LFG within King's descriptive formalism

Christopher Manning

In this paper, I will discuss how Lexical Functional Grammar (LFG: Bresnan, 1982a, etc.) can be modeled in King's (forthcoming) descriptive formalism. This paper isn't an introduction to LFG. It assumes a reading knowledge of LFG as can be gained from Kaplan & Bresnan (1982) or one of the more tutorial introductions, such as Sells (1985). Our plan of attack has three parts. Firstly, we will briefly try and get straight the ontology of LFG. Then we will examine a reformulation of LFG that (by and large) allows us to capture existing analyses, but brings the formulation closer to something that can be modeled in King's descriptive formalism.[1] Finally we will present a formalization of this model in the descriptive formalism of Chapter 3 of King (forthcoming).

The ontology of LFG. We need to get straight what is out there in the world and what our model objects are, what are denotations and what are descriptions that get interpreted. The title of Bresnan (1982a), *The Mental Representation of Grammatical Relations*, seems more likely to confuse us than help us. But in the introduction, there are some fairly clear statements of how their model of human use of language is to be constructed. Kaplan & Bresnan (1982, p. 173) adopt a *Competence Hypothesis* which postulates some form of grammar inside the mind of a human being:

> We assume that an explanatory model of human language performance will incorporate a theoretically justified representation of the native speaker's linguistic knowledge (a *grammar*).

Bresnan & Kaplan (1982, p. xxxi) explain how their model relates to this hypothesized grammar:

> [W]e assume that there is a *competence grammar* that represents native speakers' tacit knowledge of their language. . . .
>
> an information-processing model of language . . . includes . . . a component of stored linguistic knowledge K. . . .
>
> We call the subpart of K that prescribes representational operations the *representational basis* of the processing model. (The representational basis is the "internal grammar" of the model.) . . .
>
> a model satisfies the *strong competence hypothesis* if and only if its representational basis is isomorphic to the competence grammar.

The philosophical viewpoint is decidedly mentalist/cognitivist and whether the above isomorphism can really be expected or achieved is a moot point, but at any rate we have the elements of a standard scientific system, in which we are constructing a model of something in the world. Indeed, Bresnan & Kaplan explicitly adopt this sort of Popperian perspective (1982, p. xxxviii):

> As is true of the basic assumptions in any scientific theory, the validity of these postulates is not susceptible to direct empirical evaluation. Rather they stand at the center of a rich deductive system which has testable consequences at its empirical frontier.

[1] This formulation also has the advantage that it provides an adequate treatment of multiple projections, or 'levels of co-description', as when semantic representations are introduced into LFG. We will stick to 'Classical' LFG (Bresnan, 1982a) here, but see Andrews & Manning (1991) for an application of these ideas to semantic description.

In this model, then, what are the model objects (denotata), and what are the descriptions of them. One might initially be tempted to think that the grammar rules and accompanying functional equations (f-equations) are the descriptions, and that the constituent structure (c-structure) and functional structure (f-structure) pairs that they generate are the model objects. But f-structures are things constructed by an algorithm (the f-description solution algorithm (Kaplan & Bresnan, 1982, pp. 189–203)), and are best thought of as algebraic objects that describe an infinite set of model objects (any that specify at least the information contained in the f-structure). For if a lexical item is underspecified, say unmarked for number, then (in the absence of other agreement information) the resulting f-structure will also be unspecified for number. But a usage of this word in an actual sentence will have the speaker thinking of a concrete thing, which is either singular or plural. For example, in the sentence:

(1) I liked the salmon.

'salmon' is unspecified for number in the lexicon and so is the object NP in the resulting f-structure of the sentence. However a usage of this sentence in the real world will have the speaker (and hearer) linking this NP with something actual which will be either singular or plural (usually singular when this sentence is used at the dinner table and plural after a trip to the zoo). In general, f-structures are merely partial descriptions whereas model objects must be complete entities. Especially in the strongly typed system of King (forthcoming), it makes no sense to have strongly typed model objects that lack values for certain features. Moreover, it has become clearer from recent work on "parallel constraint grammars" that c- and f-structures are only two of multiple "levels of co-description" variously held to also include a semantic σ-structure, a thematic a-structure, and a prosodic p-structure (Bresnan, personal communication; Halvorsen & Kaplan, 1988; Inkelas & Zec, 1990). Rather than seeing model objects as a cartesian product of a fixed number of such structures, it seems as if it will be more enlightening to regard model objects as a separate domain and to see each level of co-description as something that puts constraints on which model objects are legal linguistic objects. This is the approach that we will adopt in this paper.

Reformulating standard LFG. This section presents a reformulation of the standard approach to the LFG formalism (Kaplan & Bresnan, 1982). The notations chosen there (phrase structure trees and attribute-value matrices) make c-structures and f-structures look like very different sorts of things. But Immediate Dominance (ID) trees (phrase structure trees without any linear ordering information) and feature structures are really the same types of things: they can both be modeled as directed acyclic graphs. And more generally, we can think of Linear Precedence (LP) rules, Immediate Dominance trees, and f-equations as all being descriptions of constraints on legal linguistic objects in a domain of model objects. And so we will develop these ideas in two stages. Firstly, rather than constructing f-structures by solving f-equations hung on a c-structure as in the standard LFG solution algorithm (Kaplan & Bresnan, pp. 189–203), we will show how it is possible to develop a representation (of algebraic objects) whereby f-structures can be found by deforming these objects (loosely, annotated c-structures). We will then begin to describe how, once we have this idea, there is no need to actually build and deform these objects; we can just think about everything as being constraints on legal linguistic objects. This idea is further developed in the next section.

