5

Cleaning up the Big Mess: Discontinuous Dependencies and Complex Determiners

PAUL KAY & IVAN A. SAG

1 Introduction

Two understudied phenomena of English[1] are intimately intertwined but, insofar as they are studied at all, are not usually related to one another. The discontinuous dependent phenomenon (DD) illustrated in (1) and the complex pre-determination (CPD) phenomenon illustrated in (2)[2] are independent. That is, each of these phenomena may occur independently of the other, as shown in (1)–(2):

(1) a. [[**so** willing to help out] **that they called early**]
b. [[[**too** far] behind on points] **to quit**]
c. [[[**more** ready] for what was coming] **than I was**]
d. [[**as** prepared for the worst] **as anyone**]
e. [[the **same** courage in the face of adversity] **as yours**]

(2) a. [[**this** delicious] a lasagna]. . .
b. [[**that** friendly] a policeman]. . .
c. [[**How** hard] a problem] (was it)?
d. [**What** a fiasco] (it was)!

[1]For their helpful comments and/or discussion regarding the ideas presented here, we would like to thank Rui Chaves, Berthold Crysmann, Charles Fillmore, Dan Flickinger, Laura Michaelis, Chris Potts, Stefan Müller, Peter Sells and Frank Van Eynde.

[2]CPD is also known as the 'Big Mess' Construction. See Berman 1974, Arnold and Sadler 1992, and Van Eynde 2007.

Sign-Based Construction Grammar.
Editors: Hans C. Boas & Ivan A. Sag.

The oddity (the ‘non-core’ property) of DD examples like those in (1) is that they appear to call for a discontinuous constituent analysis. The oddity of CPD examples like those in (2) is that they present an adjective modifying an NP (or DP), rather than a nominal (a common noun phrase or ‘$\overline{\text{N}}$’) – specifically an NP determined by the singular indefinite article *a*.

Although, as we have seen in (1) and (2), DD and CPD may appear independently, they frequently occur intertwined as in (3):

(3) a. [[[**too** heavy] a trunk] **(for me) to lift** __]
b. [[[**so** lovely] a melody] **that some people cried**]
c. [[[**more** sincere] an apology] **than her critics acknowledged**]
d. [[[**as** good] a singer] **as many professionals**]

Unsurprisingly, the initial lexical licenser determines the three-way distributional distinction displayed in (1), (2) and (3).

Licensers of DD but not CPD include those comparative governors governors listed in (4):[3]

(4) *same...as, similar...to, equal...to/with, identical... to/with,* ADJ-*er...than, rather...than, ...else...than, ...enough...that, ...other...than*

Complement-selecting adjectives, verbs, and nouns also participate in DD, as we will see. Licensers of CPD but not DD include:

(5) *this, that, how*

And licensers of both DD and CPD are listed, exhaustively we believe, in (6):[4]

(6) *so, too, more, less, as, such*

It is notable that comparative licensers are split between those that do not (e.g. (4)) and those that do (e.g. (6)) license CPD. There are licensers of CPD but not DD, of DD but not CPD, and licensers of both DD and CPD.

More than one DD can occur in a clause, as illustrated in (7):

(7) a. **so** much *more* satisfied *than* the last time **that he couldn’t stop smiling**
b. [[[**too** many *fewer*] supporters] *than her opponent* **(for her) to rely on appeals to her base**]
c. [[[[**enough** *bigger*] an audience] *than last time*] **to require standing room only**]

[3] See Huddleston and Pullum 2002: 1104.

[4] It should be noted that *such* is different from the other adjective specifiers in (6). In particular, *such*, like exclamative *what*, functions essentially as the portmanteau of a specifier and an adjective.

In examples such as (7) the multiple DDs form nested dependencies. The corresponding crossed dependencies in (8) are impossible: [5]

(8) a.***so** much *more* satisfied **that he couldn't stop smiling** *than the last time*
b.***too** many *fewer* supporters **(for her) to rely on appeals to her base** *than her opponent*
c.***enough** *bigger* an audience **to require standing room only** *than last time*

Other DDs may, however, participate with arguments or modifiers in either nested (e.g. (9b,d)) or crossed (e.g. (9a,c)) dependencies:

(9) a. Kim was [[[**more** *willing*] **than Pat is**] *to wash the dishes*].
b. Kim [[[is [**more** *willing*] now] *to wash the dishes*] **than Pat is**].
c. I [[[sent out [**more** *books*] yesterday] **than ever before**] *that I really liked*].
d. I [[[sent out [**more** *books*] yesterday] *that I really liked*] **than ever before**].

The generalization we see in these data is the following:

(10) **Generalization:** All DD licensers except *so, too,* and *enough* can participate in crossed dependencies with arguments and other dependents.

We will need to formulate the listemes for the licensers and, critically, the relevant phrasal constructions, in such a way as to account for all the observations made above, as well as some further data mentioned below.

2 Previous Proposals

There are no fully worked out analyses of DD in the syntactic literature, though there are discussions of various aspects of DD. Perhaps the most detailed of these proposal is due to Chae (1992), who extends the GPSG analysis of gap-binding by allowing a word like *too* to transmit its gap-binding potential to a higher node, e.g. to the adjective phrase *too hot* in examples like (11):

(11) This is [$_{AP}$ [**too** hot] [**to touch __**]].

In Chae's analysis, binding of the gap takes place when a nonempty SLASH specification and its appropriate licensing specification are both passed up to a single point in the tree, i.e. the AP labelled in (11).

[5] To the best of our knowledge, the first observations of contrasts of this kind are made by Williams (1974). See also Baltin 1982 and Guéron and May 1984. The relevant data is reassessed in important ways by Stucky (1987).

Flickinger and Nerbonne (1992) provide an analysis of examples like (12), proposing to allow SUBCAT (VALENCE) information to be inherited from multiple daughters in structures like (12):

(12) An [[**easy** man] [**to please** __]]$_{\overline{\mathrm{N}}}$ ...

On their proposal, an $\overline{\mathrm{N}}$ like *easy man* inherits its subcateogrization potential from both *easy* and *man* and hence can select *to please* __ as a complement.

The EXTRA feature was first proposed by Pollard in unpublished work and appears briefly in Pollard and Sag's (1994: 368), discussion of extraposition in comparative phrases. Subsequent analyses using the EXTRA feature to analyze various extraposition phenomena in English and German include Keller 1995, Van Eynde 1996, Bouma 1996, Kim and Sag 2005, and Crysmann to appear.

Kiss (2005; see also Wittenburg 1987) treats German relative clause extraposition as an anaphoric dependency, rather than a syntactic one, introducing a feature ANCHORS to pass up a set of indices from NPs within a given phrase, each of which can be associated with an extraposed relative clause at a higher level of structure. See Müller 2004 and Crysmann to appear for assessments of this and other alternative approaches.

CPD has been discussed by many researchers in the transformational literature, culminating perhaps in the work of Kennedy and Merchant (2000), who provide a useful review and a comprehensive proposal that even addresses complex pre-determiners with *of* (e.g. *how much of a difference*), which we will not discuss here. Their proposal is stated in terms of complex structures, a rich array of empty categories, and movement operations that make appeal to a constraint on phonetic form (see their footnote 28) in order to account for the most basic facts of CPD, i.e. the contrasts given in (13) below. However, Kennedy and Merchant do not include a formulation of this essential piece of their analysis.

The most successful treatment of CPD to date, in our view, is that of Van Eynde (2007).[6] A key aspect of this analysis, which we follow here in the main, is the replacement of Pollard and Sag's (1994) features MOD and SPEC by the single feature SELECT (SEL). The SELECT analysis allows Pollard and Sag's SPR feature to be eliminated, as well.

None of the proposals just mentioned provides a treatment of the interaction of DD and CPD. It turns out, however, that this interaction will follow straightforwardly from the analysis we propose here.

[6] This is an outgrowth of earlier work by Van Eynde (1998), further developing the ideas of Allegranza (1998b). See also Van Eynde 2006 and Allegranza 2007.

3 Analysis

We develop an analysis of DD, CPD, and their interaction in terms of **Sign-Based Construction Grammar** (SBCG). For a more detailed exposition of SBCG, the reader is referred to Sag this volume and the references cited there.

