An Informal Sketch of a Formal Architecture for Construction Grammar

PAUL KAY
Department of Linguistics, University of California at Berkeley, 1203 Dwinelle Hall, Berkeley, CA 94720, USA
E-mail: kay@cogsci.berkeley.edu

Abstract. A formal architecture for Construction Grammar (CG) is sketched. Modeling domain objects (constructs) are constituent structures with feature structures at the nodes, aka Feature Structure Trees (FTs). Constructions are partial descriptions of FTs, expressed as sets of constituent structure equations and path equations. Unification of constructions is defined. A construct c is licensed by a grammar G iff G contains a set of constructions whose unification fully describes c.

Key words: constraint-based, constructions, construction grammar, grammatical constructions

1. Introduction

The major empirical motivation for Construction Grammar (CG) is the need to develop a system of grammatical description in which the marked constructions (more or less 'idiomlike' forms of expression) are represented in the same formal system as the regular, 'core' patterns or rules (Fillmore et al., forthcoming; Kay and Fillmore, 1999).[1] Although many constructions are neither 'rules', in the sense in which a rule belongs to a component of grammar that is blind to all properties of words beyond their category and bar level, nor 'idioms', in the sense of belonging to a lexicon of syntactically opaque objects which simply adorn the leaves of phrase structure trees, they nonetheless have to be represented in the grammar in such a way as to fit seamlessly into the sentences in which they appear intermixed with the familiar 'rules' and 'lexical items', all of which are considered to be constructions.

Each of the sentences in (1)–(3) requires the grammar to recognize a construction which cuts across the components or strata of most or all existing modular and derivational theories, where by *construction* I intend a conventional association of any or all of the following kinds of grammatical information: syntactic, semantic — including 'pragmatic', lexical and phonological. Programmatic claims to the con-

---

[1] Charles Fillmore originated the approach to a construction-based grammar for which a formal architecture is sketched here. It is a pleasure to acknowledge the contributions to this paper of Ron Kaplan and Mary Dalrymple, without whose stimulation and advice it would not have been developed. Charles Fillmore, Mark Gawron, Andreas Kathol, J.-P. Koenig, and participants in a seminar at Stanford University in Spring 1997 have provided valuable suggestions. Versions of this material have been presented at Stanford University, the University of Konstanz, the Max Planck Institute in Nijmegen, and the 1998 Conference on Formal, Head-Driven, and Categorial Grammar in Saarbrücken. Errors and other shortcomings are the author's.

Grammars **5:** 1–19, 2002.

trary notwithstanding, no demonstration has been offered that such constructions can be reduced to a small set of abstract, module-internal principles — paramaterized or otherwise. Justification of the lack of such a demonstration by an appeal to 'core grammar' cannot succeed so long as no theory-independent characterization of core grammatical phenomena is offered, that is a characterization other than 'those phenomena accounted for by the theory of the core'.

(1) Fooled you, didn't she?

(2) Pour être beau gar, il est beau gar.
for to.be handsome guy he is handsome guy
'He is really a handsome guy.'

(3) a. You couldn't get a poor man, let alone a rich man, to wash your car, let alone wax your truck, for $20, let alone $10.
b. You couldn't get a poor man to wash your car for $20, let alone a rich man to wax your truck for $10.
c. You couldn't get a poor man, let alone a rich man, to wash, let alone to wax, your car, let alone your truck, for $20, let alone for $10.
d. You couldn't get a poor man to wash your car for $20 let alone (get) a rich man to wash your truck for $10. (Fillmore et al., 1988)

In (1) the sentence level construction will have to specify, among other things, how the notional subject of the main clause verb *fool* is recovered from the reference of the tag subject *she*. Sentences like (1) illustrate a different discourse phenomenon from diary style (Haegemann, 1990; Rizzi, 1994, 1997), where first person subjects are recoverable from knowledge of the genre. Sentences like (1) are also distinct from sentences like *Got milk?*, in which second person subjects can be recovered from context. *Got milk?* always means 'Have you got milk?' never 'Has he got milk?', 'Have they got milk?' etc. Contrariwise, in a sentence like (1) different referents for the understood subject of the main verb can, in a single context of utterance, be succesfully picked out by varying tag subjects such as *he* or *they*. Consequently, no version of 'truncation' (Rizzi, 1994, 1997) of higher structures is applicable to this class of sentences. The construction associated with this particular syntax (subjectless finite main clause followed by tag containing the missing subject) will also have to specify the particular discourse situation in which such sentences are usable and the special illocutionary forces sentences licensed by this construction have in such contexts.

Sentences like (2) also require a bi-partite sentential construction. The second piece is a finite clause of entirely vanilla character, but the initial piece is harder to classify. It consists of the preposition (or complementizer) *pour* 'for' followed by an infinitival VP. The semantics, however, is not that of an *in-order-to* purpose clause. The understood subject of the first clause (or PP) is construed arbitrarily — or perhaps as controlled by the subject of the second, main clause: the overall semantics is so non-compositional that it is hard to tell, the first clause or phrase apparently adding nothing to the proposition expressed by the main clause except intensification. The intensification semantics is compatible with the fact that the VP has to be interpretable in context as a scalar predicate (Jean-Pierre Koenig, personal communication). Crucially, the VP of the main clause has to be identical, modulo tense, with the VP of the initial clause, which fact also has to be represented in the formal description of the construction. So far as I know, the association of intensification semantics with a subordinate clause or PP essentially repeating the lexical material in the main (intensified) clause, that we find in this construction, is rare to non-existent elsewhere in French — not to mention UG.

