Sign-Based Construction Grammar

Laura A. Michaelis

1. Introduction

A foundational assumption of traditional generative grammar is that a grammar is organized in the mind of the speaker as a number of hermetically sealed modules, which in the course of a derivation hand off data one to the other.[1] Sign-Based Construction Grammar (SBCG; Sag 2010, forthcoming, Michaelis forthcoming), like the frameworks that inspire it, Berkeley Construction Grammar (BCG; Fillmore and Kay 1993, Kay and Fillmore 1999, Michaelis and Lambrecht 1996, Michaelis and Ruppenhofer 2001), and construction-based Head-Driven Phrase-Structure Grammar (Ginzburg and Sag 2001), assume no such modules, but rather that grammar is an inventory of *signs*—complexes of linguistic information that contain constraints on form, meaning and use—and that constructions are the means by which simpler signs are combined into more complex signs. The notion of construction, on this view, is a formalization, in a constraint-based

---

architecture, of the notion of construction in traditional grammar. A simple illustration of a construction is the subjectless tagged sentence shown in (1):

1. Kinda has a nice ring to it, doesn't it?

In a sentence like (1), as observed by Kay (2002), the missing subject of the main clause can only be interpreted on the basis of the reference of the tag's subject. That is, the addressee(s) of (1) must determine what *it* refers to in order to reconstruct the missing first argument of the main clause. While idiosyncratic, the biclausal construction licensing (1) shares properties with more general constructions. That is, this construction evokes signs that are licensed by other constructions, including the construction that licenses a question tag of opposite polarity to the main clause and is pronounced with rising intonation. The particular combination, arrangement and interpretation of these inherited, construction-licensed signs is, however, unique to sentences of this form: a main clause missing a subject, followed by a question tag whose pronominal subject provides the reference of the missing subject of the main clause (Kay 2002). Generalizations about constructions are captured through the interaction of a hierarchical classification of types and the type-based inheritance of grammatical constraints (Sag 2010).

The approach described here is distinct from transformational-generative grammar and its extensions, as it is a declarative, constraint-based model rather than a derivational one, and it does not assume movement, underlying structure or empty categories. In addition, SBCG associates semantic constraints and use conditions directly with the phrase-structure rules that define constructions, rather than requiring interpretations to be 'read off' syntactic representation or relegated to a 'pragmatic component'. That is,

phrase-structure rules in SBCG mean things in the same way that the words do—as a matter of linguistic convention. Like words, the phrase-structure rules of SBCG combine constraints on meaning, use and form. Finally, SBCG makes only limited use of syntactic trees, as records of the derivational histories of phrases (or, equivalently, the recursive application of syntactic rules). Trees are not objects of grammatical description in SBCG, nor are mappings between trees ('transformations'). Instead, an SBCG grammar describes the sign configurations that the grammar permits—constructions that build words from one or more lexemes and constructions that build phrases (phrasal signs) from one or more expressions. As we will discuss, these sign configurations are akin to local trees, i.e., structures consisting of a mother node and one or more daughter nodes. The SBCG program does, however, qualify as a generative grammar in the historically broader sense of aiming to provide a fully explicit account of the sentences of each language under study. This undertaking requires that the representational conventions of one's grammar cover both regular and idiomatic phrase types, as the two types interact in the licensing of sentences. A case in point is sentence (1): while the subjectless tag construction itself is idiomatic (not countenanced by the canonical English phrase-structure rules) the tag that (1) contains is presumably licensed by the same rule that licenses other tags. Thus, one can view SBCG, as well as other construction-based theories of syntax, as upholding standards of sentence description that the original proponents of generative grammar abandoned in favor of a maximally general theory (see Chomsky 1995: 435).

What is a sign in SBCG? The concept is similar to that of Saussure (1916). However, while Saussure's signs relate only form and meaning, the sign properties recognized in SBCG include phonology, (morphological) form, syntax (e.g., a word's syntactic

category and combinatoric potential), semantics (e.g., the frames that collectively define the meaning of a word, a word's referential index) and use conditions (e.g., the information-structure articulation of a phrasal type). Further, the signs of SBCG include not only words and lexemes but also phrases, including sentences; the FORM value of a phrasal sign is a list of words. Following the tradition of Unification Grammar (Schieber 1986), Generalized Phrase-Structure Grammar (Gazdar et al. 1985), Head-Driven Phrase Structure Grammar (HPSG; Pollard and Sag 1987, 1994) and allied theories, SBCG models signs as *feature structures*. Feature values include word classes like *noun*, case values like *accusative*, referential indices, lexeme classes like *intr(ansitive)-verb-lexeme*, word forms like *kick* and the binary values + and -. In this system, grammatical categories are modeled not by means of atomic symbols like V, N' or PP but rather as complexes of properties: for example, nouns specify values for the features CASE, NUMBER and GENDER and verbs specify values for inflection class (e.g., as [VFORM *finite*] or [VFORM *present-participle*]).[2] Feature structures are recursive; that is, the value of a feature may be another feature structure; for example, the value of the SYN(TAX) feature is a feature structure containing the features CAT(EGORY) and VAL(ENCE). A feature-structure value may also be a *list* of feature structures; for example, both the FRAMES feature and the VAL feature are list-valued.

All signs are modeled by feature structures, but not all feature structures model signs. Feature structures are also used in SBCG to model *constructs*. Constructs can be thought of as local trees with signs at the nodes; all constructs must be licensed by a

---

[2] Admittedly, however, traditional labels like NP and VP are often used in SBCG to abbreviate particular feature structures.

construction of the grammar (we will learn in a moment what a construction is). Formally, a construct is a feature structure containing a *mother* (MTR) feature and a *daughters* (DTRS) feature. That is, constructs are modeled as feature structures of the kind described in (2):

2. *cxt:* $\begin{bmatrix} \text{MTR } \textit{sign} \\ \text{DTRS } \textit{nelist (signs)} \end{bmatrix}$

In (2), which represents an *appropriateness declaration*, we see that a construct (abbreviated as *cxt*) must contain both a mother sign and a non-empty list of daughter signs. Intuitively speaking, a mother sign is constructed from a sequence of daughter signs. As feature structures, constructs, like signs, have *types*, notated by italic labels at the top of the brackets representing a feature structure. As in the HPSG tradition, types are organized hierarchically "[e]very feature structure of a type t must satisfy the constraints of all the supertypes of t, plus any additional constraints imposed on instances of t itself" (Sag *forthcoming*: 2). The SBCG model of constructions is in essence a theory of types. Constructions are *type constraints*:

3. *x-cxt* $\Rightarrow$ […]

Statement (3) says that if a construct is of *x* type, it must have whatever properties are represented by the three dots, in addition to the properties expressed in (2). The type constraint in (3) describes any combinatory construction: it specifies the properties that define a class of constructs, i.e., feature structures containing both the MTR and DTRS features. In particular, (3) is a *combinatory construction*: describes a particular way of

constructing a mother sign from one or more daughter signs.[3] In addition to combinatory constructions, SBCG recognizes *lexical-class constructions*. Constructions of the latter type describe classes of feature structures that correspond to words or lexemes. That is, lexical-class constructions describe classes of signs rather than classes of constructs (sign combinations). Lexical-class constructions are distinct from the *lexical entries* (lexeme descriptions) with which they combine. Both combinatory and lexical-class constructions will be discussed further in section 2.1.

Because both lexical class and constructs have types, they are subject to taxonomic organization. Sag describes (*forthcoming*: 6) describes this organization as follows:

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

As we will see, inheritance in SBCG takes over functions served by constructional inheritance in BCG. As a simple example, consider again the subjectless tagged sentence exemplified in (1). The type constraint (construction) that defines this construct type will

---

[3] It is tempting to assume that the mother signs of combinatory constructions are necessarily phrases, and that all mother signs dominate two or more daughter signs, and in fact the majority of combinatory constructions describe such construct types. However, as we will see in section 2.4, SBCG also permits unary-branching combinatory constructions, e.g., derivational constructions, in which the MTR sign is a lexeme rather than a phrase, and this MTR sign has only one daughter sign.

mention the supertype (*tagged sentence*) in its consequent clause, as the feature-structure type. This ensures that the subjectless tagged sentence will share syntactic, semantic, pragmatic and phonological (intonational) properties with its dominating type (e.g., opposite polarity of main clause and tag). Any additional properties that appear in the feature structure so typed (in particular, the covert main-clause subject) will represent properties unique to subjectless tag sentences. In this way we represent both idiosyncratic and general properties of the grammatical type. Such a type hierarchy, as we will see, enables SBCG to capture constraints are all levels of granularity.