In the introduction, we sketched a few of the more salient distributional facts about DDs. We begin the more analytical discussion with CPD structures, as illustrated in (2) and (3). As already noted, the interesting property of these structures is that they contain adjective phrases modifying determined NPs, rather than the usual adjectival modification of undetermined common noun expressions (CNPs):

(13) a. a [rotten pear] (cf. *rotten a pear)
b. a [mere bagatelle] (cf. *mere a bagatelle)
c. the [old book]
d. her [seven [lonely nights]]

The SBCG representation of the bracketed expression in (13a), a feature structure of type *head-functor-construct*, is given in Figure 1.[7] Beginning with the first daughter (specified as [FORM ⟨*rotten*⟩]) we note that the SYN value has three attributes: CATEGORY, MARKING and EXTRA As indicated, the CATEGORY (CAT) value is a feature structure of type *adj(ective)*. This feature structure includes a specification for the feature SELECT, whose value is represented by the tag '$\boxed{H}$', indicating that this value has been equated with another feature structure in the same diagram, which is also 'tagged' by $\boxed{H}$. The SELECT-based analysis provides a unified treatment of modifiers, specifiers, determiners and other 'markers' in terms of lexically-varying specifications for the SELECT feature. These in turn correspond to the varying possibilities for the second daughter (in this construction). This SELECT-based treatment of nominal modifiers is fundamental to our analysis of extraposed relative clauses, presented in (28) below. The MARKING (MRKG) value of the first daughter, *unmarked* (*unmk*), reflects the fact that adjectives are lexically specified as [MRKG *unmk*]. And following Van Eynde (2007), the mother's MRKG value is identified with that of the functor daughter.[8]

[7] Constructs are in essence local trees whose nodes are fully resolved feature structures. (Formally, they are functions from the domain (MTR, DTRS), where MTR (MOTHER) is sign-valued and the value of DTRS (DAUGHTERS) is a list of signs.) Constructs are the local structures associated with phrases like [*red ball*], [*the* [*red ball*]], or [*put Fido out*]. They are represented as boxed local trees whose nodes are AVMs. Combinatory constructions are (partial) descriptions of constructs, that is, constraints that characterize a class of constructs. They are represented as AVMs with a MTR and a DTRS feature, not as trees. Since constructions are descriptions, not feature structures, they are never boxed.

[8] Note that the features HEAD, LOCAL, and NONLOCAL, familiar from much work in HPSG, are not just being supressed in our displays. They have in fact been eliminated from the grammar.

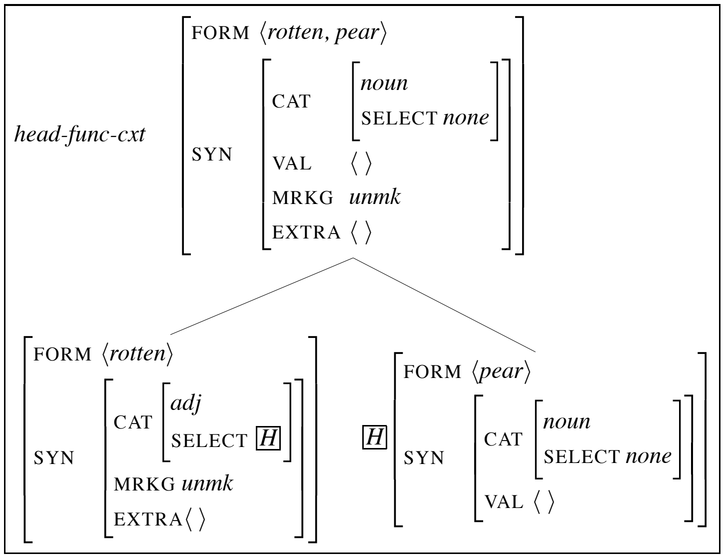


FIGURE 1 A Head-Functor Construct

The EXTRA feature plays a central role in the present discussion. It is a nonlocal, list-valued feature that provides the mechanism for a wide range of extrapositions (in line with the arguments offered by Keller, Van Eynde, and Bouma), including those illustrated in (14):[9]

(14) a. **It** seems **that your hair is burning**.
(extraposition from subject)
b. They regret **it** very much **that we could not hire Mosconi**.
(extraposition from object)
c. I am **unwilling** when sober **to sign any such petition**.
(extraposition of VP complement)
d. He **lowered** the nitro bottle gently **onto the floor**.
(extraposition of PP complement)
e. **An article** appeared yesterday **about the situation in Kazakhstan**.
(extraposition of PP modifier)
f. **A man** walked in **who was wearing striped suspenders**.
(extraposition of relative clause)

We continue to speak of 'nonlocal' features, however, by which we mean features that transmit information about an unbounded dependency. The SBCG theory of 'head' features involves the feature HD-DTR, achieving much the same effect as the theory in Ginzburg and Sag 2000.

[9]We will not attempt to establish this broad claim in the present paper, but we intend the EXTRA feature and the constructions that mention it eventually to cover all the data in (14).

The EXTRA feature thus works much like SLASH (GAP): A lexical entry or lexical construction requires an item on the EXTRA list of a sign. For example, the comparative listeme *more* specifies a *than*-phrase on its EXTRA list. When this sign serves as the daughter of some phrasal construct, its non-empty EXTRA specification becomes part of the mother's EXTRA list and this continues until a higher structure (a *head-extra-construct*) realizes the item as a constituent sign whose mother's EXTRA list is free of the now realized ('extraposed') item. We will see how this works in detail below. For the moment we note that in a *hd-func-cxt* like the one shown in Figure 1, the mother inherits the EXTRA value from the non-head (functor) daughter.

The second daughter ([FORM ⟨*pear*⟩]) is the head daughter, as indicated by the boxed *H* preceding the outer brackets. Its CAT value, as indicated, is a feature structure of type *noun* and its VALENCE (VAL) value is the empty list. The mother sign ([FORM ⟨*rotten, pear*⟩]) of this construct, following Van Eynde, inherits its CAT and VAL specifications from the head daughter and its MRKG and EXTRA values from the functor (non-head) daughter. The construction that licenses this construct is the Head-Functor Construction, shown in (15):[10,11]

(15) **Head-Functor Construction:** (↑*headed-cxt*):

$$\textit{head-func-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \text{SYN} \; X \,!\, \begin{bmatrix} \text{MRKG} & M \\ \text{EXTRA} & L \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \left\langle \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{CAT} \; [\text{SELECT} \; H] \\ \text{MRKG} \; M \\ \text{EXTRA} \; L \end{bmatrix} \end{bmatrix}, \; H\!:\![\text{SYN} \; X] \right\rangle \\ \text{HD-DTR} & H \end{bmatrix}$$

This construction specifies the inheritance by the mother of the MRKG and EXTRA values from the functor (first) daughter, as we observed in the *rotten pear* construct in Figure 1.[12] It also specifies that the mother's SYN value is otherwise identical to that of the head (second) daughter. As explained in Sag this volume, the notation

[10] Discussion of the semantics employed in our analysis is postponed until section 4.

[11] Van Eynde (2006, 2007) couches his proposal in terms of phrasal types, using the framework of Ginzburg and Sag (2000). For convenience, we refer to his phrasal type constraints as SBCG constructions. The reader should also be aware that Van Eynde proposes multiple subtypes of his head-functor phrasal type, a complication we do not investigate here.

[12] Capital letter tags (e.g. 'X') denote feature structure variables; 'L' tags denote variables ranging over (possibly empty) lists of feature structures.

$$\begin{bmatrix} \text{FEATURE}_0 & X ! \begin{bmatrix} \text{FEATURE}_1 \ldots \\ \ldots \\ \text{FEATURE}_n \ldots \end{bmatrix} \end{bmatrix}$$

means that the value of the feature FEATURE_0 is identical to that of the variable X, which appears elsewhere in the larger description, except for the values of the features $\text{FEATURE}_1 \ldots \text{FEATURE}_n$. The Head-Functor Construction thus requires that the SYN values of the mother and head daughter be identical except for the values of EXTRA and MRKG. The mother's value for these features is constrained only in terms of the functor daughter, as indicated in (15). Note that this analysis entails, since CAT is part of SYN, that the mother's CAT value will be identical to the head-daughter's CAT value. Among other things, this guarantees that when a functor combines with a given expression, the latter passes up its SELECT specification to the phrase it projects. Thus, *unhappy camper* will be [SELECT *none*] because *camper* is, *unusually happy* will be [SELECT CNP] because *happy* is, and so forth. The Head-Functor Construction thus licenses adjectivally modified nominals and determined noun phrases, among other local structures, correctly assigning to each an appropriate MRKG, EXTRA, and SELECT value.

We now turn our attention to the CPD phenomenon we illustrated in (2)–(3) above. Here a problem arises, since we cannot use the Head-Functor Construction to license CPD noun phrases like [[*so big*] [*a mess*]]. This is because (1) ordinary adjectives, like *big* or *rotten*, select only undetermined nominals (CNPs), as illustrated in (13a,b), and (2) since SELECT is a CAT feature, the Head-Functor Construction would incorrectly require the SELECT value of *so big* to be the same as that of *big*, which would prevent it from modifying *a mess*.

Van Eynde (2007) has proposed a constructional HPSG solution to this problem at the level of the NP. That is, to license a noun phrase like [[*so big*] [*a mess*]] Van Eynde proposes a construction whose mother is a noun phrase and whose first daughter is an adjective phrase marked 'degree', which necessitates that it contain a degree modifier from the list given in (6), excluding *such* (which is lexically specified to select a singular, indefinite NP). In Van Eynde's (2007) 'Big Mess' construction, which is distinct from his Head-Functor construction, the adjectival daughter does **not** select the nominal head; rather the Big Mess construction specifies merely that the indices of the two daughters are identified. And because only indices are identified in Van Eynde's analysis, there appears to be no way to allow idiom dependencies (multiword expressions) to interact properly with complex predeterminers, as illustrated in (16):

(16) a. Let's not make too big a deal out if it.

b. ... stop raising the topic of ... It's so red a herring it looks like a baboon's arse.

c. Romney is one of the four founders of Bain Capital. He can't be that empty a suit.