The syntactic variation under identity of interpretation represented by the variants of (3) is, in its details, restricted to sentences employing the conjunction *let alone*. There are other multiple focus constructions with somewhat similar, although not identical, syntactic properties (see Fillmore et al., 1988). The syntactic variation may very roughly be summarized as follows. For each multiple focus *let alone* sentence of the form, like (3d), $...X_1X_2...X_n$ *let alone* $Y_1Y_2...Y_n$, where each pair $< X_i, Y_i >$ denotes a pair of contrastively matched foci (e.g., <rich man, poor man>), any other sentence obeying the form $...X_1...X_i$ *let alone* $Y_1...Y_i$, $X_{i+1}...X_m$ *let alone* $Y_{i+1}...Y_m$, ..., $X_j...X_n$ *let alone* $Y_j...Y_n(m < j < n)$ has the same meaning as the original. This fact has to be formally represented in the construction licensing sentences containing the conjunction *let alone*.

Similar points have recently been made with regard to a wide variety of constructions. A few suggestive examples are presented below: (See the cited references for discussion.)

(4) Big Louie sees you with the loot and he puts out a contract on you. (Culicover and Jackendoff, 1997)

(5) Bill slept the afternoon away. (Jackendoff, 1997)

(6) Frank dug his way out of the prison. (Goldberg, 1995)

(7) Tout le monde lui savait une maladie incurable.
everyone 3sg.dat knew a disease incurable
'Everyone knew he had an incurable disease.' (Koenig, 1995, see also Ruwet, 1992)

(8) What's it doing raining on my birthday. (Kay and Fillmore, 1999; Kay, 1995; Pullum, 1973)

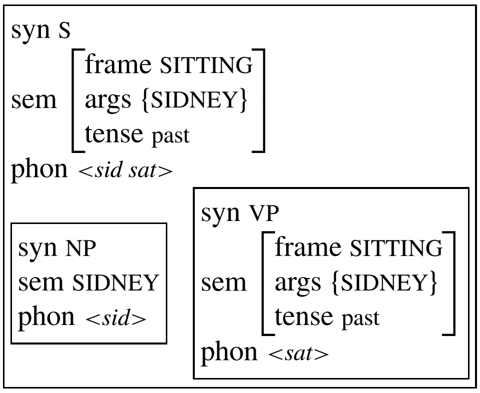
(9) Larry und Arzt!
Larry and doctor
'Larry (be) a doctor?!' (Lambrecht, 1990)

(10) It's AMAZING the people you SEE here. (Michaelis and Lambrecht, 1996)[2]

The original representation system for Construction Grammar (CG) was a system that represents constructions (items of grammar) and constructs (items of language: sentences, phrases and words) in 'box diagrams', which depict constituent structure as nested boxes, with the syntactic, semantic, lexical and phonological information associated with each constituent written in the box representing that constituent.

In the example in (11) below, a highly oversimplified representation of the sentence *Sid sat*, the outer box represents the sentence as a constituent, the small, left-daughter box indicating the subject (*Sid*) constituent and the right-daughter box representing the predicate (*sat*) constituent. The feature structure enclosed only in the larger box, at the top, says that (1) syntactically we have a sentence, (2) semantically, (a) the frame (relation) is SITTING, (b) the unique argument is SIDNEY and (c) the tense is past, and (3) the phonology of the sentence is schematically represented as <*sid sat*>. The entries in each of the smaller, daughter, boxes can be read in the same way.

(11)



The above example is of the *construct* (piece of English) *Sid sat*. The grammar of English licenses this construct by unifying or combining certain constructions. Constructions are also represented by box diagrams and the unification of the constructions licensing a construct was described as 'overlaying' or 'combining' constructions so that the parts they share match up and the parts they don't share are added to the whole.

This paper takes a first step toward specifying what kind of formal objects these box diagrams are pictures of and what is really meant by 'matching up', 'combining' or 'unifying' two box diagrams, each of which gives a partial description of a construct (bit of language), into something that gives a fuller description of this

---

[2] For arguments supporting a constructional approach which are based less on the non-modularity of individual constructions than on the web of relationships, that is partial identities, among different constructions, see Fillmore (1998), Goldberg (1995), Kathol (1997), Kathol (2000), Koenig (1994), Malouf (1996), Sag (1997).

construct, until finally we have licensed a sentential construct (i.e., a sentence). The paper concludes with a brief discussion of similarities and differences of CG with HPSG and LFG.

The empirical objects accounted for by a construction grammar — sentences, phrases and words — are taken to have the form of constituent structure (CS) trees with feature structures (FSs) at the nodes (Shieber, 1992; Andrews and Manning, 1999). A construction is defined in terms of a set of simultaneous constraint equations, describing a class of such hybrid structures. A hybrid structure which represents a fully parsed word, phrase or sentence of a language is called a construct of that language. Licensing a structure that satisfies more than one construction consists in putting together the relevant set of constructions (interpreted as a conjunction). The box notation of CG (Fillmore et al., forthcoming; Kay and Fillmore, 1999) is preserved by translating back and forth between that notation and an algebraic notation that looks much like LFG functional equations.

We first define FSs, then feature structure trees (FTs). The latter are the 'hybrid' structures: CS trees with FSs at the nodes. Modeling-domain objects (constructs) are FTs. CG modeling-domain objects are, thus, somewhat like those of LFG except that, like HPSG, phonological, semantic, categorial and 'bar' information are included in the FSs: no separate semantic and phonological modules and no labeling function, like LFG's λ, are posited. There is nothing corresponding to the up and down arrows of LFG. Also, LFG emphasizes the separateness of C-structures, F-structures and rules of correspondence between them, while CG emphasizes the unity of the construction, which, as in HPSG types (Pollard and Sag, 1987, 1994, and especially Sag, 1997), combines CS and non-CS aspects in a single unit.