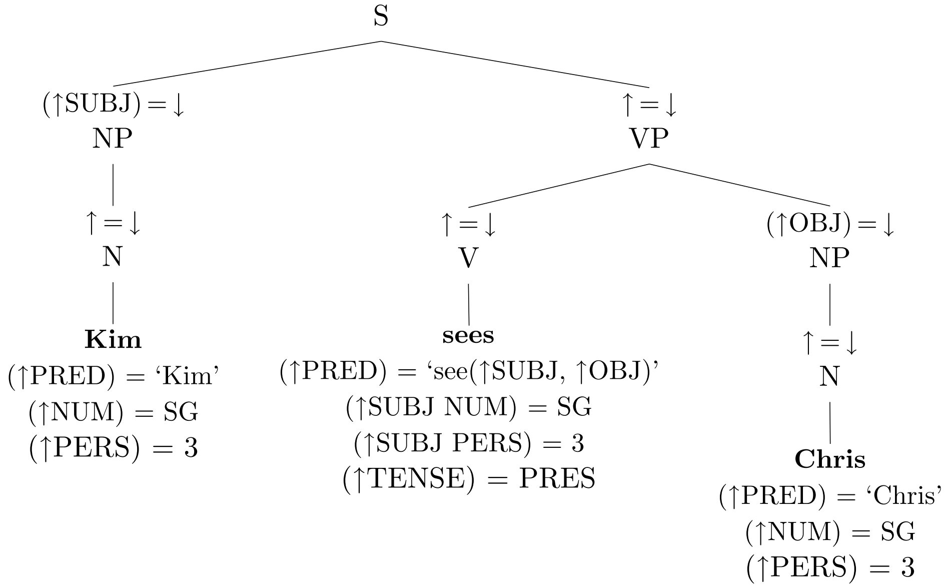
Consider the annotated c-structure (3), which we might derive for *'Kim sees Chris.'* from the toy LFG English grammar (2):

(2) a. $$\mathrm{S} \quad \rightarrow \quad \begin{array}{cc} \mathrm{NP} & \mathrm{VP} \\ (\uparrow \mathrm{SUBJ}) = \downarrow & \uparrow = \downarrow \end{array}$$

$$\begin{array}{llcc} \text{b. VP} & \rightarrow & \text{V} & \text{NP} \\ & & \uparrow\!=\!\downarrow & (\uparrow\text{OBJ})\!=\!\downarrow \end{array}$$

$$\begin{array}{llc} \text{c. NP} & \rightarrow & \text{N} \\ & & \uparrow\!=\!\downarrow \end{array}$$

(3)



Let us regard $(\uparrow\text{SUBJ})\!=\!\downarrow$ as a not inconvenient way of writing the piece of feature structure $\begin{bmatrix}\text{SUBJ} & []\end{bmatrix}$, where the inner and outer brackets are the node and its mother respectively. Then we can begin to think of the annotated phrase structure rules in (2) as a partial specification of a feature structure (or equivalently, a local subtree in a directed acyclic graph (DAG)). However, if we interpreted $\uparrow\!=\!\downarrow$ as meaning [ ], where the brackets are a node and its mother, although we could produce f-structures, we would no longer be preserving the immediate dominance system of the c-structure. So let us introduce a new feature H (for syntactic head). Then we can represent (2a), except for the LP information it encodes, as:

(4) $$\begin{bmatrix} \text{CAT} & \text{S} \\ \text{SUBJ} & \begin{bmatrix}\text{CAT} & \text{NP}\end{bmatrix} \\ \text{H} & \begin{bmatrix}\text{CAT} & \text{VP}\end{bmatrix} \end{bmatrix}$$

Moreover, with the same proviso about ignoring linear precedence information, we can represent the whole annotated c-structure (3) as a feature structure, as shown below:

$$(5)\quad \left[\begin{array}{ll} \text{CAT} & \text{S} \\ \text{SUBJ} & \left[\begin{array}{ll} \text{CAT} & \text{NP} \\ \text{H} & \left[\begin{array}{ll}\text{CAT} & \text{N}\\ \text{PRED} & \text{‘Kim’} \\ \text{NUM} & \text{SG}\\ \text{PERS} & 3\end{array}\right]\end{array}\right] \\ \text{H} & \left[\begin{array}{ll} \text{CAT} & \text{VP} \\ \text{H} & \left[\begin{array}{ll} \text{CAT} & \text{V}\\ \text{SUBJ} & \left[\begin{array}{ll}\text{NUM}&\text{SG}\\ \text{PERS}&3\end{array}\right]\\ \text{PRED} & \text{‘see}\langle\text{—, —}\rangle\text{’}\\ \text{OBJ} & \text{—} \\ \text{TENSE} & \text{PRES}\end{array}\right]\\ \text{OBJ} & \left[\begin{array}{ll}\text{CAT}&\text{NP}\\ \text{H} & \left[\begin{array}{ll}\text{CAT} & \text{N}\\ \text{PRED} & \text{‘Chris’} \\ \text{NUM} & \text{SG}\\ \text{PERS} & 3\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]$$

This assemblage of the ‘information’ (or constraints) supplied by the grammar rules and lexical items we will call an unresolved object. This algebraic object can also equivalently be represented as a DAG where the information is represented as directed labeled arcs and nodes (but it is a little bit more difficult to typeset, so its construction is left to the reader, as an exercise).

