Chapter 2

Background: Construction-HPSG

This chapter presents the theoretical background necessary to develop and evaluate the syntactic analyses of AAVE copula absence in Chapter 3. I will be assuming a theory that might be called 'Construction-HPSG', roughly Head-Driven Phrase Structure Grammar as presented in Pollard and Sag 1994, augmented with a theory of constructions, as in Sag 1997 and Ginzburg and Sag to appear, as well as the theory of lexical alternations developed in Koenig 1999. This theory is a particularly interesting one in which to compare the various analyses because of its surface-orientation.

Sections 2.1–2.2 give a brief introduction to HPSG. Section 2.3 reviews the analyses in HPSG and related frameworks that have shown that phonologically empty linguistic categories are not required for a range of linguistic phenomena. This section concludes with a brief overview of surface-orientation as a design feature of grammar. Section 2.4 introduces constructions, and Section 2.5 presents a theory of the properties of English auxiliaries.

2.1 The basics: types, constraints, and the core grammar

HPSG conceptualizes language as a system of signs, or pairings of (phonological and syntactic) form with (semantic and pragmatic) meaning. The grammar is merely a static description of what signs are in the language, along with a basic mechanism for combining the signs.

More formally, an HPSG consists of a set of types, arranged into a multiple inheritance hierarchy, a set of constraints on those types, and an initial symbol. Each type is a (partial) description of a kind of linguistic object, such as a word or a phrase.[1] These types can be combined according the feature logic to create descriptions of larger linguistic objects. Any combination of types that satisfies the initial symbol is a description of a grammatical sentence of the language.

To make this concrete, let us consider a simple example. The analysis of (1) requires three word types (for *Kim, likes* and *bagels*) and two phrasal types (one to combine *likes* and *bagels* and one to combine *Kim* and *likes bagels*).

(1) Kim likes bagels.

Consider first the (partial) description of *likes* given in Figure 2.1.[2] This description is formalized as an attribute-value matrix (avm). The first attribute (or feature) is PHON(onology) and its value is a phonological representation of the sign. For ease of exposition, I will systematically use English orthography as a place holder for a more adequate phonological representation. Thus, the value of the feature PHON in Figure 2.1 is the English orthography 'likes'. The next feature is SYNSEM, specifying the syntactic and semantic properties of the word. Inside SYNSEM, the feature HEAD gives the part of speech (in this case, *verb*), and the further information that this verb is finite. The values of the next three features (SUBJ, SPR, and COMPS) specify that this verb is looking for one complement and a subject, but no specifier.[3] The boxed numbers (tags) in the values of SUBJ and COMPS indicate structure-sharing with the value of another feature, the argument structure (ARG-ST). The argument structure

[1] As we will see below, not all types are descriptions of linguistic units in this sense.

[2] This description and those that follow are partial in two respects: First, for ease of exposition, I am abstracting away from such issues as quantification, pied-piping, and extraction (although the latter will be addressed below) and therefore the avms shown lack certain features posited for those analyses. On quantification, see Copestake et al. 1999. On pied-piping, see Sag 1997 and Ginzburg and Sag to appear. Second, and more technically, the linguistic descriptions found in a grammar are usually underspecified to some extent with respect to the fully specified models that satisfy them. See Pollard and Sag 1994 for discussion.

[3] These are list-valued features, as indicated by the angle brackets. $\langle\ \rangle$ indicates the empty list.

$$
\begin{bmatrix}
\textit{word} & \\
\textsc{phon} & \langle \text{likes} \rangle \\
\textsc{synsem} &
\begin{bmatrix}
\textit{synsem} & \\
\textsc{local} &
\begin{bmatrix}
\textit{local} & \\
\textsc{cat} &
\begin{bmatrix}
\textit{category} & \\
\textsc{head} & \begin{bmatrix} \textit{verb} & \\ \textsc{form} & \textit{fin} \end{bmatrix} \\
\textsc{subj} & \left\langle \boxed{1}
\begin{bmatrix}
\textit{synsem} & \\
\textsc{local} &
\begin{bmatrix}
\textit{local} & \\
\textsc{cat} &
\begin{bmatrix}
\textit{category} & \\
\textsc{head} & \begin{bmatrix} \textit{noun} & \\ \textsc{agr} & \textit{3sg} \end{bmatrix} \\
\textsc{subj} & \langle\,\rangle \\
\textsc{spr} & \langle\,\rangle \\
\textsc{comps} & \langle\,\rangle
\end{bmatrix} \\
\textsc{cont} & \begin{bmatrix} \textit{content} & \\ \textsc{index} & \boxed{4} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\right\rangle \\
\textsc{spr} & \langle\,\rangle \\
\textsc{comps} & \left\langle \boxed{2}
\begin{bmatrix}
\textit{synsem} & \\
\textsc{local} &
\begin{bmatrix}
\textit{local} & \\
\textsc{cat} &
\begin{bmatrix}
\textit{category} & \\
\textsc{head} & \textit{noun} \\
\textsc{subj} & \langle\,\rangle \\
\textsc{spr} & \langle\,\rangle \\
\textsc{comps} & \langle\,\rangle
\end{bmatrix} \\
\textsc{cont} & \begin{bmatrix} \textit{content} & \\ \textsc{index} & \boxed{5} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\right\rangle \\
\textsc{arg-st} & \left\langle \boxed{1}, \boxed{2} \right\rangle
\end{bmatrix} \\
\textsc{cont} &
\begin{bmatrix}
\textit{content} & \\
\textsc{index} & \boxed{3} \\
\textsc{key} & \boxed{6} \\
\textsc{rels} & \left\langle \boxed{6}
\begin{bmatrix} \textit{like\_rel} & \\ \textsc{event} & \boxed{3} \\ \textsc{arg1} & \boxed{4} \\ \textsc{arg2} & \boxed{5} \end{bmatrix},
\begin{bmatrix} \textit{t-overlap\_rel} & \\ \textsc{arg1} & \boxed{3} \\ \textsc{arg2} & \textit{now} \end{bmatrix}
\right\rangle
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Figure 2.1: Lexical entry for *likes*

is a list of all of the dependents selected by this head, ordered in terms of obliqueness.[4] The substructure labelled $\boxed{1}$ (i.e., that which is also on the SUBJ list) specifies that the first syntactic argument of *likes* is a third singular noun phrase with the index $\boxed{4}$.[5] Here the term 'noun phrase' is shorthand for the specifications [HEAD *noun*] and [SUBJ $\langle\ \rangle$, SPR $\langle\ \rangle$, COMPS $\langle\ \rangle$]. That is, noun phrases are phrases headed by nouns with all of their valence requirements satisfied. Accordingly, the argument structure in Figure 2.1 is often abbreviated as in Figure 2.2.

Returning to the *synsem* of *likes*, the CONT(ent) value gives the semantics, in the style of Minimal Recursion Semantics (Copestake et al. 1999). Ignoring such issues as quantification, the semantics consists of two *relation*s, the *like_rel* and the *t(emporal)-overlap_rel*. The last of these is a representation of present tense,[6] stating that the situation described by the *like_rel* ($\boxed{3}$) overlaps with the present moment. The *like_rel* itself is a binary relation, whose two arguments ($\boxed{4}$ and $\boxed{5}$) are identified with the semantic indices of the two syntactic arguments. In the abbreviated notation shown in Figure 2.2, the tags corresponding to the semantic indices are given as subscripts to the arguments. The *content* value also has a feature KEY. This feature serves as a pointer to the main relation contributed by the word. Therefore its value is identified with one of the relations on the RELS list (here, $\boxed{6}$, the *like_rel*).

The label in italics at the top left of each pair of square brackets in Figure 2.1 gives the type of that substructure. The definition of each type includes which features are relevant to it (e.g., for the type *word*, the relevant features are PHON, SYNSEM and ARG-ST) as well as the type(s) of permissible values of each feature. For example, the value of SYNSEM is always of type *synsem* or one of its subtypes. When the value of a feature in a given description is the most general one possible for that feature, it is usually not indicated, as in Figure 2.2. As noted above, features can also take lists

[4]Argument structure is the locus of the binding theory, among other things. See Manning 1996 and Manning and Sag 1998 for the justification of Argument Structure as a level independent of both semantics and the valence features SUBJ/SPR/COMPS.

[5]Following Copestake et al. (1999), I am adopting a feature INDEX inside CONT. This feature is identified with the EVENT variable of the KEY for verbs and with the INST variable of the KEY for nouns. This feature is necessary because selecting heads need access to these variables in order to incorporate them properly into their own semantic relations.

[6]Others are possible.

as their values. Although this is not explicitly indicated in Figure 2.1, the lists are always lists of some particular type of object. For example, the value of the features SUBJ, SPR, COMPS and ARG-ST is always a list of *synsems*, and the value of RELS is always a list of *relations*.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{likes} \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \textit{fin} \end{bmatrix} \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle \boxed{2} \rangle \\ \text{ARG-ST} & \left\langle \boxed{1}\text{NP}[\textit{3sg}]_{\boxed{4}}, \boxed{2}\text{NP}_{\boxed{5}} \right\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{3} \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{like\_rel} & \\ \text{EVENT} & \boxed{3} \\ \text{ARG1} & \boxed{4} \\ \text{ARG2} & \boxed{5} \end{bmatrix}, \begin{bmatrix} \textit{t-overlap\_rel} & \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \textit{now} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 2.2: Abbreviated lexical entry for *likes*

The lexical entries for *Kim* and *bagels* are given in Figures 2.3 and 2.4. Note that *Kim* has no valence requirements (SUBJ/SPR/COMPS are empty), and *bagels* has only an optional specifier (indicated by the parentheses around $\boxed{3}$ in its SPR value).

Now, the subcategorization information specified for *likes* is inert without phrase types to recognize it and pair it with nouns in the syntax. The first of these phrase types is the *head-complements-phrase*, shown in Figure 2.5. In HPSG, phrases, like words, are pairings of form and meaning. Accordingly, they have the features PHON and SYNSEM, just like words. Phrases differ from words, however, in that they encode constituent structure, via the HEAD-DTR and NON-HEAD-DTRS features.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{ Kim } \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{3sg} \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{ARG-ST} & \langle\ \rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{1} \\ \text{KEY} & \boxed{2} \\ \text{RELS} & \left\langle \boxed{2} \begin{bmatrix} \textit{named\_rel} & \\ \text{INST} & \boxed{1} \\ \text{ARG} & \text{Kim} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 2.3: Abbreviated lexical entry for *Kim*

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{ bagels } \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{AGR} & \textit{pl} \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \langle\ (\boxed{3})\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{ARG-ST} & \langle\ (\boxed{3}\text{DetP})\ \rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{1} \\ \text{KEY} & \boxed{2} \\ \text{RELS} & \left\langle \boxed{2} \begin{bmatrix} \textit{bagel\_rel} & \\ \text{INST} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 2.4: Abbreviated lexical entry for *bagels*

$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{head-comps-ph}} \\
\text{PHON} & \boxed{\text{C}} \oplus \boxed{\text{D}} \oplus \ldots \oplus \boxed{\text{N}} \\
\text{SYNSEM} & \left[\begin{array}{ll} \text{LOCAL} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \boxed{1} \\ \text{SUBJ} & \boxed{\text{A}} \\ \text{SPR} & \boxed{\text{B}} \\ \text{COMPS} & \langle\ \rangle \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{INDEX} & \boxed{2} \\ \text{KEY} & \boxed{3} \\ \text{RELS} & \boxed{\text{F}} \oplus \boxed{\text{M}} \oplus \ldots \oplus \boxed{\text{Z}} \end{array}\right] \end{array}\right] \end{array}\right] \\
\text{HEAD-DTR} & \left[\begin{array}{ll} \text{PHON} & \boxed{\text{C}} \\ \text{SYNSEM} & \left[\begin{array}{ll} \text{LOCAL} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \boxed{1} \\ \text{SUBJ} & \boxed{\text{A}} \\ \text{SPR} & \boxed{\text{B}} \\ \text{COMPS} & \textbf{sts}(\boxed{\text{E}}) \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{INDEX} & \boxed{2} \\ \text{KEY} & \boxed{3} \\ \text{RELS} & \boxed{\text{F}} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right] \\
\text{NON-HEAD-DTRS} & \boxed{\text{E}} \left\langle \left[\begin{array}{ll} \text{PHON} & \boxed{\text{D}} \\ \text{SYNSEM} & \left[\ \ldots\ \text{RELS}\ \boxed{\text{M}} \right] \end{array}\right], \ldots \left[\begin{array}{ll} \text{PHON} & \boxed{\text{N}} \\ \text{SYNSEM} & \left[\ \ldots\ \text{RELS}\ \boxed{\text{Z}} \right] \end{array}\right] \right\rangle
\end{array}\right]
$$

Figure 2.5: Head-complements phrase (abbreviated)

In Figure 2.5, the syntactic properties of the mother (that is, the value of SYNSEM | LOCAL | CAT) are systematically related to those of the head daughter. In particular, the HEAD, SUBJ and SPR values are shared between the two.[7] Only the COMPS value differs, as the mother is specified as [COMPS $\langle\ \rangle$]. This signifies that all of the complement requirements of the head have been realized within the phrase. In particular, they have been realized because each *synsem* that the head selects for (each *synsem* on its COMPS list) is identified with the SYNSEM value of one of the non-head daughters. (Figure 2.5 makes use of the 'sts' notation. The function **sts** (for 'signs-to-synsems') takes a list of *sign*s as its argument and returns the corresponding list of *synsem*s.) Whatever other valence requirements (SUBJ or SPR) there may be are not satisfied within this phrase, but 'passed up' to the next level. This is how subcategorization requirements are enforced.

The phonology and semantics of the phrase are constructed from the phonology and semantics of all of the daughters. In particular, the PHON value of the phrase is PHON value of the head daughter ($\boxed{C}$) followed by the PHON value of each of the non-head daughters ($\boxed{D}$ through $\boxed{N}$).[8,9] Similarly, the RELS value, or collection of semantic relations, of the mother is the append of the RELS value of the head daughter ($\boxed{F}$) to those of the non-head daughters ($\boxed{M}$ through $\boxed{Z}$). The INDEX and KEY values are identified with those of the head daughter.

Note that this formulation of the head-complements phrase is completely general: complement-taking words of any category can be the head daughter, and any number of selected complements can be accommodated. In the case of (1), the head daughter is the word *likes*, which selects for only one complement. Unifying the description of *likes* from Figure 2.1 with the value of HEAD-DTR in Figure 2.5 gives the description in Figure 2.6.

Since the description of *bagels* unifies with (that is, does not conflict with) the one non-head daughter called for by Figure 2.6, it can be combined with that structure

[7]Boxed tags with capital letters (e.g., $\boxed{A}$) are when the identified values are lists.

[8]Here and in what follows, the symbol $\oplus$ represents the 'append' relation:

$$\langle\ a, \ldots, b\ \rangle \oplus \langle\ c, \ldots, d\ \rangle = \langle\ a, \ldots, b, c, \ldots, d\ \rangle$$

[9]Other relationships between the PHON features are possible.