Constructions are descriptions of classes of FTs, expressed as sets of constraint equations. Some of these equations express the constituent structure, as relations of dominance and precedence between nodes. Other equations are path equations which identify the CS node to which the beginning FS of each path is anchored. So we have equations that say things like "(The FS at node-1)'s syntax's head's cat is v." (Cf. Kaplan and Bresnan, 1982: 182ff; Kaplan, 1995.)

2. Primitive Notions

The primitive notions of the current exposition are attributes, atomic values, phonological representations, and (constituent structure) nodes.

3. Feature Structures

(12) Let the expression $(D \to R)$ denote the set of partial functions from domain D to range R.

We define features structures recursively as follows:

(13) A: a finite set of attributes,
V: a finite set of atomic values,
P: a denumerable set of phonological representations.

An arbitrary member f of a set F is a *feature structure* (FS) iff $F \in (A \to (V \cup P \cup F \cup \wp(F)))$.[3]

A feature structure is a partial function from a finite set of attributes to a range consisting of objects of any of the following four types: atomic values, phonological representations, feature structures, sets of feature structures.

4. Feature Structure Trees (FTs)

(14) The symbol '$<$' is used to denote the notion 'precedes'.

(15) N: a finite set of CS nodes,
F: a finite set of FSs,
M(other): a partial function in $(N \to N)$,
$<$ ('precedes'): a strict partial order on N,
ϕ: a function in $(N \to F)$.

A five-tuple $t =< N, <, F, M, \phi >$ is a *Feature Structure Tree* (FT) iff:
(i) Exactly one node $r \in N$ has no mother in N.
(ii) Every node $n \in N$ other than r has exactly one mother.
(iii) For distinct nodes $n_i, n_j \in N$, if $M(n_i) = M(n_j)$, then $(n_i < n_j)$ or $(n_j < n_i)$.
(iv) For $n_i, n_j \in N$, if $M(n_i) < M(n_j)$, then $n_i < n_j$.

FTs are the formal objects which represent constructs. Note that no variables exist in the space of FTs and all precedence relations are specified by conditions (iii) and (iv).[4]

5. Descriptions (Equations)

Let's start with a construct expressed in CG box notation and see what a description of it will look like in equation form:[5]

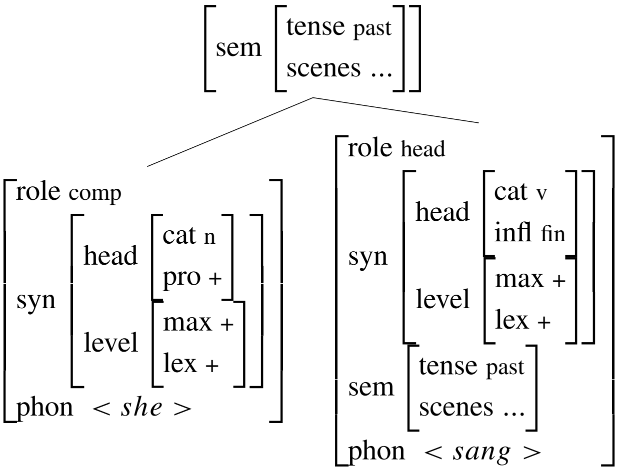
---

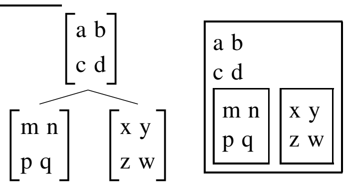
[3] '$\wp(F)$' denotes the power set of F.

[4] If n_i and n_j are not in a relation of dominance, (i-iv) assure that either $n_i < n_j$ or $n_j < n_i$. The graph structure of t is a single-rooted tree with no crossed or converging branches. So (A) if there are nodes n_a, n_b in N such that n_a is or dominates n_i, n_b is or dominates n_j, and $n_a < n_b$, then $n_i < n_j$ and (B) for any distinct pair of nodes n_i, n_j not in a dominance relation there must exist a pair of sister nodes n_a, n_b, dominating (or identical to) n_i and n_j, respectively. Since n_a, n_b are ordered by (15iii), n_i and n_j are ordered by (A).

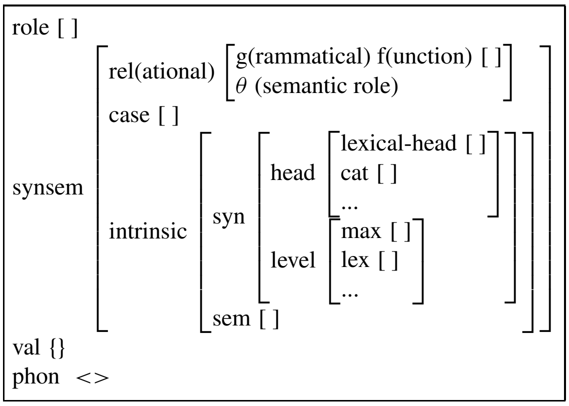
[5] CG often uses nested boxes rather than a structure of branching nodes to represent constituent structure. Thus, the two diagrams below are notational variants for a single FT. In the present paper, constructs are represented in branching node diagrams and constructions in box diagrams:

(16) *She sang.*





This paper will not be concerned with the feature geometry of any particular implementation of the CG approach. In order to simplify each example as much as possible, no attempt has been made to have all examples imply a single, consistent feature geometry. This paper is concerned only with the formal architecture in which CG grammars are expressed, not with the particulars of any one grammar or with a general theory of CG grammars. Also, to simplify some diagrams, paths are abbreviated to their last one or two attributes (e.g., diagram (31)), since the full path is in principle recoverable from knowledge of the feature geometry. One version of a CG feature geometry which seems to work fairly well for English is summed up in the following diagram.