Here the dependency between, e.g. idiomatic *bit* and idiomatic *deal*, must be transmitted across the determiner *a*. This is accomplished in our analysis of idioms (see Kay and Sag 2012) via the feature SELECT.

We present here a related analysis that operates inside the adjective phrase, rather than at the NP level. This choice encodes a different intuition, namely that the special property of the CPD phenomenon is the apparent divergence of the selectional potential of an AP from that of its lexical head. On this view, *big* selects an undetermined noun phrase, but *so big* selects a singular, indefinite NP. The selectional process is the same as in normal adjectival modification: once the special AP *so big* is constructed to select an NP rather than a CNP, the AP and the NP are combined by the familiar Head-Functor Construction. The need for a special construction arises only in building the AP.[13]

The CPD construct *so big* is shown in Figure 2. Starting with the first daughter ([FORM ⟨*so*⟩]), we note that its category is *adverb* and that it selects its right sister, indicated by the tag '[1]'. The *so*-constituent is specified as [MRKG *deg*], which is a lexical property of all and only the lexical items listed in (6), other than *such*. The EXTRA list contains a single item, which is a *that*-marked clause. The second daughter ([FORM ⟨*big*⟩]) is of category *adjective* and selects an unmarked nominal head. The mother of this construct ([FORM ⟨*so, big*⟩]) inherits its MRKG and EXTRA values from the first daughter, as in a *head-func-cxt*. Another similarity with head-functor constructs is the identification of the type of mother's CAT value (*adj*) with that of the second daughter. But here the parallelism with the Head-Functor Construction breaks down; we note that the SELECT values of the mother and second daughter differ. In particular, since the head daughter reflects the SELECT restriction of the lexical item *big*, i.e. [MRKG *unmk*], it must be an adjectival expression that would normally modify an undetermined nominal. By contrast, the mother's SELECT value is a nominal sign specified as [MRKG *a*],

[13]Our account, unlike Van Eynde's, provides a uniform treatment of CPD APs (*so big*) and lexical expressions, e.g. *what*, *such*, and *many*, which may appear in pre-determiner position (*what/such/many a fool*). That is, *what*, *such*, and *many* can bear exactly the same SELECT value as the phrases licensed by the CPD Construction formulated in Figure 3. Although these words select bare plurals (*Such fools!*), which CPD APs do not, all these facts could presumably be accommodated in a lexicon with multiple constraint inheritance. Although there is considerable lexical idiosyncrasy in this domain, as Van Eynde observes, we are not aware of further data that would distinguish on empirical grounds between our analysis and an appropriate extension of Van Eynde's.

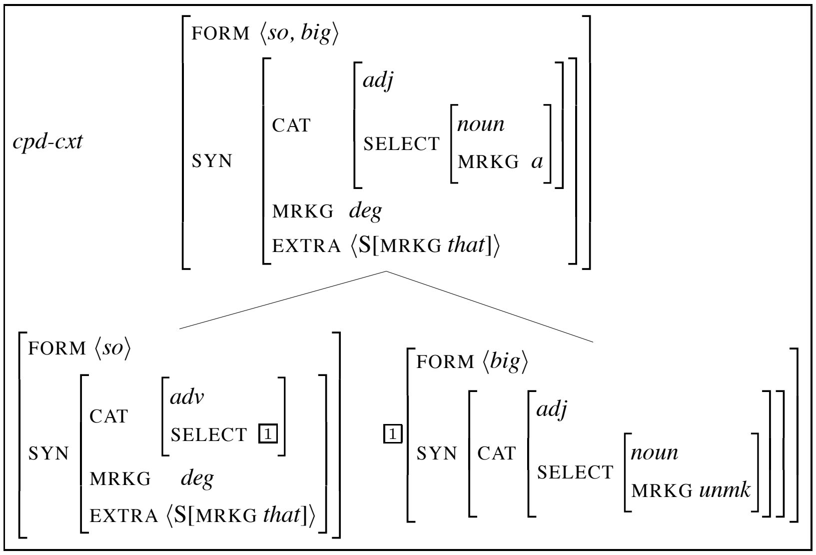


FIGURE 2 A Complex Pre-Determiner Construct

$$\textit{cpd-cxt} \Rightarrow \left[\begin{array}{ll} \text{MTR} & \left[\text{SYN} \left[\begin{array}{ll} \text{CAT} & X\,!\left[\text{SELECT} \left[\text{SYN} \left[\begin{array}{l}\text{CAT } \textit{noun} \\ \text{MKG } \textit{a}\end{array}\right]\right]\right] \\ \text{MKG} & \textit{deg} \\ \text{EXTRA} & L \end{array}\right]\right] \\ \text{DTRS} & \left\langle \left[\text{SYN} \left[\begin{array}{ll} \text{CAT} & [\text{SELECT } H] \\ \text{MKG} & \textit{deg} \\ \text{EXTRA} & L\end{array}\right]\right], H \right\rangle \\ \text{HD-DTR} & H : \left[\text{SYN} \left[\text{CAT } X : \left[\begin{array}{l} \textit{adj} \\ \text{SELECT} \left[\text{SYN}\left[\begin{array}{l}\text{CAT } \textit{noun} \\ \text{MKG } \textit{unmkd}\end{array}\right]\right]\end{array}\right]\right]\right] \end{array}\right]$$

FIGURE 3 Complex Pre-Determiner Construction (↑*headed-cxt*) and hence the constructed expression must modify an NP determined by the article *a*.

The Complex Pre-Determiner Construction, closely related to the Head-Functor Construction, is sketched in Figure 3.[14] The CAT values of the head daughter and the mother are identified (in terms of X ! [...]) except for the values of SELECT. The mother's SELECT value is required to be an NP specified as [MRKG *a*], i.e. an NP containing the determiner *a*. The head daughter's SELECT value, by contrast, is specified as [MRKG *unmk*], which ensures that the second daughter will be an undetermined NP (a CNP). As in the construct that we have just considered (Figure 2), the MRKG and EXTRA values of the first daughter and the mother are identified. The first daughter is specified as [MRKG *deg*], identifying it as one of the lexical licensers of the CPD phenomenon.

A noun phrase like *so big a mess* is licensed as follows. The CPD Construction in Figure 3 licenses the construct *so big* shown in Figure 2. Because this construct is well-formed, its mother, the phrase sketched in (17), is licensed by the grammar:

(17)
$$\left[\begin{array}{ll} \textit{phrase} & \\ \text{FORM} & \langle \textit{so, big}\rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \textit{adj} & \\ \text{SELECT} & \left[\text{SYN} \left[\begin{array}{ll}\text{CAT} & \textit{noun}\\ \text{MRKG} & \textit{a}\end{array}\right]\right]\end{array}\right] \\ \text{MRKG} & \textit{deg} \\ \text{EXTRA} & \langle \text{S[MRKG } \textit{that}]\rangle \end{array}\right]\end{array}\right]$$

The NP *a mess* is licensed via the Head-Functor Construction (see (15) above), with the determiner *a* passing up its [MRKG *a*] specification to the constructed NP:

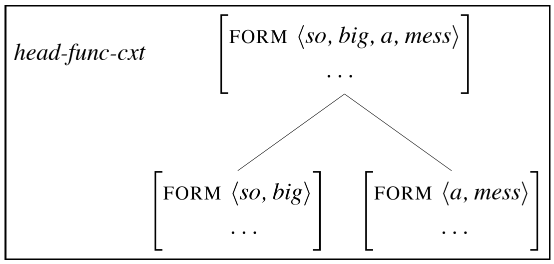
(18)
$$\left[\begin{array}{ll} \textit{phrase} & \\ \text{FORM} & \langle \textit{a, mess}\rangle \\ \text{SYN} & \left[\begin{array}{ll} \text{CAT} & \textit{noun} \\ \text{MRKG} & \textit{a} \\ \text{VAL} & \langle\ \rangle \end{array}\right]\end{array}\right]$$

Therefore, the Head-Functor Construction allows the combination of *so big*

[14] It is possible to factor out the similarities between these two sister constructions in terms of constraints associated with a common supertype, simplifying the formulation of both constructions. We leave such refinements for further research.

and *a mess*, with the former selecting the latter. The resulting construct is shown in (19):

(19)



And the mother of the construct in (19) has the properties shown in (20): [15]

(20)

$$\begin{bmatrix} \text{FORM} & \langle \textit{so, big, a, mess} \rangle \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \textit{noun} \\ \text{SELECT} & \textit{none} \end{bmatrix} \\ \text{VAL} & \langle\ \rangle \\ \text{MRKG} & \textit{deg} \\ \text{EXTRA} & \langle \text{S}[\textit{that}] \rangle \end{bmatrix} \end{bmatrix}$$

that-clause@*that*-clause Having illustrated the analysis of constructs like *so big a mess*, we now need to account for an extraposed *that*-clause, extraposed in the sense that while its introduction is dependent on the presence of *so*, it is linearized after the noun *mess*. Moreover, it need not appear immediately after *mess*, as shown in (21):

(21) [[[**so** big a mess] resulted from the meeting of the committee on the seventeenth of August] **that it took hours to clean it up**].