$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{head-comps-ph}} \\
\textsc{phon} & \boxed{\text{C}} \oplus \boxed{\text{D}} \\
\textsc{synsem} & \left[\begin{array}{ll}
\textsc{local} & \left[\begin{array}{ll}
\textsc{cat} & \left[\begin{array}{ll}
\textsc{head} & \boxed{1} \\
\textsc{subj} & \boxed{\text{A}} \\
\textsc{spr} & \boxed{\text{B}} \\
\textsc{comps} & \langle\ \rangle
\end{array}\right] \\
\textsc{cont} & \left[\begin{array}{ll}
\textsc{index} & \boxed{2} \\
\textsc{key} & \boxed{3} \\
\textsc{rels} & \boxed{\text{E}} \oplus \boxed{\text{F}}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\textsc{head-dtr} & \left[\begin{array}{ll}
\textsc{phon} & \boxed{\text{A}}\langle \text{ likes } \rangle \\
\textsc{synsem} & \left[\begin{array}{ll}
\textsc{local} & \left[\begin{array}{ll}
\textsc{cat} & \left[\begin{array}{ll}
\textsc{head} & \boxed{1}\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{verb}} \\ \textsc{form} & \textit{fin} \end{array}\right] \\
\textsc{subj} & \boxed{\text{A}}\langle \text{ NP}[\textit{3sg}]_{\boxed{4}} \rangle \\
\textsc{spr} & \boxed{\text{B}}\langle\ \rangle \\
\textsc{comps} & \langle \boxed{6}\text{NP}_{\boxed{5}} \rangle
\end{array}\right] \\
\textsc{cont} & \left[\begin{array}{ll}
\textsc{index} & \boxed{2} \\
\textsc{key} & \boxed{3} \\
\textsc{rels} & \boxed{\text{E}}\left\langle \begin{array}{l}
\boxed{3}\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{like\_rel}} \\ \textsc{event} & \boxed{2} \\ \textsc{arg}1 & \boxed{4} \\ \textsc{arg}2 & \boxed{5} \end{array}\right], \\
\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{t-overlap\_rel}} \\ \textsc{arg}1 & \boxed{2} \\ \textsc{arg}2 & \textit{now} \end{array}\right]
\end{array}\right\rangle
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\textsc{non-head-dtrs} & \left\langle \left[\begin{array}{ll}
\textsc{phon} & \boxed{\text{D}} \\
\textsc{synsem} & \boxed{6}\left[\textsc{local} \mid \textsc{cont} \mid \textsc{rels}\ \boxed{\text{F}}\right]
\end{array}\right] \right\rangle
\end{array}\right]
$$

Figure 2.6: Head-complements phrase headed by *likes*

in a similar fashion. The PHON and SYNSEM values of the resulting description are shown in Figure 2.7.[10] (To keep this figure small enough to handle, the daughters features are suppressed.)

$$\begin{bmatrix} \textit{head-comps-ph} \\ \text{PHON} & \langle \text{likes, bagels} \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{verb}\\ \text{FORM} & \textit{fin}\end{bmatrix} \\ \text{SUBJ} & \langle \text{NP}[\textit{3sg}]_{\boxed{4}} \rangle \\ \text{SPR} & \langle\, \rangle \\ \text{COMPS} & \langle\, \rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{2} \\ \text{KEY} & \boxed{3} \\ \text{RELS} & \left\langle \begin{array}{l} \boxed{3}\begin{bmatrix}\textit{like\_rel}\\ \text{EVENT} & \boxed{2}\\ \text{ARG1} & \boxed{4} \\ \text{ARG2} & \boxed{5}\end{bmatrix}, \\ \begin{bmatrix}\textit{t-overlap\_rel} \\ \text{ARG1} & \boxed{2} \\ \text{ARG2} & \textit{now}\end{bmatrix}, \\ \begin{bmatrix}\textit{bagel\_rel}\\ \text{INST} & \boxed{5}\end{bmatrix}\end{array}\right\rangle \end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Figure 2.7: Head-complements phrase *likes bagels*

The head-subject phrase, which combines *Kim* with *likes bagels*, is shown in Figure 2.8. As before, the syntactic features of the mother of this phrase (those under SYNSEM | LOCAL | CAT) are systematically related to the syntactic features of the head daughter. The HEAD, SPR and COMPS values are shared between the two. Further, the comps value of the head daughter must be empty. This ensures that

[10]In Figure 2.7, information that before was spelled out within the HEAD-DTR value is now spelled out within PHON and SYNSEM. Since the tags represent structure-sharing, this is only a notational change.

$$
\left[\begin{array}{ll}
\textit{head-subj-ph} & \\
\text{PHON} & \boxed{B} \oplus \boxed{A} \\
\text{SYNSEM} & \left[\begin{array}{ll} \text{LOCAL} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \boxed{1} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \boxed{C} \\ \text{COMPS} & \boxed{D} \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{INDEX} & \boxed{2} \\ \text{KEY} & \boxed{3} \\ \text{RELS} & \boxed{E} \oplus \boxed{F} \end{array}\right] \end{array}\right] \end{array}\right] \\
\text{HEAD-DTR} & \left[\begin{array}{ll} \text{PHON} & \boxed{A} \\ \text{SYNSEM} & \left[\begin{array}{ll} \text{LOCAL} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \boxed{1}\left[\begin{array}{ll} \textit{verb} & \\ \text{FORM} & \textit{fin} \end{array}\right] \\ \text{SUBJ} & \langle \boxed{4} \rangle \\ \text{SPR} & \boxed{C} \\ \text{COMPS} & \boxed{D}\langle\ \rangle \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{INDEX} & \boxed{2} \\ \text{KEY} & \boxed{3} \\ \text{RELS} & \boxed{E} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right] \\
\text{NON-HEAD-DTRS} & \left\langle \left[\begin{array}{ll} \text{PHON} & \boxed{B} \\ \text{SYNSEM} & \boxed{4}\left[\text{LOCAL} \mid \text{CONT} \mid \text{RELS}\ \boxed{F}\right] \end{array}\right] \right\rangle
\end{array}\right]
$$

Figure 2.8: Head-subject phrase (abbreviated)

heads combine first with their complements, giving the constituent structure [*Kim* [*likes bagels*]] rather than [[*Kim likes*] *bagels*]. The SUBJ value of the mother is empty, as the subject is realized within the phrase. As before, the phrasal type enforces the selectional restrictions of the head: the SYNSEM value of the non-head daughter (the subject) is identified with the *synsem* on the head daughter's SUBJ list. This identification means that any requirements (such as agreement features) specified by the head daughter of its subject must be compatible with the subject. Similarly, the semantic index of the subject is available to the head daughter and (through a series of such identities, including $\boxed{3}$ in Figure 2.5) can be incorporated as an argument of a relation contributed by the head verb.[11]

In our analysis of (1), the head daughter of the head-subject phrase is the head-complements phrase *likes bagels*. Unifying the description of this phrase with the head daughter value of Figure 2.8 gives the description in Figure 2.9. (The HEAD-DTR and NON-HEAD-DTRS features of the head-complements phrase have again been left out for reasons of space.)

Since the description of *Kim* unifies with (that is, does not conflict with) the one non-head daughter called for by Figure 2.9, it can be combined with that structure in a similar fashion. The PHON and SYNSEM values of the resulting description are shown in Figure 2.10.

Figure 2.11 gives a 'translation' of the analysis of this sentence into the more familiar tree notation. In this figure, the nodes are labelled with the CAT values of the words and phrases postulated in the analysis. From this figure, it is apparent that constituent structure is minimal. Note also how the valence requirements of the verb get 'canceled off' as one moves up the tree.

[11]Unlike the head-complements phrase, this version of the head-subject phrase requires a finite verb as its head. As such, it corresponds to Sag's (1997) finite head-subject phrase. Sag notes that a complete grammar of English may require other types of head-subject phrase.

$$
\begin{bmatrix}
\textit{head-subj-ph} & \\
\textsc{phon} & \boxed{B} \oplus \boxed{A} \\
\textsc{synsem} & \begin{bmatrix} \textsc{local} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \boxed{1} \\ \textsc{subj} & \langle\ \rangle \\ \textsc{spr} & \boxed{C} \\ \textsc{comps} & \boxed{D} \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textsc{index} & \boxed{2} \\ \textsc{key} & \boxed{3} \\ \textsc{rels} & \boxed{E} \oplus \boxed{F} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\
\textsc{head-dtr} & \begin{bmatrix} \textsc{phon} & \boxed{A}\langle \text{ likes, bagels } \rangle \\ \textsc{synsem} & \begin{bmatrix} \textsc{local} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \boxed{1}\begin{bmatrix} \textit{verb} & \\ \textsc{form} & \textit{fin} \end{bmatrix} \\ \textsc{subj} & \langle\ \boxed{4}\text{NP}[\textit{3sg}]_{\boxed{5}}\ \rangle \\ \textsc{spr} & \boxed{C} \\ \textsc{comps} & \boxed{D}\langle\ \rangle \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textsc{index} & \boxed{2} \\ \textsc{key} & \boxed{3} \\ \textsc{rels} & \boxed{E}\left\langle \begin{array}{l} \boxed{3}\begin{bmatrix} \textit{like\_rel} & \\ \textsc{event} & \boxed{2} \\ \textsc{arg1} & \boxed{5} \\ \textsc{arg2} & \boxed{6} \end{bmatrix}, \\ \begin{bmatrix} \textit{t-overlap\_rel} & \\ \textsc{arg1} & \boxed{2} \\ \textsc{arg2} & \textit{now} \end{bmatrix}, \\ \begin{bmatrix} \textit{bagel\_rel} & \\ \textsc{inst} & \boxed{6} \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\
\textsc{non-head-dtrs} & \left\langle \begin{bmatrix} \textsc{phon} & \boxed{B} \\ \textsc{synsem} & \boxed{4}\begin{bmatrix} \textsc{local} \mid \textsc{cont} \mid \textsc{rels} & \boxed{F} \end{bmatrix} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Figure 2.9: Head-subject phrase headed by *likes bagels*

$$
\begin{bmatrix}
\textit{head-subj-ph} & \\
\text{PHON} & \langle \text{ Kim, likes, bagels } \rangle \\
\text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix}
\text{CAT} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \textit{fin} \end{bmatrix} \\
\text{SUBJ} & \langle\ \rangle \\
\text{SPR} & \langle\ \rangle \\
\text{COMPS} & \langle\ \rangle
\end{bmatrix} \\
\text{CONT} & \begin{bmatrix}
\text{INDEX} & \boxed{2} \\
\text{KEY} & \boxed{3} \\
\text{RELS} & \left\langle \begin{array}{l}
\begin{bmatrix} \textit{named\_rel} & \\ \text{INST} & \boxed{5} \\ \text{ARG} & \text{Kim} \end{bmatrix}, \\
\begin{bmatrix} \textit{like\_rel} & \\ \text{EVENT} & \boxed{2} \\ \text{ARG1} & \boxed{5} \\ \text{ARG2} & \boxed{6} \end{bmatrix}, \\
\begin{bmatrix} \textit{t-overlap\_rel} & \\ \text{ARG1} & \boxed{2} \\ \text{ARG2} & \textit{now} \end{bmatrix}, \\
\begin{bmatrix} \textit{bagel\_rel} & \\ \text{INST} & \boxed{6} \end{bmatrix}
\end{array} \right\rangle
\end{bmatrix}
\end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 2.10: Head-subject phrase *Kim likes bagels*

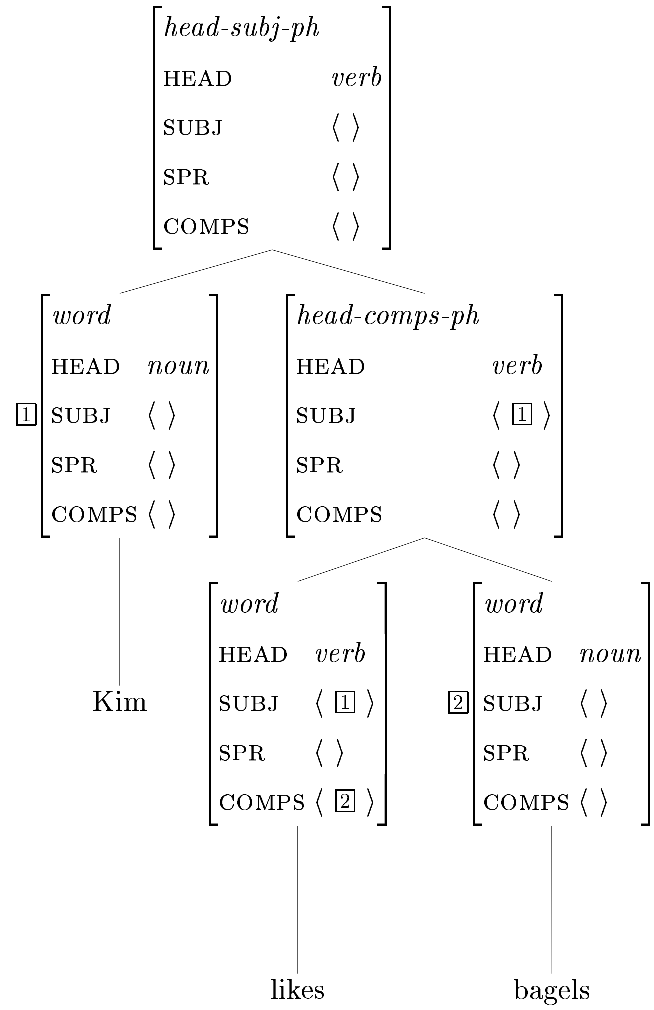

Figure 2.11: Tree version of *Kim likes bagels*

In this section, I have presented the analysis of this sentence in a bottom up fashion. It is important to note, however, that the partial descriptions of words and phrases used are defined in such a way that they may be combined in any order. That is, because this is a declarative, constraint-based grammar, one could get exactly the same result in any number of ways. For example, one could combine *Kim* with the NON-HEAD-DAUGHTER value of the *head-subj-ph*, then combine the *head-comps-ph* with the HEAD-DAUGHTER value of the *head-subj-ph*, then combine *bagels* with NON-HEAD-DAUGHTER value of the *head-comps-ph*, and plug *likes* in last as the head daughter of the *head-comps-phrase*. As long as all of the constraints provided by the partial description are consistent, they'll give the same result in any order. If any constraints conflict, it won't be possible to combine the types in any order.

2.2 Hierarchies

The types posited in the preceding discussion are quite detailed, as required by a precise formulation of the grammar. However, much of that detail is shared by different types, and one would like to capture such generalizations. In HPSG, this is accomplished by organizing the types into a hierarchy. For example, the *head-subj-ph* and the *head-comps-ph* are both subtypes of a more general type, *headed-phrase*. This mini type hierarchy is depicted as in Figure 2.12.

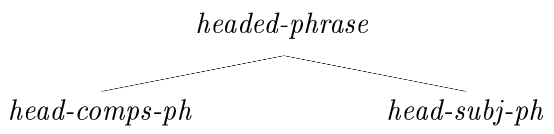

Figure 2.12: Miniature type hierarchy of headed phrases

The type *headed-phrase* bears all of the constraints that its subtypes have in common, as shown in Figure 2.13. In particular, *headed-phrase* specifies the identification of the HEAD and SPR values between mother and head daughter,[12] and the relationship of the PHON and CONT values of the mother to those of the daughters.[13] The subtypes inherit these constraints and add some of their own. This is illustrated for the *head-comps-ph* in Figure 2.14. Unifying all of the constraints given in Figure 2.13 with those given in Figure 2.14 gives exactly Figure 2.5 (modulo the specific numbers or letters chosen for tags, which are immaterial anyway).

$$
\begin{bmatrix}
\textit{headed-phrase} & \\
\text{PHON} & (\boxed{E} \oplus \ldots \oplus \boxed{N}) \bigcirc \boxed{A} \\
\text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{HEAD} & \boxed{1}\\ \text{SPR} & \boxed{B}\end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{2} \\ \text{KEY} & \boxed{3} \\ \text{RELS} & \boxed{D}\oplus\boxed{M}\oplus\ldots\oplus\boxed{Z}\end{bmatrix}\end{bmatrix}\end{bmatrix}\\
\text{HEAD-DTR} & \begin{bmatrix}\text{PHON} & \boxed{A}\\ \text{SYNSEM} & \begin{bmatrix}\text{LOCAL} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{HEAD}&\boxed{1}\\ \text{SPR}&\boxed{B}\end{bmatrix}\\ \text{CONT} & \begin{bmatrix}\text{INDEX}&\boxed{2}\\ \text{KEY}&\boxed{3}\\ \text{RELS}&\boxed{D}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\\
\text{NON-HEAD-DTRS} & \left\langle \begin{bmatrix}\text{PHON}&\boxed{E}\\ \text{SYNSEM} & [\ldots\ \text{RELS}\ \boxed{M}]\end{bmatrix}, \ldots \begin{bmatrix}\text{PHON}&\boxed{N}\\ \text{SYNSEM}&[\ldots\ \text{RELS}\ \boxed{Z}]\end{bmatrix}\right\rangle
\end{bmatrix}
$$

Figure 2.13: Headed phrase

[12]The relationship of the SUBJ and COMPS values of the mother to those of the head daughter can't be specified by *headed-phrase* as they differ for the two subtypes. Likewise, once a *head-spr-phrase* is included in the grammar, the same issue will arise for the SPR feature, and *headed-phrase* will not constrain the value of this feature. However, with a minimal use of defaults, Sag (1997) is able to state the general preservation of all valence features save those that are being canceled off as a constraint on *headed-phrase*.