The next section of this chapter, section 2, will focus on the formal architecture of SBCG, its historical origins and motivations, its descriptive applications and the manner in which it diverges from BCG. I will begin section 2 by addressing the historical connection between SBCG and BCG. I will then focus on specific components of its formalism: locality (section 2.1), variable-grain description (section 2.2), inheritance (section 2.3) and the use of non-branching constructions to capture inflectional and derivational processes (section 2.4). The concluding section, section 3, will address what insights SBCG offers to descriptive linguists who are not formalists.

2. The Formal Architecture of SBCG

For many years, the principle reference work available to construction grammarians working in the BCG tradition has been an unpublished (but widely circulated) course reader, Fillmore and Kay (1995). It outlines the model that we have here referred to as Berkeley Construction Grammar (BCG). Among the notable features of this work is its treatment of the interaction between verb-level argument-structure constructions (e.g., passive and ditransitive) and partially specified lexical entries (i.e., lexical entries in

which thematic roles lack grammatical-function assignments) This analysis dispenses with linking overrides of the kind that Goldberg (1995) requires to model voice alternations. Fillmore and Kay propose, for example, that the semantics of transfer associated by Goldberg (1995) with a verb valence containing containing an agentive subject and two object arguments (recipient and theme, respectively)—in other words, the active-voice ditransitive valence—is instead associated with a single linking constraint: the theme argument is realized as a *nominal oblique* (an oblique argument with direct rather than prepositional coding). This 'stripped down' ditransitive construction (referred by Fillmore and Kay as the *Nominal Oblique Theme* construction) is capable of unifying directly with either the Active construction (in which case the recipient argument is linked to the grammatical function *object*) or the Passive construction (in which case the recipient argument is linked to the grammatical function *subject*).[4] The Fillmore and Kay work also demonstrates that the mechanism of lexeme-construction unification allows one to describe English nominal and verbal syntax

---

[4] Although both Goldberg (1995) and Fillmore and Kay (1995) describe linking constraints with respect to grammatical functions like *subject* and *object* rather than case roles like *nominative* and *accusative*, it should be noted that the role played by grammatical functions in these models are served instead by case roles in SBCG, which eschews grammatical functions in favor of a ordered ARG-ST (argument-structure) list that encapsulates a ranking akin to Keenan and Comrie's (1977) Accessibility Hierarchy. This ranking plays a role in the SBCG binding constraints, which are stated as constraints on ARG-ST lists.

without recourse to the unary branching phrases and 'inaudible' determiners of 'syntactocentric', X'-based approaches. A salient example of this mode of analysis is found in the treatment of nominals. Nouns denoting unbounded substances and aggregates, i.e., mass nouns (like *butter*) and plural nouns (like *horses*), are analyzed as unmarked with regard to the binary feature MAX(IMAL); this move ensures that such nouns can either (a) unify with one of various determination constructions—all of which require a [MAX-] head daughter)—thus licensing examples like *the butter* or *some butter*, or (b) with one of various-head complement constructions (e.g., the Verb Phrase construction, the Preposition Phrase construction)—all of which require a [MAX+] complement—thus licensing phrases like *I avoid butter* and *in butter*. This analysis provides an elegant alternative to the prevalent X'-based treatment, in which certain lexical nouns are viewed as phrases (NPs) based on their external rather than internal syntax. The BCG model of nominal syntax is more intuitive than this one, in that it reserves the label 'phrase' for branching structures (constructs).

Despite its elegance and intuitive appeal, however, the BCG model also has certain dysfunctional properties: it represents constructions using a cumbersome nested-box notation and permits an undefined degree of recursion, as in the treatment of long-distance dependencies. Further, the repertoire of features that appear in construction descriptions is loosely organized and apparently open-ended. In addition, while Fillmore and Kay 1995 persuasively argue that formal and semantic commonalities among constructions can be captured by means of inheritance relations (rather than, say, transformations), the work does not provide any means of notating such taxonomic relationships, other than notations that appear sporadically in construction diagrams.

Construction grammarians seeking a more comprehensive and principled system of formal representation were inclined to look to an allied declarative model, HPSG (Pollard and Sag 1987, 1994). Like BCG, HPSG treats words and phrasal patterns as similar kinds of form-meaning pairings, uses feature structures to model semantic and syntactic classes of grammar objects and specifies a structured inventory of sign types that constitutes a multiple-inheritance network.

SBCG represents an attempt to "expand the empirical coverage of HPSG, while at the same time putting BCG on a firmer theoretical footing" (Sag *forthcoming*: 1). Like BCG, and the Cognitive Linguistics-influenced version of Construction Grammar proposed by Goldberg (1995, 2006), SBCG is a theory of constructional meaning: it assumes that rules of syntactic combination (descriptions of local trees) are directly associated with interpretive and use conditions, expressed by semantic and pragmatic features that attach to the mother or daughter nodes in these descriptions (Kay and Michaelis *forthcoming*; Sag *forthcoming*). This amounts to the claim that syntactic rules can have meanings. This claim sets Construction Grammar apart from prevailing models of meaning composition. Such theories are based on a principle that Jackendoff (1997: 48) describes as the "doctrine of syntactically transparent composition". According to this doctrine, "[a]ll elements of content in the meaning of a sentence are found in the lexical conceptual structures [...] of the lexical items composing the sentence". To propose a construction-based model of semantic composition like SBCG is not, however, to deny the existence of syntactically transparent composition. It is instead to treat it, in accordance with Jackendoff (1997: 49), as a "default in a wider array of options". That is, whenever a class of expressions can be viewed as licensed by a context-free phrase

structure rule accompanied by a rule composing the semantics of the mother from the semantics of the daughter, a construction-based approach would propose a construction that is functionally equivalent to such a rule-to-rule pair. But the constructional approach also enables us represent linguistic structures in which the semantics of the mother does not follow entirely from the semantics of the daughters, as in the case of idiomatic expressions like *throw in the towel*. In the following four subsections, we will discuss four features that make SBCG a robust formalism for construction-based syntax. The features are: (1) locality, (2) variable-grain description, (3) a model of inheritance that offers reduced stipulation and enhanced data coverage in comparison to BCG and (4) the use of unary branching constructions to capture derivational and inflectional morphology.

2.1 Locality

In SBCG, the phrase types in the target language are described by means of combinatory constructions. Combinatory constructions describe constructs—signs that are built from one or more distinct signs. Constructions in SBCG take the form of type constraints. A type constraint, as mentioned in section 1, is a conditional statement that tells us what properties a feature structure will have if it is an instance of the type in question. Intuitively, constructs are local trees (mother-daughter configurations) with feature structures (specifically, signs) at the nodes. Constructions can describe only such mother-daughter dependencies and not, e.g., mother-granddaughter dependencies (Sag *forthcoming*, 2007).

As mentioned in section 1, a local tree is distinct from a derivation tree. Derivation trees record the process of phrase construction through the recursive expansion of phrasal nodes, and can, of course, have many more than two levels. While derivation trees can be

used to describe the recursive licensing of signs in SBCG, such trees are not objects of grammatical description in SBCG. Only those trees that qualify as constructs are in the language model. As discussed in section 1, a construct is modeled in SBCG as a feature structure that contains a mother (MTR) feature and a daughters (DTRS) feature. The value of the MTR feature is a sign and the value of the DTRS feature a list of one or more signs. Let us now recall the characterization of signs from section 1. A sign, as in the Saussurean tradition, is a form-meaning pairing, but the concept has a rigorous formal implementation in SBCG. A sign is a feature structure. A sign assigns values to the following features, among others:

- PHON: a phonological phrase
- FORM: a list of the formatives (words or morphemes) that comprise the expression
- ARG-ST: a ranked list of a lexical expression's arguments (i.e., only lexical signs have this feature)
- SYN: CAT and VAL(ENCE)
- SEM: INDEX and FRAMES
- CNTXT: BACKGROUND (BCKGRND, including the set of presuppositions associated with a construction type), contextual-indices (C-INDS; identities of speaker and addressee), TOPIC and FOCUS (pragmatic roles sharing referential indices with elements on the ARG-ST list)

Several of these features deserve further discussion here. Let us begin with the features that pertain to a predicator's combinatoric potential, VAL and ARG-ST. As mentioned previously, the feature VAL lists those argument signs that an expression has yet to combine with syntactically, whether that expression is a word or a phrase. The feature VAL is closely related to the feature ARG-ST—both of these features can take an empty list of signs as their values. The VAL list of a lexeme is identical to its ARG-ST list, provided that the lexeme's arguments are locally instantiated (i.e., not extracted), and overtly expressed (see section 2.4 for a discussion of the SBCG treatment of null-instantiated arguments and Sag 2010 for a treatment of extraction). However, while only lexemes

carry the ARG-ST feature, both lexemes and phrases have VAL sets, whose members are subtracted at each mother node, as the head expression's syntactic sisters are supplied. Thus while the verbal lexeme *devour* has two NPs on its VAL list (i.e., <NP, NP>) the VP *devoured the competition* (licensed by the combinatory construction that combines heads with their complements) has a singleton VAL list, <NP>, signaling that only the subject requirement remains to be satisfied.