The mechanism for realizing extraposed elements and the positions in which they can be realized will occupy much of our attention in the remainder of this paper.

We noted that in both the Head-Functor Construction and the Complex Pre-Determiner Construction the mother inherits its EXTRA value from the first daughter. Because of this, the listeme for *so*, sketched in (22) is an essential ingredient in the *so-that* dependency:

[15]Following Müller (2009), predicative NPs are created via a unary ('pumping') construction whose daughter is a nonpredicative NP. This provides a straightforward account of predicative uses, e.g. examples like *She is so big a fan that she bought season tickets, Kim is too honest a guy to do that*, etc.

(22)
$$\begin{bmatrix} \text{FORM} & \langle \textit{so} \rangle \\ \text{SYN} & \begin{bmatrix} \text{CAT} & [\text{SELECT}\ [\text{SYN}\ [\text{EXTRA}\ L_1]]] \\ \text{EXTRA} & L_1 \oplus \langle \text{S}[\textit{that}] \rangle \end{bmatrix} \end{bmatrix}$$

The listeme for *so* ensures that its EXTRA list includes a *that*-clause appended to (⊕) the EXTRA list of the element that *so* selects. That is, *so* says in effect: 'my EXTRA list consists of the EXTRA list of the element I select followed by a *that*-clause'. Various constructions, including the Complex Pre-Determiner Construction, specify the EXTRA value of the mother in terms of the EXTRA values of the daughters. In the case of the constructions we have seen so far – and also the Subject-Predicate Construction, presented below – the mother's EXTRA value is identified with the EXTRA value of the first daughter. Often the EXTRA list of the selected element will be empty, as in the case of *big*. The result is that when *so* and *big* are combined, the EXTRA value of the mother (*so big*) is just the singleton list containing S[*that*], as in Figure 2 and (17) above. The EXTRA values of both *a* and *mess* are the empty list, so the EXTRA value of *a mess* is also the empty list. Hence, the EXTRA value of *so big a mess* will consist of the single item S[*that*], which originated on the EXTRA list of the listeme for *so*, got 'passed up' to *so big* by the Complex Pre-Determiner Construction, and then again to *so big a mess* by the Head-Functor Construction.

How do extraposed elements exit the EXTRA list and get realized in the sentence? The extraposition analysis we are proposing follows previous GPSG/HPSG treatments of nonlocal dependencies.[16] At the site of introduction, lexical or constructional constraints ensure that the unrealized element corresponds to an element of the list serving as the value of the nonlocal feature **F** (SLASH (GAP) or, in this case, EXTRA). The nonempty specification for **F** is introduced (lexically, as we have seen) at a given point in an analysis tree. General principles then require that the **F**-specifications be inherited by the mothers of successively larger constructs – these phrases form the middle of the nonlocal dependency. Certain constructions then license the presence of a daughter that has a nonempty **F**-specification, typically introducing a new phrase (the filler in the case of filler-gap dependencies) that is identified with a member of its sister's **F**-value (at the top of the nonlocal dependency). The construction realizing extraposed elements, the Head-Extraposition Construction,[17] is formulated in (23):

[16] See Gazdar 1981, Pollard and Sag 1994, Bouma et al. 2001, and Levine and Hukari 2006.

[17] See Pollard and Sag 1994, Keller 1995, Van Eynde 1996, Bouma 1996, Kim and Sag 2005 and Crysmann to appear.

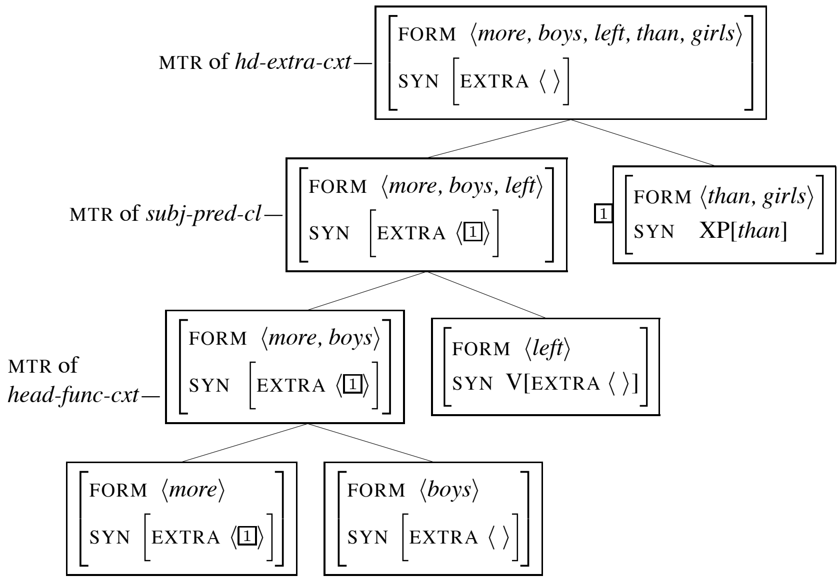


FIGURE 4 A Head-Extraposition Analysis Tree

(23) **Head-Extraposition Construction: (↑*headed-cxt*)**

$$hd\text{-}extra\text{-}cxt \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \text{SYN } Y\,!\begin{bmatrix}\text{EXTRA} & L\end{bmatrix}\end{bmatrix} \\ \text{DTRS} & \left\langle H : \begin{bmatrix}\text{SYN } Y : \begin{bmatrix}\text{EXTRA} & \langle X \rangle \oplus L\end{bmatrix}\end{bmatrix}, X \right\rangle \\ \text{HD-DTR} & H \end{bmatrix}$$

The Head-Extraposition Construction in (23) realizes the initial element of the EXTRA list of the head (first) daughter as the second daughter. The EXTRA list of the mother is the EXTRA list of the head daughter minus the element realized as the second daughter. This means that the order of elements on a non-singleton EXTRA list corresponds to the linear order of those elements in a binary-branching head-extraposition derivation.

The combination of the three lexical and constructional processes is exemplified by the analysis tree in Figure 4.[18] Starting at the lower left, we see that *more*, in combining with *boys*, records on its EXTRA list the requirement for

[18]Note that an analysis tree like this is not a structure licensed directly by the grammar. Rather, each local tree it contains is directly licensed. For this reason, there is no box around the tree as a whole, which is simply a sketch of a 'proof' one might give to illustrate that the grammar licenses the particular sign at the top of the analysis tree.

a *than*-phrase, represented by the tag '1', adding this element to the empty EXTRA list of its selected sister *boys*. The Head-Functor Construction identifies the EXTRA list of its functor daughter *more* with that of the mother of the construct it licenses (*more boys*). When *more boys* and *left* combine in accordance with the Subject-Predicate Construction, the EXTRA list of the first (non-head) daughter *more boys* also becomes the EXTRA list of the mother *more boys left* (because the EXTRA list of the head daughter must be empty) – see below. The construct combining *more boys left* and *than girls* is licensed by the Head-Extraposition Construction (see (23)), which realizes the sole member of the head daughter's EXTRA list (the XP[*than*]) as the second daughter *than girls* of the highest construct in Figure 4. The EXTRA list of this construct's mother is the empty list.

Extraposed elements obey certain ordering restrictions, as we saw in examples (7)–(9) above. In order to specify where extraposed elements can be realized we need to consider further constructions. First, we note that some extraposed complements, either arising within the VP or extraposed from the subject, can be permuted with arguments of predicates (as in (24)) and also with other extraposed elements, such as relative clauses (as in (25)):

(24) a. Kim was **more** *willing* **than Pat** *to wash the dishes*.
b. Kim was **more** *willing to wash the dishes* **than Pat**.
c. I sent out **more** *books* yesterday *that I really liked* **than ever before**.
d. I sent out **more** *books* yesterday **than ever before** *that I really liked*.

(25) a. **More** *books* arrived *that I actually liked* **than I expected**.
b. **More** *books* arrived **than I expected** *that I actually LIKED*.

As noted earlier, not all extraposed elements have this property. In particular, as summarized in (10) above, complements of *too, so* and *enough* do not permute with arguments or other extraposed dependents, as shown again by the examples in (26):

(26) a. The boys are **so** *proud* now *of their achievements* **that they've become unbearable**.
b.*The boys are **so** *proud* now **that they've become unbearable** *of their achievements*.
c. Nichelle is **so** much *taller* now *than Beavis* **that people think she's in middle school**.
d.*Nichelle is **so** much *taller* now **that people think she's in middle school** *than Beavis*.