[13]Here I've represented the PHON value of the mother as the list of PHON values of the non-head daughters with the PHON value of the head daughter shuffled in ($\bigcirc$) somewhere. The subtype *head-comps-ph* specifies that the head daughter's phonology comes first. A more general solution to this problem is to use linear precedence constraints, as in Pollard and Sag 1994.

$$\begin{bmatrix} \textit{head-comps-ph} & \\ \text{PHON} & \boxed{A} \oplus \textit{list} \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{SUBJ} & \boxed{B} \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{HEAD-DTR} & \begin{bmatrix} \text{PHON} & \boxed{A} \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{SUBJ} & \boxed{B} \\ \text{COMPS} & \mathbf{sts}(\boxed{C}) \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{NON-HEAD-DTRS} & \boxed{C}\textit{list}(\textit{sign}) \end{bmatrix}$$

Figure 2.14: Head-complements phrase, as a subtype

Although the type *headed-phrase* is a part of the grammar, it is never used directly in the analysis of any sentence. In general, only 'maximal' word and phrase types (types with no subtypes) are so used.

In addition to organizing the types so that generalizations can be stated on supertypes, the hierarchy also plays a role in determining compatibility (i.e., whether two types can unify). For example, consider Sag and Wasow's (1999) sub-hierarchy of (Standard) English agreement types, given in Figure 2.15.[14]

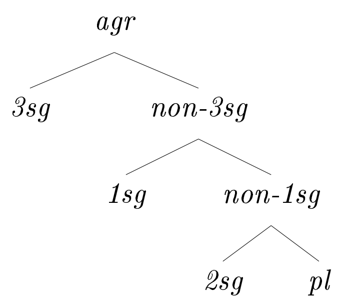

Figure 2.15: (Standard) English agreement types

These types are potential values of the feature AGR of nouns (cf. Figure 2.3,

[14]The features appropriate to these types are PERSON, NUMBER and GENDER. The first two are declared as appropriate for *agr*, and thus for all of its subtypes. In English, GENDER is only relevant for *3sg*.

page 10). Inflected verbs constrain the possible AGR values of their subjects. *Likes* will constrain its subject to be [AGR *3sg*] (cf. Figure 2.1). Since *3sg* and *non-3sg* are sister types (and have no common subtypes), *non-3sg* and any of its subtypes are incompatible with the *3sg* constraint imposed by the verb. On the other hand finite *like* requires that it's subject be [AGR *non-3sg*].[15] This is compatible with any subtype of *non-3sg*, so that an NP specified as, say, *1sg* could appear as the subject of *like*.[16]

Returning to the lexical types, much of the information given (especially for *likes* in Figure 2.1) represents generalizations that hold across all verbs or across subclasses of verbs. Here again, the generalizations are captured by factoring them out as constraints on supertypes. Beginning with the work of Flickinger et al. (1985), Flickinger (1987) and Pollard and Sag (1987), lexical types have been organized into a multiple inheritance hierarchy. Multiple inheritance hierarchies allow types to have more than one supertype and therefore express cross-cutting generalizations.

To take a simplified example, in English, valence possibilities cross-cut the part of speech distinctions. To represent this, part of the hierarchy for English could look like Figure 2.16. The subtypes of $\boxed{\text{VALENCE}}$ specify ARG-ST values (the feature name ARG-ST is omitted for reasons of space), while the subtypes of $\boxed{\text{POS}}$ specify HEAD values. (This is, of course, not an exhaustive listing of the subtypes in either dimension.) The next level down in the hierarchy specifies word classes, such as *verb-trans* which each inherit the constraints from one part of speech type and one valence type. Individual lexemes are subtypes of these word classes. Davis (1996) and Davis and Koenig (2000) extend this idea to a hierarchy of types that specify the linking between semantic arguments and the elements of ARG-ST. In a fully worked out grammar, all that needs to be stipulated for most lexical entries is the word class the entry belongs to, its phonological form, and the semantic relation it contributes.[17]

[15]Note that this is not true of the homophonous base form.

[16]For ease of exposition, I am representing agreement features as head features. In fact, agreement is much more complicated, involving both semantic agreement (mediated by indices) and syntactic agreement. For discussion, see Pollard and Sag 1994, Kathol 1999 and Bender and Flickinger 1999.

[17]The exceptions are words with idiosyncratic properties (defective paradigms, unusual complementation patterns, etc.). It would be superfluous to posit word classes in these cases each populated by only one lexical item. Instead, the lexical entries can inherit from more general word classes and

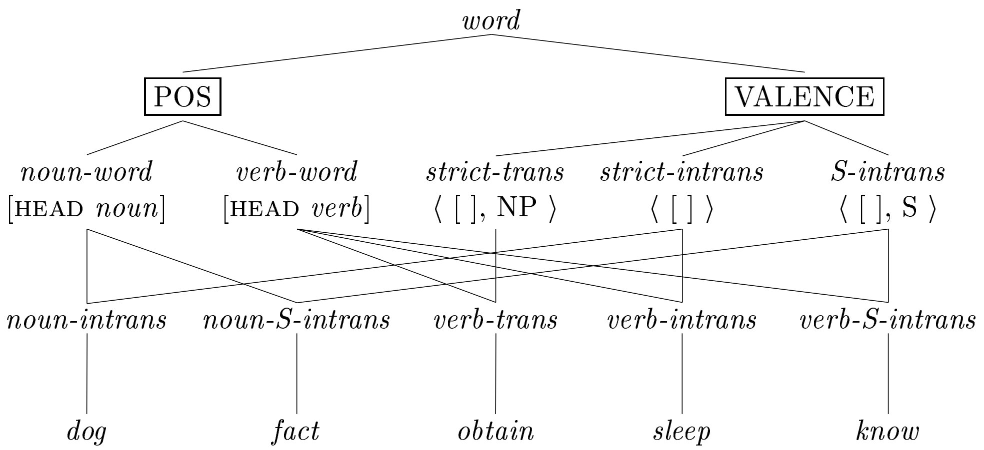

Figure 2.16: Multiple inheritance hierarchy

All of the rest of the complex information required (cf. Figure 2.1, page 7) is inherited from supertypes.

Note that it is rare in such multiple inheritance hierarchies to find all possible combinations of the supertypes instantiated. For example, in Figure 2.16, *strict-trans* has no mutual subtypes with *noun-word*. This is one sense in which the type hierarchy represents linguistically relevant *sub*generalizations.

Taking a completely underspecified type such as *object* as the root of the hierarchy, all of the types in the grammar, including lexical types, phrasal types and substructure types such as *synsem* or *3sg*, can be organized into one hierarchy. Figure 2.17 shows how this would look for the types introduced so far, plus a few contrasting types, such as *inf*(initival) which contrasts with *fin*(inite) under *form*. Because of space limitations, the subhierarchy under *word* is not included in this figure. Please refer to Figure 2.16 above.

This section has reviewed how types are organized into a hierarchy, and the information that that organization embodies. To summarize, the hierarchy allows for the expression of (possibly cross-cutting) generalizations and a specification of (in)compatibility between types.

specify the idiosyncratic information.

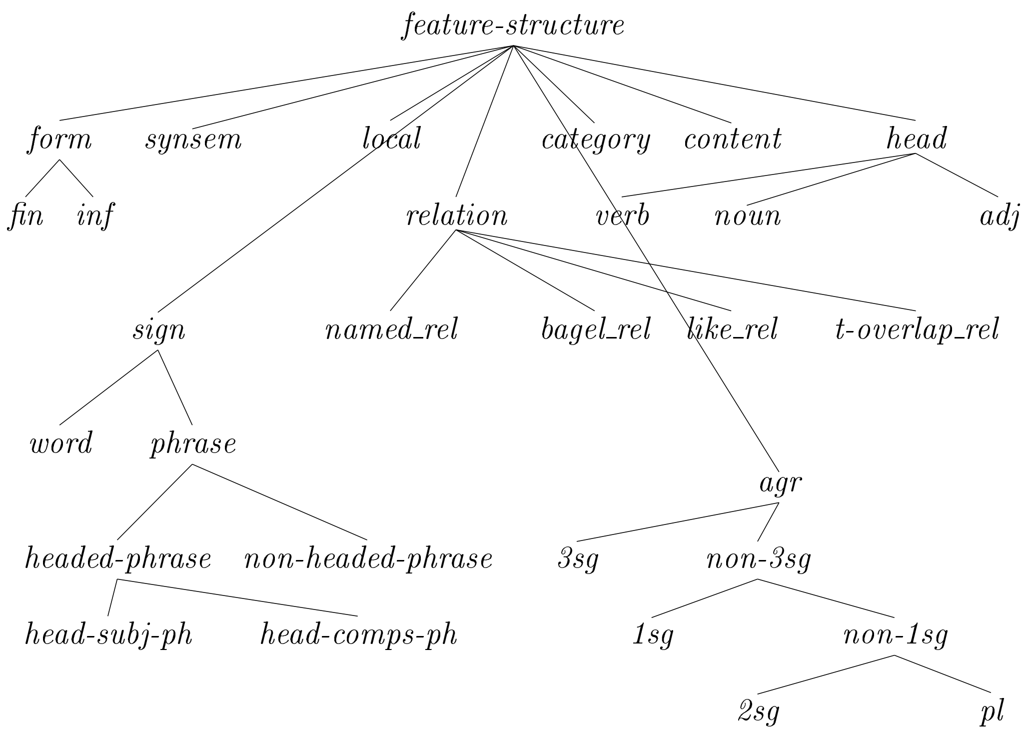

Figure 2.17: Hierarchy for types introduced so far

2.3 Grammar without empty categories

In this section, I will review lexicalist, constraint-based analyses of phenomena for which phonologically empty elements have been posited in other frameworks. In all three cases, it will be shown that empty categories are not required. These analyses are not original to HPSG, although I will give HPSG versions of them. The phenomena discussed are raising and control, pro-drop, and extraction/long-distance dependencies.

2.3.1 Raising and control

'Raising' and 'control' refer to the phenomena illustrated in (2) and (3), respectively:

(2) Kim tends to annoy Sandy.

(3) Kim tries to annoy Sandy.

In (2), *tends* provides a one-place relation, taking the proposition 'Kim annoys Sandy' as its sole argument. In (3), *tries* provides a two-place relation between Kim and the proposition 'Kim annoys Sandy'. In both cases, the subject of the higher verb is also interpreted as the subject of the lower verb. Transformational analyses of these phenomena (beginning with Rosenbaum 1967) posited underlying structures which included the sentence *Kim annoy Sandy.* The subject of the sentence is then either moved to be the subject of the higher verb (raising) or deleted under identity with the subject of the higher verb ('equi-NP-deletion'/control). More modern versions of these analyses (e.g., Chomsky 1981) still have movement in the raising sentence (leaving behind an NP-trace) and posit the phonologically empty element PRO as the embedded subject of the control sentence.

Bresnan (1978) suggests that instead the relationship of the matrix subject to the embedded verb could be mediated by the lexical entry of the raising or control verb (i.e., of *tries* or *tends*) itself. In HPSG, Pollard and Sag (1994) develop this analysis as shown in Figures 2.18 and 2.19, which give abbreviated lexical entries for *tend* and *try* respectively.[18]

$$
\begin{bmatrix}
\textit{lexeme} & \\
\textsc{stem} & \langle \text{ tend } \rangle \\
\textsc{synsem} & \begin{bmatrix} \textsc{local} & \begin{bmatrix}
\textsc{cat} & \begin{bmatrix}
\textsc{head} & \textit{verb} \\
\textsc{subj} & \langle \boxed{1}\text{NP} \rangle \\
\textsc{comps} & \left\langle \begin{bmatrix} \textsc{local} & \begin{bmatrix}
\textsc{cat} & \begin{bmatrix}
\textsc{head} & \begin{bmatrix} \textit{comp} & \\ \textsc{form} & \textit{inf} \end{bmatrix} \\
\textsc{subj} & \langle \boxed{1} \rangle \\
\textsc{comps} & \langle \ \rangle
\end{bmatrix} \\
\textsc{cont} & \begin{bmatrix} \textsc{index} & \boxed{2} \end{bmatrix}
\end{bmatrix} \end{bmatrix} \right\rangle
\end{bmatrix} \\
\textsc{cont} & \begin{bmatrix}
\textsc{index} & \boxed{3} \\
\textsc{key} & \boxed{4} \\
\textsc{rels} & \left\langle \boxed{4} \begin{bmatrix} \textit{tend\_rel} & \\ \textsc{event} & \boxed{3} \\ \textsc{arg1} & \boxed{2} \end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 2.18: Raising: A lexical entry for *tend*

$$
\begin{bmatrix}
\textit{lexeme} & \\
\text{STEM}\ \langle \text{ try } \rangle & \\
\text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{SUBJ} & \langle \text{ NP}_{\boxed{4}} \rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{comp} & \\ \text{FORM} & \textit{inf} \end{bmatrix} \\ \text{SUBJ} & \langle \begin{bmatrix} \text{INDEX} & \boxed{4} \end{bmatrix} \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{3} \\ \text{KEY} & \boxed{5} \\ \text{RELS} & \left\langle \boxed{5} \begin{bmatrix} \textit{try\_rel} & \\ \text{EVENT} & \boxed{3} \\ \text{ARG1} & \boxed{4} \\ \text{ARG1} & \boxed{2} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 2.19: Control: A lexical entry for *try*

Both *tend* and *try* select for a phrase headed by an infinitival complementizer (i.e., *to*), which has all of its complements saturated but which is still looking for a subject. The raising verb *tend* identifies its own subject ($\boxed{1}$) with the subject of its complement. The control verb *try* simply coindexes its own subject with the subject of its complement ($\boxed{4}$).[19] In either case, the index of the NP realized as the subject of the matrix verb gets passed along to the infinitival CP. In order to see how the index of *Kim* gets passed all the way to *annoy*, something needs to be said about the complementizer *to*. *To* itself is treated as a raising complementizer, with the lexical entry in Figure 2.20. As a semantically vacuous element, not only does *to* raise the subject of its complement, it also 'raises' or adopts the semantics, by identifying its INDEX value with that of its complement.

Using the familiar tree-structure notation, these lexical entries give rise to constituent structures like that shown in Figure 2.21. The lower clause needs no subject position, as the structure sharing in the lexical entries for the matrix verbs establishes all of the necessary identities. Since there is no subject position, there is no need for it to be filled with an NP-trace or PRO.[20]

---

[18]Because these types are descriptions of the uninflected lexemes, certain information that one would find in the descriptions of inflected words, such as tense information, is left out.