The semantic features IND and FRAMES require comment here as well. The value of the IND feature is the referential index of an expression, represented as a variable. This index is assigned to an individual when the expression bearing the index is a NP or to a situation when the expression bearing the index is a clause, verb or VP. The value of the FRAMES feature is the set of predications that jointly express the meaning of a sign. The term *frame* is understood as it is in Fillmore's frame semantics (Baker et al. 2003): a scene (whether dynamic or static) involving one or more participants. Each predicate in the list of frames is represented by a typed feature structure, whose features are the semantic roles required by that frame. For example, the feature structure that represents the meaning of the lexeme *eat* bears the label *eat-fr(ame)*. The features inside each feature structure of the type *frame* (e.g., EATER, EXPERIENCER, or simply ARG) take indices as their values. The major features described here are exemplified in the lexeme entry shown in figure 1:

$$
\begin{bmatrix}
lexeme \\[4pt]
\text{FORM} \ < break > \\
\text{ARG - ST} \ \langle \text{NP}i, \text{NP}x \rangle \\
\text{SYN} \begin{bmatrix} \text{CAT } verb \\ \text{VAL } \langle \text{NP}i, \text{NP}x \rangle \end{bmatrix} \\[6pt]
\text{SEM} \begin{bmatrix} \text{IND } e \\ \text{FRAMES} \left\langle \begin{bmatrix} break\text{ –}fr \\ \text{AGENT } i \\ \text{THEME } x \end{bmatrix}, \begin{bmatrix} rigid\text{ –}object\text{ –}fr \\ \text{ARG } x \end{bmatrix} \right\rangle \end{bmatrix}
\end{bmatrix}
$$

Figure 1. A lexeme entry

The lexeme represented here is the verb *break*; its ARG-ST and VAL values are identical (although we will see a case in which they diverge in section 2.4). The meaning of this verb is represented by two frame-type feature structures. The first is labeled *break-fr*. The features inside this feature structure, AGENT and THEME, represent the semantic roles assigned by the *break*-frame. The second is labeled *rigid-object-fr*. This frame captures an entailed property of the theme argument of *break*. Following the tradition of Minimal Recursion Semantics (Copestake et al. 2005), SBCG uses a 'flat' representation of the frames that jointly comprise the meaning of an expression. That is, there are no embedding relationships among the frames in the FRAMES list. The frames is this list are bound together solely by means of coreference relationships among their arguments. For example, in figure 1, the theme argument of the *break-frame* and the sole argument of the *rigid-object-frame* are coindexed.

The representation in figure 1 is of a single sign; recall, however, from above that both signs and sign combinations (constructs) are represented by feature structures in SBCG. By modeling phrase types as feature structures, SBCG captures properties common to lexemes and phrase types in a way that BCG did not. As mentioned,

according to the BCG vision, the grammar is an inventory of trees (nested boxes) of arbitrary depth. By contrast, argument-structure constructions like the Transitive construction are represented in BCG by feature structures, as in figure 2:

$$\begin{bmatrix} \text{SYN } [\text{VOICE } \textit{active}] \\ \text{VAl} \left\{ \text{REL} \begin{bmatrix} \text{GF } \textit{obj} \\ \text{DA } \text{-} \end{bmatrix} \right\} \end{bmatrix}$$

Figure 2. The Transitive construction as per BCG (Fillmore and Kay 1995)

The construction shown in figure 2 expresses a constraint on transitive lexemes: each such lexeme assigns the grammatical relation *object* to one argument in its valence set, provided that this argument is not the highest ranking or 'distinguished' argument. The Transitive construction presumably represents a class of lexemes (those that take direct objects), but it is intuitively unclear why a lexeme description like that in figure 2 should qualify as a construction, as it does not contain nested boxes. SBCG, by contrast, proposes two types of constructions: the aforementioned combinatory constructions, which describe properties of phrase types, and lexical-class constructions, which describe properties shared by classes of lexemes (like *devour*) and words (like *devoured*). The only difference between lexical-class constructions and combinatory constructions is the type name in the antecedent of the type constraint. Because both words and phrases are signs, the two can be described in the same way. This is shown by Figures 3-4, which illustrate, respectively, a lexical-class construction and a combinatory construction.

$$\text{applicative} - \text{lexeme} \Rightarrow \begin{bmatrix} \textit{trans} - \textit{verb} - \textit{lxm} \\[6pt] \text{ARG} - \text{ST} \ \langle \text{NP}_x, \ \text{NP}_z, \ \text{PP} \left[\textit{with} \right]_y \rangle \\[6pt] \text{SEM} \ | \ \text{FRAMES} \begin{bmatrix} \textit{saturation - fr} \\ \text{ACTOR} \ x \\ \text{THEME} \ y \\ \text{SURFACE} \ z \end{bmatrix} \end{bmatrix}$$

Figure 3. The Applicative lexical-class construction

$$\text{subj} - \text{pred} - \text{cxt} \Rightarrow \begin{bmatrix} \textit{phrase} \\ \text{MTR} \begin{bmatrix} \text{SYN} \left[\text{VAL} \ < \ > \right] \\ \text{MRKG} \ M : \textit{unmk} \end{bmatrix} \\ \text{DTRS} \ <X, \ H> \\ \text{HD - DTR} \ \left[\text{SYN} \begin{bmatrix} \text{CAT} \ [\text{VF} \ \textit{fin}] \\ \text{VAL} \ <X> \\ \text{MRKG} \ M : \textit{unmk} \end{bmatrix} \right] \end{bmatrix}$$

Figure 4. The Subject-Predicate combinatory construction

The Applicative construction, shown in figure 2, describes the lexeme class to which the verbs *fill* and *cover* belong, as illustrated by (4-5):

4. She filled the bathtub with champagne.

5. They covered the wall with a sheet.

The lexeme class described by the construction in figure 3 is a subtype of the transitive-lexeme class, as indicated by the typing of the feature structure to the right of the arrow. As shown by the ARG-ST list, verbs of this lexeme class express the theme argument as a PP headed by *with*. The semantic constraints associated with this lexeme class are as indicated by the frame labeled *saturation-fr* in the FRAMES list. This frame is intended to capture the resultant-state entailment that the theme occupies a critical mass of points

within a planar region (Michaelis and Ruppenhofer 2001). The Applicative construction also describes one of the classes to which the verbs *spray* and *load* belong: the lexical entries of these variable verbs lack a specified ARG-ST list, making them compatible with both the Applicative and Oblique-Goal lexical-class constructions.

The Subject-Predicate construction, shown in figure 4, licenses basic declarative clauses. As described in figure 4, a subject-predicate construct consists of two daughter signs, the second of which is a finite verbal sign that selects for the first sign by means of its VAL feature. As shown in this figure, the mother of a subject-predicate construct has an empty VAL list, indicating that it is a complete predication.

In sum, SBCG captures properties common to lexical items and phrases by describing both as feature structures. Combinatory constructions describe sign configurations (via the MTR and DTRS features), while lexical-class constructions describe single signs. But signs and sign configurations are both types of feature structures and their descriptions thus denote classes of feature structures..

