text{prep } \text{none}] \\ \text{VAL} & L \end{bmatrix} \end{bmatrix} \end{bmatrix} \]
**Head-Functor Construction:** (↑headed-cxt):

\[ MTR \begin{bmatrix} \text{SYN } X \! [\text{MRKG } M] \end{bmatrix} \equiv \text{DTRS} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{CAT} \left[ \text{SELECT } Y \right] \end{bmatrix}, Y : \left[ \text{SYN } X \right] \end{bmatrix} \text{ HD-DTR } Y \]

**Aux-Initial Construction:** (↑headed-cxt):

\[ MTR \begin{bmatrix} \text{SYN} \left[ \text{VAL } ( \right) \end{bmatrix} \equiv \text{DTRS} \begin{bmatrix} V \begin{bmatrix} \text{CAT} \left[ \text{INV } + \right] \end{bmatrix} \end{bmatrix} \oplus \begin{bmatrix} \text{MTR} \begin{bmatrix} \text{SYN} \left[ \right. \end{bmatrix} \end{bmatrix} \]  

**Filler-Head Construction** (↑headed-cxt):

\[ MTR \begin{bmatrix} \text{SYN } X_1 \! [\text{GAP } L] \end{bmatrix} \equiv \text{DTRS} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{WH} \end{bmatrix}, H \end{bmatrix} \text{ HD-DTR } H : \begin{bmatrix} \text{SYN } X_1 : \text{GAP} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{SEM} \left[ \text{IND } \alpha \right] \end{bmatrix} \text{ STORE } \Sigma \end{bmatrix} \end{bmatrix} \oplus L \]

**Interrogative Construction** (↑core-cl):

\[ \begin{bmatrix} \text{MTR} \begin{bmatrix} \text{question} \end{bmatrix} \end{bmatrix} \equiv \text{DTRS} \begin{bmatrix} \text{SEM} \begin{bmatrix} \text{PARAMS } \Sigma_1 \end{bmatrix} \end{bmatrix} \text{ HD-DTR } \begin{bmatrix} \text{STORE } \Sigma_2 = \Sigma_1 \end{bmatrix} \]  

\[ \text{DTRS} \begin{bmatrix} \text{list} \left[ \text{REL } \{ \} \right] \end{bmatrix} \]
**Nonsubject Wh-Interrogative Construction**

(↑filler-head-ext & ↑interrogative-cl):

\[
\begin{align*}
\text{MTR} & \quad \text{SEM} \quad \text{FRAMES} \quad \text{PARAMS} \quad \text{PROP} \quad \{\pi, \ldots\} \quad \oplus \quad L \\
\text{SYN} & \quad \text{CAT} \quad \text{nonverbal} \\
\text{WH} & \quad \{\pi\} \\
\text{DTRS} & \quad \text{SYN} \quad \text{CAT} \quad \text{INV} \quad \text{IC} \quad X \\
\text{VAL} & \quad \langle \rangle \\
\text{SEM} & \quad \text{LTOP} \quad l \\
\text{FRAMES} & \quad L
\end{align*}
\]
References


Beyssade, Claire and Marandin, Jean-Marie. 2007. French Intonation and Attitude Attribution. In Pascal Denis, Eric McCready, Alexis Palmer and Brian Reese (eds.),


Mexico.
July 10, 2011

Fillmore, Charles J. and Kay, Paul. 1996. Construction Grammar Coursebook, unpub-
lished manuscript, UC Berkeley.


Fillmore, Charles J., Lee-Goldman, Russell R. and Rhodes, Russell. this volume. SBCG and the FrameNet Construction.


Harris, Zellig S. 1946. From morpheme to utterance. Language 22(3), 161–183.


Kay, Paul and Sag, Ivan A. ms. Argument Idioms and Idiom Arguments. Ms, Stanford: CSLI.
Keenan, Edward. 2002. Some Properties of Natural Language Quantifiers: General-


Sag, Ivan A. and Boas, Hans C. this volume. Introduction.


Thrúinsson, Höskuldur. 1979. On Complementation in Icelandic, Outstanding Disser-