For ease of exposition, the syntactic arguments are specified in the SUBJ and COMPS values and the ARG-ST is suppressed. Since certain mismatches are allowed between ARG-ST and the valence features, lexeme descriptions usually don't stipulate anything about the value of SUBJ and COMPS. For more on this, see the discussion of pro-drop and extraction below.

[19]Thus there are two differences between raising and control verbs in HPSG: whether or not the subject plays a semantic role, and whether the subject is identified with, or merely coindexed with, the subject of the complement. This latter distinction is motivated by quirky case phenomena in Icelandic, where quirky case selected by the lower verb is only realized in the matrix clause if the matrix verb is a raising, rather than control, verb. On quirky case in Icelandic, see Andrews 1982 and Andrews 1990.

[20]Pollard (1989) posits a subtype of *synsem* called *PRO*, which appears on SUBJ and ARG-ST lists in environments where phrase-structure PRO appears as a subject on other accounts. The crucial difference is that *PRO* never occupies a phrase structural position. (See also Sag 1997:451.)

$$
\begin{bmatrix}
\textit{lexeme} & \\
\text{STEM} & \langle \text{ to } \rangle \\
\text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{comp} \\ \text{SUBJ} & \langle \boxed{1}\text{NP} \rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{FORM} & \textit{base}\end{bmatrix} \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle\, \rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{2} \\ \text{RELS} & \langle\, \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 2.20: Raising: A lexical entry for *to*

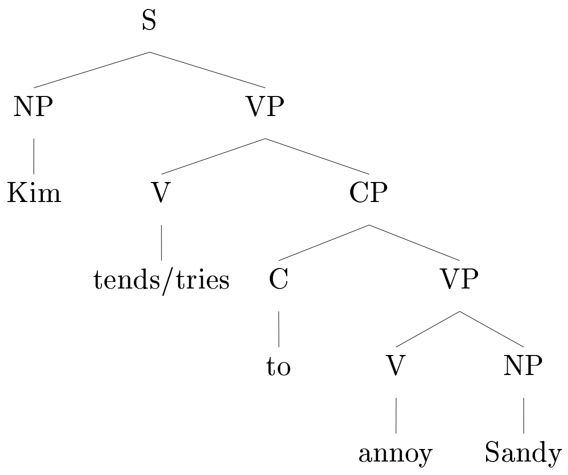

Figure 2.21: Constituent structure in raising/control sentences

2.3.2 Pro-drop

A second type of phonologically null element that has been posited is *pro*. This 'little pro', or phonologically empty pronoun, is posited whenever (NP) arguments are left unexpressed but aren't controlled by some other NP in the sentence (Chomsky 1981, Jaeggli 1982, Rizzi 1982). Classic examples are null subjects in Italian (4) and missing arguments in Japanese (5).

(4) Italian (Balari 1992:4)
Bisogna que parta subito
is.necessary that leave.1/3SG immediately
'It is necessary that I/he/she leave(s) immediately.'

(5) Japanese (Manning and Sag 1999:65)
Naoki-ga mi-ta.
Naoki-NOM see-PAST
'Naoki saw (it).'

However, Bresnan (1982) shows that one does not need to posit a phonologically empty pronoun (that occupies a position in the phrase structure) in order to account for this phenomenon. If a level of argument structure is available, it is sufficient to encode the pronominal properties there.

Thus, on the analysis sketched by Manning and Sag (1999), heads with 'missing' arguments have a mismatch between their ARG-ST value and the value of their valence features.[21] Thus *mita* 'saw' in (5) is subject to the constraints shown in Figure 2.22.

Such entries can be systematically produced either by lexical rule, or by underspecifying the relationship between ARG-ST and the valence features, on the analogy of Bouma et al.'s (in press) analysis of extraction (reviewed in §2.3.3 below). The constraint that relates the valence features to ARG-ST is called the Argument Realization Principle (ARP), and it can be formulated to allow for different kinds of mismatches. Figure 2.23 gives the ARP for a language like Japanese which allows any argument

[21] Balari's (1992) analysis predates the notion of argument structure in HPSG. Balari proposes instead that null subjects arise when an element appears on a SUBJ list, but is not realized in the phrase structure (or controlled by some other NP in the phrase structure).

$$\left[\text{SYNSEM}\ \left[\text{LOCAL}\ \left[\begin{array}{ll}\text{CAT} & \left[\begin{array}{ll}\text{SUBJ} & \langle\, \boxed{1}\, \rangle \\ \text{COMPS} & \langle\,\rangle \\ \text{ARG-ST} & \langle\, \boxed{1}\text{NP}[\mathit{nom}]_{\boxed{2}},\ \text{NP}[\mathit{pro}]_{\boxed{3}}\, \rangle\end{array}\right] \\ \text{CONT} & \left[\text{KEY}\ \left[\begin{array}{ll}\multicolumn{2}{l}{\mathit{see\_rel}} \\ \text{ARG1} & \boxed{2} \\ \text{ARG2} & \boxed{3}\end{array}\right]\right]\end{array}\right]\right]\right]$$

Figure 2.22: Some constraints on *mita* with a missing object

to be missing. This constraint states that, for all signs of type *word*, the ARG-ST value must equal the SUBJ value followed by the COMPS value, with possibly some *pronoun_synsems* shuffled in. Figure 2.24 gives the ARP for a language like Italian which allows subjects to be missing.[22] This constraint states that, for all signs of type *word*, the ARG-ST value must equal the SUBJ value, plus a possibly empty list of *pronoun_synsems*, plus the COMPS value.[23,24]

$$\left[\begin{array}{ll}\multicolumn{2}{l}{\mathit{word}} \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{CAT} & \left[\begin{array}{ll}\text{SUBJ} & \boxed{A} \\ \text{COMPS} & \boxed{B} \\ \text{ARG-ST} & \boxed{A} \oplus \boxed{B} \bigcirc \mathit{list}(\mathit{pron\_synsem})\end{array}\right]\end{array}\right]$$

Figure 2.23: Argument Realization Principle for Japanese-type language

[22]As Balari (1992) argues at length, the picture is somewhat messier: even Italian has some subjects (e.g., second person singular subjects in embedded subjunctives) which cannot be dropped. However, I see no reason that an Argument Realization analysis could not be scaled up to include for such constraints.

[23]$\ominus$ indicates the relation of list subtraction. In the following statement, Z is a list that results from removing some number (possibly zero) of elements from X while preserving the order of the elements with respect to each other. Y is the list of elements removed.

$$\mathrm{X} \ominus \mathrm{Y} = \mathrm{Z}$$

It follows that in Figure 2.24, the first part of the argument structure ($\boxed{A}$) is the SUBJ list plus some number of *pronoun_synsems*. The list $\boxed{A}$ must also be constrained to contain at most one element.

[24]Both of these ARPs will need to be revised to allow for extraction as well as pro-drop. See note 31 on page 38.

$$\begin{bmatrix} \textit{word} & \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{CAT} & \begin{bmatrix} \text{SUBJ} & \boxed{\text{A}} \ominus \textit{list}(\textit{pron\_synsem}) \\ \text{COMPS} & \boxed{\text{B}} \\ \text{ARG-ST} & \boxed{\text{A}} \oplus \boxed{\text{B}} \end{bmatrix} \end{bmatrix}$$

Figure 2.24: Argument Realization Principle for Italian-type language

These two examples (raising/control and pro-drop) show that the information encoded by argument structure is akin to that encoded by the deep structure in early versions of transformational grammar. Argument structure encodes the possible syntactic arguments of a head and their 'underlying' obliqueness relationships. There are different possible relationships between argument structure and the surface configurations the head can appear in.[25] The advantage of a theory with a level like argument structure (or f-structure in LFG) is that the similarities between the different realizations of the arguments of the same head can be described without positing phonologically null syntactic elements.

2.3.3 Extraction

Since Wasow 1972 (see also Lasnik 1992), transformational analyses of extraction phenomena (topicalization, fronted *wh*-questions, *tough*-movement) have posited an empty category, called a trace, at the bottom of the extraction dependency. For example, the movement that creates the extraction dependency in (6) relates the underlying structure (a) to the surface structure (b) in Figure 2.25.[26] The matching subscripts on *bagels* and the trace in the surface structure indicate that *bagels* binds the trace.

(6) Bagels I know Kim likes.

Traceless analyses of extraction have been proposed in various frameworks by Ades and Steedman (1982) (Categorial Grammar), Gazdar et al. (1984) (GPSG), Kaplan

[25] See Manning and Sag 1998 for a discussion of the range of possible relationships between ARG-ST and the valence lists.

[26] Since this discussion is focussed on traces, Figure 2.25 abstracts away from the CP/IP notation and the associated empty categories.

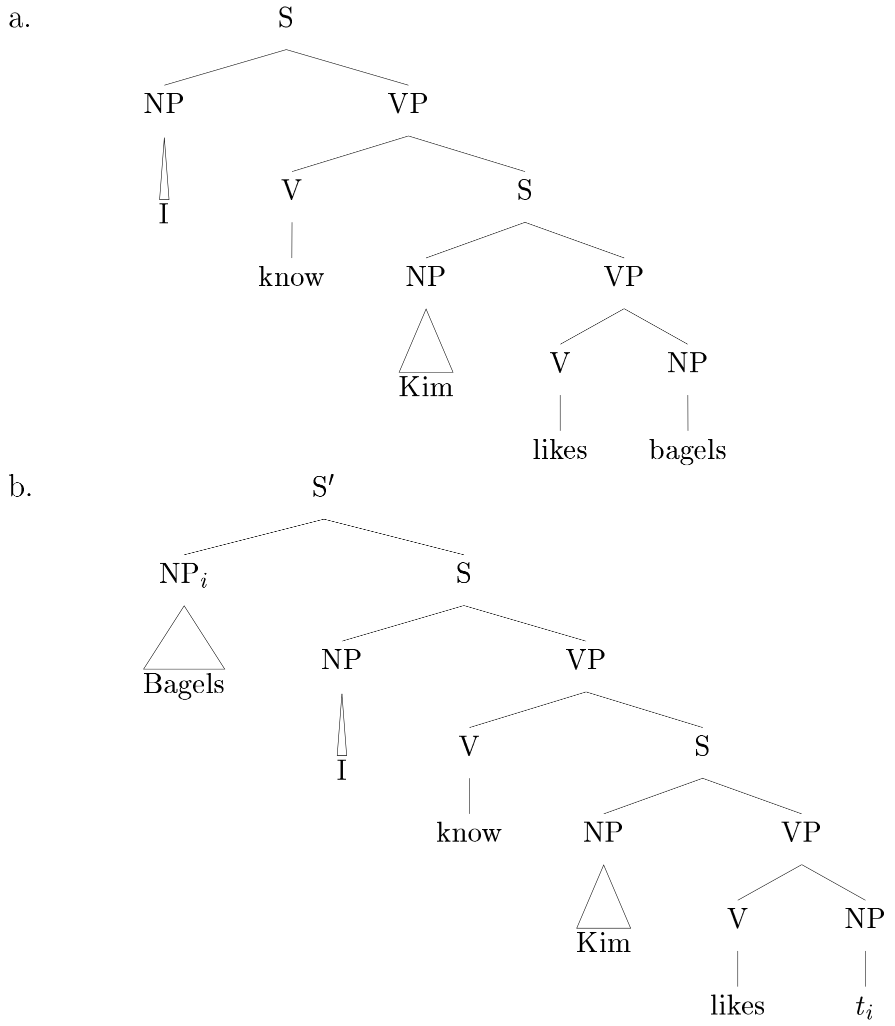

Figure 2.25: Movement analysis of extraction

and Zaenen (1988) (LFG), and Pollard and Sag (1994) (HPSG), among others. The history of the GPSG/HPSG version of traceless extraction proceeded in two steps. The first step was Gazdar's (1981) demonstration that a context-free grammar with complex category labels could model extraction, without recourse to transformations. The essence of Gazdar's analysis is that nodes dominating a trace carry that information 'up the tree' to the subtree that combines with the filler of the trace. That is, a sentence with an NP gap in it is a different category (S/NP or 'S slash NP') from a sentence with no gap (just plain S), and there is a special phrase structure rule that combines a filler NP with an S/NP. Under this analysis, the single structure representing (6) is as in Figure 2.26.[27]

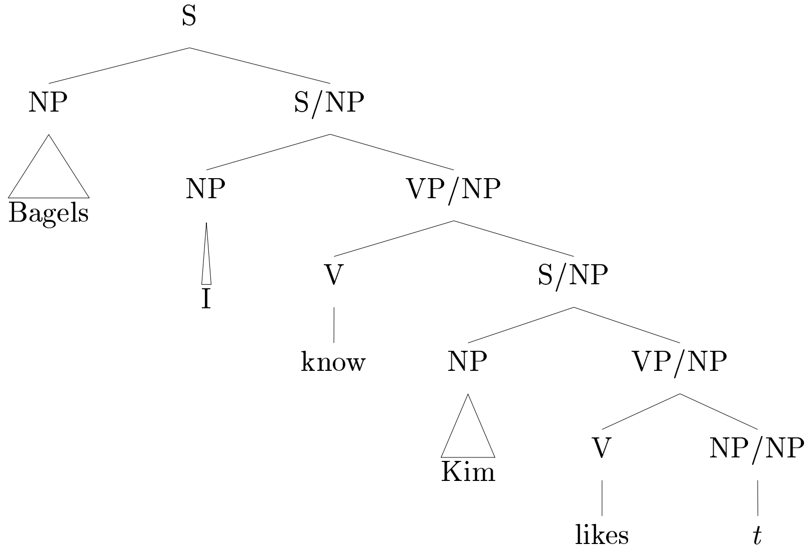

Figure 2.26: Slash analysis of extraction

When viewed in this light, extraction constructions involve three separate parts: the bottom, where something is missing, the middle, where that information is transmitted, and the top, where the filler is found (Gazdar et al. 1985:138). Since there is no movement on this analysis, it is possible to consider other formulations of the

[27] *Bagels* and the trace (*t*) are not coindexed this time, as "*t* is a dummy element [which] serves no semantic function ... " (Gazdar 1981:162) Gazdar's system includes separate semantic rules associated with each phrase structure rule which conspire to link up the interpretation of *bagels* with an argument of *likes*' relation.

bottom of the dependency which do not involve a trace or even a phrase structural position that a trace could occupy. The various proposals to this effect constitute step two.

Pollard and Sag's (1994), Ch. 9 proposal (refined by Sag (1997) and Bouma et al. (in press)) makes use of the potential for mismatch between the argument structure and the valence features. In particular, they posit word types such as the one in Figure 2.27.[28]

$$
\begin{bmatrix}
\textit{word} & \\
\text{PHON} \ \langle \text{likes} \rangle & \\
\text{SYNSEM} & \begin{bmatrix}
\text{LOCAL} & \begin{bmatrix}
\text{CAT} & \begin{bmatrix}
\text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \textit{fin} \end{bmatrix} \\
\text{SUBJ} & \langle \boxed{1} \rangle \\
\text{SPR} & \langle\ \rangle \\
\text{COMPS} & \langle\ \rangle \\
\text{ARG-ST} & \left\langle \boxed{1}\text{NP}[\textit{3sg}]_{\boxed{4}}, \begin{bmatrix} \textit{gap-ss} & \\ \text{LOCAL} & \boxed{2}\begin{bmatrix} \text{CONT} \mid \text{INDEX} & \boxed{5} \end{bmatrix} \\ \text{SLASH} & \left\{\boxed{2}\right\} \end{bmatrix} \right\rangle
\end{bmatrix} \\
\text{CONT} & \begin{bmatrix}
\text{INDEX} & \boxed{3} \\
\text{KEY} & \boxed{6} \\
\text{RELS} & \left\langle \boxed{6}\begin{bmatrix} \textit{like\_rel} & \\ \text{EVENT} & \boxed{3} \\ \text{ARG1} & \boxed{4} \\ \text{ARG2} & \boxed{5} \end{bmatrix}, \begin{bmatrix} \textit{t-overlap\_rel} & \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \textit{now} \end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix} \\
\text{SLASH} & \left\{\boxed{2}\right\}
\end{bmatrix}
\end{bmatrix}
$$

Figure 2.27: Lexical entry for *likes* with an extracted complement

[28] This is an example of complement extraction. The same mechanism can be extended to adjunct extraction (see Pollard and Sag 1994:384–388 and Bouma et al. in press). For a discussion of traceless subject extraction, see §3.4.3 below.