The principle that governs the licensing of language objects in SBCG is the Sign Principle. According to the Sign Principle of SBCG (Sag *forthcoming*) a sign is *lexically licensed* if it satisfies (corresponds to) a lexical entry and *constructionally licensed* if it is the mother sign of some construct type.[5] This means that one can verify the

---

[5] Note that according to the Sign Principle, a *lexical* sign can be *constructionally* licensed, if it corresponds to the MTR sign of a derivational or inflectional construction (such constructions will be discussed in section 2.4). In fact, the only lexical signs that are licensed exclusively by lexical entries are those that are not 'produced' by (i.e., not the mothers of) derivational or inflectional constructions.

grammaticality of a phrase based only on the properties of its topmost (MTR) feature structure, since these properties include identifying information about that node's daughters (e.g., the frames on the MTR's FRAMES list). The analysis tree portrays graphically how the grammar licenses this sign, but the model of the phrase is just the single sign, i.e., it is the root feature structure of the analysis tree.

2.2. Variable Granularity

As is widely recognized by proponents of Construction Grammar and exemplar-based approaches (e.g., Bybee 2007), many grammatical generalizations are not very general. The ISIS construction, exemplified in (6) below, is a case in point (small caps indicate points of prosodic prominence; | indicates a pause):

6. See I—I agree with that, but my whole PROBLEM is | is that I really DON'T like Bush.

    As discussed by Brenier and Michaelis (2005), ISIS is one of several strategies that speakers use to announce a forthcoming proposition (e.g., *I really don't like Bush)* by means of a 'set up' clause (e.g., *my whole problem is*) whose subject is an informationally light noun phrase like *the main thing*, *the problem*, *the real issue*, or *the question*. Analyzed as a syntactic amalgam by Brenier and Michaelis, ISIS contains two adjacent tensed forms of the verb *be*, the first of which is typically accented. Brenier and Michaelis argue that this idiomatic pattern is designed to solve a syntax-to-phonology mapping problem: unlike the standard alternative structure (7), ISIS contains an unbroken verb phrase and an unaccented copula:

7. The thing IS | I always carry my checkbook.

But in solving a phonological problem, ISIS creates a syntactic one: the ISIS 'set up' has no identifiable syntactic category—it is more than a verb phrase but less than a full

clause. While functionally motivated, the ISIS construction features several fine-grained constraints that are inexplicable on semantic or syntactic grounds. One of these concerns possible tense combinations: while the BE1 slot may be filled by the words *was*, *is* or *being*, the BE2 slot is invariantly filled by the word *is*.[6]

8. The real question **was is** are we getting a reasonable return on our investment.

Independent-clause (IC) exclamatives provide another example of a fine-grained constraint (Sag forthcoming):

9. GOD, *(I can't believe) who they HIRED/where they WENT!

As (9) shows, IC and subordinate-clause exclamatives differ with regard to the syntactic category of the filler daughter: *who* and *where* are not possible filler daughters of IC exclamatives in English, although they are in some other languages (Michaelis 2001). It is essential to represent such category restrictions if one is to prevent one's grammar from overgenerating.

How does SBCG avoid overgeneration? It treats nodes, and in particular the MTR nodes of constructs, as feature structures—not category labels.[7] A description of a feature structure is a set of properties. As property sets, feature-structure descriptions follow the

---

[6] Brenier and Michaelis' observations are confined to the Switchboard corpus of American English telephone conversations.

[7] Certainly, X' approaches occasionally use feature structures in place of category labels, but the relevant feature inventories do not offer much descriptive power: the features are limited to syntactic ones, and the feature values are all atomic—there is no embedding of feature structures in other feature structures.

logic of set inclusion: the more properties in the description, the smaller the class of language objects that description picks out. For example, the feature set that describes an IC exclamative (e.g., *What fools!*) includes that which defines the filler-head construction, shown in figure 5. Inclusion relations among feature-structure descriptions allow us to model constructs at each step along the idiomaticity continuum, with an array of constructions of correspondingly graded generality.

$$
\textit{filler -hd -cxt} \Rightarrow \begin{bmatrix} \text{MTR} \begin{bmatrix} \text{SYN [VAL } L1] \end{bmatrix} \\ \text{DTRS } \langle [\text{SYN } X], H \rangle \\ \\ \text{HD -DTR } H : \begin{bmatrix} \textit{phrase} \\ \text{SYN} \begin{bmatrix} \text{CAT } \textit{verbal} \\ \text{VAL } L1 \end{bmatrix} \\ \text{GAP } < [\text{SYN } X] > \ \oplus \ L2 \end{bmatrix} \end{bmatrix}
$$

Figure 5. The Filler-Head construction (based on Sag forthcoming)

2.3. A Generalized Account of Inheritance

A leading insight of Construction Grammar from its inception is that grammar rules are not procedures but category descriptions, and as such, subject to taxonomic organization. Such taxonomies, which came to be known in the Construction Grammar literature as *inheritance networks* (see, e.g., Goldberg 1995), provide for cross-cutting generalizations about constructions. The idea, simply put, is that a construct can be an instance of multiple types at once. Goldberg (1995) posited two major inheritance relations: the *instance relation* and the *subpart relation*. Both relations are illustrated by the Extraposed Exclamative construction, as in (10):

10. It's amazing what she said.

Under Goldberg's conception of inheritance, the Extraposed Exclamative would be regarded as an instance of the Extraposition construction that contains as a subpart the *Wh*-Interrogative Clause construction .While inheritance networks of this nature offer a way to describe semantic and syntactic commonalities among signs and constructs without recourse to derivations, it has remained unclear just how such cross-constructional generalizations are to be represented. Should they be represented by a stipulation in a construction x, 'inherit construction y', as per Fillmore & Kay 1995? Or should they by represented by typed links in radial-category diagrams, as per Lakoff, 1987, Goldberg 1995 and Michaelis and Lambrecht 1996? Both strategies have an *ad hoc* flavor. The 'inherit' stipulation looks like a feature, but it is never made clear what types of feature structures contain it. Typed links do not appear in the feature structures used to represent constructions, so one cannot know from looking at a given construction what its 'relatives' are. In SBCG, by contrast, constraint inheritance is described by the hierarchy of types. Grammatical objects of all kinds, including phrase types, are represented by feature structures, and these feature structures belong to a taxonomy of types. One can therefore determine what constraints are shared by what construct types by consulting the descriptions of the MTR and DTR signs of constructions, where the common properties are captured by features or feature-structure types, as appropriate. Rather than being stipulated, inheritance relations are encoded in each construction's 'DNA'. To understand how this works, we must recall that a construction is a conditional statement defining the properties that are common to all instances of a given feature-structure type. The general schema for a combinatory construction is shown in (11), repeated from (3) above:

11. *x-cxt* $\implies$ […]

Thus, rather than positing instance links, SBCG allows a construction to define a construct type A, and another construction to define a subtype of A by mentioning A in its consequent clause.[8] An example is given in figure 6:

$$inv - excl - cxt \Rightarrow \begin{bmatrix} ai \text{ - } cxt \\ ... \end{bmatrix}$$

Figure 6. The Inverted Exclamative construction

The construction described in figure 6 is the Inverted Exclamative construction, exemplified by (12-13):

12. Man, was I ever wrong about that one!

13. Jesus, can that guy talk or what?

As indicated in figure 6, the MTR sign of this construction is that of an Auxiliary-Initial construct. That construct type is described by the type constraint in figure 7:

$$ai \text{ -} cxt \Rightarrow \begin{bmatrix} hd \text{ - } cxt \\ MTR \begin{bmatrix} SYN [VAL < >] \end{bmatrix} \\ DTRS \langle X_0, X_1, ... X_n \rangle \\ \\ HD \text{ -} DTR \; H \; : \; X_0 \begin{bmatrix} word \\ SYN \begin{bmatrix} CAT [INV+] \\ VAL \langle X_1, ... X_n \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 7. Auxiliary Initial construction (based on Sag forthcoming))

According to the construction in figure 7, an Auxiliary-Initial construct is a headed construct that qualifies as clause (i.e., a phrase whose head daughter is a verb and whose

---

[8] Strictly speaking, mentioning a type as an antecedent in a description does not create it. Rather, it is the type hierarchy that creates types.

valence set is empty). The head daughter of this headed construct is a verb that appears in initial position (as indicated by its place in the DTRS list) and is marked by the feature [INV+] (a distinct constraint ensures that [INV+] words are finite auxiliaries). Because the Inverted Exclamative construct described in figure 6 is of the type *Auxiliary-Initial construct*, the former will inherit all of the properties of the feature structure (a type of construct) described in figure 7, as well as additional properties represented by the ellipses in figure 6.