In the CG notation used here, empty brackets, braces, etc., specify a variable of the appropriate type. For example, { } denotes an arbitrary set rather than the null set, [ ] an unspecified FS.

(This representation of the construct is very partial, but we will pretend that it is complete.) We can write a description of (certain aspects of) the construct shown in (16) in the following way. First we define the constituent structure as consisting of a mother and two daughters.

5.1. CS EQUATIONS

(17) a. $M(n_2) = n_1$[6]

b. $M(n_3) = n_1$

c. $n_2 < n_3$

5.2. FS EQUATIONS

Using the LFG type of notation for path equations, based on the idea that paths represent successive functional application, we can write, for example:

(18) $\phi(n_1)$ sem tense = past

Equation (18) can be read '(the feature structure assigned by ϕ to n_1)'s sem's tense is past.' We can now proceed to a full decription of *She sang* (to the degree that its structure is specified in (16)):

(19) a. $\phi(n_2)$ role = comp

b. $\phi(n_2)$ syn head cat = n

c. $\phi(n_2)$ syn head pro = +

d. $\phi(n_1)$ sem tense = $\phi(n_3)$ sem tense

Equation (19d) represents the kind of identity of information that is talked about in Fillmore et al., forthcoming as unification, and represented with unification variables that look like '#2'. When we come to the Subject-Predicate (S-P) construction, we will see that in the box diagram (31) this identity of information is represented by the occurrence of two instances of '#2[ ]'.

Although it is not shown in (16), we know that a fuller description of the construct *She sang* will also satisfy the following equation.

(20) $\phi(n_3)$ sem scenes $\subseteq \phi(n_1)$ sem scenes[7]

---

[6] More self-consciously: $\exists N, n_1, n_2[(n_1 \in N) \,\&\, (n_2 \in N) \,\&\, (M(n_2) = n_1)]$.

[7] The scenes value in CG is roughly analogous to the LISZT value in HPSG minimal recursion semantics (MRS). Broadly, CG semantic representation is similar in spirit and form to MRS (Copestake, Flickinger and Sag. ms., Copestake, Flickinger, Malouf, Riehemann and Sag. ms.). Substantively, (20) means roughly that the semantics of a sentence will include everything in the semantics of its verb phrase.

(Of course in this particular case, the more informative Equation (21) also holds. But (20) is interesting because it will be true of every S-P construct.)

(21) $\phi(n_3)$ sem scenes = $\phi(n_1)$ sem scenes

A few more equations descriptive of *She sang*, just to give a little more of the flavor:

(22) a. $\phi(n_2)$ phon = $< she >$

b. $\phi(n_3)$ role = head

c. $\phi(n_3)$ syn head cat = v

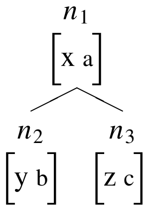
d. $\phi(n_3)$ phon = $< sang >$

We pretend Equations (17–22) fully characterize the construct *She sang*. (There are two levels of pretense here (1) the pretense that diagram (16) fully characterizes the construct *She sang* and (2) the pretense that Equations (17–22) fully characterize diagram (16).[8]) Any proper subset p of this set of equations will also describe (be satisfied by) *She sang* but will not describe *She sang* fully, in the sense that there will exist equations which are satisfied by *She sang* which are not in p (or implied by p). For example, if p is the set of equations (17–22c), p describes *She sang* only partially because (22d) is satisfied by *She sang* and is not in p (or implied by p).

6. Constructions and Licensing

In describing an FT, such as that shown in (23a), we are forced, for bookeeping purposes, to put numerical labels (or something equivalent) at the nodes, so as to be able to identify individual nodes in stating constraint equations such as those in (23b–23g), but if, for example, we had stated (23a) as (23h), yielding the equations in (23i–23n), we would have given a description of a structure which we would like to consider identical to the structure in (23a):

(23) a.



---

[8] One respect in which Equations (17–22) do not fully characterize the structure in diagram (16) is that these equations don't tell us that n_2 and n_3 are the only daughters of n_1. We could, for example, define predicates for the description language that would allow us to specify n_2 as the leftmost daughter of n_1 and n_3 as the rightmost daughter of n_1. More generally, we allow the description language to contain elements that are not part of the direct representation of modeling domain objects, e.g., negation, disjunction and material implication.

b. $\phi(n_1)x = a$

c. $\phi(n_2)y = b$

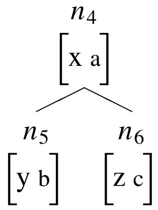
d. $\phi(n_3)z = c$

e. $M(n_2) = n_1$

f. $M(n_3) = n_1$

g. $n_2 < n_3$

h.



i. $\phi(n_4)x = a$

j. $\phi(n_5)y = b$

k. $\phi(n_6)z = c$

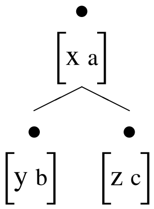
l. $M(n_5) = n_4$

m. $M(n_6) = n_4$

n. $n_5 < n_6$

Actually, in writing either (23a) or (23h), the structure we are really after is adequately represented in (24):

(24)



If we look back to the definition of FTs in (15), we note that no mention is made of particular numerical labels for nodes. Numerical node labels were introduced only to facilitate the writing of descriptive equations. They are not an essential part of the model-domain structures. At some point we will need to free our characterizations of FTs from particular assignments of identifying numerals to nodes. Let us call sets of equations of the kind exemplified in (23b–23g) and (23i–23n) ‘descriptions’.