The lexical item in Figure 2.27 is not seeking any complements (its COMPS list is empty), but it does have a non-empty SLASH value:[29] The LOCAL value of its second argument is entered into the SLASH set of this argument (and by Sag's (1997) lexical amalgamation version of slash-passing) into the SLASH set of the verb. This information will be transmitted up the tree and eventually identified with the LOCAL value of the filler. Since the LOCAL value contains both the CAT and CONT substructures, any syntactic or semantic requirements the verb places on its second argument will also be transmitted.

Translating again to the more familiar tree notation, the constituent structure on this analysis is as in Figure 2.28. This constituent structure involves no trace and no phrase-structural position for the complement of *likes*.

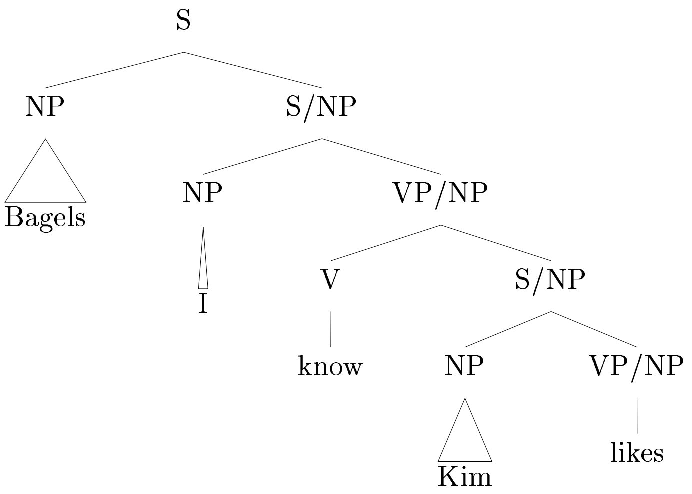

Figure 2.28: Traceless analysis of extraction

Pollard and Sag (1994) related lexical entries such as in Figure 2.27 to more basic lexical entries by means of a lexical rule. Bouma et al. (in press) model the relationship between the two types of lexical entries by instead underspecifying the relationship between ARG-ST and the valence features. In particular, they posit a

[29]The feature SLASH shown in Figure 2.27 is feature-based encoding of Gazdar's complex symbols with '/'.

constraint on the type *word* such as in Figure 2.29.[30] This constraint allows for a mismatch between the COMPS list and the non-initial members of the ARG-ST list. In particular, it states the the COMPS list may be shorter than the corresponding part of the ARG-ST list ($\boxed{2}$), just in case the elements 'removed' are unified with the type *gap-ss*. The type *gap-ss* (a subtype of *synsem*) identifies its LOCAL value with the sole element of its SLASH set, as in Figure 2.30, starting off the slash dependency.

$$\begin{bmatrix} \textit{word} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{SUBJ} & \boxed{A} \\ \text{COMPS} & \boxed{B} \ominus \textit{list(gap-ss)} \\ \text{ARG-ST} & \boxed{A} \oplus \boxed{B} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 2.29: Argument Realization Principle

$$\begin{bmatrix} \textit{gap-ss} & \\ \text{LOCAL} & \boxed{1} \\ \text{SLASH} & \left\{ \boxed{1} \right\} \end{bmatrix}$$

Figure 2.30: *gap-ss*

Both the descriptions of *likes* in Figure 2.2 (page 9) and in Figure 2.27 satisfy the constraint in Figure 2.29, so both are licensed.[31]

[30]This presentation follows Bouma et al. (in press) in abstracting away from the feature SPR for simplicity, but differs in that their feature DEPS is replaced by ARG-ST. The value of DEPS is the append of the value of ARG-ST plus a (possibly empty) list of adjuncts. This is crucial for Bouma et al.'s (in press) uniform account of adjunct and complement extraction.

Note also that the constraint on (English) verbs that their SUBJ lists be at most one element long is stated elsewhere in the type hierarchy.

[31]The Argument Realization Principle for a null complement language such as Japanese would have to also include the constraint in Figure 2.23, page 32. The result would look like this:

$$\begin{bmatrix} \textit{word} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{SUBJ} & \boxed{A} \\ \text{COMPS} & \boxed{B} \ominus \textit{list(gap-ss)} \\ \text{ARG-ST} & \boxed{A} \oplus \boxed{B} \bigcirc \textit{list(pron\_synsem)} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

2.3.4 Summary

This section has reviewed lexicalist, constraint-based analyses which show that empty categories are not required in the description of raising/control, pro-drop, or even extraction. In the last case especially, there is both linguistic and psycholinguistic evidence to prefer the traceless analysis.[32]

The most striking piece of linguistic evidence, as noted by Sag (1998), is the Conjunct Constraint subclause of Ross's (1967)b Coordinate Structure Constraint:

(7) In a coordinate structure, no conjunct may be moved.

This is illustrated by the examples in (8), from Sag 1998.

(8) a. *Which of her books did you read both [[a review of __] and [__]]?
b. *Which of her books did you find [[__] and [a review of __]]?
c. *Which rock legend would it be ridiculous to compare [[__] and [__]]? (cf. Which rock legend would it be ridiculous to compare __ with himself?)

Sag notes that these facts follow directly from a traceless analysis of extraction:

> A *Wh*-gap is simply a position where an element selected by a head (whether complement or adjunct) fails to be realized [...].
>
> The elements that are coordinated, i.e. the conjuncts of a coordinate structure, must be syntactic constituents (or perhaps sequences thereof).
>
> Conjunctions are not heads; rather, coordinate structures instantiate an independent construction type.
>
> Therefore, *wh*-gaps, which are not constituents, can never be conjuncts. (Sag 1998:5)

The psycholinguistic evidence is due to Pickering and Barry (1991) who note the difference in difficulty of processing between (9) and (10).

(9) That's the prize which we gave [every student capable of answering every single tricky question on the details of the new and extremely complicated theory about the causes of political instability in small nations with a history of military rulers] __ .

[32]For arguments against putative evidence for traces, see Sag and Fodor 1994 and Sag 1998.

(10) We gave [every student capable of answering every single tricky question on the details of the new and extremely complicated theory about the causes of political instability in small nations with a history of military rulers] [a prize].

Pickering and Barry (1991) argue that if the interpretation of the extracted element in (9) is processed only once the (putative) trace is encountered, it should be as difficult to process as (10). They take the fact that (9) is relatively easy to process as evidence that the interpretation of the extracted element is processed at the selecting head (*gave*), i.e., evidence against a trace-based theory of extraction.

More generally, not only traces, but all empty elements run counter to the surface-orientation of lexicalist, constraint-based theories of grammar. According to Sag and Wasow (1999), a grammar is surface-oriented if it "provides a reasonably simple structure that is directly associated with the string of words that constitute each sentence" (Sag and Wasow 1999:219). Surface-orientation is one of the design features of what they call 'performance-plausible competence grammar'. It is motivated by evidence that language processing is incremental, with hearers producing partial structures and partial interpretations of an incoming speech stream syllable by syllable. In order for a model of competence grammar to be able to be embedded in a performance system that can model this kind of incremental processing, the less abstract the syntactic representations, and the more closely said representations are tied to the surface string, the better. A parser that had empty categories in its lexicon must allow for their appearance at every point in the surface string. Surely this would interrupt incremental processing. At the very least, it would be inefficient.[33]

2.4 Constructions

Early work in HPSG (e.g., Flickinger et al. 1985, Flickinger 1987, Pollard and Sag 1987 and Pollard and Sag 1994) emphasized moving grammatical information into complex

[33]Johnson and Kay (1994) show that a parser can terminate even given empty categories as possible elements of the input string as long as all empty categories are sponsored by some other lexical item. That is, the lexical entries for the pronounced lexical items would specify how many empty categories each pronounced item could sponsor, bounding the number of empty categories that need to be considered. Guaranteed termination and efficiency are two different issues, however.

lexical entries (arranged in a hierarchy) that interacted with maximally simple phrase structure schemata. These schemata (which numbered only six in Pollard and Sag 1994) projected the information borne by the lexical entries into the rich variety of sentences found in language.

Although this strategy had impressive initial success, later work in HPSG (e.g., Sag 1997 and Ginzburg and Sag to appear) as well as work in Construction Grammar (e.g., Fillmore 1987, Fillmore 1988 and Fillmore et al. 1988; see also Zwicky 1994) has shown that as one moves away from the 'core' aspects of grammar, more complex phrase structure schemata (constructions) are required. In fact, this move is completely compatible with the notion of sign based grammar, as developed in Pollard and Sag 1994: If words and phrases are both just types of signs, and signs are pairing of form and meaning, then there is no reason for phrasal signs not to include more interesting pairings of form and meaning than the bare, simple phrase structure schemata assumed before. Further, as shown by Sag (1997), by extending the idea of the multiple inheritance hierarchy from words to phrases, cross-constructional generalizations can be captured in a theory with multiple, specific phrase structure types. In this section, I will briefly review two examples of the use of constructions in linguistic analysis: The analyses of the correlative conditional construction in Fillmore (1987, 1988) and McCawley (1988), and Sag's (1997) constructional analysis of relative clauses in English.

2.4.1 Correlative conditionals

The English correlative conditional construction, illustrated in (11–14), provides one of the most compelling arguments for the need for a notion of construction in grammatical theory. This construction is peripheral in the sense that its intricacies go beyond what can be captured by most models of "core" grammar, but far from peripheral in that it does not seem to be restricted to rarely used registers. The examples in (11) and (12) come from the Callhome corpus of spoken American English and the examples in (13) and (14) come from the North American News Text corpus, and in particular from the New York Times.[34] Further, this construction is not some peculiarity of English alone. As McCawley (1988) argues, German and Mandarin have constructions that fulfill similar functions and similar, although not identical, constraints.

(11) Anyway, the more he hears about the community, the more he hears it's not so great.

(12) And also I think the sunnier, the more bright it is, the fuzzier it comes out for some reason.

(13) It was a fragile hope that began as an impossibility and gradually assumed a reality all its own; the further she marched through the draw, the more it seemed that a 10th championship was not only her desire but her destiny.

(14) He said the study found that most accused doctors were between ages 40 and 69. "The older the physician, the more likely the resolution would be surrender of the license," Winn wrote.

Fillmore (1987, 1988) and McCawley (1988), discuss numerous properties of this construction, including the following:[35]

- It has two daughters.
- Each daughter is a filler-gap construction, with a left-dislocated filler.
- The filler of each daughter is the word *the* followed by a comparative phrase.
- The antecedent is optionally a negative polarity context (15).

(15) a. The more you do any of that, the easier it gets. (Fillmore 1987:165)
b. The more noises I hear, the more sure I am that someone is here.
c. #The more noises I hear, the more sure I am that anyone is here.

[34] Both of these corpora are available from the Linguistic Data Consortium: http://www.ldc.upenn.edu.

[35] This presentation is restricted to the non-inverted (antecedent first) variety of correlative conditional, for simplicity.

- The *will* future is disallowed in the antecedent (16).

(16) The faster you (*will) drive, the sooner you'll get there.
(McCawley 1988:177)

- The copula can be omitted, just in case its complement is the extracted element and the correlative conditional is of the generic type of correlative conditional (17).

(17) a. The more outrageous a politician's promises (are), the bigger his vote count (is).
b. The more obnoxious Fred *(is), the less attention you should pay to him.
c. The happier the customers *(are) behaving, the more things you should try to sell them. (McCawley 1988:178)

Some of these properties follow from the fact that correlative conditionals are a type of conditional, and some are the properties of comparatives, but some are peculiar to this construction. Fillmore and McCawley argue that the only way to capture these properties is with a construction-specific rule of some sort (a grammatical construction in the technical sense for Fillmore and a construction-specific transformation for McCawley). McCawley notes that the other alternative is to localize the properties of this construction in some of the words involved. However, as he argues, this is unsatisfactory: An account that localized the properties of the construction in some lexical item (*the*? a comparative operator? some kind of empty category?) would miss the connections that correlative conditionals bear to conditionals, on the one hand, and comparatives on the other. Only in a network of related constructions can one represent both the peculiarities of correlative conditionals and the generalizations that they participate in.

2.4.2 Relative clauses

Sag (1997) develops a hierarchy of phrase types which accounts for a wide variety of relative clauses in English. On his analysis (see also Ginzburg and Sag to appear), the type *phrase* is elaborated along two dimensions, CLAUSALITY and HEADEDNESS, as shown in Figure 2.31. According to this hierarchy, all phrases are either non-clausal, or belong to one of the four major clause types (imperative, declarative, interrogative or relative).[36] These clause types each dominate a number of subtypes. It is on the supertypes like *inter-cl* that the message value (illocutionary force) is fixed. The *rel-cl* supertype states that all instances express *propositions*, and further that they modify nouns ([MOD [HEAD *noun*]]). The hierarchy in Figure 2.31 also states that every *phrase* is either non-headed (*non-hd-ph*), or belongs to one of the types of headed phrase, such as the *fin-head-subj-phrase* or the *head-filler-phrase*.

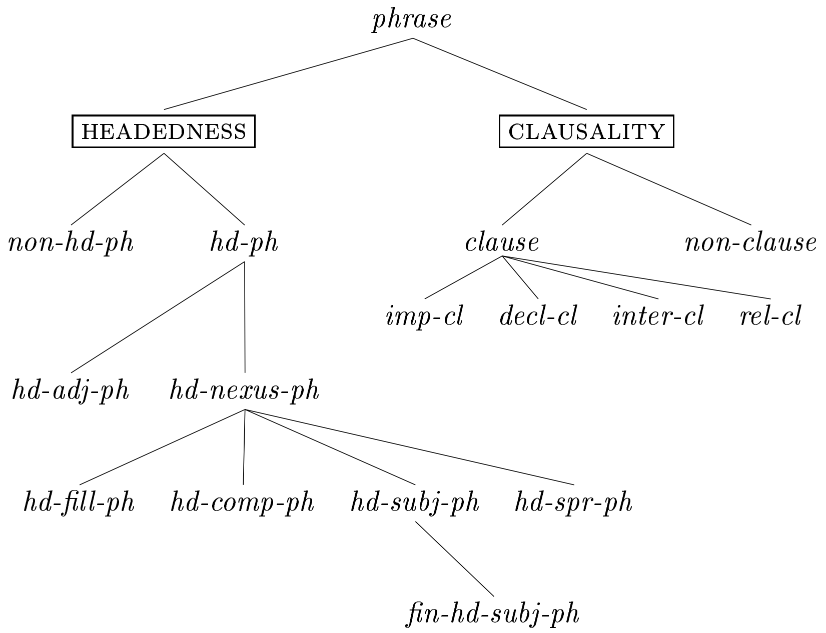

Figure 2.31: Sag's (1997) hierarchy of phrase types

Individual constructions inherit from both sides of this hierarchy, and may or may not add further constraints of their own. In Sag's system, the type *relative-clause* has two immediate subtypes: *wh-relative-clause* and *non-wh-relative-clause*. The distinguishing feature of all *wh* relative clauses is that their non-head daughter contains a *wh* word which is coindexed with the noun the relative clause modifies.