Two things need to be explained about the data of (24)–(26): (1) the fact just mentioned, that comparative complements permute while *so, too* and *enough* complements don't, and (2) the prior fact that some extraposed complements permute with elements that are patently extraposed. We account for the latter fact – the possibility of crossed dependencies like (24a) and (25b) – by postulating two unary lexical constructions. The first 'moves' arguments from the VAL list to the EXTRA list; the second allows nouns to be constructed that have a relative clause on their EXTRA list.[19] A first approximation of these constructions is given in (27) and (28):[20,21]

(27) **Complement Extraposition Construction:**

$$
\textit{comp-extra-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \langle \text{NP} \rangle \oplus L_1 \\ \text{EXTRA} & L_2 \oplus \langle X \rangle \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \left\langle \begin{bmatrix} \textit{word} & \\ \text{SYN} & \begin{bmatrix} \text{VAL} & \langle \text{NP} \rangle \oplus (\text{L}_1 \bigcirc \langle X \rangle) \\ \text{EXTRA} & L_2 \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}
$$

(28) **Nominal Modifier Extraposition Construction:**

$$
\textit{nm-extra-cxt} \Rightarrow \begin{bmatrix} \text{MTR} & \begin{bmatrix} \textit{word} & \\ \text{FORM} & L_0 \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \textit{noun} \\ \text{VAL} & L \\ \text{EXTRA} & \langle [\text{SYN}[\text{CAT}[\text{SEL}\ Z]]] \rangle \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \left\langle Z: \begin{bmatrix} \textit{word} & \\ \text{FORM} & L_0 \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \textit{noun} \\ \text{VAL} & L \\ \text{EXTRA} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}
$$

[19] A relative clause otherwise functions as a nominal modifier, selecting the nominal it modifies via SELECT; see Sag 2010.

[20] In (27), $\bigcirc$ denotes the 'shuffle' relation, as opposed to the 'append' relation ($\oplus$) used in (28) and in (22) and (23) above. For example, if $L_1 = \langle a, b \rangle$ and $L_2 = \langle c \rangle$, then any of the following lists satisfies the description '$L_1 \bigcirc L_2$', while only the list in (ii) satisfies '$L_1 \oplus L_2$':

(i) $\langle c, a, b \rangle$, $\langle a, c, b \rangle$

(ii) $\langle a, b, c \rangle$

In addition, the following lists satisfy neither description, as they fail to respect the original ordering of elements in L_1:

(iii) $\langle c, b, a \rangle$, $\langle b, c, a \rangle$, $\langle b, a, c \rangle$

For further discussion and motivation, see Reape 1994.

[21] (27)–(28) are both postinflectional constructions (Sag this volume) and hence need not stipulate as much information as we include here for expository convenience.

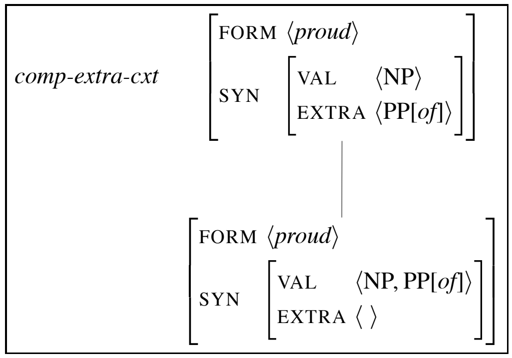


FIGURE 5 A Complement Extraposition Construct

The Complement Extraposition Construction 'pumps' a daughter (one that is a 'predicator') with a nonsubject valent *X* to sanction a predicator (the mother) where *X* appears as the last element of the EXTRA list and is absent from the VAL list. As the final element on the EXTRA list, *X* is the last element on the list to be realized by the Head Extraposition Construction [(23) above] and hence appears in the sentence after any other elements realized from this list. Similarly, the Nominal Modifier Extraposition Construction in (28) pumps a nominal Z with an empty EXTRA list to a nominal whose EXTRA list consists of an element specified as [SYN[CAT[SEL Z]]] – that is, an expression modifying Z.[22] As illustrated in section 4 below, this ensures that the extraposed element functions semantically exactly as it would if it were a modifier adjacent to the nominal.

Because the Head-Extraposition Construction is binary, only one extraposed element is introduced at each level of structure. Hence, multiple extrapositions involve a nested, left-branching tree structure. Multiple extraposition dependencies typically arise when one of these dependencies interacts with an extraposition dependency induced lexically (by *so*, *more*, etc.). A construct of type *comp-extra-cxt* (a postinflectional construct, as already noted) is illustrated in Figure 5, where the daughter's VAL list contains a PP[*of*], and its EXTRA list is empty. The mother's VAL list contains only the subject of *proud* – the PP[*of*] appears on the EXTRA list.

Let us now return to the fact that, unlike other extraposed dependents (such as *than-* or *as*-phrases), *so*, *too* and *enough* dependents never participate in crossed dependencies. We account for this via the listemes shown in (29):

[22] Further conditions must be imposed on Z, whose nature we cannot explore here.

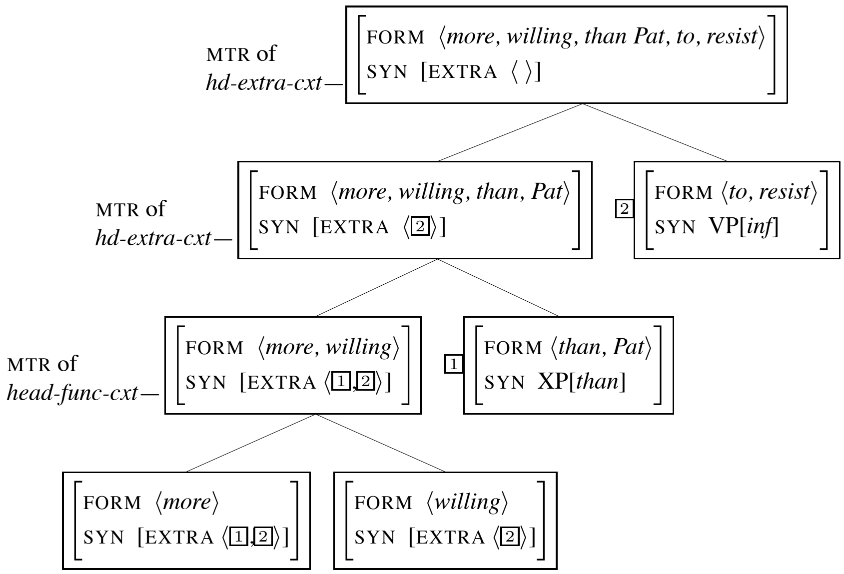


FIGURE 6 A Multiple Extraposition Analysis Tree

(29) a. $\left[\begin{array}{l}\text{FORM } \langle \textit{so} \rangle \\ \text{SYN } \left[\begin{array}{l}\text{CAT } \left[\text{SELECT [SYN [EXTRA } L_1 \text{]]}\right] \\ \text{EXTRA } L_1 \oplus \langle \text{S}[\textit{that}] \rangle\end{array}\right]\end{array}\right]$

b. $\left[\begin{array}{l}\text{FORM } \langle \textit{more} \rangle \\ \text{SYN } \left[\begin{array}{l}\text{CAT } \left[\text{SELECT [SYN [EXTRA } L_1 \text{]]}\right] \\ \text{EXTRA } L_1 \bigcirc \langle \text{XP}[\textit{than}] \rangle\end{array}\right]\end{array}\right]$

We have already seen that *so* adds its S[*that*] dependent at the right end of the EXTRA list, ensuring that it will be realized highest (hence latest, rightmost) in the structure of any element realized from the same list. Note that the entry for *more* is partially similar, but with the important difference that the XP[*than*] dependent is added not at the end, but at an arbitrary position within the selected element's EXTRA list (as specified by the use of $\bigcirc$, rather than $\oplus$). This arrangement allows dependents of comparative expressions to be realized either earlier (hence lower, to the left) or later (hence higher, to the right) than other elements realized from their EXTRA list, as illustrated in Figures 6 and 7. *So*/*too*/*enough* dependents cannot similarly precede other extraposed elements because of the lexical constraint imposed in the SELECT and EXTRA specifications shown in (29a).