Having considered the treatment of 'instance links' in SBCG, let us look at the SBCG analog to Goldberg's subpart link. Because SBCG is a localist theory of syntax, as described in section 2.1, its treatment of subpart relations will necessarily diverge significantly from that found in BCG works. Constructions are configurations of signs rather than configurations of constructs; therefore, a construction cannot include another construction in its DTRS list and a construction—a description of a class of constructs—can make no reference to the daughters of a construct's daughters. Such a practice would be no more acceptable in SBCG than would placing a phrase-structure rule in the expansion of another phrase-structure rule in a context-free grammar, e.g., *VP → V (PP→ P PP). If we cannot represent one construction as containing another, how then are 'subpart' relations to be represented in SBCG? The example in (14) will serve to illustrate SBCG's approach to 'subpart' relations:

14. Never have I seen one.

Looking at the construct type illustrated in (14), Inverted Negative Adverb Preposing, we might intuitively say that it contains two daughters, the first of which is a negative adverb and the second of which is the MTR of an Auxiliary-Initial construct type. However, as

discussed above, constructions cannot contain other constructions. Instead, we would say that the head daughter of a particular construct type shares one or more features with the MTR of some other construct type. To see how this works, let us look at the Inverted Negative Adverb Preposing construction, shown in figure 8:

$$
\textit{inv-neg-adverb cxt} \Rightarrow
\begin{bmatrix}
\text{MTR} \begin{bmatrix} \text{CAT } \textit{verb} \\ \text{VAL } <\ > \end{bmatrix} \\
\text{DTRS} \left\langle \begin{bmatrix} \text{SYN } [\text{CAT } \textit{adverb}] \\ \text{SEM } [\text{FRAME } \textit{neg}]] \end{bmatrix}, H \right\rangle \\
\text{HD} - \text{DTR } H : \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{CAT } \textit{verb} \\ \text{INV } + \end{bmatrix} \\ \text{VAL} <\ > \end{bmatrix}
\end{bmatrix}
$$

Figure 8. The Inverted Negative Adverb Preposing construction

The construct type shown in figure 8 has a head-daughter sign with the property [INV+]. This feature is shared by the Auxiliary Initial construct type, shown in Figure 6: its head daughter is also [INV+]. Thus, SBCG captures daughter properties shared by constructions in a manner that is consistent with localist assumptions.

But adhering to locality is not the only reason to replace a conception of inheritance based on subpart relations with one based on head-daughter features. As I. Sag observes (p.c.), one cannot apply the subpart-based conception of inheritance to the (bracketed) head daughters in (15-17):

15. How many books [you read and I buy]!

16. Never before [have so many people attended that we ran out of room].

17. She [fortunately almost never complains].

Under a subpart-based conception of inheritance, the Interrogative Exclamative construction (e.g., *How nice it was!*) would inherit its second daughter from the Subject-Predicate construction. This analysis would not, however, cover examples like (15), in which the second daughter consists of conjoined clauses. How is the head daughter of the Interrogative Exclamative characterized in SBCG? It simply has the features [VFORM *fin*] and [INV-]. Similarly, under a BCG conception of inheritance, the Inverted Negative Adverb Preposing construction, exemplified by (14) above, would inherit its second daughter from the Auxiliary-Initial construction, depicted in figure 7. Such an analysis would not, however, work for (16), whose right daughter is licensed not by the Auxiliary-Initial construction but by the Head-Extraposition construction (Kay and Sag *forthcoming*). Examples like (16) are not problematic for analyses of the sort proposed by SBCG. These simply require that the second daughter of the Negative Adverb Preposing construction is a clause specified as [INV+]. Finally, example (17) undermines the BCG assumption that the Subject-Predicate construction inherits the Head-Complement construction as its head daughter. Because in (17) there are adverbial expressions preceding the matrix verb (*complains*), the head daughter would be licensed by a modification construction, and not the Head-Complement construction. If, however, we assume, in line with SBCG, that the Subject-Predicate construction merely constrains its second daughter to be [VFORM *fin*], (17) is unproblematic. In sum, while it may be conceptually appealing to refer to clauses like (15-17) as inheriting (or even 'containing') other constructions as their head daughters, accounting for the full array of head-daughter phrases that we actually encounter requires a feature-based formalization.

2.4. Unary Branching Constructions

SBCG makes extensive use of unary branching constructions—descriptions of constructs in which a MTR feature structure dominates a single daughter feature structure—to model inflectional and derivational processes. This might at first appear to be a step backwards, since the progenitor of SBCG, BCG, stringently avoided unary branching analyses in either nominal or verbal syntax. The BCG treatment of nominal syntax was discussed in the introduction to section 2: recall that BCG presumes both mass and plural noun lexemes to be unmarked for the MAXIMAL feature, allowing these nouns to either unify with the head daughter of a determination construction, in which case the noun assumes the value [MAX-], or combine directly with the Head-Complement construction, in which case the noun assumes the value [MAX+]. In the latter case, a mass- or plural-noun complement (e.g., *in **water***, *gather **coins***) is not considered to be an N exhaustively dominated by a NP, as in the traditional X' approach, but rather a noun that is *lexically* unmarked for maximality, and thus able to take on the feature [MAX+] when unifying with the Head-Complement construction (whose complement daughter is required to be [MAX+]). Similarly, BCG treats intransitive verb lexemes like *disappear* as [MAX+]; this move allows such verbs to unify directly with the Subject-Predicate construction, which requires its head daughter to be [MAX+].[9] This treatment thus eschews the traditional X'

---

[9] It should be noted that SBCG too avoids unary branching analyses in VP syntax, since what is called a VP in other theories is in SBCG simply a verb that has only a subject argument left on its VAL list. However, the underspecification analysis proposed within BCG for nominal syntax is not available in SBCG, which disallows unmarked feature-

analysis in which an intransitive-verb predicate (e.g., *The stain **disappeared***) is represented by a VP exhaustively dominating a V.

At the same time, certain BCG analyses do require non-branching domination. In particular, BCG employs derivational ('box over box') constructions to model nominal coercion phenomena, exemplified by (18-19):

18. Give me **a butter**. (mass → count coercion)

19. Add some tomato. (count→ mass coercion)

In the case of (18), Fillmore and Kay (1995: Ch. 4) propose a derivational construction (the Mass-to-Count construction) in which a noun lexeme type bearing a negative value for the feature BOUNDED is dominated by a noun lexeme type bearing the opposite value. This derived noun is then able to unify directly with the Indefinite Determination construction, which requires a [BOUNDED+] head daughter. Correspondingly, a Count-to-Mass construction is presumed to license the mass-noun lexeme *tomato*, which can then unify directly with the [BOUNDED-] head daughter of the Partitive Determination construction in (19). Thus, while coercion phenomena are elsewhere described as cases of semantic-conflict resolution (see, e.g., De Swart 1998), in which the construction overrides feature-values of the lexeme in the interest of verb-construction unification (Michaelis 2004), the BCG analyses of (18-19) presume no such conflict. Instead, derivational constructions generate new lexemes bearing boundedness values appropriate to the 'coercive' context, e.g., the indefinite article in (18), the partitive article *some* in

---

values. Instead, SBCG presumes a derivational construction that converts (or 'pumps') mass and count nouns to DET+ (i.e., determined) nouns.

(19).[10] Thus, the use of unary branching constructions to model derivational processes in SBCG finds a clear precedent in BCG.

As mentioned, unary branching constructions are used to model both inflectional and derivational processes in SBCG. Sag (*forthcoming*: 32) describes this situation as follows (emphasis mine):

> The properties of a word—a verb, say—are partly determined by a lexical entry (a lexeme description in the lexicon), partly by lexeme-class constructions, and **in addition by one or more of a set of derivational and inflectional constructions**. Following tradition, derivational constructions allow lexemes to be formed from other lexemes; inflectional constructions allow words to be constructed from lexemes.