[36]This hierarchy could obviously be extended to include other clause types.

Non-*wh* relative clauses consist of a single slashed head daughter. The item in the slash set of the head daughter is coindexed with the modified noun.

All of the relative clauses considered,[37] inherit from one of the two relative clause types, and some subtype of *headed-phrase*. For example, *wh-subject-relative-clause* (18) inherits from both *wh-relative-clause* and *fin-head-subject-phrase*.

(18) the baker who ate all of the cream puffs

The type *simple-infinitive-relative-clause* (19) inherits from both *non-wh-rel-cl* and *head-comps-phrase*.

(19) the baker to blame for the missing cream puffs

For the details of the analysis, the reader is referred to Sag 1997. What is important here is that with this system Sag is able to capture the generalizations involving all relative clauses and subgeneralizations involving only some types of relative clause, while accounting for such intricate phenomena as pied-piping and the ordering of different types of relative clauses should they modify the same noun. Further, this constructional analysis makes no appeal to phonologically empty relative complementizers assumed in many GB analyses. As such, it is yet another example of a constraint-based analysis obviating the need for a type of empty category.

2.4.3 Summary

To some extent the existence of both constructions and rich lexical entries leaves us with an overabundance of analytical tools. That is, while it is often possible to show that some phenomenon requires the positing of a construction (especially if one is avoiding positing phonologically empty elements), many phenomena could be analyzed as either the effects of an additional (set of) constructions or as some type of lexical alternation. (The exceptions are lexical alternations with morphological expression: These are required to be treated lexically by the Lexical Integrity

[37] Sag does not extend his analysis to free relatives, but notes that such an extension is possible.

Hypothesis of Bresnan and Mchombo 1987, widely accepted by Construction Grammarians.[38]) For example, subject-auxiliary inversion could be a matter of a lexical alternation that places the subject on the complements list (Warner 1993 and others) or a matter of a construction that realizes both the subject and the complements after the auxiliary (Pollard and Sag 1994, Fillmore 1999). Further, constructional analyses, especially those of 'peripheral' phenomena that involve restricted classes of words or other complications, REQUIRE rich lexical entries to represent this information. The situation will be ameliorated somewhat if lexical rules are abandoned in favor of modeling lexical alternations with a type hierarchy. The type-hierarchy system appears to be somewhat more restricted than lexical rules. If so, then a larger class of phenomena may become clearly constructional.

This overabundance of analytical tools is not, however, a defect in the theory. Rather, it is an avenue to new discoveries. That constructions are necessary is shown by the examples discussed above. As discussed in Chapter 6, a construction-based approach to syntax looks to be the most promising for integrating sociolinguistic variation into the theory of grammar. Perhaps it will eventually be possible to develop a set of heuristics for determining when a phenomenon reflects the existence of a construction, and when it reflects a lexical alternation; perhaps both analyses exist in different people's grammars.

2.5 Properties of English auxiliaries

The final aspect of HPSG that I would like to review is the treatment of four major properties of English auxiliaries: negation, inversion, ellipsis and tag questions. Warner (1993, 2000), Kim and Sag (1995), Kim (2000), Bender and Flickinger (1999), and Sag and Wasow (1999) analyze the properties in terms of lexical alternations

[38]In the Construction Grammar literature, morphological rules are also called 'constructions' but crucially obey the Lexical Integrity Hypothesis in that morphological constructions are distinct from syntactic constructions: the constructions that alter the morphological form of a word do not also combine it with other words.

involving the auxiliaries.[39] Inversion has also, however, been analyzed as a special construction or phrase structure schema (Pollard and Sag 1994, Fillmore 1999, Sag 1999). Sag (1999) also proposes a constructional analysis of ellipsis.

In this presentation, I will draw on Warner's (2000) presentation of the negation facts within a multiple inheritance hierarchy and Sag's (1999) alternative constructional account of ellipsis and inversion. I will also update Bender and Flickinger's (1999) analysis of tag questions as a construction related to the inversion and ellipsis constructions. This presentation will focus on the standard variety represented in the literature cited, with some reference to AAVE.

Before turning to the analysis of these properties of auxiliaries it is necessary to first consider what a basic (non-negated, non-inverted, etc.) lexical entry for an auxiliary looks like. As forms of *be* are most relevant here, they will be used to illustrate the whole class.

Following Ross (1967)a and Gazdar et al. (1982), auxiliaries are analyzed as taking VP complements.[40] In the theory assumed here, it follows from this that auxiliaries are raising (or control) verbs. Thus the lexical entry for *be* given in Figure 2.32 is similar in the relevant respects to that given for *tend* in Figure 2.18, page 27.

Be also differs from *tend* in two respects: First, while both *be* and *tend* are verbs, *be* is also an auxiliary verb. They will thus be distinguished by a feature AUX. If all of the auxiliary properties are treated as lexical alternations, it is sufficient to mark *tend* as [AUX −] and *be* as [AUX +]. However, Sag's (1999) constructional analysis reinterprets the feature AUX. Accordingly, I have not given *be* an AUX specification in Figure 2.32 and postpone further discussion of this feature until we get to the analysis of negation below.

More importantly, while *tend* takes an infinitival CP complement, *be* selects for a predicative phrase (coded here as any subject-seeking, complement-saturated constituent that is [PRED +]). Predicative phrases include predicative NPs (20), APs

[39] Lexical alternations have been treated within HPSG variously as lexical rules (Flickinger et al. 1985, Pollard and Sag 1987, Flickinger 1987) and particular interactions of types within a multiple inheritance hierarchy (Riehemann 1999, Kathol 1999, Koenig 1999, Davis and Koenig 2000, Warner 2000).

[40] See also Akmajian et al. (1979) where non-finite auxiliaries take VP complements, although finite auxiliaries still instantiate an Aux node.

$$
\begin{bmatrix}
\textit{lexeme} & \\
\text{PHON} & \langle \text{ be } \rangle \\
\text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{ARG-ST} & \left\langle \boxed{1}, \begin{bmatrix} \text{LOCAL} \mid \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{PRED} & + \end{bmatrix} \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{SPR} & \langle \, \rangle \\ \text{COMPS} & \langle \, \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{KEY} & \boxed{2} \\ \text{RELS} & \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 2.32: Lexical entry for *be*

(21), and PPs (22), and progressive and passive participial VPs (23–24).

(20) Kim is a doctor.

(21) Kim is tired.

(22) Kim is at a party.

(23) Kim is sleeping.

(24) Kim was left behind.

Note that predicative NPs differ slightly in their range of possible internal syntactic structure from non-predicative NPs (25).[41] Similarly, not all adjectives can be used predicatively (26).

(25) a. Dana was advisor to the committee.
b. Mary is too much of a fool to take seriously.
c.*I ran after advisor to the committee.
d.*Too much of a fool to take seriously was appointed to the committee.

[41] From Warner 1993:74.

(26)*This suggestion is mere.

Traditionally, this range of complementation patterns has been broken up into the copula (taking NP, AP, and PP complements), progressive *be* and passive *be*. However, Lapointe (1980), Falk (1984), and Warner (1993) all argue for collapsing all of these complementation patterns as in Figure 2.32. Warner gives two main arguments for this position: First, the passive and progressive participles occur without *be*, with progressive and passive 'meaning' (27–28).[42] (In the case of passive, this meaning is actually just an alternative linking of semantic to syntactic arguments.) Therefore, there is reason to attribute passive or progressive 'meaning' to special *be* lexemes that select these participles as complements.

(27) What, Major droning on again!

(28) Anyone thought to be hiding contraband will be interrogated.

Second, all combinations of complement types shown in (20–24) can appear coordinated as the complement of one single *be*. Some of these combinations are illustrated in (29).[43]

(29) a. He is [$_{\text{AP}}$ very angry indeed] and [$_{\text{VP[prog]}}$ throwing furniture about the room].
b. Paul is [$_{\text{PP}}$ already in the car] and [$_{\text{VP[prog]}}$ waiting for you].
c. I'm [$_{\text{VP[prog]}}$ still expecting to go] and [$_{\text{AP}}$ very keen about the prospect].
d. Paul is [$_{\text{AP}}$ horribly misshapen], [$_{\text{NP}}$ a creature of darkness], and [$_{\text{VP[pass]}}$ thought to practice witchcraft]. Please don't ask him round again.
e. The contraband was [$_{\text{PP}}$ inside the wheel arch] and [$_{\text{VP[pass]}}$ thought to be safely hidden].
f. John was [$_{\text{VP[pass]}}$ put into a state of turmoil by the decision] and [$_{\text{VP[prog]}}$ hoping for its reversal].

[42](27) is from Warner 1993:75.

[43]Falk (1984) attributes this type of example to Thomas Wasow. The examples in (29) are from Warner 1993:24.

The same holds true for AAVE, both in sentences with overt forms of *be* and in copulaless sentences. (30) illustrates the claim for copulaless sentences.[44]

(30) a. Paul already in the car and waitin for you.
b. Paul red as a beet and lookin like he wants to/wanna disappear.
c. We hopin to go and too pumped about it.
d. %Paul ugly, a crook, and known to smoke crack. Don't you ever invite him again.

We now turn to the first property of English auxiliaries: sentential negation.

2.5.1 Negation

Sentential negation in English is expressed by *not* immediately following a finite auxiliary, or by special negated forms of the auxiliaries:[45]

(31) a. Kim is not happy.
b. Kim isn't happy.

Sentential negation is to be distinguished from constituent negation (Klima 1964). In (32), the first *not* represents sentential negation, and the second, constituent negation:

(32) Kim [can [not] [not eat donuts]].

Warner (1993), Kim and Sag (1995), Kim (2000) and Warner (2000) argue for an analysis of (uncontracted) sentential negation in which *not* is an argument of the auxiliary. That is, examples like (31) involve auxiliaries like the one sketched in Figure 2.33, which gives rise to constituent structures like the one shown in Figure 2.34.[46]

The evidence for this analysis includes the behavior of *not* in VP ellipsis, the distribution of *do*, and lexical idiosyncrasy in the relative scope of the auxiliary and *not*.

[44] (30d) was accepted by two consultants, rated marginal by one, and rejected by one. The consultant who rejected it rejects passives with the null copula in general, and accepts (30d) with overt *is*. The one who rated it as marginal remarked that it is better with a pronominal subject. (30c) was only checked with three consultants, and accepted by two and rated as slightly marginal by the other.

[45] See Zwicky and Pullum 1983 for arguments that *-n't* is an affix and not a clitic.

[46] Again, this is a *lexeme* type. Further information needs to be filled in by inflectional types to make a word that can be used in the syntax.

$$\begin{bmatrix} \textit{lexeme} \\ \text{PHON } \langle \text{ is } \rangle \\ \text{SYNSEM} \mid \text{LOCAL} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{FORM} & \textit{fin} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}, \text{ Adv}[\textit{not}], \begin{bmatrix} \text{LOCAL} \mid \text{CAT} \begin{bmatrix} \text{HEAD} & [\text{PRED } +] \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{SPR} & \langle\,\rangle \\ \text{COMPS} & \langle\,\rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 2.33: *Be* with sentential negation

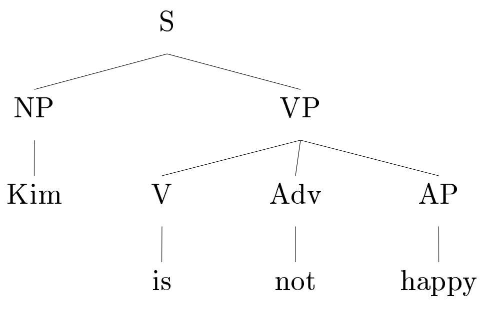

Figure 2.34: Constituent structure of sentential negation

Unlike other VP initial adverbs, *not* may be stranded by ellipsis. Further, *not* may only be stranded by ellipsis when it is preceded by a FINITE auxiliary. That is, only SENTENTIAL negation can be stranded.[47]

(33) a. Kim said he could have heard the news, but Lee said that he could not.
b. *Kim said he could have heard the news, but Lee said that he could have not.
c. *Kim has written a novel, but Lee has never.

In a theory where ellipsis does not involve empty categories, the *not*-as-complement analysis of sentential negation predicts these facts. (33b) and (33c) are out because the modifier (*never* or *not*) has no constituent to attach to. In (33a), on the other hand, *not* is not a (syntactic) modifier, but rather a complement of *could*.

This analysis of sentential negation also provides part of an account of the phenomenon called '*do* support' (Chomsky 1955, 1991). Unstressed (i.e., non-emphatic) *do* occurs in all of the auxiliary constructions, including sentential negation, but not elsewhere:

(34) a. Kim did not leave.
b. Did Kim leave?
c. Sandy didn't consider leaving, but Kim did.
(=Kim did consider leaving).
d.*Kim did leave.
e. Contrary to popular belief, Kim DID leave.

This pattern can be broken down into three facts: (1) Main verbs cannot participate in these constructions. (2) When there is no other auxiliary present, *do* is 'inserted'. (3) Unstressed auxiliary *do* does not appear elsewhere. The first two facts are accounted for by restricting sentential negation and the other auxiliary constructions to the class of auxiliary verbs and by positing a semantically vacuous auxiliary *do*. The first move rules out sentences like (35), and the second ensures that the sentences in (34a-c) have the same meanings as (35a-c) would were they grammatical.[48]

[47] Examples from Kim and Sag 1995:310–311.

[48] See Kim 2000 for arguments that this general approach is better than a transformational '*do*-support' approach.

(35) a.*Kim left not.
b.*Left Kim?
c.*Sandy didn't consider leaving, but Kim (did) consider.

Sag (1999) proposes to account for the third property by reinterpreting the feature AUX. AUX is a head feature of verbs. In much previous work, all auxiliaries were specified as [AUX +] and all main verbs being specified as [AUX −]. On Sag's reinterpretation, main verbs are still [AUX −], but most auxiliaries are now [AUX *bool*], that is, underspecified for AUX. The exception is auxiliary *do*, which is specified as [AUX +].[49] Certain phrasal signs make reference to this feature. The inversion, ellipsis, and tag constructions discussed below all require [AUX +] heads. Negation, still treated as a lexical alternation, also only applies to [AUX +] words. However, Sag's *finite-verb-phrase* (a subtype of *head-complement-phrase*) instead constrains the AUX and NEG values to be the same. That is, the head of a finite verb phrase may be [AUX +] just in case it is also negated (see Figure 2.35).[50]

$$\begin{bmatrix} \textit{finite-verb-phrase} & \\ \text{HEAD-DTR} \mid \text{SYNSEM} \mid \text{LOCAL} \mid \text{CAT} \mid \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AUX} & \boxed{1} \\ \text{NEG} & \boxed{1} \\ \text{INV} & - \end{bmatrix} \end{bmatrix}$$

Figure 2.35: *finite-verb-phrase*, subtype of *head-comps-phrase*

Since most auxiliaries are underspecified for AUX, they can appear in this construction with or without negation. Non-emphatic *do*, on the other hand, is always [AUX +], and as such can only head this construction when it is negated. It is still free to participate in the other constructions (inversion, ellipsis and tag) because they

[49]Auxiliary *do* is not to be confused with main verb *do*, which is always [AUX −]. They also differ in that auxiliary *do* only has finite forms.

(i) Kim didn't do it.
(ii) *Kim should do run.

[50]See the discussion of inversion in §2.5 below for an explanation of the INV feature.

are not subject to the constraint in Figure 2.35. Emphatic *do* is unspecified for AUX, so it can occur as the head of a *finite-verb-phrase* without being negated.

(36) Kim DID leave.