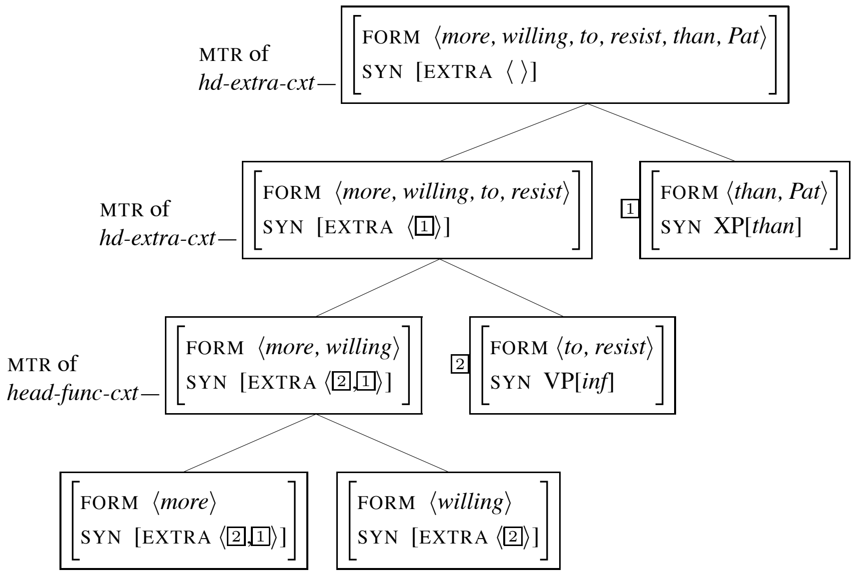


FIGURE 7 Another Multiple Extraposition Analysis Tree

We have seen that *so*/*too*/*enough* dependents must follow comparative dependents if they reside on the same EXTRA list. However, if the comparative licensor is within the subject NP and the *so*/*too*/*enough* licensor is within the VP of a subject-predicate clause, then the *so*/*too*/*enough* dependent must linearly precede the *than*-phrase (extraposition is bounded by the VP):

(30) a. **More** girls were _so_ happy _that they cheered_ **than boys**.
b.***More** girls were _so_ happy **than boys** _that they cheered_.

We account for this interaction by formulating the Subject-Predicate Construction as shown in (31):

(31) **Subject-Predicate Construction:**

$$subj\text{-}pred\text{-}cl \Rightarrow \begin{bmatrix} \textit{headed-cxt} \\ \text{MTR} \begin{bmatrix} \text{SYN} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{VFORM} & \textit{fin} \\ \text{INV} & - \end{bmatrix} \\ \text{VAL} \quad \langle\ \rangle \\ \text{EXTRA} \quad L \end{bmatrix} \end{bmatrix} \\ \text{DTRS} \left\langle Y{:}[\text{EXTRA}\ L\,],\ H{:}\begin{bmatrix} \text{SYN} \begin{bmatrix} \text{VAL} & \langle Y \rangle \\ \text{EXTRA} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

The Subject-Predicate Construction licenses headed constructs with a mother and two daughters. The mother is required to be non-inverted and finite, with an empty VAL list and, crucially in the present context, to have an EXTRA list that is identified with that of the first (subject) daughter. In addition, the subject daughter satisfies the subject valence requirement (*Y*) of the head VP daughter and the EXTRA list of the latter must be empty, ensuring that any extraposed elements that arise within the VP of a Subject-Predicate construct are realized within that VP, via a lower application of the Head Extraposition Construction. Put differently, the VP head daughter of a subject-predicate construct acts as a barrier for extraposition.

Finally, we note that it is not just subject-predicate clauses that inherit the extraposition potential of their first daughter. This is also true of filler-gap constructions (see Sag 2010):

(32) a. [[[How many **more** talents] did she have] **than the other candidate**]?
b. [[[Which **candidates**] do you support] **who have signed the legislation**]?
c. [[[How many **soups**] he had sampled] **that he didn't like**]!
d. [[[**So** eager] was he to see the comet] **that he stayed up all night**].

4 Semantics

Thus far, we have neglected the matter of semantics. In fact, the semantics of DD and CPD are relatively straightforward in SBCG.[23] Note first that a complement is lexically assigned a semantic role in any given verbal lexeme. Since the Complement Extraposition Construction (see (27) above) simply 'moves' the complement from the verb's VAL list to its EXTRA list, it leaves all the semantic connections intact. Thus the semantics of both (33a) and (33b) is that shown in (34),[24] which is rendered in (35) with generalized quantifiers in a Davidsonian predicate logic:[25]

(33) a. Kim [affirms [that Pak left] [to Judge Lee]].
b. Kim [[affirms [to Judge Lee]] [that Pak left]].

[23]See Sag this volume for explantion of the Minimal Recursion Semantics (MRS) used here, which is based on Copestake et al. 2005.

[24]This is not quite true, as the frames of the extraposed clause appear later on the FRAMES list than they do in the analogous non-extraposed sentence. But order of elements on FRAMES lists (better regarded as multisets) has no semantic significance. Hence the meanings assigned to (33a) and (33b) are identical.

[25]Note that 's'-variables range over situations (essentially Davidsonian events) and 'l'-variables range over feature structure labels. For explanation of the treatment of tense in terms of situation arguments, see Sag this volume. Present-tense situation variables are left unbound here, though nothing hinges on this choice.

(34)
$$\begin{bmatrix} \text{LTOP} & l_0 \\ \text{FRAMES} & \left\langle \begin{bmatrix} \textit{affirming-fr} & \\ \text{LBL} & l_0 \\ \text{SIT} & s_0 \\ \text{SPEAKER} & \textit{Kim} \\ \text{ADDRESSEE} & \textit{Judge-Lee} \\ \text{MESSAGE} & l_1 \end{bmatrix}, \begin{bmatrix} \textit{exist-fr} & \\ \text{LBL} & l_1 \\ \text{BV} & s_1 \\ \text{RESTR} & l_2 \\ \text{SCOPE} & l_3 \end{bmatrix}, \begin{bmatrix} \textit{past-fr} & \\ \text{LBL} & l_2 \\ \text{ARG} & s_1 \end{bmatrix}, \begin{bmatrix} \textit{leaving-fr} & \\ \text{LBL} & l_3 \\ \text{SIT} & s_1 \\ \text{THEME} & \textit{Pak} \end{bmatrix} \right\rangle \end{bmatrix}$$

(35) $\textit{affirm}(s_0, \textit{Kim}, \textit{Judge-Lee}, exist(s_1, past(s_1), leave(s_1, Pak)))$

Sentences containing extraposed nominal modifiers, allowed in virtue of the Nominal Modifier Extraposition Construction in (28) above, are also assigned a semantics that is identical to that of their unextraposed counterparts, where the relative clause combines with the NP via the Head-Functor Construction, as in (36a):

(36) a. [[the **food**] **that Pat loves**] arrived.

b. [[the **food** arrived] **that Pat loves**].

This follows because the Head-Functor Construction, which attaches the relative clause to the head noun, applies lower in the analysis tree than the constructions licensing extraposition of the relative clause. The modification relation thus established is unaffected by the extraposition process. Hence, when the modifier is in extraposed position (as in (36b)), its SELECT value is identified with the same NP (*the food*) that the modifier's SELECT value is identified with in the head-functor construct (as in (36a)). This guarantees that both (36a) and (36b) have the semantics shown in (37) (again, modulo order of elements on the FRAMES list):

(37)
$$\begin{bmatrix} \text{LTOP} & l_0 \\ \text{FRAMES} & \left\langle \begin{bmatrix} \textit{the-fr} & \\ \text{LBL} & l_0 \\ \text{BV} & x \\ \text{RESTR} & l_1 \\ \text{SCOPE} & l_3 \end{bmatrix}, \begin{bmatrix} \textit{food-fr} & \\ \text{LBL} & l_1 \\ \text{ENTITY} & x \end{bmatrix}, \begin{bmatrix} \textit{love-fr} & \\ \text{LBL} & l_1 \\ \text{SIT} & s_3 \\ \text{EXPERIENCER} & \textit{Pat} \\ \text{CONTENT} & x \end{bmatrix}, \begin{bmatrix} \textit{exist-fr} & \\ \text{LBL} & l_3 \\ \text{BV} & s_1 \\ \text{RESTR} & l_4 \\ \text{SCOPE} & l_5 \end{bmatrix}, \begin{bmatrix} \textit{past-fr} & \\ \text{LBL} & l_4 \\ \text{ARG} & s_1 \end{bmatrix}, \begin{bmatrix} \textit{arriving-fr} & \\ \text{LBL} & l_5 \\ \text{SIT} & s_1 \\ \text{THEME} & x \end{bmatrix} \right\rangle \end{bmatrix}$$

(38) shows the equivalent expression in a Davidsonian predicate logic:

(38) $the(x, food(x) \,\&\, love(s_3, Pat, x), exist(s_1, past(s_1), arrive(s_1, x)))$

The equivalence of extraposed and nonextraposed counterparts holds for all the examples of DD discussed in this paper, by similar reasoning. The examples with *so*, *such*, *too*, and the like may be treated in terms of a generalized quantifier, e.g. *so-fr*, that binds the degree variable associated with the expression that *so* modifies, whose restriction is the semantics of that modified expression, and whose scope (the ENABLED-SITUATION in FrameNet terms) is determined by the broader sentential context.

The meaning of a *so*-sentence can thus be represented as in (39):

(39) Max is so clean that he squeaks.

(40) $so(\delta, clean(s_0, Max, \delta), squeak(s_1, he))$

We assume the *so-fr* quantifier is existential in nature. Thus (40) is to be understood roughly as follows:

(41) There is a degree δ such that Max is δ clean and that fact has as a consequence that he squeaks.

The listeme for *so* can now be formulated as in (42):[26]

[26] Of course some of the constraints in (42) may not be particular to *so*, but rather to a lexical class to which *so* belongs. In this case, the listeme for *so* could be simplified, moving a share of the work to a lexical class construction.