How then do we get to the desired f-structure (6) from (5)?

$$(6)\quad \left[\begin{array}{ll} \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{‘Kim’} \\ \text{NUM} & \text{SG}\\ \text{PERS} & 3\end{array}\right] \\ \text{PRED} & \text{‘see}\langle\text{—, —}\rangle\text{’} \\ \text{TENSE} & \text{PRES} \\ \text{OBJ} & \left[\begin{array}{ll}\text{PRED} & \text{‘Chris} \\ \text{NUM} & \text{SG}\\ \text{PERS} & 3\end{array}\right]\end{array}\right]$$

We do this via the notion of a projection, which we formulate as follows:[2]

[2]The notation $(e\ g) = x$ and so on represents relationships in the standard notation for f-descriptions (also known as f-equations: Kaplan & Bresnan, 1982, pp. 180–183). It says that the value of feature g in f-structure e is x.

(7) A projection is defined by:

(a) A shared set of attributes
(b) A squish set of attributes

The operation of a projection is defined in two stages as follows:

(i) A (declarative) relationship of equality of attribute values is set up as follows:
For a projection, P, with $f \in$ shared(P) and $g \in$ squish(P),
and for a feature-structure node, e, if $(e\ g) = x$ then $(e\ f) = (x\ f)$.

(ii) From the top, recursively, throw away all feature-value pairs for which the feature is not in the shared set.

The result of performing (i), above, to an algebraic object, we will call a resolved object. Sometime 'unification grammarians' can think of lexical items and c-structure rules putting various features and values in an object and the operation (i) then unifying the values of shared set attributes linked across squish set attributes. Performing (ii) for any projection will yield the information relevant to just that projection.

In our example, all attributes except CAT are f-structure attributes, and so:

squish(f) = {H}
shared(f) = {PRED, TENSE, NUM, PERS, SUBJ, OBJ, ...}

Here shared(f) contains all features except H and CAT, but if we added other c-structure features (such as perhaps BAR) or some semantic structure attributes, then we would not want these attributes to appear in the f-projection but might want them included in other projections, such as a semantic projection, σ.[3] The operation (i), above, will now duplicate the effect of the old $\uparrow = \downarrow$ equation on the f-projection. The resolved form of (5) is along the lines of (8). Features have 'spread' upwards from lexical entries in such a way that agreement and concord is being checked, just as with standard f-structures. Indeed, for a sentence to be grammatical, this resolved object must be consistent, just as one normally requires f-structures to be consistent.[4]

Applying (ii) to (8) will yield the conventional f-structure (6)—just throw away all the CATs and Hs and their values. However, this is just an aesthetic operation that produces the conventional f-structures that people are used to seeing by 'forgetting' all other information in our algebraic object. Whether we perform (ii) or not is irrelevant to whether our grammar deems a model object a legal linguistic object. All the useful work is done by (i): this part of a projection is a substantive theory that partially determines denotations in our model domain. Conversely, (ii) just provides a subset display of related information that can be helpful to a human reader. Hence it is quite sufficient to model only part (i) in our formalization of projections, below.

[3]Since at the moment CAT is the only feature not spreading along the f-projection, this notion of projections might seem unduly complicated. If we just put CAT to one side, it would be much easier to simply identify *everything* across H features. Indeed, one can produce f-structures perfectly well by this means and such a procedure was proposed in Manning (1986). However, this technique does not generalize well when one wishes to make use of multiple projections. Again, see Andrews & Manning (1991) for motivation of this more complicated notion of projection and exemplification of its use.

[4]By consistent, we mean that a normal attribute, such as CASE can only have one value. If f-equations attempt to assign it two different values, then the sentence is deemed ungrammatical.

$$
(8)\quad \left[\begin{array}{ll}
\text{CAT} & \text{S} \\
\text{SUBJ} & \left[\begin{array}{ll} \text{CAT} & \text{NP} \\ \text{H} & \left[\begin{array}{ll} \text{CAT} & \text{N}\\ \text{PRED} & \text{‘Kim’} \\ \text{NUM} & \text{SG}\\ \text{PERS} & 3\end{array}\right] \\ \text{PRED} & \text{‘Kim’}\\ \text{NUM} & \text{SG} \\ \text{PERS} & 3 \end{array}\right]\\
\text{PRED} & \text{‘see}\langle\text{—, —}\rangle\text{’} \\
\text{TENSE} & \text{PRES} \\
\text{OBJ} & \left[\begin{array}{ll} \text{CAT} & \text{NP} \\ \text{H} & \left[\begin{array}{ll} \text{CAT} & \text{N}\\ \text{PRED} & \text{‘Chris’} \\ \text{NUM} & \text{SG}\\ \text{PERS} & 3\end{array}\right] \\ \text{PRED} & \text{‘Chris’}\\ \text{NUM} & \text{SG} \\ \text{PERS} & 3 \end{array}\right]\\
\text{H} & \left[\begin{array}{ll} \text{CAT} & \text{VP} \\ \text{SUBJ} & [\ \dots\ ] \\ \text{PRED} & \dots \\ \text{TENSE} & \text{PRES} \\ \text{OBJ} & [\ \dots\ ] \\ \text{H} & [\ \dots\ ] \end{array}\right]
\end{array}\right]
$$

Once we have got this far, there is no need to view ‘resolution’ as an operation that changes one algebraic object into another. Rather we can set up our theory of grammar as a system of constraints on model objects that are English language objects (or Modern Icelandic objects, etc.). A standard f-equation annotation will say that if a piece of a model object is built using a certain phrase structure rule, then the indicated relationship must hold (for example, a particular lexical item may be of feminine gender). Our notion of projections is then just another set of constraints that all English language objects must obey. Any English language object must have the same value for f-projection shared attributes on either side of an H relationship. It is this understanding of reformulated LFG that we will formalize in the next section.