As an example of a construction that licenses the 'construction' of a word from a lexeme, consider the Preterit construction (described by Sag *forthcoming*). A construct licensed by the Preterit construction is given in figure 9 (certain details of Sag's representation have been eliminated for ease of exposition):

---

[10] This analysis might be regarded as the constructional equivalent of the interpolated-operator model proposed by Jackendoff (1997: Ch. 3) within a modularist framework.
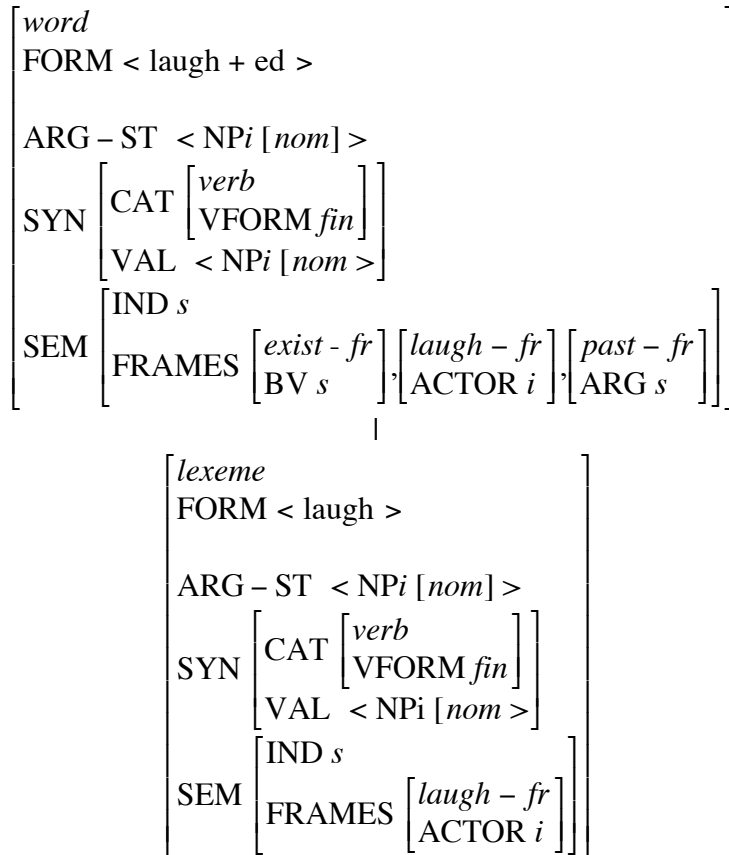
$$
\begin{bmatrix}
\textit{word} \\
\text{FORM} < \text{laugh} + \text{ed} > \\[6pt]
\text{ARG} - \text{ST} \; < \text{NP}i \, [\textit{nom}] > \\[4pt]
\text{SYN}
\begin{bmatrix}
\text{CAT}
\begin{bmatrix}
\textit{verb} \\
\text{VFORM} \, \textit{fin}
\end{bmatrix} \\
\text{VAL} \; < \text{NP}i \, [\textit{nom} >
\end{bmatrix} \\[12pt]
\text{SEM}
\begin{bmatrix}
\text{IND} \; s \\
\text{FRAMES}
\begin{bmatrix} \textit{exist - fr} \\ \text{BV} \; s \end{bmatrix} ,
\begin{bmatrix} \textit{laugh} - \textit{fr} \\ \text{ACTOR} \; i \end{bmatrix} ,
\begin{bmatrix} \textit{past} - \textit{fr} \\ \text{ARG} \; s \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

$$|$$

$$
\begin{bmatrix}
\textit{lexeme} \\
\text{FORM} < \text{laugh} > \\[6pt]
\text{ARG} - \text{ST} \; < \text{NP}i \, [\textit{nom}] > \\[4pt]
\text{SYN}
\begin{bmatrix}
\text{CAT}
\begin{bmatrix}
\textit{verb} \\
\text{VFORM} \, \textit{fin}
\end{bmatrix} \\
\text{VAL} \; < \text{NP}i \, [\textit{nom} >
\end{bmatrix} \\[12pt]
\text{SEM}
\begin{bmatrix}
\text{IND} \; s \\
\text{FRAMES}
\begin{bmatrix} \textit{laugh} - \textit{fr} \\ \text{ACTOR} \; i \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Figure 9. An inflectional construct licensed by the Preterit construction

The Preterit construction constructs a word from a lexeme, where the MTR word has the following properties: first, the verb form (VFORM) is *finite*; second, the subject valence member's CASE value is *nominative* (as required of all subjects of finite verbs); third, the FRAMES list contains, in addition to the daughter lexeme's frame (i.e., the *laugh-frame*), a 'pastness' frame and an existential quantifier frame (*exist-fr*). The bound-variable (BV) argument of the quantifier frame is identified with the situation specified in the daughter lexeme's frame, as indicated by the index s. This situation is also the sole argument of the *past-frame*.

Derivational constructions are used in SBCG to model lexeme-lexeme relations. The MTR in a derivational construct is a lexeme; this MTR has one or more lexeme DTRS. Derivational constructions are used to model not only derivational morphology

(e.g., compounding, morphemes that change the syntactic categories of words) but also valence-changing constructions. Valence-changing constructions include the Caused Motion construction, as described by Goldberg (1995: Ch. 7). Goldberg views the Caused Motion construction as a verbal lexeme type that licenses an agent, a theme and a path. She points out that the construction combines with two types of verbs: those like transitive *move*, which match the valence of the construction, as in (20), and those like *sneeze*, which do not, and therefore must undergo valence augmentation, as in (21):

20. She moved the piano into the foyer.

21. She sneezed the foam off the beer.

Rather than assuming a verb-construction valence-reconciliation mechanism in cases like (21), Sag (*forthcoming*), inspired by Kay (2005), posits a derivational construction with an effect similar to that of Goldberg's Caused Motion construction. This derivational construction is shown in figure 10.[11]

$$
caused\text{-}motion\text{-}cxt \Rightarrow
\begin{bmatrix}
deriv\text{-}cxt \\[4pt]
\text{MTR} \begin{bmatrix}
\text{FORM } <X> \\
\text{ARG-ST } \left\langle \text{NP}i, \text{NP}j, \begin{bmatrix} \text{PP} \\ \text{VAL} <\text{NP}k> \end{bmatrix} \right\rangle \\
\text{SYN } Y \\
\text{SEM} \begin{bmatrix}
\text{IND } s \\
\text{FRAMES} \left\langle \begin{bmatrix} caused\text{-}motion\text{-}fr \\ \text{EFFECTOR } i \\ \text{THEME } j \\ \text{LOCATION } k \end{bmatrix} \oplus L \right\rangle
\end{bmatrix}
\end{bmatrix} \\[8pt]
\text{DTRS} \left\langle \begin{bmatrix}
\text{FORM } <X> \\
\text{ARG-ST } \langle \text{NP}i,...\rangle \\
\text{SYN } Y \\
\text{SEM} \begin{bmatrix} \text{IND } s \\ \text{FRAMES } L \end{bmatrix}
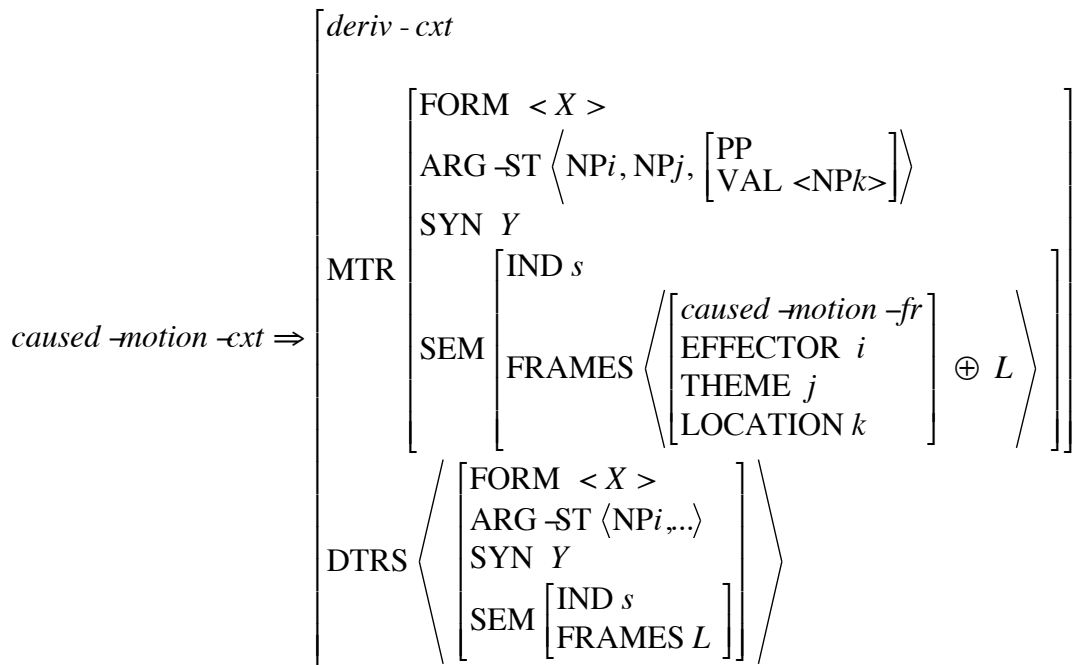\end{bmatrix} \right\rangle
\end{bmatrix}
$$