(25) A *description* is a set of constraint equations which makes use of numerical labels to identify individual nodes.

We want to be able to say that technically distinct descriptions like those of (23b–23g) and (23i–23n) are mere ‘renumberings’ of each other.

(26) For two descriptions D, D', we will say that D' is a renumbering of D iff there is a one-one function g from node labels to node labels such that, for d the set of node labels of D and d’ the set of node labels of D’, (i) d’ is the image of d under g and (ii) there is a one-one mapping of the equations of D onto the equations of D' such that D' has the same equations as D except that each occurrence of a node label d_i in an equation of D is replaced in the corresponding equation of D' by d'_i, its image under g.

The definition in (26) will serve two purposes, one formal and one empirical. On the formal side, when we get around to combining two descriptions, call them D_1 and D_2, we will define their combination in terms of arbitrary renumberings of D_1 and D_2. This procedure will have the result that all logically possible identifications of a node in D_1 with a node in D_2 may appear in the combination (subject to whatever constraints are explicitly placed on such identifications in the definition of combination). Consequently, in combining descriptions we won’t have to bother with renumbering nodes, making sure that D_1 and D_2 have disjoint sets of node labels to start out with, or anything of that sort. The empirical consequence of definition (26) is that it allows us to eliminate node labels from that part of the formal apparatus which is intended to be psychologically plausible. Numerical node labels are used by linguists in talking to each other, but these labels are not part of what the theory attributes to speakers. Speakers are held to combine, not descriptions — which contain node labels serving only to enable the analyst to keep the nodes straight — but arbitrary renumberings of descriptions, that is, descriptions of structures like (24).

(27) A description D is *satisfiable* iff there is at least one FT t such that for each equation $d \in D$, t satisfies d.[9]

Any satisfiable description is a potential construction. The grammar of a language tells us which descriptions are the constructions of a given language. (The grammar also tells us the memberships of A, V, and (somehow) P.)

‘Combining’ constructions involves taking the union of the corresponding descriptions, subject to certain conditions to be described presently. Combining

---

[9] We have used the notion of satisfaction intuitively in the informal text and now do so in the semi-formal presentation, assuming that an appropriate model theory can be provided. The intuition is exemplified by the fact that the FT of (23a) satisfies each of the equations of (23b–23g). (Since definition (27) refers only to an arbitrary description D and does not introduce the notion of renumbering, no problem of vagueness with regard to numbering arises.)

constructions produces sets of equations of increasing cardinality, providing increasingly detailed descriptions. We think of a sufficiently detailed description as licensing a particular FT. A description's licensing (or not licensing) an FT is defined in part in terms of the FT's satisfying (or not satisfying) the description. However, the relation between a licensing description and the FT it licenses must include, but not be restricted to, satisfaction of the former by the latter. A description may, as we have noted, only partially describe an FT which satisfies it.

Intuitively we want a licensing description to be a full, not merely a partial, description of the licensed FT: the description and the model-object (FT) it licenses should specify the 'same information'. That is, the licenser should contain no information not embodied in the model-object and the model-object should embody no information not contained in the licenser.

(28) A satisfiable description D fully describes an FT t, iff:
(i) t satisfies each equation $d \in D$ and
(ii) for any description D', if t satisfies each equation $d' \in D'$, then $D \rightarrow D'$.

Property (i) ensures that D contains no information not embodied in t. Property (ii) ensures that t embodies no information not contained in D.[10] The relation defined in (28) is sometimes described by calling the model object the 'minimal satisfier' of the description; the motivation for this terminology is that the model-object t is the least detailed object which the description D describes: if you added any more information to t, then D would no longer describe t (Kaplan and Bresnan, 1982: 201–203).[11] Our preferred terminology for the relation defined in (28) will be that D 'licenses' t.

(29) For a description D and an FT t, if D fully describes t we say equivalently that D licenses t or that t is a minimal satisfier of D.

We want the grammar (A, V, P and the constructions) to provide us with a description that licenses each construct in the language and with no description which licenses any FT that is not a construct of the language. This desideratum of a construction grammar corresponds to the familiar desideratum of generative grammar: a grammar of a language should account for all the sentences of the language, provide an analysis of the structure of each, and rule out all non-sentences.[12] Put slightly differently, the object of the game is to select A, V, P and

---

[10] Suppose t embodies all the information in D plus that contained in a set of constraint equations E (where E is not implied by D). The description $D \cup E$ is then satisfied by t, violating (ii). So D does not fully describe t.

[11] This is the approach taken by Kaplan and Bresnan (1982: 201–203) to the problem dealt with in (28ii). The Kaplan-Bresnan approach is to make sure that t is the 'smallest' object that satisfies D while the approach taken in (28ii) is to make sure that D is the 'biggest' description that t satisfies.

[12] On this usage, one does not speak of an 'ambiguous sentence' but of an 'ambiguous word string, which corresponds to more than one sentence'. Nothing important turns on this terminological point. Also, a construction grammar licenses words and phrases, as well as sentences.

the constructions so that every FT which is a construct of the language is licensed by some combination of constructions and every combination of constructions that licenses an FT licenses a construct.

Having established what we want to do with combinations of descriptions, namely license constructs, we need to define combination of descriptions. A major part of the definition of combination of descriptions will involve forming unions of the sets of constraint equations that constitute the descriptions being combined. Also, in combining descriptions we will need to avail ourselves of the freedom of node identification afforded by the notion of an arbitrary renumbering of a description (recall definition 26). (For expository convenience, we define the $satisfaction\ set$ of a description D as the set of models satisfying D.)