Formalizing LFG with King's (forthcoming) Descriptive Formalism. The standard formulation of LFG (Kaplan & Bresnan, 1982) is quite removed from the HPSG view of grammar (Pollard & Sag, 1987) that King (forthcoming) formalizes in his book. But this new formulation is much closer and we can attempt to apply his methods to it. Our first problem, though, is to determine a type (or variety) system for LFG. Traditionally, LFG has been loosely typed. There is a simple type system for the values of attributes (Kaplan & Bresnan, 1982, pp. 177–180). Values are of three types:

(a) Atomic Value

(b) Semantic Form

(c) f-substructure

The first is the type of the value of attributes like CASE and TENSE, the second of PRED (alone), the third of grammatical functions like OBJ and TOPIC. However, f-structures themselves are untyped and any sort of information can be 'added' as specified by f-equations. Now this is obviously too unconstrained. Intuitively the attribute CASE is totally inappropriate for the f-structure correspondent of a verb, and the attribute TENSE is inappropriate for the f-structure correspondent of a noun. But there is another problem that we will relate to this one, which is how we are going to capture the Principles of Completeness and Coherence (Kaplan & Bresnan, 1982, p. 211–214) in King's formalism. The most apparent way to do this is to build these principles into the variety system and signature. I will explain how to do this momentarily, but since the variety system will thus be constructed out of the attribute system, let us first look at our set of attributes.

Assume a finite set Θ of theta roles. For concreteness, we might take:

$$\Theta = \{\text{Agent, Beneficiary, Recipient, Experiencer, Instrument, Theme, Patient, Locative}\}$$

Then we will construct our attribute set, $\mathcal{A}$, thus:

$$\begin{array}{rcl}
\Xi & = & \{\text{SUBJ, OBJ, COMP, XCOMP}\} \;\cup \\
 & & \{\text{OBJ}_\theta \;\mid\; \theta \in \Theta\} \;\cup \\
 & & \{\text{OBL}_\theta \;\mid\; \theta \in \Theta\} \;\cup \\
\mathcal{T} & = & \{\text{CASE, TENSE, ASPECT, PERS, FORM, NUM}, \ldots\} \\
\mathcal{A} & = & \Xi \cup \mathcal{T} \;\cup\; \{\text{H, ADJ, XADJ, PRED}\}
\end{array}$$

I'm not actually going to come up with a definitive list of *atomic-valued features* (CASE, TENSE, etc.), and nor am I going to incorporate a complete variety system for them into my signature, but it should be very obvious to see how this would be done. We will just allow all of these atomic-valued features in any f-structure. Further, in our signature, we will allow any 'atomic value' (all atomic values are actually modeled as varieties in King's formalism) as the value of any atomic-valued feature, whereas in reality, clearly, 1, 2 and 3 are appropriate values for PERS; SG and PL for NUM etc.

Modeling Completeness and Coherence. Rather than explicitly exhausting SUBCAT lists and relying on semantic constraints, as in HPSG, LFG relies on the principles of Completeness and Coherence (Kaplan & Bresnan, 1982, pp. 211–214) to handle the subcategorization of arguments by verbs (and other predicators, such as prepositions). These principles rely on a distinction between something 'being there' and 'not being there'. For example, the lexical entry of a verb might require that the f-structure that it becomes part of contains a SUBJ and an OBJ, each of which contains a PRED. Since King's system is strongly typed (everything of a certain type must have or lack any attribute; see King, Chapter 3), the most straightforward way to model these principles is to build them into the variety system (without changing the formalism, the only other way seems to be to add extra bogus attributes such as THEMATIC: {YES, NO}). So, the model object equivalents of f-structures that require different grammatical functions (GFs) to satisfy Completeness or Coherence must be of a different type. We must build up a variety system that allows us to do this. Part of this involves us also being able to distinguish between thematic and nonthematic GFs (Bresnan, 1982b, p. 289: most nouns are thematic and are specified for a PRED feature, but the few nonthematic nouns (*it* and *there* in English) lack a PRED feature, and can thus appear in nonthematic contexts like *There is a chance of rain*). So let us define the following set of varieties (minimal types):

$$\mathcal{V}_v = \left\{ v_{\xi,\eta} \;\mid\; \xi \in \{+,-\}^{(\Xi)},\ \eta \in \{t,n\} \right\}$$

ξ, above, is a partial function from our set of subcategorizable GFs to $\{+,-\}$.[5] We will represent such a function in a non-standard, but hopefully fairly intuitive, format as a set of the GFs for which the function is defined each annotated with a superscript which gives the value of the function for that GF. An object of one of the varieties defined above will subcategorize for the GFs for which the function ξ is defined. Further, if $\xi(\mathrm{GF})$ is $+$ then the GF is thematic and vice versa. The second subscript indicates whether we are dealing with a terminal (t) or a nonterminal (n)—this will correspond to whether the node is specified for the H attribute. So two example varieties from $\mathcal{V}_v$ are $v_{\{\mathrm{SUBJ}^-\},n}$ and $v_{\{\mathrm{SUBJ}^+,\ \mathrm{OBJ}^-,\ \mathrm{XCOMP}^+\},t}$.

As well as these varieties (minimal types), it will be convenient to introduce some non-minimal types where any or all of the GFs lack a superscript. The semantics of this will be that such a GF is subcategorized for, but it is not being specified whether the GF is thematic or not. This happens regularly in lexical entries. For example, the semantic form for *believe* in LFG is 'believe⟨(↑SUBJ), (↑XCOMP)⟩(↑OBJ)' and the translation of this will introduce the type specification that the f-structure in which it occurs is $v_{\{\mathrm{SUBJ}^+,\ \mathrm{XCOMP}^+,\ \mathrm{OBJ}\},t}$. Whether or not the OBJ is thematic actually depends on the subcategorization (PRED-value) of the verb *believe* takes as a complement (XCOMP). We can understand any statement about such a non-minimal type as a meta-expression that expands into the obvious disjunction involving only minimal types (varieties).