Figure 10. The Caused Motion construction

In figure 10, we see that the MTR lexeme adds the *caused-motion-frame* to the FRAMES list of the daughter lexeme, as well as augmenting the ARG-ST list of the daughter lexeme. The three members of the MTR's ARG-ST list are coindexed with the appropriate members of its FRAMES list. Notice that only one sign is required to appear on the daughter lexeme's ARG-ST list, that which represents the effector argument. This ensures that both transitive verbs like *move* and intransitive verbs like *sneeze* can unify with the

---

[11] Sag (*forthcoming*) does not provide a formal representation of the Caused Motion construction; figure 10 is based on his representation of the Verb-Way construction.

daughter-lexeme description. Thus, this construction licenses both 'valence-match' tokens like (20) and 'valence-mismatch' tokens like (21).[12]

In addition to valence-augmenting derivational constructions like the Caused Motion construction, SBCG posits valence-reducing derivational constructions, like Passive. As an additional example of a valence-reducing derivational construction, consider the Null Complementation construction described by Michaelis (*forthcoming*), based on a treatment by Kay (2004). The Null Complementation construction builds a lexeme with a covert (null-instantiated) valence member from a lexeme with an *optionally* covert valence member. A construct licensed by the Null Complementation construction is shown in figure 11.

---

[12] Note that the Caused Motion construction produces an active-voice lexeme. An additional derivational construction, Passive, is required to license passive predications like *The foam was sneezed off the beer*. As mentioned, Passive is a valence-reducing derivational construction.
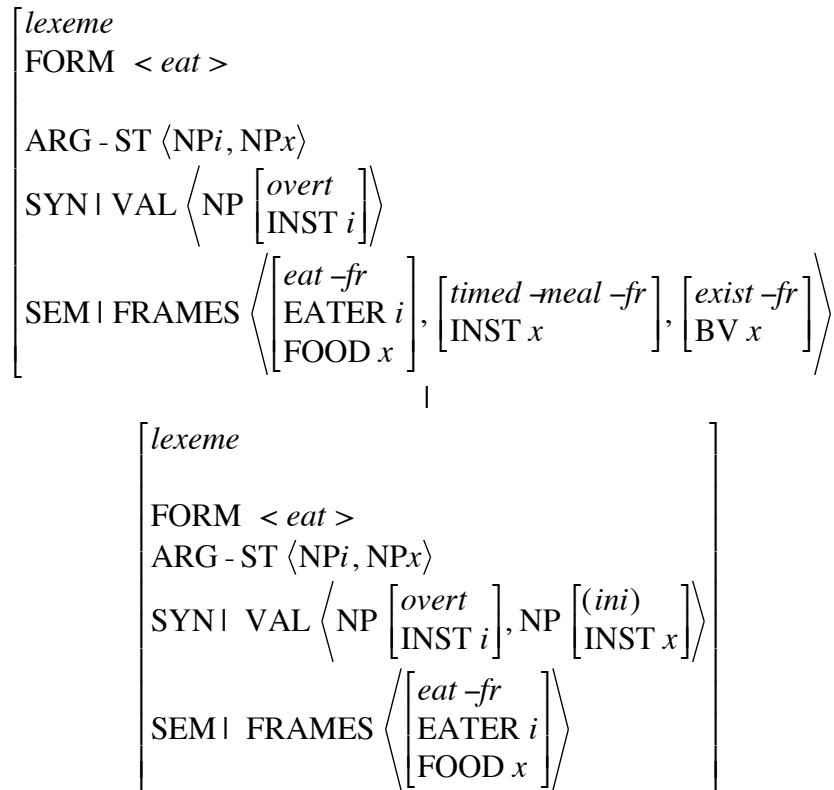
$$
\begin{bmatrix}
lexeme \\
\text{FORM} \;\; <eat> \\[6pt]
\text{ARG-ST} \;\langle \text{NP}i, \text{NP}x\rangle \\[4pt]
\text{SYN}\,|\,\text{VAL}\;\left\langle \text{NP}\begin{bmatrix} overt \\ \text{INST } i\end{bmatrix}\right\rangle \\[10pt]
\text{SEM}\,|\,\text{FRAMES}\;\left\langle \begin{bmatrix} eat\text{ --}fr \\ \text{EATER } i \\ \text{FOOD } x\end{bmatrix}, \begin{bmatrix} timed\text{ --}meal\text{ --}fr \\ \text{INST } x\end{bmatrix}, \begin{bmatrix} exist\text{ --}fr \\ \text{BV } x\end{bmatrix}\right\rangle
\end{bmatrix}
$$

$|$

$$
\begin{bmatrix}
lexeme \\[6pt]
\text{FORM} \;\; <eat> \\
\text{ARG-ST} \;\langle \text{NP}i, \text{NP}x\rangle \\[4pt]
\text{SYN}\,|\,\;\text{VAL}\;\left\langle \text{NP}\begin{bmatrix} overt \\ \text{INST } i\end{bmatrix}, \text{NP}\begin{bmatrix} (ini) \\ \text{INST } x\end{bmatrix}\right\rangle \\[10pt]
\text{SEM}\,|\,\;\text{FRAMES}\;\left\langle \begin{bmatrix} eat\text{ --}fr \\ \text{EATER } i \\ \text{FOOD } x\end{bmatrix}\right\rangle
\end{bmatrix}
$$

Figure 11. A construct licensed by the Null Complementation construction

In the construct shown in figure 11, the DTR lexeme is the verb *eat*, which licenses two arguments, an eater and a food. The sign types of its valence members are, respectively, *overt* and *(ini)*. According to the hierarchy of sign types laid out by Kay (2004), this *(ini)* sign type may resolve to either an *overt* sign or an *ini* sign (where *ini* stands for 'indefinite null instantiation'). In the construct shown in figure 11, the sign type has resolved to *ini*. The reason that we do not see this *ini* sign on the valence list of the MTR lexeme is that an *ini* sign is a type of *covert* sign; as such, it is subject to the constraint in (22):

22. *covert* $\Rightarrow$ *sign* & [FORM $<>$]

The type constraint in (22) ensures that the type *ini* has no form value. A sign with no form value will not appear on a lexeme's VAL list, since the VAL value is a list of *overt*

signs. As a subtype of the type *null-comp*, the *ini* sign type is also subject to the constraint in (23):

$$23.\ \textit{null-comp} \Rightarrow \textit{covert} \ \& \ \left[\text{SEM} \begin{bmatrix} \text{INDEX } x \\ \text{FRAMES} \left\langle \begin{bmatrix} \textit{quantifier –frame} \\ \text{BV } x \end{bmatrix} \right\rangle \end{bmatrix} \right]$$

The constraint in (23) ensures that if there is a sign of the type *null-comp*, it will have a quantifier frame on its FRAMES list. This quantifier frame takes a bound variable (BV) as its argument, and this bound variable shares its index with the covert sign. The constraint in (24) further ensures that the quantifier frame is in particular an existential-quantifier frame:

24. *ini* ⇒ *null-comp* & [SEM [FRAMES <*exist-fr*>]]

The constraint in (24) captures the existential interpretation of a missing argument in sentences like (25):

25. I've eaten ø.

Sentence (25) means something like 'I've eaten some food at a canonical meal time' rather than 'I've eaten *that* food at a canonical meal time'.[13]

Described in procedural terms, the Null Complementation construction builds an *eat* lexeme in which the food argument is missing from the verb's VAL set but remains part of its FRAMES set, where it is the bound variable of an existential quantifier. It is important to note, however, that while the mother and daughter lexemes have distinct VAL sets, they

---

[13] The 'canonical meal time' implication associated with null-complement predications like (25) is captured by the addition of the *timed-meal* frame to the FRAMES list of the MTR.

have the same ARG-ST set: the food participant appears on the ARG-ST list of the MTR lexeme even though it does not appear on the MTR lexeme's VAL list. As a result, the food argument can be the 'controller' of a secondary predicate in contexts like (26-27):

26. Her dog eats ø from a china bowl.

27. Eat ø chilled.

In both (26) and (27), the food argument is covert (i.e., missing from the verb's VAL list), but it nevertheless shares its referential index with the argument of a nonverbal predicate—the PP *from a china bowl* in (26) and the AP *chilled* in (27). This type of coindexation is referred to as *coinstantiation* in the Construction Grammar tradition. Coinstantiation lexemes are described by the type constraint in (28):

28. *coin-lexeme* $\Rightarrow$ *lexeme* & [ARG-ST $<...XP_i...$[ARG-ST $<controlee_i>$]$>$]

The lexeme *eat* can satisfy the constraint in (28) because its ARG-ST set is the same whether it has an overt or covert second argument.[14]

This brief look at the workings of the Preterit and Null Complementation constructions has shown us that constructions in SBCG are not limited to those that license constructs (combinatory constructions) and lexical classes (lexical-class constructions), but include as well those that license lexeme-word relations (inflectional constructions) and lexeme-lexeme relations (derivational constructions).