(30) A combination C of two satisfiable descriptions D, E is a satisfiable description $D' \cup E'$ such that:
(i) D' is a renumbering of D,
(ii) E' is a renumbering of E, and
(iii) the satisfaction set of C is the intersection of the satisfaction set of D' and the satisfaction set of E'.

Certain things should be noted about definition (30). First, when the combination C fully describes an FT, D' and E' will necessarily have at least one node in common. Secondly, the definition of renumbering (26) allows us to 'match up' nodes in D and nodes in E in every possible way consistent with the dominance and precedence relations already specified in D and E.[13] Thirdly, not every pair of satisfiable descriptions is combinable. The combined description must be such that there is at least one FT that satisfies it. Moreover, there must be a way of assigning node match-ups such that condition (iii) of (30) is not violated. Fourthly, combination of descriptions is not an operation: there can be more than one combination of the same two descriptions. An example would be a case in which one description specifies something with the shape $[_S$ NP $[_{VP}$ V NP]] and the other description specifies something with the shape $[_{NP}$]. The second description can describe either the subject or object NP of the first. Actually, the isolated NP could also be neither of these; it might, for example, be the subject of a higher clause of which the $[_S$ NP $[_{VP}$ V NP]] structure is a clausal complement. But if non-overlapping descriptions are combined 'initially', they will have to be connected 'eventually' if they are to contribute to a description that licenses a construct, which is an FT. The words 'initially' and 'eventually' are put in scare quotes because licensing is declarative: a licensing description can be 'put together' by combining all the constructions that go into the 'finished' description in any order. One can likewise think of parsing a construct as decomposing it into the constructions whose combination constitutes its licensing description.

---

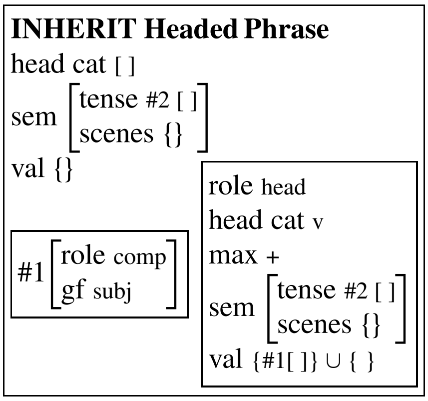
[13] To prevent the licensing of non-constructs (overgeneration) we have to be sure to set up the constructions in a way that will prevent non-constructs from being licensed. But this is just the problem that every grammarian has in making sure the hypothesized rules don't overgenerate.

6.1. EXAMPLE: THE SUBJECT-PREDICATE (S-P) CONSTRUCTION

Some of the constraints (equations) in a construction will come from more general principles, which are simply more general constructions that are inherited by more specific constructions. An important subset of these constraints are of the (rough) form: If a construction specifies a daughter node to be ‘role head’ and it inherits the more abstract construction we’re talking about, then the head daughter’s ... x is related to the mother’s ... x by the relation y (e.g., identity, subset).

In the diagram for a construction, we list at the top of the outer box in **boldface** the names of any constructions which are inherited by the one being diagrammed. For example, in the diagram of the Subject-Predicate construction in (31), it is indicated in this way that the Headed Phrase construction (35) is inherited. The notation in (31) **INHERIT Headed Phrase** does not denote an attribute-value pair. It may be thought of simply as a ‘macro’ which ‘calls’ the Headed Phrase construction to contribute all its information, i.e., all its equations, to the Subject-Predicate construction. More interestingly, the notation **INHERIT Headed Phrase** may be interpreted as expressing one link in an inheritance hierarchy of constructions which, in its entirety, represents the major grammatical generalizations of the language under study, for example, the generalization of English — and other languages — that lexical (and phrasal) heads share many syntactic and semantic properties with their phrasal mothers.

(31) Subject-Predicate (S-P) Construction (abbreviated)



Equations (32–34c) express information presented diagrammatically in Diagram (31). Equations (32) express constituent structure information about the construction. Equations (33) express path information which is peculiar to the S-P construction, specifically not inherited from the Headed Phrase (or any other construction). Equations (34a–34c) express information contained in the S-P construction that is inherited from the Headed Phrase construction, see (35).[14]

---

[14] (33a) and (33f) correspond to ‘#1’ and ‘#2’ in (31), respectively.

(32) **CS Equations**

a. $M(n_2) = n_1$

b. $n_2 < n_3$

c. $M(n_3) < n_1$

(33) **Equations proper to S-P**

a. $\phi(n_2) \in \phi(n_3)$ val

b. $\phi(n_2)$ role = comp

c. $\phi(n_2)$ ... gf = subj

d. $\phi(n_3)$ role = head

e. $\phi(n_3)$ ... cat = v

f. $\phi(n_1)$ ... sem tense = $\phi(n_3)$ ... sem tense

(34) a. $\phi(n_1)$ ... head cat = $\phi(n_3)$ ... head cat
(head feature percolation; ‘#2’ in 35)

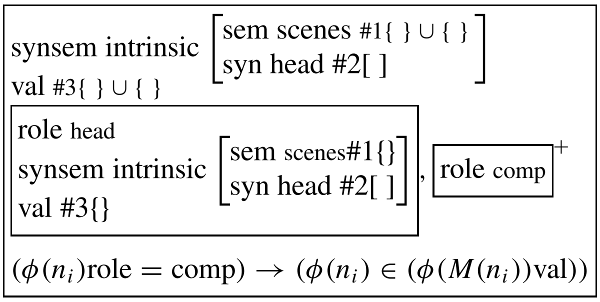
b. $\phi(n_3)$ val $\subseteq \phi(n_1)$ val
(valence of head daughter is a subset of valence of mother; ‘#3’ in 35)

c. $\phi(n_3)$ ... sem scenes $\subseteq \phi(n_1)$ ... sem scenes
(scenes of head daughter are a subset of scenes of mother; ‘#1’ in 35)