In addition, we need to distinguish certain other varieties. As alluded to above, we are going to gloss over some of the details here, but we certainly need varieties for atomic values and semantic forms. So let us assume a set $\mathcal{V}_a$ of atomic value varieties:

$$\mathcal{V}_a = \{1,\ 2,\ 3,\ \mathrm{SG},\ \mathrm{DU},\ \mathrm{PL},\ \mathrm{NOM},\ \mathrm{ACC},\ \mathrm{S},\ \mathrm{NP},\ \mathrm{VP},\ \mathrm{V},\ \ldots\}$$

The semantic forms that are the values of PRED are conventionally written on one line, but are a complex object, pieces of which are themselves f-structures (attribute-value matrices).[6] We need to work out a format for representing these semantic forms. Let us tentatively adopt the following representation for a semantic form like [PRED 'believe⟨SUBJ, XCOMP⟩OBJ']:

(9)
$$\left[\mathrm{PRED} \quad \begin{bmatrix} \mathrm{RELN} & \text{believe} \\ \mathrm{THEMATIC} & \langle \_\_, \_\_ \rangle \\ \mathrm{NONTHEMATIC} & \langle \_\_ \rangle \end{bmatrix} \right]$$

[5]Making ξ a partial fraction effectively allows us to 'cheat' as there can be three values for each GF: $+$, $-$ and undefined.

[6]Although authors (including me) often just write GF names in a semantic form, theoretically these positions are meant to be unified with the values of the subcategorized grammatical functions in the enclosing f-structure that have those names (Kaplan & Bresnan, 1982, p. 189; see also Andrews, 1990, pp. 214–215).

We make the predicate name the value of the RELN attribute and represent the thematic and nonthematic arguments as two lists. We will borrow the method for handling lists from King (forthcoming, Section 3.3). For the PRED value of a word that lacks arguments (like nouns when used as referential objects), we will keep the same representation but just assume that both the lists THEMATIC and NONTHEMATIC are empty. The treatment of relation names is actually somewhat problematic. King's formalism assumes that the varieties partition the model space (i.e., each object is of one and only one variety). This is not a problem. The varieties introduced so far are clearly disjoint and we can if necessary cover the rest of the model space with one additional variety. But King's formalism also requires that the number of varieties is finite and the varieties of the RELN attribute (informally, its values) do not seem to satisfy this restriction. This is not a problem specific to LFG. The HPSG formalism of Pollard & Sag (1987, forthcoming) also has a RELN attribute with an apparently unbounded number of language particular values (see for example Pollard & Sag, 1987, p. 84 ff., p. 93, p. 95 ff.) and these all become varieties in King's system. Indeed, it gets worse as Pollard & Sag (1987) decide each relation has its own inventory of roles, which gives us another brush with the infinite (pp. 85–86 ff.). Each parameterized state of affairs (in which these relations and roles appear) would also have to be an object of a different type. One way to maintain a finitary basis to the variety system is to adopt the idea of semantic decomposition into a finite number of basic semantic concepts (which will be varieties). One approach to this is developed in Jackendoff (1983, 1990), who mentions the need for a finitary basis to the human conceptual system as one argument for his approach (1990, pp. 8–9, 37–41). Andrews & Manning (1991) discuss integrating Jackendovian semantic representations into LFG in place of the standard semantic forms.[7]

This has all been a long way of saying that we wish to put to one side the details of the representation of semantic forms. In our translation of LFG, each PRED value will introduce a statement about what variety it should appear in, and this variety specification will be sufficient to implement Completeness and Coherence. Below we will include our tentative representation of PRED-values, but to some extent we wish to marginalize this issue as one for further work, concerning not only the formalization of LFG, but other grammatical formalisms like HPSG as well. So let us just say that $\mathcal{V}_s$ is our set of varieties used to implement PRED-values. We will specify just one of its members, semanticform, which is the type of a feature structure specified for the three attributes RELN, THEMATIC and NONTHEMATIC, as above. However, we will not include all the other details of semantic representation in our signature, below.

We have introduced types for predicators (like verbs and prepositions), but not yet for non-predicators like nouns. Nouns and noun phrases will be of just four types, depending on whether they have a PRED or not and whether they are terminals or not. These types will be:

$$\mathcal{V}_n = \{n_{+,t},\ n_{-,t},\ n_{+,n},\ n_{-,n}\}$$

where the first subscript denotes whether the variety has a PRED or not and the second denotes whether it is a terminal or a nonterminal. So let our variety system be:

$$\mathcal{V} = \mathcal{V}_v \ \cup\ \mathcal{V}_a \ \cup\ \mathcal{V}_s \ \cup\ \mathcal{V}_n$$

Then our signature, $\mathcal{S} : \mathcal{V} \times \mathcal{A} \rightarrow pow(\mathcal{V})$, will look roughly like this:[8]

[7] Actually f-structures aren't very good for semantic interpretation (c.f. Halvorsen & Halvorsen, 1988; Andrews & Manning, 1991) and so in Andrews & Manning (1991), the feature PRED is done away with altogether, and the semantics is described using new attributes and a separate semantic projection σ.

[8] Variables such as η that appear in the signature are taken to be implicitly universally quantified over the appropriate domain ($\{+,-\}$ for η, etc.).