Finally, a lexical analysis of sentential negation is apparently required by the idiosyncratic nature of scopal relations. As first noted by Horn (1972) and explored in detail by Warner (2000), the scope of sentential negation with respect to the semantic relation of modals varies from auxiliary to auxiliary. Assuming the interpretations of the various auxiliaries that Warner does, *can, could,* deontic *may, dare, will, would,* and *need* take wide scope negation, while *must, ought, shall, should* and epistemic *may* and *might* take narrow scope negation. (37) and (38)[51] exhibit wide and narrow scope negation, respectively. Note that the polarity of the tag in (37) forces the sentential negation reading as opposed to the constituent negation (narrow scope) reading. In (38), the ellipsis shows that this is also an instance of sentential negation, even though the negation takes narrow scope.

(37) Paul could not have worked as hard, could he?
[Scope: not(possible)]

(38) — Paul should not have been drinking, should he?
— No, he should not.
[Scope: obligation(not)]

As illustrated in (39) and (40), the auxiliaries exhibit the same scope properties with *-n't*:

(39) Paul couldn't have worked as hard, could he?
[Scope: not(possible)]

(40) Paul shouldn't have been drinking, should he?
[Scope: obligation(not)]

[51] (37–40) are from Warner 2000:176–177.

Warner (2000) also notes that the wide and narrow-neg-scope auxiliaries do not fall into natural semantic classes. He is able to group most of the narrow scope auxiliaries as contributing either a *obligation_rel* or a *may-epistemic_rel*, but even this disjunctive classification has exceptions: *need*, which is covered by this classification belongs in fact to the wide-scope class.[52] Treating the relative scope of *not* and the auxiliaries as a lexical matter makes such lexical idiosyncrasies expected. Further, it allows for the constraints concerning scope to be stated fairly simply, as shown below.

Before turning to the remaining auxiliary properties, I will briefly present a formalized version of Warner's analysis. This version departs somewhat in its details from Warner's own formalization, but remains the same in spirit.

Because the scopal relations for an auxiliary remain the same under both forms of negation, Warner proposes a cross-classification, shown in Figure 2.36.[53,54]

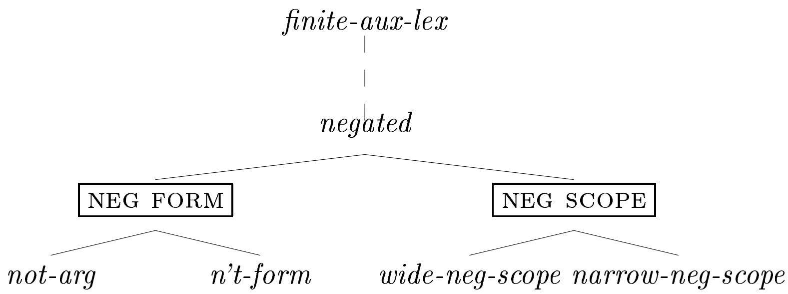

Figure 2.36: Negation subhierarchy

The type *negated* specifies semantic negation. The types under $\boxed{\text{NEG SCOPE}}$ specify the relative scope of negation and the auxiliary, while the types under $\boxed{\text{NEG FORM}}$ specify whether the negation is realized as a *not* complement or as the affix *n't*.

Where I begin to depart from Warner's formalization is in the location of this subhierarchy within the larger hierarchy. For Warner, these are all *lexeme* types. However, the morphology of the *-n't* forms (especially *isn't, wasn't, hasn't*, etc.) show

[52]Warner (arbitrarily) includes *be*, *have*, and *do* in the wide-neg-scope class.

[53]For Warner, this cross-classification is embedded in a larger one, in which all auxiliaries are classified along two dimensions, $\boxed{\text{NEGATION}}$ and $\boxed{\text{INVERSION}}$. Since inversion is treated constructionally here, the hierarchy shown in Figure 2.36 is actually just a subhierarchy of Warner's.

[54]The dashed line connecting *finite-aux-lex* and *negated* indicates that there are intermediate subtypes.

that the type that adds *-n't* can't be a lexeme type: the stem that *-n't* attaches to is a fully inflected word. Warner doesn't go into inflection, but placing the *n't-form* type within the *lexeme* subhierarchy presupposes a simplified view of (inflectional) morphology that almost works for English. On this view, there is a feature STEM in addition to PHON, and lexemes specify only STEM values. Then inflectional types, such as *3sg-present-verb*, specify a relationship between the STEM and PHON values. As Koenig (1999) argues at length, for morphological systems more complex than English inflectional morphology, a constituent structure approach is necessary.[55] In Koenig's model, morphological constituent structure is modeled by types that have a DAUGHTER feature which takes other lexemes as its value.[56]

The most general part of Koenig's hierarchy of signs is given in Figure 2.37.[57] As usual, there are two main subtypes of *sign*: *phrase* and *lexeme*. This latter type is usually named *word*, but Koenig reserves the name *word* for a subtype of *lexeme*, namely those signs that are syntactic atoms—fully inflected words that can go 'out' into the syntax. The other subtypes of *lexeme* are *root* and *stem*. Signs of type *stem* are morphologically complex, but not syntactic atoms. That is, they are incomplete words. Both *word*s and *stem*s are subtypes of *complex-lexeme*, so they both inherit the specification in Figure 2.38. Since the value of the DAUGHTER feature is of type *lexeme*, the embedded sign can be a *word, stem* or *root*. Signs of type *root* are morphological atoms, that is, they are monomorphemic. This type corresponds to the (implicit) type *lexeme* in Warner's (2000) system. That is, Warner's *negated* subhierarchy would be within the *root* subhierarchy. However, the *root* subhierarchy includes much cross-classification (including the part of speech and valence dimensions), but no morphologically complex types.

[55] Koenig also argues that a constituent structure approach alone isn't sufficient. His model incorporates both multiple inheritance hierarchies and morphological constituent structure.

[56] More precisely, Koenig's *complex-lexeme*s have a μ-STRUC feature which takes a value of type μ-*struc*, and μ-*struc*s have a DAUGHTER feature. Since the rest of the μ-STRUC doesn't concern us here, I will use the simpler feature geometry in Figure 2.38.

[57] The $\boxed{\text{MORPH-PHON}}$ partition contains types that model phonological alternations. This aspect of Koenig's theory is not relevant to the discussion at hand.

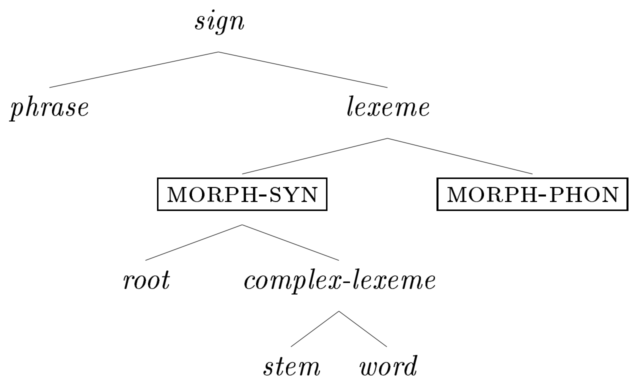

Figure 2.37: Koenig's hierarchy of signs

$$\begin{bmatrix} \textit{complex-lexeme} & \\ \textsc{daughter} & \textit{lexeme} \end{bmatrix}$$

Figure 2.38: Constituent structure of complex lexemes

As noted above, the affix *n't* attaches to fully inflected words. As such, it seems best to model it as a subtype of *word*.[58] This is shown in Figure 2.39. The type *negated* subsumes two partitions or dimensions. This means that any word belonging to the type *negated* must belong to one type from each of $\boxed{\textsc{neg form}}$ and $\boxed{\textsc{neg scope}}$.

Figure 2.40 shows the constraints on the type *negated*.[59] The first thing to note is that this type is only compatible with daughters that are finite non-negated auxiliaries. The mother will also be a finite auxiliary, but specified as [NEG +]. The differing specifications for NEG on the mother and daughter prevent the daughter from also being of type *negated*, i.e., block this type from 'applying' recursively to its

[58]It is tempting to preserve the simpler STEM/PHON model of morphology for English by listing out all of the inflected forms of auxiliaries as separate lexemes. After all, *be* is irregular and as such needs all of its forms listed somehow anyway. However, the are many different uses of *be*, several of which may need their own lexeme types. It would therefore be best if the various forms of *be* were modeled with inflectional types that could 'apply' to all of the different lexemes *be*. In this case, the STEM/PHON system won't work: The inflectional types would already fix the relationship between STEM and PHON, and the *n't-form* type couldn't affect the PHON value further.

[59]For all of these *word* subtypes, I am assuming that a good deal of information, including much of the CONTENT values, is 'copied up' from the daughter to the mother. Ideally, such identities could be stated on a supertype, perhaps as high as *word* itself.

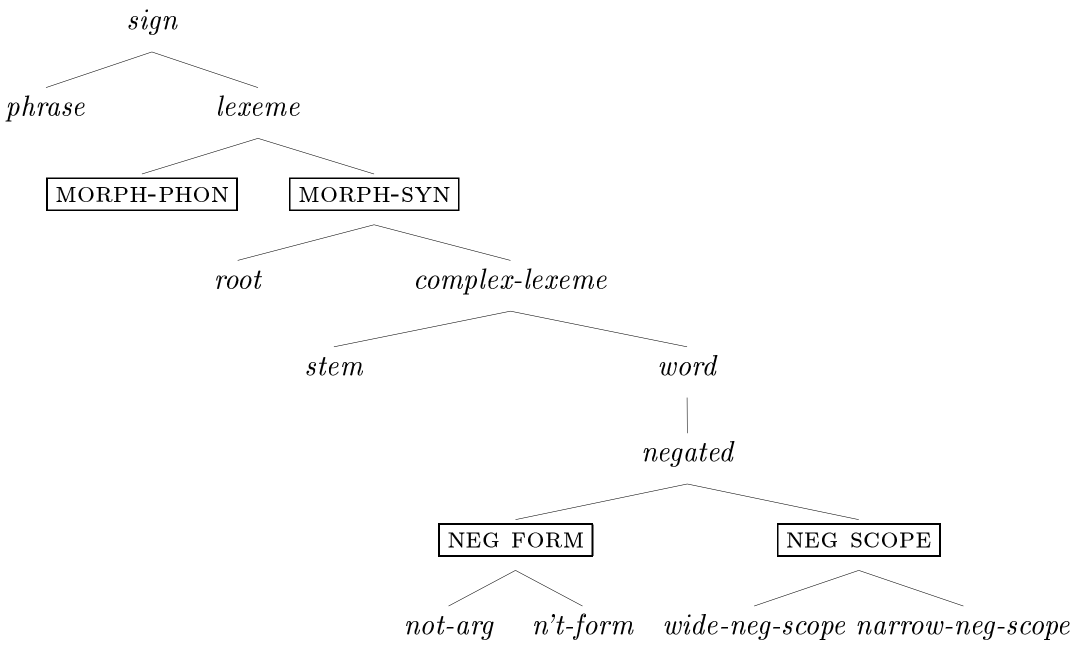

Figure 2.39: Integrating auxiliary negation into Koenig's hierarchy of signs

own output. Note that this means that the type can't be headed, as the head values conflict.[60]

The type *negated* also provides the adverb *not*. However, *not* is not placed directly on the argument structure list. Taking advantage of Koenig's (1999) rearrangement of the ARG-ST feature, the *not* is placed on the ADD-ARG list. The intuition behind these features is that SEM-ARG contains the semantically selected arguments, ADD-ARG contains any additional arguments (such as the expletive subject in the extraposition type), and ARG-LIST is some combination of the two. ARG-LIST is also what constraints such as the argument realization principle should refer to.[61] In Figure 2.40, the SEM-ARG value of the mother is identified with the ARG-LIST value of the daughter. It wouldn't do to put the *not* on the ARG-LIST at this point, because *n't-form* in fact requires that the *not* not appear on the ARG-LIST. However, it is advantageous to introduce *not* on the supertype *negated*, as that way the scope types

[60]It may be possible to make the type headed anyway, with appropriate use of defaults.

[61]Outside of this subsection, I will return to the more familiar ARG-ST notation, with the understanding that this is just an abbreviation for ARG-ST | ARG-LIST.

$$
\begin{bmatrix}
\textit{negated} & \\
\textsc{synsem} & \begin{bmatrix} \textsc{local} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \begin{bmatrix} \textsc{form} & \textit{fin} \\ \textsc{aux} & + \\ \textsc{neg} & + \end{bmatrix} \\ \textsc{arg-st} & \begin{bmatrix} \textsc{sem-arg} & \boxed{A} \\ \textsc{add-arg} & \langle\ \text{Adv}[\textit{not}]\ \rangle \end{bmatrix} \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textsc{key} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\
\textsc{daughter} & \begin{bmatrix} \textit{word} & \\ \textsc{synsem} \mid \textsc{local} & \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{head} & \begin{bmatrix} \textsc{form} & \textit{fin} \\ \textsc{aux} & + \\ \textsc{neg} & - \end{bmatrix} \\ \textsc{arg-st} & \begin{bmatrix} \textsc{arg-list} & \boxed{A} \end{bmatrix} \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textsc{key} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 2.40: Constraints on the type *negated*

can also refer to this element.

Figures 2.41 and 2.42 give the constraints on the types *n't-form* and *not-arg* respectively. The type *n't-form* identifies the SEM-ARG and ARG-LIST values, effectively keeping *not* off the ARG-LIST. Since the word *not* won't appear in the phrase structure anywhere, its semantic contribution has to be incorporated some other way. *N't-form* accomplishes this by adding the RELS value of *not* ($\boxed{C}$) to its own RELS value.[62] The phonological changes are modeled here as a function $f_{n't}$, which takes the phonology of the daughter as its input. The simpler PHON value $\boxed{D} \oplus \langle$ n't $\rangle$ would give the wrong result for *don't, won't, mustn't* and *shan't*. Further, given the function $f_{n't}$, one can stipulate that it is undefined for *am* (cf. **amn't*).

$$\begin{bmatrix} \textit{n't-form} & \\ \text{PHON} & f_{n't}(\boxed{D}) \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{ARG-ST} & \begin{bmatrix} \text{SEM-ARG} & \boxed{A} \\ \text{ADD-ARG} & \langle \begin{bmatrix} \ldots \mid \text{RELS} & \boxed{C} \end{bmatrix} \rangle \\ \text{ARG-LIST} & \boxed{A} \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{RELS} & \boxed{B} \oplus \boxed{C} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{DAUGHTER} & \begin{bmatrix} \text{PHON} & \boxed{D} \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} \mid \text{CONT} \mid \text{RELS} & \boxed{B} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 2.41: Constraints on the type *n't-form*

The type *not-arg* (Figure 2.42) is somewhat simpler. Since it has no phonological effects, the PHON values of the mother and daughter are identified ($\boxed{B}$). Further, as *not* will appear as an independent word in the phrase structure when auxiliaries inheriting from this type are used, nothing special need be said about the semantics of *not*. The type *not-arg* does, however, place the *not* onto the ARG-LIST of the mother ($\boxed{3}$).