---

[14] The type constraint shown in (28) has been simplified for expository purposes. While only lexical items have the feature ARG-ST, coinstantiation constructions commonly contain *phrasal* secondary predicates. A workable version of the coindexation constraint in (28) would require the XARG feature, described by Sag (*forthcoming*).

3. Conclusion

Although SBCG cannot be divorced from the formal conventions that it uses to represent lexemes, constructions and the hierarchical relations among types, it also offers insights to Construction Grammarians whose work is not primarily formal. The leading insight of SBCG, simply put, is that the lexicon provides a model for the syntax-semantics interface. Lexical-class constructions, which define classes of lexemes or words, and combinatory constructions, which define classes of phrases, are both constraints on feature structures. In phrasal constructions, a list-valued feature of the mother is used to represent the property of having the daughters it does. Further, the constructions that embody derivational and inflectional processes are not distinct in kind from the constructions that build phrases. Thus, rather than seeing syntax, semantics and lexicon as independent modules, with the lexicon characterized as a set of idiosyncratic form-meaning associations, SBCG proposes a lexicon structured by hierarchically organized lexical classes and extends this model to relations among phrasal classes.

Taxonomic organization is not the only thing that words and constructions have in common: constructions mean what they mean in the same way that words do. Like words, constructions may invoke semantic, pragmatic and phonological conditions simultaneously. As an example of an idiomatic pattern with highly particular intonational phonology, consider the exclamatory construction that Michaelis and Lambrecht (1996) refer to as the Antitopic Exclamative. In this construction, a preclausal interjection receives prosodic prominence and the following clause receives the intonational contour of a right-dislocated phrase, as in (29-31):

29. GOD it's hot.

30. MAN I'm tired.

31. DAMN you're good.

The point here is that, as Croft and Cruse (2002: 247) put it, "[c]onstructions, like the lexical items in the lexicon, are 'vertical' structures that combine combine syntactic, semantic and even phonological information (for the specific words in a construction, as well as any unique prosodic features that may be associated with a construction". The more general point, as expressed by Culicover and Jackendoff (2005: 15) is that "[t]here is a continuum of grammatical phenomena from idiosyncratic (including words) to general rules of grammar".

References

Baker, Collin F., Charles J. Fillmore and Beau Cronin. 2003. The Structure of the Framenet Database. *International Journal of Lexicography* 16: 281-296.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Copestake, Ann, Daniel Flickinger, Carl J. Pollard and Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation* 3: 281–332.

Croft, William and D.A. Cruse. 2002. *Cognitive Grammar*. Cambridge: Cambridge University Press.

Culicover, Peter and Ray Jackendoff. 2005. *Simpler Syntax*. Oxford: Oxford University Press.

De Swart, Henriëtte. 1998. Aspect Shift and Coercion. *Natural Language and Linguistic*

Theory 16. 347-385.

Fillmore, Charles J., Paul Kay and Mary C. O'Connor. 1988. Regularity and Idiomaticity in Grammatical Constructions: The Case of Let Alone. *Language* 64: 501-538.

Fillmore, Charles J. and Paul Kay. 1995. *Construction Grammar Coursebook.* Unpublished ms., Department of Linguistics, University of California, Berkeley.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum and Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Ginzburg, Jonathan and Ivan Sag. 2000. Interrogative Investigations: The Form, Meaning and Use of English Interrogatives. Stanford, CA: CSLI Publications.

Goldberg, Adele. 1995. *Constructions: A Construction Grammar Approach to Argument Structure.* Chicago: University of Chicago Press.

Goldberg, Adele. 2006. *Constructions at Work*. Oxford: Oxford University Press.

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

Kay, Paul. 2002. English Subjectless Tagged Sentences. *Language* 78: 453-81

Kay, Paul. 2004. Null Complementation Constructions. Unpublished ms., University of California, Berkeley.

Kay, Paul. 2005. Argument-Structure Constructions and the Argument-Adjunct Distinction. In M. Fried and H. Boas, (eds.), *Grammatical Constructions: Back to the Roots*. Amsterdam: Benjamins. 71-98.

Kay, Paul and Charles J. Fillmore. 1999. Grammatical Constructions and Linguistic Generalizations: The What's X Doing Y Construction. *Language* 75: 1-33.

Kay, Paul and Laura A. Michaelis. forthcoming. Constructions and Compositionality. In C. Maienborn, K. von Heusinger and P. Portner, (eds.), *Semantics: An International Handbook of Natural Language Meaning, HSK Handbooks of Linguistics and Communication Science Series: 23: Semantics and Computer Science*. Berlin: Mouton de Gruyter.

Kay, Paul and Ivan A. Sag. forthcoming. How Hard a Problem would this be to Solve? *Proceedings of the Ninth Conference on Head-Driven Phrase Structure Grammar*.

Keenan, Edward and Bernard Comrie 1977. Noun Phrase Accessibility and Universal Grammar. *Linguistic Inquiry* 8: 63-69.

Lakoff, George. 1987. *Women, Fire and Dangerous Things: What Categories Reveal about the Mind*. Chicago: University of Chicago Press.

Michaelis, Laura A. 2001. Exclamative Constructions. In M. Haspelmath, E. König, W. Österreicher and W. Raible (eds.), *Language Universals and Language Typology: An International Handbook*. Berlin: Walter de Gruyter. 1038-1050.

Michaelis, Laura A. forthcoming. Making the Case for Construction Grammar. In H. Boas and I. Sag (eds.), *Sign-Based Construction Grammar*. Stanford, CA: CSLI Publications.

Michaelis, Laura A. 2004. Type-Shifting in Construction Grammar: A Unified Model of Aspectual Coercion. *Cognitive linguistics* 15. 1-67.

Michaelis, Laura A. and Knud Lambrecht. 1996. Toward a Construction-Based Model of Language Function: The Case of Nominal Extraposition. *Language* 72: 215-247.

Michaelis, Laura A. and Josef Ruppenhofer. 2001. *Beyond Alternations: A Constructional Account of the Applicative Pattern in German*. Stanford, CA: CSLI Publications.

Pollard, Carl and Ivan A. Sag. 1987. *Information-based Syntax and Semantics. Volume 1: Fundamentals*. Stanford, CA: CSLI Publications.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press

Ruppenhofer, Josef. 2004. *The Interaction of Valence and Information Structure*. Unpublished doctoral dissertation, Department of Linguistics, University of California, Berkeley.

Sag, Ivan A. 2007. Remarks on Locality. In S. Müller (ed.), *Proceedings of the HPSG07 Conference, Stanford University*. Stanford, CA: CSLI Publications. 394-414.

Sag, Ivan A. forthcoming. Sign-Based Construction Grammar: An Informal Synopsis. In H. Boas and I. Sag (eds.), *Sign-Based Construction Grammar*. Stanford, CA: CSLI Publications.

Sag, Ivan A. 2010. English Filler-Gap Constructions. *Language* 86.

Saussure, Ferdinand de. 1916. *Cours de Linguistique Générale*. C. Bally and A. Sechehaye and A. Riedlinger, (eds.). Lausanne and Paris: Payot.

Schieber, Stuart. 1986. *An Introduction to Unification-Based Approaches to Grammar*. Stanford: CSLI Publications.