For $\alpha \in \mathcal{A}$	$\mathcal{S}\langle \text{atomic}, \alpha\rangle$	$= \emptyset$
For $\alpha \in \mathcal{A}$	$\mathcal{S}\langle \text{semanticform}, \alpha\rangle$	$=$ *This is being left unspecified!*
For $v \notin \{\text{atomic, semanticform}\}$,		
for $\alpha \in \mathcal{T}$	$\mathcal{S}\langle v, \alpha\rangle$	$= \mathcal{V}_a$
For $v \in \{v_{\xi,\eta}, n_{+,\eta}\}$	$\mathcal{S}\langle v, \text{PRED}\rangle$	$= \{\text{semanticform}\}$
otherwise	$\mathcal{S}\langle v, \text{PRED}\rangle$	$= \emptyset$
For $\gamma \in \Xi$,	$\mathcal{S}\langle v_{\xi,\eta}, \gamma\rangle$	$= \{n_{\xi(\gamma),\eta}\}$, if $\xi(\gamma)$ is defined
		$= \emptyset$, otherwise (ξ undefined on γ)
for $n \in \mathcal{V}_n$	$\mathcal{S}\langle n, \gamma\rangle$	$= \emptyset$
For elements of $\mathcal{V}_v$	$\mathcal{S}\langle v_{\xi,n}, \text{H}\rangle$	$= \{v_{\xi,n},\ v_{\xi,t}\}$
	$\mathcal{S}\langle v_{\xi,t}, \text{H}\rangle$	$= \emptyset$
and for elements of $\mathcal{V}_n$	$\mathcal{S}\langle n_{\pi,n}, \text{H}\rangle$	$= \{n_{\pi,n},\ n_{\pi,t}\}$
	$\mathcal{S}\langle n_{\pi,t}, \text{H}\rangle$	$= \emptyset$

The game plan should now be becoming somewhat clearer. So let U be our set of linguistic objects, and we define the functions V and A in the obvious way. For $u \in U$, $V(u)$ is the variety of u. And for $A : \mathcal{A} \to U^{(U)}$, $A(a)(u)$ is the value of attribute a of object u when this exists, and undefined otherwise. Then, by construction, $\langle U, V, A\rangle$ is a $\mathcal{S}$-structure.[9] We will now illustrate how LFG can be translated into this descriptive formalism. Because Completeness and Coherence is built right into the signature, the only Universal Principle of 'Classical LFG' (Bresnan, 1982a) will be the one that implements the f-projection defined above, but we will additionally have the expected grammar rules and lexicon.

Translating annotated grammar rules. Consider again the phrase structure rule:

(10) $$\text{S} \quad \to \quad \underset{(\uparrow\text{SUBJ}) = \downarrow}{\text{NP}} \quad \underset{\uparrow = \downarrow}{\text{VP}}$$

We will translate this as follows:

$$\begin{aligned} d'_1 \quad = \quad & \dagger\text{CAT} \sim \text{S} \wedge \\ & \dagger\text{SUBJ CAT} \sim \text{NP} \wedge \\ & \dagger\text{H CAT} \sim \text{VP} \end{aligned}$$

All phrase structure rules will apply to objects that are of nonterminal varieties (ones in $\mathcal{V}_v$ and $\mathcal{V}_n$ with the value n for their second subscript). However, we do not need to specify this explicitly, as it follows from the last equation in the above example (the specification of the head) by the definition of our signature (last four lines) and Condition S1 (King, forthcoming, Section 3.1).

Imagine we translated all our phrase structure rules $d_1, \ldots, d_n$ similarly into $d'_1, \ldots, d'_n$. Then our theory of immediate dominance would be:

$$\text{c-rules} = \left\{ (\dagger\text{H} \approx \dagger\text{H}) \to \bigvee\{d'_1, \ldots, d'_n\} \right\}$$

That is, if an object is a nonterminal (phrasal object), then it must obey one of the phrase structure rules. We will not deal with linear precedence (LP) here, but we will also assume a theory, lprec, of linear precedence which we could imagine as a set of global LP rules as in HPSG and as modeled in King (forthcoming, Section 3.3), or as separate linear precedence conditions on each phrase structure rule.

[9] This can't really be proved in a manner that is not trivial and virtually circular. We would need to assume King's conditions M1 and M2 (Section 3.2) and then adapt the proof in Section 3.3.

The f-projection. Let shared(f) $= \Xi \cup \mathcal{T} \cup \{$PRED, ADJ, XADJ$\}$. Then for all $\alpha \in$ shared(f), let:

$$d_\alpha = \dagger\mathrm{H} \approx \dagger\mathrm{H} \rightarrow \dagger\alpha \approx \dagger\mathrm{H}\alpha$$

and then set:

$$\text{f-proj} = \{d_\alpha \mid \alpha \in \text{shared(f)}\}$$

Then $u \in [\![\text{f-proj}]\!]_{\mathcal{S}} \iff u$ has a consistent f-projection. This captures the same sense of feature spreading and consistency that is caused in standard LFG by the $\uparrow = \downarrow$ notation. But here, we could also easily define other projections to handle the semantics and so on (Andrews & Manning, 1991).