(42)

$$
\begin{bmatrix}
\text{FORM} & \langle \textit{so} \rangle \\
\text{SYN} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{SELECT} & \begin{bmatrix} \text{SYN} & \begin{bmatrix} \text{CAT} & [\text{LID}\ \langle[\text{DEGREE}\ \delta]\rangle] \\ \text{EXTRA} & L_1 \end{bmatrix} \\ \text{SEM} & [\text{LTOP}\ l_1] \end{bmatrix} \end{bmatrix} \\ \text{EXTRA} & L_1 \oplus \left\langle \begin{bmatrix} \text{SYN} & [\text{MRKG}\ \textit{that}] \\ \text{SEM} & [\text{LTOP}\ l_2] \end{bmatrix} \right\rangle \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{LTOP} & l_0 \\ \text{FRAMES} & \left\langle \begin{bmatrix} \textit{so-fr} & \\ \text{LBL} & l_0 \\ \text{BV} & \delta \\ \text{RESTR} & l_1 \\ \text{SCOPE} & l_2 \end{bmatrix} \right\rangle \end{bmatrix}
\end{bmatrix}
$$

This listeme fixes the degree argument of the LID of the selected phrase as the bound variable (BV value) of the *so-frame*,[27] the LTOP of the selected phrase as its restriction (RESTR value), and the LTOP of the extraposed *that*-clause as its SCOPE value. These identities thus establish the basic connections essential to the meaning of a *so*-sentence and do so in a manner that is independent of the order in which these elements are realized. In particular, sentences like (43a-b) will be assigned identical meanings, modulo order of elements on the FRAMES list:

(43) a. Bo [[[was [**so** tired]] **that he fainted**] yesterday].
b. Bo [[[was [**so** tired]] yesterday] **that he fainted**].

The semantic composition of the relevant analysis trees are sketched in Figures 8 and 9.

In the case of multiple extraposed clauses, there is also no effect on the semantics. Pairs like (9c-d) above, repeated here as (45a-b), receive identical semantic treatment (modulo order of frames):

(44) a. I [[[sent out [**more** *books*] yesterday] **than ever before**] *that I really liked*].
b. I [[[sent out [**more** *books*] yesterday] *that I really liked*] **than ever before**].

[27] An LID value contains the main frame of the sign it occurs in and so contains all the arguments of that frame. We assume that *so* selects a gradable adjective, which contains an unbound degree variable in its semantics.

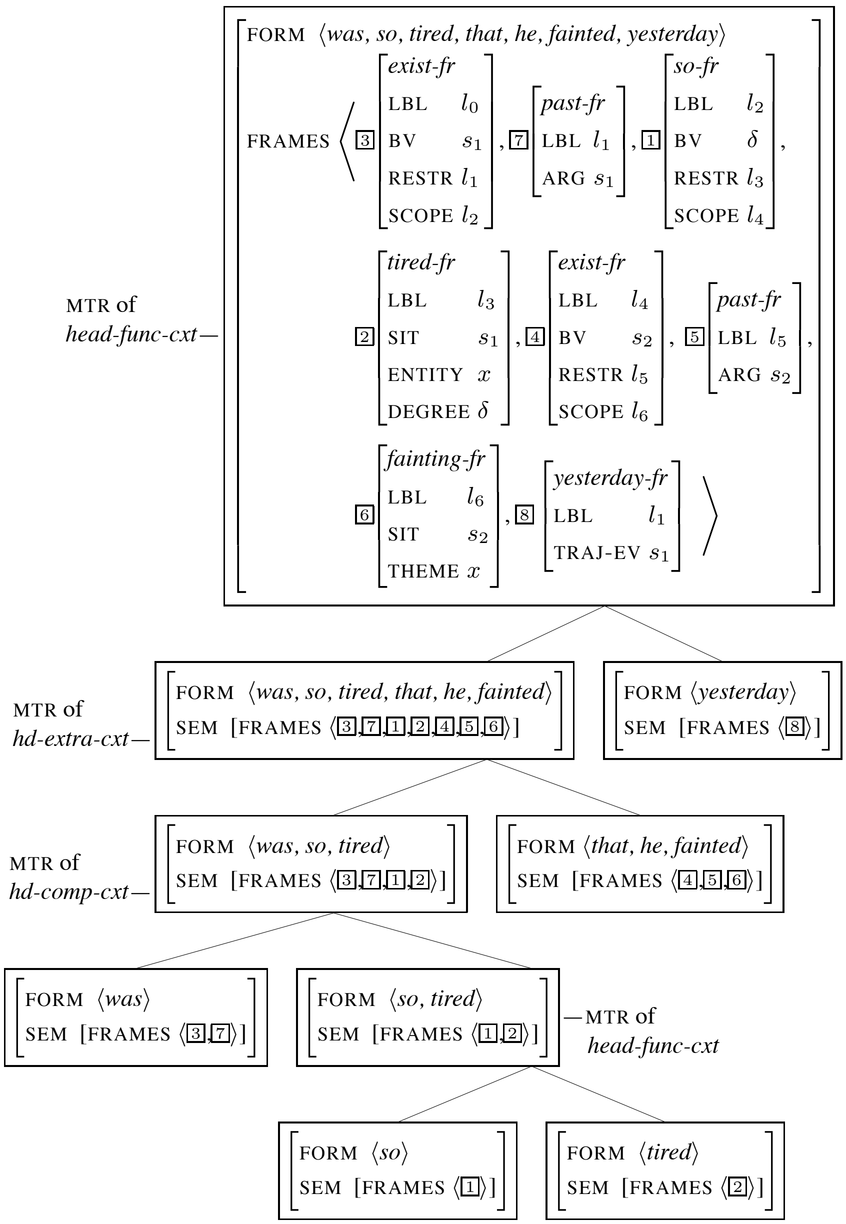


FIGURE 8 Semantic Analysis Tree: Adjunction Over Extraposition

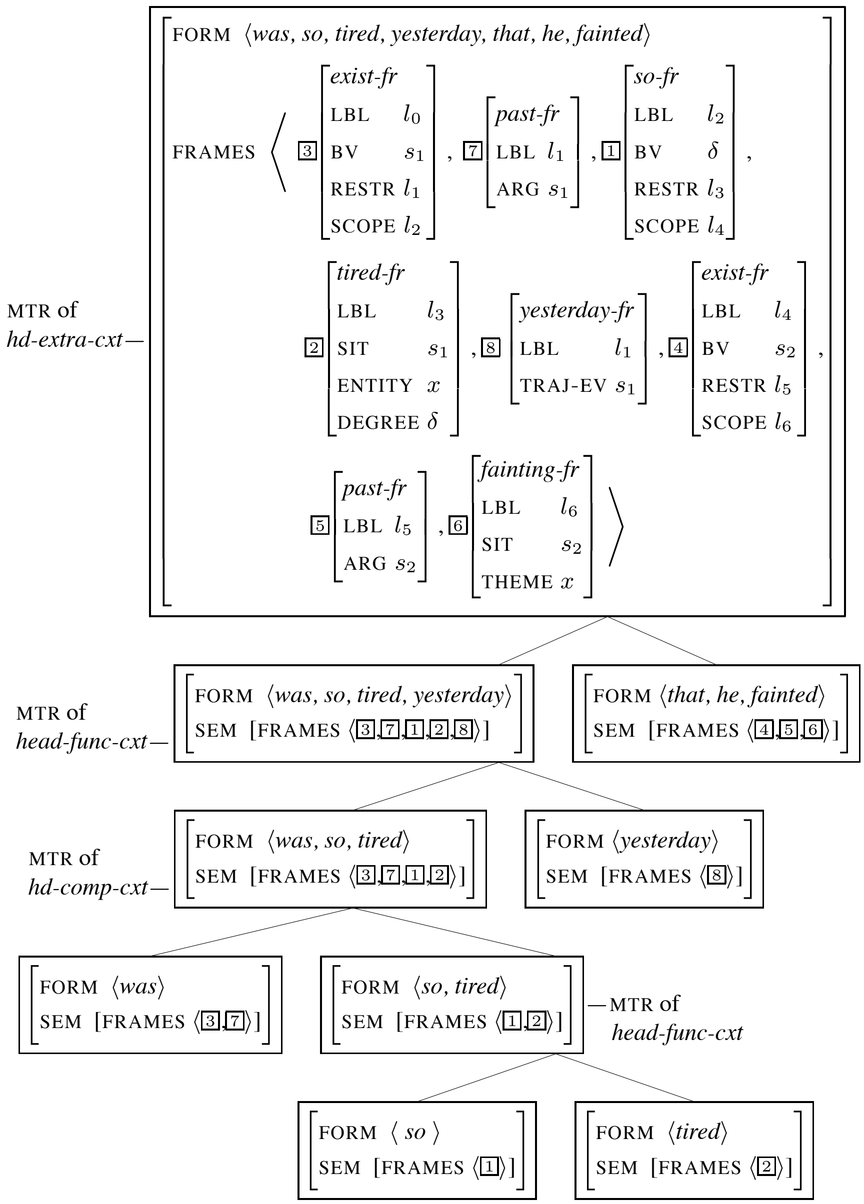


FIGURE 9 Semantic Analysis Tree: Extraposition Over Adjunction

The treatment of comparatives is similar to that of *so*-sentences, though these introduce further issues of binding and ellipsis that are beyond the scope of this paper.