7. Constructional Inheritance

The Headed Phrase (HP) construction, which is inherited by the S-P construction, among others, follows in diagram form:

(35) Headed Phrase (HP) (abstract) construction

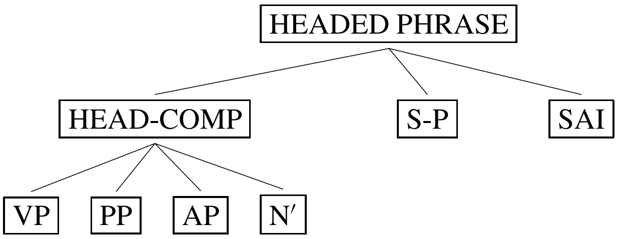


The constraint expression at the bottom of diagram (35) says that the feature structure of each comp daughter unifies with (has the same structure as) a member of the valence value of the mother. This constraint is analogous to the HPSG

Valence Principle. There are cases, such as this, where it is more convenient to express a constraint directly in algebraic form than to create for it a special convention in the box diagram mode of representation. The capability of expressing constraints on constructions in algebraic form constitutes the most important contribution of the present formulation to the practical, descriptive use of CG box notation in the representation of constructions. The box notation, without algebraic constraints, can now be seen as a visually convenient, but less than fully expressive, abbreviatory convention for the expression of a set of constituent structure and path-constraint equations. The box notation suffices for the representation of constructs, model objects, but the descriptive language allows predicates that are not conveniently expressed in box notation, material implication being perhaps the most obvious example.

All the properties of (35) are inherited by each of the more detailed constructions which license verb phrases, preposition phrases, adjective phrases, and ‘n-bar’ constituents, as well as the S-P construction, which licences subject-predicate clauses, and SAI, which licenses inverted clauses. The comma in (35) indicates that HP doesn’t stipulate linear order between head and complement(s). The HEAD-COMPLEMENT construction, which inherits HP and is in turn inherited by VP, PP, AP and N-’bar’, places the head before the complements.

(36)



(37) A construction C_1 *inherits* a distinct construction C_0 iff for some renumbering C_0' of C_0, C_0' is a subset of C_1.[15]

We distinguish a ‘compiled’ grammar, within the full grammar, which contains only the maximal constructions, that is, those constructions which have no heirs (the leaves of the inheritance hierarchy). This is the grammar that we propose is most relevant to actual speakers’ and hearers’ ‘on line’ production and interpretation of utterances. If a description D contains (implies) a construction C_1 and licenses a construct t, if there is a construction C_0 which C_1 inherits, then the description $D \cup C_0$ is equivalent to D and will also license t. But since $D \cup C_0$ is not in the compiled grammar, we don’t let this worry us.

---

[15] Consequently, C_1 inherits C_0 if the set of models satisfying C_1 is a subset of the set of models satisfying C_0'.

7.1. RECURSIVE LICENSING

Unlike LFG phrase structure rules and lexical items and unlike HPSG maximal types, distinct maximal constructions can span the same (piece of) FT. For example, the English VP construction, which provides for a lexical verb followed by an arbitrary number of constitutents (subject to valence restrictions), can unify with a construction specifically licensing a VP displaying the 'heavy NP shift' property. In order to specify an explicit recursive licensing procedure for sentences, we need some way to deal with this overlap of constructions. We wish to reduce the set of constructions of a grammar to a set of construction-like objects (let's call them CLOs) with the property that in licensing a given sentence, exactly one CLO licenses each node. To obtain the set of CLOs from the set of constructions C: (1) form the power set of the set of constructions $\wp(C)$; (2) for each set of constructions in $\wp(C)$, attempt to unify all the members, matching the root nodes; (3) throw away all the sets that don't unify; (4) the remainder is the set of CLOs.

8. CG, LFG and HPSG

The relation of constructional inheritance has an empirical interpretation analogous to that of subtype in HPSG. Often a CG analysis can be directly translated into an HPSG analysis (and vice versa) by positing an HPSG type for each CG construction and assuming isomorphism between the relevant part of the type hierarchy in HPSG and a combination of the relevant part of the constructional inheritance hierarchy together with certain on-line combinations in CG.

The formal conception of CG architecture sketched in these notes is, however, more similar to that of LFG. A separation is maintained between constituency and other syntactic properties in the modeling space. The paths which populate the description space in CG are quite similar to those of LFG. One difference between LFG and CG is the way the full set of equations characterizing a model object is factored by the grammar. In LFG the factoring is into two subsets, one a set of (annotated) PS equations (= rules), the other a set of lexical items (containing semantic forms). Various principles ('up and down arrow' computations) provide for combining these into the full set of equations licensing the construct. CG, on the other hand, designates as linguistic units small subsets of equations (the constructions); these contain both C-structure and non-C-structure information. CG resembles HPSG in this regard, in endowing the constructions (for HPSG: types) with semantic, including 'pragmatic', and phonological information, in addition to the syntactic information.

A formal difference between CG and (I think) both LFG and HPSG is that CG admits only actual words, phrases and sentences (constructs) as elements of the modeling domain. In HPSG terms, this would roughly amount to saying that the only modeling domain objects are signs. Abstractions over actual linguistic objects (words, phrases, sentences, i.e., constructs) are not considered to be part of the modeling domain in CG.