The lexicon. We will exhibit how to model a couple of lexical items. These examples include the crucial type specifications that model Completeness and Consistency. So consider a verb like *believe* with lexical entry (11):

(11) believes, V, ($\uparrow$PRED) = 'believe$\langle$($\uparrow$SUBJ), ($\uparrow$XCOMP) $\rangle$($\uparrow$OBJ)'
($\uparrow$TENSE) = PRES
($\uparrow$SUBJ PERS) = 3
($\uparrow$SUBJ NUM) = SG

Recall that I will supply expressions that fill in our tentative semantic form representation (using King's notation for representing lists), but that, for our purposes, the key thing to notice is the variety specification that is introduced on the basis of the PRED value and which captures the subcategorization of the verb:

$$\begin{aligned}
l'_1 \;=\; & \dagger\mathrm{CAT} \sim \mathrm{V} \wedge \\
& \dagger\mathrm{PRED\ RELN} \sim \mathrm{believe}\ \wedge \\
& \dagger\mathrm{PRED\ THEMATIC\ FIRST} \approx \dagger\mathrm{SUBJ}\ \wedge \\
& \dagger\mathrm{PRED\ THEMATIC\ REST\ FIRST} \approx \dagger\mathrm{XCOMP}\ \wedge \\
& \dagger\mathrm{PRED\ THEMATIC\ REST\ REST} \sim \mathrm{elist}\ \wedge \\
& \dagger\mathrm{PRED\ NONTHEMATIC\ FIRST} \approx \dagger\mathrm{OBJ}\ \wedge \\
& \dagger\mathrm{PRED\ NONTHEMATIC\ REST} \sim \mathrm{elist}\ \wedge \\
& \dagger \sim v_{\{\mathrm{SUBJ}^+,\ \mathrm{XCOMP}^+,\ \mathrm{OBJ}\},t}\ \wedge \\
& \dagger\mathrm{TENSE} \sim \mathrm{PRES} \wedge \\
& \dagger\mathrm{SUBJ\ PERS} \sim 3 \wedge \\
& \dagger\mathrm{SUBJ\ NUM} \sim \mathrm{SG}
\end{aligned}$$

A normal noun, such as *dog* would translate something like this:

$$\begin{aligned}
l'_2 \;=\; & \dagger\mathrm{CAT} \sim \mathrm{N} \wedge \\
& \dagger\mathrm{PRED\ RELN} \sim \mathrm{dog} \wedge \\
& \dagger\mathrm{PRED\ THEMATIC} \sim \mathrm{elist} \wedge \\
& \dagger\mathrm{PRED\ NONTHEMATIC} \sim \mathrm{elist} \wedge \\
& \dagger \sim n_{+,t} \wedge \\
& \dagger\mathrm{PERS} \sim 3 \wedge \\
& \dagger\mathrm{NUM} \sim \mathrm{SG}
\end{aligned}$$

While a nonthematic noun such as nonthematic *there* would perhaps translate to:[10]

$$\begin{aligned} l'_3 \;=\; & \dagger\text{CAT} \sim \text{N} \wedge \\ & \dagger\text{FORM} \sim \text{there} \wedge \\ & \dagger \sim n_{-,t} \end{aligned}$$

Imagine we translated all our lexical items $l_1, \ldots, l_m$ similarly into $l'_1, \ldots, l'_m$. Then our theory of the lexicon would be:

$$\text{lexicon} = \{(\neg(\dagger\text{H} \approx \dagger\text{H}) \rightarrow \bigvee\{l'_1, \ldots, l'_m\})\}$$

This says that every terminal (things unspecified for the H attribute) must be licensed by one of the lexical items in the language.

A Theory of English. Let

$$\mathcal{E} = \text{c-rules} \;\cup\; \text{lprec} \;\cup\; \text{f-proj} \;\cup\; \text{lexicon}$$

Then $\mathcal{E}$ is our theory of LFG-English. The reader should be able to confirm that our variety system and signature are working with the lexical item specifications to ensure that only model objects with the obvious analogues of f-structure Consistency, Completeness and Coherence are being admitted as objects of LFG-English. We have not filled in the details of the treatment of adjuncts, prepositional phrases and so on, but what we have outlined models most of the essential features of LFG. Since King's descriptive formalism supports disjunctions, these would pose no problem, and there is nothing to stop us representing so called 'set-valued' features like ADJ(UNCT) simply as lists, again adopting the formalism of King (forthcoming, Section 3.3).[11] But let us turn finally to the largest hole in our coverage.

Constraining Equations. One of the exotica of the LFG armamentarium has been the distinction between *defining equations* and *constraining equations* (Kaplan & Bresnan, 1982, pp. 207–210). Although we have talked considerably about 'constraints', above, what we have been modeling has been LFG's defining equations. The intuitions behind defining equations and constraining equations depend crucially on the ideas of adding 'information' to a structure and thus being able to say whether something is or isn't there. A defining equation puts some information in an f-structure, while a constraining equation merely checks what something else has (or hasn't) put in an f-structure (but it doesn't license that information). Thus, a constraining equation is only satisfied if some other equation has defined a feature as having the same value. As such, this conception is fundamentally antithetical to the formalism that King (forthcoming) provides. In King's formalism (inheriting ideas from HPSG), there is a strong type system, and any model object always has values for all attributes appropriate to its type and the above distinction makes no sense. The notion of information has been systematically expunged and replaced by the simple concept of the denotation of descriptions.

Can we incorporate constraining equations into our formalism? Firstly, we should realize that not all constraining equations are the same with regards to this formalism. Negative constraining equations are not a problem. We can simply translate a negative constraining equation as the

[10] The FORM feature is assumed in much early LFG work (Kaplan & Bresnan 1982, p. 213), but its status has always been somewhat dubious...