Finally, the variation in linear order introduced by the Complex Pre-Determiner Construction also has no semantic effect. For example, (45) is assigned the semantics in (46):

(45) [[[[**so** tall] [a man]] arrived] **that Lee laughed**].

(46) $so(\delta, exist(x, man(x)\ \&\ tall(x,\delta)\ \&\ exist(s_2, past(s_2), arrive(s_2, x))),$
$exist(s_1, past(s_1), laugh(s_1, Lee)))$

5 Conclusion

In this paper, we have seen that the complex pre-determination ('Big Mess') phenomenon and the discontinuous dependency phenomenon are independent – either may occur in a sentence without the other. Nevertheless we find them frequently intertwined because there are seven listemes (*so*, *too*, *more*, *less*, *as*, *such*, and *how*) that contain features which play key roles in both constructions. The CPD phenomenon requires a special construction (in our analysis or the alternative suggested in Van Eynde 2007); the DD phenomenon follows from the properties of two distinct classes of lexical licensors and the grammatical mechanisms that govern extraposition in general. The details of the distribution of DD complements derive from the interaction of (1) a general construction for realizing elements of the EXTRA list, (2) specifications on phrasal constructions determining the contents of the mother's EXTRA list as a function of the EXTRA lists of the daughters, and (3) the constraints specified in the listemes for relevant lexical licensors.

The subtleties of DD, CPD and their interaction have led us to an analysis that embodies one of the fundamental tenets of Berkeley Construction Grammar:

> One cannot analyze an idiomatic construction without simultaneously discovering and setting aside all the aspects of the data that are NOT licensed by the construction one is studying. To know what is idiomatic about a phrase one has to know what is nongeneral and to identify something as nongeneral one has to be able to identify the general. In grammar, the investigation of the idiomatic and of the general are the same; the study of the periphery is the study of the core-and vice versa. The picture that emerges from the consideration of special constructions ... is of a grammar in which the particular and the general are knit together seamlessly (Kay and Fillmore 1999).

We believe that the general approach we have adopted here has allowed us to develop a precise analysis of these phenomena that abstracts the significant generalizations they present, to elucidate their interactions with other aspects of grammar, and to thereby explicate the interaction of the idiosyncratic, the general, and the extensive and finely graded area in between.

References

Allegranza, Valerio. 1998b. Determination and Quantification. In *Linguistic Specifications for Typed Feature Structure Formalisms*, vol. 10 of *Studies in Machine Translation and Natural Language Processing*, pages 281–314. Luxembourg: Official Publications of the European Communities.

Allegranza, Valerio. 2007. *The Signs of Determination. Constraint-Based Modelling Across Languages*, vol. 16 of *SABEST: Saarbrücker Beiträge zur Sprach- und Translationswissenschaft*. Frankfurt am Main: Peter Lang.

Arnold, Doug and Louise Sadler. 1992. Non-Modifying Adjectives in HPSG. In *Working Papers in Language Processing*, vol. 35. Dept. of Languages and Linguistics, U. Essex.

Baltin, Mark R. 1982. A Landing Site Theory of Movement Rules. *Linguistic Inquiry* 13:1–38.

Berman, Arlene. 1974. *Adjectives and adjective complement constructions in English*. Ph.D. thesis, Harvard University.

Bouma, Gosse. 1996. Extraposition as a Nonlocal Dependency. In *Proceedings of Formal Grammar 96*, pages 1–14. Prague.

Bouma, Gosse, Rob Malouf, and Ivan A. Sag. 2001. Satisfying Constraints on Extraction and Adjunction. *Natural Language and Linguistic Theory* 1(19):1–65.

Chae, Hee-Rahk. 1992. *Lexically Triggered Unbounded Discontinuities in English: An Indexed Phrase Structure Grammar Approach*. Ph.D. thesis, The Ohio State University.

Copestake, Ann, Dan Flickinger, Carl J. Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation* 3(4):281–332.

Crysmann, Berthold. To appear. On the Locality of Complement Clause and Relative Clause Extraposition. In G. Webelhuth, M. Sailer, and H. Walker, eds., *Rightward Movement in a Comparative Perspective*. Amsterdam: John Benjamins.

Flickinger, Daniel P. and John Nerbonne. 1992. Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns. *Computational Linguistics* 19(3):269–309. Walter Dalemans and Gerald Gazdar, eds., *Inheritance and Natural Language Processing*, special issue.

Gazdar, Gerald. 1981. Unbounded Dependencies and Coordinate Structure. *Linguistic Inquiry* 12:155–84.

Ginzburg, Jonathan and Ivan A. Sag. 2000. *Interrogative Investigations: The form, meaning, and use of English interrogatives*. Stanford: CSLI Publications.

Guéron, Jacqueline and Robert May. 1984. Extraposition and Logical Form. *Linguistic Inquiry* 15(1):1–31.

Huddleston, Rodney D. and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge: Cambridge University Press.

Kay, Paul and Charles Fillmore. 1999. Grammatical Constructions and Linguistic Generalizations: The *What's X Doing Y?* Construction. *Language* 75.1:1–33.

Kay, Paul and Ivan A. Sag. 2012. A Lexical Theory of Phrasal Idioms. Unpublished manuscript, Stanford University.

Keller, Frank. 1995. Towards an Account of Extraposition in HPSG. In *Proceedings* *of the 7th Conference of the EACL*, pages 301–306. Dublin. student session.

Kennedy, Christopher and Jason Merchant. 2000. Attributive comparative deletion. *Natural Language and Linguistic Theory* 18:89–146.

Kim, Jong-Bok and Ivan A. Sag. 2005. Variations in English Object Extraposition. In *The Proceedings of the 41st Annual Meeting of the Chicago Linguistic Society*, pages 192–212. Chicago: CLS.

Kiss, Tibor. 2005. Semantic Constraints on Relative Clause Extraposition. *Natural Language and Linguistic Theory* 23(2):281–334.

Levine, Robert D. and Thomas Hukari. 2006. *The Unity of Unbounded Dependency Constructions, no. 166*. CSLI Lecture Notes. Stanford: CSLI Publications.

Müller, Stefan. 2004. Continuous or Discontinuous Constituents? A Comparison between Syntactic Analyses for Constituent Order and Their Processing Systems. *Research on Language and Computation* 2(2):209–257. Special Issue on Linguistic Theory and Grammar Implementation.

Müller, Stefan. 2009. On Predication. In S. Müller, ed., *The Proceedings of the 16th International Conference on Head-Driven Phrase Structure Grammar*. Stanford: CSLI Publications.

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

Reape, Mike. 1994. Domain Union and Word Order Variation in German. In J. Nerbonne, K. Netter, and C. J. Pollard, eds., *German in Head-Driven Phrase Structure Grammar*, no. 46 in *CSLI Lecture Notes*, pages 151–197. Stanford: CSLI Publications.

Sag, Ivan A. 2010. English Filler-Gap Constructions. *Language* 86:486–545.

Sag, Ivan A. This volume. Sign-Based Construction Grammar: An informal synopsis.

Stucky, Susan. 1987. Configurational Variation in English: A Study of Extraposition and Related Matters. In G. Huck and A. Ojeda, eds., *Discontinuous Constituency*, no. 20 in Syntax and Semantics, pages 337–405. New York: Academic Press.

Van Eynde, Frank. 1996. An HPSG Treatment of *It*-Extraposition Without Lexical Rules. In G. Durieux, W. Daelemans, and S. Gillis, eds., *CLIN VI. Papers from the Sixth CLIN Meeting*, pages 231–248. University of Antwerp, Center for Dutch Language and Speech.

Van Eynde, Frank. 1998. The Immediate Dominance Schemata of HPSG: A deconstruction and a reconstruction. In P. Coppen, H. van Halteren, and L. Teunissen, eds., *Computational Linguistics in the Netherlands 1997. Selected Papers from the Eighth CLIN Meeting*, pages 119–133. Amsterdam/Atlanta: Rodopi.

Van Eynde, Frank. 2006. NP-Internal Agreement and the Structure of the Noun Phrase. *Journal of Linguistics* 42(1):139–186.

Van Eynde, Frank. 2007. The Big Mess Construction. In S. Müller, ed., *The Proceedings of the 14th International Conference on Head-Driven Phrase Structure Grammar, Stanford University*, pages 415–433. Stanford: CSLI Publications.

Williams, Edwin S. 1974. *Rule Ordering in Syntax*. Dissertation, MIT.

Wittenburg, Kent. 1987. Extraposition from NP as Anaphora. In G. Huck and A. Ojeda, eds., *Discontinuous Constituency*, no. 20 in Syntax and Semantics, pages 428–445. New York: Academic Press.