8.1. ARCHITECTURE AND THEORY

What has been sketched here is a formal architecture in which a theory of construction grammar can be expressed, not a theory of construction grammar. Within this architecture any sort of construction can be expressed; for example, we could express a construction for a transitive sentence whose subject begins with a vowel and whose object contains a relative clause specifying the color of its denotatum. A properly constrained constructional theory of grammar will have something to say about the likelihood of such a construction being encountered in any language. But that is a matter of theory, not of underlying architecture. To be sure, an architecture imposes constraints on the theories that can be expressed within it, but we should not expect a theory of grammar to follow from the notation in which it is expressed any more than we expect a theory of planetary motions to follow from the notation of differential equations.

References

Andrews A. and C. C. Manning. Information-spreading and levels of representation, CSLI Report 93–176, 1993.

Copestake A., D. Flickinger, R. Malouf, S. Riehemann and I. A. Sag. Translation using Minimal Recursion Semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation* (TMI-95), Leuven, Belgium, 1995.

Copestake A., D. Flickinger, I. A. Sag and C. J. Pollard. Minimal Recursion Semantics: An introduction. ms. Stanford: Center for the Study of Language and Information, 1999.

Culicover P. W. and R. Jackendoff. Semantic Subordination despite Syntactic Coordination, *Linguistic Inquiry*, 28(2): 195–217, 1997.

Fillmore C. J. Inversion and constructional inheritance. In G. Webelhuth, J.-P. Koenig and A. Kathol, editors, *Lexical and Constructional Aspects of Linguistic Explanation*, volume 1 of *Studies in Constraint-Based Lexicalism*, 113–128, CSLI Publications, Stanford, 1998.

Fillmore C. J., P. Kay, A. Kathol and L. Michaelis. *Construction Grammar*, CSLI Publications, Stanford, forthcoming.

Fillmore C. J., P. Kay and M. C. O'Connor. Regularity and idiomaticity in grammatical constructions: The case of *let alone*, *Language*, 64(3): 501–538, 1988.

Goldberg A. E. *Constructions*, University of Chicago Press, Chicago and London, 1995.

Haegeman L. Non-overt subjects in diary contexts. In J. Mascaró and M. Nespor, editors, *Grammar in progress*, John Benjamins, Amsterdam, 1990.

Jackendoff R. Twistin' the night away, *Language*, 73(3): 534–559, 1997.

Kaplan R. The formal architecture of Lexical Functional Grammar. In M. Dalrymple, R. Kaplan, J. Maxwell III and A. Zaenen, editors, *Formal Issues in Lexical-Functional Grammar*, 7–27, CSLI Publications, Stanford, 1995.

Kaplan R. and J. Bresnan. Lexical Functional Grammar: A formal system for grammatical representation. In *The Mental Representation of Grammatical Relations*, 173–281, MIT Press, Cambridge, 1982. Also appeared in T. M. Dalrymple et al., *Formal Issues in Lexical-Functional Grammar*, 29–130, CSLI Publications, Stanford, 1995.

Kathol A. Concrete minimalism of German. In F.-J. d'Avis and U. Lutz, editors, *Zur Satzstruktur des Deutschen*, number 90 in Arbeitsberichte des SFB 340, 81–106, SFB 340, Tübingen/Stuttgart, 1997.

Kathol A. *Linear Syntax*, Oxford University Press, Oxford, 2000.

Kay P. Construction Grammar. In J. Verschueren, J.-O. Östman and J. Blommaert, editors, *Handbook of Pragmatics*, John Benjamins, Amsterdam, 1995.

Kay P. and C. J. Fillmore. Grammatical constructions and linguistic generalizations: The *What's x doing y?* construction, *Language*, 75(1): 1–33, 1999.

Koenig J.-P. A. *Lexical underspecification and the syntax/semantics interface*. Ph. D. dissertation, University of California, Berkeley, Department of Linguistics, 1994.

Koenig J.-P. A. Mapping constructions as word-templates: Evidence from French. In C. Burgess, K. Dziwirek and D. Gerdts, editors, *Grammatical Relations: Theoretical Approaches to Empirical Questions*, 249–270, CSLI Publications, Stanford, 1995.

Lambrecht K. What, me, worry?: Mad Magazine sentences revisited. In *Proceedings of the Sixteenth Annual Meeting of the Berkeley Linguistics Society*, 215–229, 1990.

Malouf R. A constructional approach to English verbal gerunds. In *Proceedings of the Twenty-Second Annual Meeting of the Berkeley Linguistics Society*, 255–266, 1996.

Michaelis L. A. and K. Lambrecht. Toward a construction-based theory of language function: The case of nominal extraposition, *Language*, 72(2): 215–247, 1996.

Pollard C. J. and I. A. Sag. *Information-based Syntax and Semantics*, Vol. 1, CSLI Lecture Notes Series No. 13, CSLI Publications, Stanford, 1987.

Pollard C. J. and I. A. Sag. *Head-Driven Phrase Structure Grammar*, University of Chicago Press and Stanford, CSLI Publications, Chicago, 1994.

Pullum G. K. What's a sentence like this doing showing up in English? *York Papers in Linguistics*, 3: 113–115, 1973.

Rizzi L. Some notes on linguistic theory and language development: The case of root infinitives. *Language Acquisition*, 3: 371–393, 1994.

L. Rizzi. The fine structure of the left periphery. In L. Haegeman, editor, *Elements of Grammar: A Handbook of Generative Syntax*, Kluwer, Dordrecht, 1997.

Ruwet N. Le datif épistémique en français et la Condition d'opacité de Chomsky. In *Grammaire des insultes et autres études*, 172–204, Seuil, Paris, 1982.

Sag I. A. English relative clause constructions, *Journal of Linguistics*, 33(2): 431–484, 1997.

Shieber S. *Constraint-Based Grammar Formalisms*, MIT Press, Cambridge, MA, 1992.