[11] In addition to the three types mentioned above, Classical LFG recognizes a fourth type, features whose value is a set of f-structures (Kaplan & Bresnan, 1982, pp. 215–218). These sets are used to model GFs which can have multiple exponents, such as multiple adjectives or adverbs modifying a single noun or sentence.

appropriate statement in King's descriptive formalism. Existential and negative existential constraints are also not that much of a problem, as we can easily extend the ideas we used to model Completeness and Coherence above. An existential constraint, such as (↑CASE), can be translated into a claim that the appropriate object is of a type which has a CASE value, and the opposite for a negative existential constraint. This leaves the distinction between positive defining and constraining equations, such as:

(SUBJ CASE) $=_c$ DAT
(SUBJ CASE) = DAT

Is this distinction useful? It is vital to certain LFG analyses, such as Neidle's (1982) analysis of Russian Case. It allows her to capture something like the essence of the GB Case Filter (Chomsky, 1981): although NPs in Russian are marked for CASE (a constraining equation), to appear in a sentence their (abstract) Case must be licensed by the verb (a defining equation). Constraining equations are also used in Andrew's (1982, 1990) analysis of Modern Icelandic. Sag, Karttunen & Goldberg (forthcoming) try to produce an alternative analysis of Icelandic Case that doesn't involve constraining equations (so that it can be handled in a monotonic fashion in HPSG). However, their alternative solution seems to require a certain amount of 'machinery' (a DCASE feature which doubles the normal CASE feature; regular case-marking verbs (only) unify the two) and Andrews (personal communication) suggests that it does not handle all the Icelandic data, such as agreement of functionally controlled complements with quirky case marked NPs (at a minimum we would also seem to need to add DNUM and DGEND features doubling NUM and GEND, but I haven't tried to work this analysis through in detail). So positive constraining equations do not appear to be a superfluous entity that we can just expunge without offending anybody. It does not seem that constraining equations are vital to writing extensionally adequate grammars but simply that they usefully capture a distinction one wants to have in one's linguistic theory.[12] Indeed, Kaplan & Bresnan (1982, pp. 208, 210) largely foresaw Sag et al.'s approach of using "ad hoc features" and wrote about a different phenomenon:

> Introducing a special interpretation for f-description statements [that is, constraint equations, CM] is not strictly necessary to account for these facts. We could allow only the defining interpretation of equations and still obtain the right pattern of results by means of additional feature specifications.... There are two objections to the presence of such otherwise unmotivated features: they make the formal system more cumbersome for linguists to work with and less plausible as a characterization of the linguistic generalizations that children acquire.... [W]e have chosen an explicit notational device to highlight the conceptual distinction between definitions and constraints.

So then, can we model constraining equations? I think that, technically, the answer is yes. If we exclude only constraining equations referring to predicates (which aren't used) then the number of values of attributes is fixed and finite. Thus we could further extend our type system so that every value of every attribute gave us a different type. All constraining equations could then be expressed as statements about the variety of an object. However, even if one did not already think so before, by the time one has adopted a move like this, I think it should be clear that the fit between the theory and the mathematics that we are using to formalize it has become rather strained. The

[12] This is clearly an unsatisfactory way to leave things, from a mathematical point of view. However, (i) I am aware of no algorithmic method for changing a grammar with constraining equations to one using only defining equations, but (ii) it does not seem as if the admittance of constraining equations increases the weak generative capacity of LFGs.

system of varieties was never meant to be a bloated entity that has been co-opted to modeling constraining equations. So there is a real issue here that remains to be addressed.

Bibliography

Andrews, A. D. (1982). The Representation of Case in Modern Icelandic. In Bresnan (1982a), pp. 427–503.

Andrews, A. D. (1990). Case structures and control in Modern Icelandic. In J. Maling & A. Zaenen (Eds.), *Syntax & Semantics 24: Modern Icelandic Syntax*, pp. 187–234. Tokyo: Academic Press.

Andrews, A. D. & Manning, C. D. (1991). Projections in LFG. Unpublished manuscript. The Australian National University and Stanford University.

Bresnan, J. (Ed). (1982a). *The Mental Representation of Grammatical Relations*. Cambridge, MA: The MIT Press.

Bresnan, J. (1982b). Control and Complementation. In Bresnan (1982a), pp. 282–390.

Bresnan, J. & Kaplan, R. M. (1982). Introduction: Grammars as Mental Representations of Language. In Bresnan (1982a), pp. xvii–lii.

Inkelas, S. & Zec, D. (Eds.). (1990). *The Phonology-Syntax Connection*. Chicago: University of Chicago Press.

Jackendoff, R. (1983). *Semantics and Cognition*. Cambridge, MA: MIT Press.

Jackendoff, R. (1990). *Semantic Structures*. Cambridge, MA: MIT Press.

Halvorsen, P.-K. & Kaplan, R. M. (1988). Projections and Semantic Description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, Tokyo.

Kaplan, R. M. & Bresnan, J. (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Bresnan (1982a), pp. 173–281.

King, P. J. (forthcoming). *A Logical Formalism for Head-driven Phrase Structure Grammar*. Stanford, CA: CSLI.

Manning, C. D. (1986). A neater graph-theoretic formalism for LFG. Unpublished manuscript. The Australian National University.

Neidle, C. (1982). Case Agreement in Russian. In Bresnan (1982a), pp. 391–426.

Pollard, C. & Sag, I. (1987). *Information-Based Syntax and Semantics Vol. 1*. Stanford, CA: CSLI.

Pollard, C. & Sag, I. (forthcoming). *Information-Based Syntax and Semantics Vol. 2*. Stanford, CA: CSLI.

Sag, I. A., Karttunen, L. & Goldberg, G. (forthcoming). A lexical analysis of Icelandic case. In Sag & Szabolcsi (Eds.), *Lexical Matters*. Stanford, CA: CSLI.

Sells, P. (1985). *Lectures on Contemporary Syntactic Theories*. Stanford, CA: CSLI.