[62] $\boxed{C}$ in Figure 2.41 must be the RELS value of *not* because *n't-form* inherits from *negated* the constraint [ADD-ARG $\langle$ Adv[*not*] $\rangle$].

$$\begin{bmatrix} \textit{not-arg} & \\ \text{PHON} & \boxed{B} \\ \text{SYNSEM} & \begin{bmatrix}\text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{ARG-ST} & \begin{bmatrix} \text{SEM-ARG} & \langle \boxed{2} \rangle \oplus \boxed{A} \\ \text{ADD-ARG} & \langle \boxed{3} \rangle \\ \text{ARG-LIST} & \langle \boxed{2}, \boxed{3} \rangle \oplus \boxed{A}\end{bmatrix}\end{bmatrix} \\ \text{CONT} & \begin{bmatrix}\text{RELS} & \boxed{C}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{DAUGHTER} & \begin{bmatrix}\text{PHON} & \boxed{B} \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{CONT} \mid \text{RELS} & \boxed{C}\end{bmatrix}\end{bmatrix}$$

Figure 2.42: Constraints on the type *not-arg*

Figures 2.43 and 2.44 show the constraints on *wide-neg-scope* and *narrow-neg-scope* respectively. Both pick up the relevant features the possibility that something scopes in between the handle of the *not_rel* from the ADD-ARG value,[63] as this is the only place it is uniformly available in both *n't-form* and *not-arg*. The different scope relationships are represented as one relation taking the other as its argument. In *wide-neg-scope*, the handle of the auxiliary's main relation ($\boxed{1}$) is identified with the argument of the *not_rel*. In *narrow-neg-scope*, the handle of the *not_rel* is identified with the argument of the auxiliary. Further, the argument of the *not_rel* is identified with the handle of the complement of the auxiliary ($\boxed{2}$).[64] The KEY specifications are meant to constrain each type to the relevant auxiliaries. As noted by Warner, these classes still have exceptions, which will need to be listed in some way.

[63] The '...' before KEY in these avms indicate an abbreviated feature path.

[64] This is still abstracting away from the possibility of quantifiers. In fact, the relevant HANDLE values are constrained to be out-scoped by or equal to the relevant ARG values, to allow for the possibility that some quantifier scopes in between. (This is why handles, and not indices, must be used for this purpose. For details, see Copestake et al. 1999.) This also means that the constraints on *narrow-neg-scope* do not contradict the constraint (on the base entries for the auxiliaries) that the handle of the auxiliary's complement is out-scoped by or equal to the argument of the auxiliary's relation.

$$
\begin{bmatrix}
\textit{wide-neg-scope} \\
\text{SYNSEM} \mid \text{LOCAL} \begin{bmatrix}
\text{CAT} \begin{bmatrix} \text{ARG-ST} \mid \text{ADD-ARG} \ \left\langle \begin{bmatrix} \ldots \mid \text{KEY} \begin{bmatrix} \text{ARG} & \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\
\text{CONTENT} \begin{bmatrix} \text{HANDLE} & \boxed{1} \\ \text{KEY} & \begin{bmatrix} \neg \ (\textit{oblig\_rel} \vee \textit{may-epistemic\_rel}) \end{bmatrix} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Figure 2.43: Constraints on the type *wide-neg-scope*

$$
\begin{bmatrix}
\textit{narrow-neg-scope} \\
\text{SYNSEM} \mid \text{LOCAL} \begin{bmatrix}
\text{CAT} \begin{bmatrix} \text{ARG-ST} \mid \text{ADD-ARG} \left\langle \begin{bmatrix} \ldots \mid \text{KEY} \begin{bmatrix} \text{HANDLE} & \boxed{1} \\ \text{ARG} & \boxed{2} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{SEM-ARG} \left\langle [\,], \begin{bmatrix} \ldots \mid \text{KEY} \begin{bmatrix} \text{HANDLE} & \boxed{2} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\
\text{CONTENT} \begin{bmatrix} \text{KEY} & \begin{bmatrix} \textit{oblig\_rel} \vee \textit{may-epistemic\_rel} \\ \text{ARG} \ \boxed{1} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Figure 2.44: Constraints on the type *narrow-neg-scope*

Ain't

At this point, a few words are in order about the status of *ain't* in AAVE and how it fits in with the analysis of sentential negation reviewed here. Weldon (1994) argues, on the basis of patterns of variation as well as specific attested utterances, that *ain't* serves as an inflected form of the auxiliaries *be*, *have*, and past tense *do*. Examples (41–43), from Weldon 1994:362–363 illustrate each of these use of *ain't*.

(41) They ain't 'posed to know we taping.

(42) He ain't started my grill yet.

(43) I ain't give you none, Boo, did I?

The alternative analysis Weldon argues against is Debose's (1994) proposal that *ain't* (as well as *not*) is a tense/aspect neutral auxiliary, separate from *be*, *have*, and *do*.

The syntactic evidence that Weldon adduces for her claim is as follows. First, *ain't* can appear in matrix polar questions and tag questions—environments that require a tensed auxiliary.[65] In this, it contrasts with *not*, although *ain't* and *not* might appear to occupy the same position in (46).

(44) a. Ain't that little boy cute?
b. *Not that little boy cute?

(45) a. That stuff is still in the refrigerator, ain't it?
b. *That stuff is still in the refrigerator, not it?

(46) a. They ain't best friends.
b. They not best friends.

Secondly, when *ain't* is used as a past tense form of *do*, it gets the matching tag, as shown in (43) above. Tags that might go with other uses of *ain't* won't work here (Weldon 1994:387).

(47)*I ain't give you none, Boo, do I?/am I?/have I?

[65](44) and (45) are from Weldon 1994:378.

n't-form

reg-n't-form *ain't-form*

Figure 2.45: Two subtypes of *n't-form*

It's true that forms of *do* are the default for tags in the sense that tags on sentences headed by main verbs always take *do*. However, if *ain't* were truly a tense/aspect neutral auxiliary, one would expect it to be able to take *have*, *be* or present tense *do* tags regardless of the 'use' it was being put to in the matrix clause.

Finally, if *ain't* is indeed a form of *do* only in the past tense,[66] then it is not, in fact, completely neutral with respect to tense.

The most straightforward way to include these facts in the analysis of negation reviewed here is through the function $f_{n't}$. This is the function that determines the phonological form of auxiliaries with contracted *n't*. Even without *ain't*, this function must accommodate idiosyncratic phonology, e.g., *won't*/**willn't* (Zwicky and Pullum 1983). It can easily be extended to allow *ain't* as a possible output form for the inputs *is, are, has, have* and *did*.[67]

Another possibility is that the type *n't-form* in Figure 2.41 has two subtypes, as shown in Figure 2.45. On this conception, the type *n't-form* would leave its PHON value unspecified. The type *reg-n't-form*[68] would be specified as [PHON $f_{n't}(\boxed{D})$], assuming the same $f_{n't}$ as the earlier *n't-form* did. The type *ain't-form* would instead be specified as [PHON *ain't*]. This type would have to require feature specifications on its DAUGHTER that restrict it to *be*, *have* and past tense *do*. Alternatively, the PHON value of *ain't-form* could be given by a function $f_{ain't}$. $f_{ain't}$(have), etc., would give *ain't*, while $f_{ain't}$(would), etc., would be undefined. Note that, since this is a morphological type, positing one type that can produce the *ain't* forms of *have, be* and *do* is not the same as positing a tense/aspect neutral auxiliary *ain't*. The *ain't*

[66]Weldon (1994) finds no examples of *ain't* in present-tense *do* contexts in her data.

[67]Such morphological functions are not presumed to provide a unique output for every input, but rather can give two or more variant outputs for one input.

[68]So named because the relationship between *will* and *won't* is at least more predictable than the relationship between *has* and *ain't*.

form of *have* would retain the semantic and morphosyntactic features of *have* (with the exception of the person/number features), and likewise for the *ain't* forms of *be* and *do*.

The advantage of positing the type *ain't-form* over dealing with *ain't* within the function $f_{n't}$ is that *ain't* is not merely an alternative form of *hasn't/isn't*, etc. It is highly marked in its social/stylistic value. Consider, for example, this description of one use of *ain't* from Anna Quindlen's column entitled "It's the Cult of Personality" in *Newsweek*, August 14, 2000.

> Brokaw and Bush, two guys just standing around talking. Shirtsleeves. Sunshine. Fence posts. You get the idea. The candidate was hunkered down at the ranch, going *mano a mano* for a couple of endless, empty minutes with the anchorman. The governor showed off his Yiddish—"kibitzing" was how he described what he was doing with his father, the former president—but the seminal moment was in fluent good-ole-boy. "I know you are a pretty good fisherman," Bush said to Brokaw, who was angling for the name of the as yet unknown vice-presidential nominee. "Yes, you are, but I ain't catching."
>
> Take a good look at that verb, fellow voters, and consider what the meaning of "ain't" ain't. It ain't good English, of course, and it ain't necessarily an entirely natural locution for a graduate of Andover and Yale, even by way of west Texas. What it is is a marker for the most important issue of this election. Relaxed, a little irreverent, down-home: that "ain't" is supposed to communicate a whole tractor load of material about the Bush personality.

Of course, this is a description of the social and stylistic value of *ain't* in a certain kind of white speech. There's no reason to expect it to be the same in AAVE. However, given the stigmatization and stereotyping of *ain't* it would be surprising for it to have no social or stylistic value in AAVE. If, as argued in Chapter 6, social value is indeed a matter of grammar, the type *ain't-form* provides a place to record it.

2.5.2 Inversion

The next property of English auxiliaries is inversion. Following Pollard and Sag (1994), Fillmore (1999) and Sag (1999), I adopt here a constructional analysis of subject-auxiliary inversion. On this analysis, inverted structures are built by a separate phrase structure rule that realizes both the subject and the complement(s) after the auxiliary. A preliminary version of the specific rule assumed here (to be revised in the discussion of ellipsis below) is given in Figure 2.46.[69]

$$\left[\begin{array}{ll} \textit{sai-phrase} & \\ \text{PHON} & \boxed{\text{B}} \oplus \textit{list} \\ \text{SYNSEM} & \left[\begin{array}{ll} \text{LOC} \mid \text{CAT} & \left[\begin{array}{ll} \text{SUBJ} & \langle\,\rangle \\ \text{COMPS} & \langle\,\rangle \end{array}\right]\end{array}\right] \\ \text{HD-DTR} & \left[\begin{array}{ll} \text{PHON} & \boxed{\text{B}} \\ \text{SYNSEM} & \left[\begin{array}{ll} \text{LOC} \mid \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{INV} & + \\ \text{FORM} & \textit{fin} \end{array}\right] \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{COMPS} & \textbf{sts}(\boxed{\text{A}}) \end{array}\right]\end{array}\right]\end{array}\right] \\ \text{NON-HD-DTRS} & \langle \boxed{1} \rangle \oplus \boxed{\text{A}} \end{array}\right]$$

Figure 2.46: Subject-auxiliary inversion construction (preliminary version)

The crux of this analysis is the feature INV, introduced by Gazdar et al. (1982) and further justified in Gazdar et al. (1985). The *sai-phrase* requires an [INV +] element as its head. All main verbs are specified as [INV −]. It follows that the *sai-phrase* will always be an auxiliary and the construction need not specify [AUX +]. Both INV and AUX are necessary features, however: While most auxiliaries are underspecified for the feature INV, there are exceptions both ways. The auxiliary *better* (48) is [INV −], while 1st singular *aren't* (49) is [INV +] (Sag 1999).[70]

(48) a. You better not do that.
b.*Better I do that?.

[69]This construction is a subtype of *headed-phrase* and, as such, inherits the constraints shown in Figure 2.13, page 21. The constraint on the PHON value of this construction interacts with the constraints it inherits from *headed-phrase* to ensure that the PHON value of an *sai-phrase* is the PHON value of the head daughter followed by the PHON values of the non-head daughters (i.e., the subject and then the complements).

[70]Sag also cites futurate *shall* and *mightn't* with wide scope negation as [INV +] forms.

(49) a. Aren't I lucky?
b.*I aren't lucky.

In order to capture the facts in (49), and to block VPs (as opposed to lexical verbs) from heading the *sai-phrase* (50), the *finite-verb-phrase* construction must constrain its head to be [INV −] (cf. Figure 2.35, page 53).

(50)*[[Is happy] Kim]?

Note that subject-auxiliary inversion is not limited to matrix polar questions. In a more complete grammar, this *sai-phrase* would be a supertype to a polar question construction, an inverted conditional construction (51), and a blessings/curses construction (52) (Fillmore 1999).[71]

(51) Had we known better, we would never have come.

(52) a. May all your wildest dreams come true.
b. May your teeth fall out on your wedding day.

Likewise, it is involved in the negative inversion constructions in both Standard English (Fillmore 1999) and AAVE (Labov et al. 1968, Sells et al. 1996).[72]

(53) Never have we seen such a sight.

(54) Can't nobody say nothin to dem peoples.

[71] These constructions, of course, impose further constraints, including the restriction to certain auxiliaries for the inverted conditional construction (i) and the blessings/curses construction (ii).

(i) *Did we know better, we would never have come.
(ii) Will all your teeth fall out.

While grammatical, (ii) is not an instance of the blessings/curses construction.

[72] (54) is a naturally occurring example collected by Sells et al. (1996):592.

2.5.3 Ellipsis

Following Sag (1999), I adopt a constructional analysis of ellipsis. The description of *ellipsis-phrase* is given in Figure 2.47. This construction, a subtype of *headed-phrase*, is similar to the *head-complements-phrase* in that it (potentially) realizes some complements, but passes up the subject requirement. However, the relationship between the COMPS specification of the head daughter and the NON-HD-DTRS value is less direct than in the *head-complements-phrase*. Here, the NON-HD-DTRS ($\boxed{B}$) correspond to whatever is left of the COMPS list after some (non-zero) number of complements are elided (*nelist*(*elided*)).

$$\begin{bmatrix} \textit{ellipsis-phrase} & \\ \text{SS} \mid \text{LOC} \mid \text{CAT} & \begin{bmatrix}\text{SUBJ} & \boxed{A}\\ \text{COMPS} & \langle\,\rangle\end{bmatrix} \\ \text{HD-DTR} \mid \text{SS} \mid \text{LOC} \mid \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{verbal} & \\ \text{AUX} & +\end{bmatrix}\\ \text{SUBJ} & \boxed{A}\\ \text{COMPS} & \textbf{sts}(\boxed{B}) \oplus \textit{nelist}(\textit{elided})\end{bmatrix}\\ \text{NON-HD-DTRS} & \boxed{B}\end{bmatrix}$$

Figure 2.47: Ellipsis construction

That ellipsis can affect some but not all of the complements of an auxiliary is shown by the examples in (55–58), from Warner 2000:205.[73]

(55) Are there any first-year students angry about their grades?
—Yes, there are some.
—No, there aren't any.

It is also possible to leave out the first complement of the *there*-copula, as shown in (56–58), from Warner 2000:205–206.

(56) Are there any first-year students angry about their grades?
—No, but there are upset with their teachers.

[73]These examples all involve non-locative second complements. As Warner shows, the locative second complements appear to be optional, so that eliding them doesn't place the same requirements on the preceding linguistic context that ellipsis usually does.

(57) He didn't tell me there was any treasure buried in the garden, but I wonder if there is hidden in the orchard.

(58) There is a complex variable hidden in the first formula, so I wonder if there is lurking in the second.

Warner takes these to be examples of ellipsis, but it is not clear how one could distinguish examples like (56) through (58) from examples of pseudo-gapping, like (59), from Miller (1992:94).

(59) John spoke to Mary more often than Peter did to Ann.

Accordingly, I will assume that (56–58) are instances of a separate construction (pseudo-gapping), and that ellipsis can only affect a right-most sublist of the COMPS list.

However, as shown in (60), *not* cannot be elided, even when it appears as the complement of an auxiliary and everything after it is elided.

(60) a. Kim likes bagels, but Sandy does not.
b. Kim does not like bagels, and Sandy does not, either.
c. #Kim likes bagels, but Sandy does.
d. #Kim does not like bagels, and Sandy does, either.

To capture this fact, the elided elements in an instance of *ellipsis-phrase* are unified with the type *elided*, shown in Figure 2.48. Since *not* is arguably the only word selected as a complement by English auxiliaries, the ellipsis of *not* can be blocked by ensuring that *elided* only unifies with the *synsem*s of phrases. The constraints shown in Figure 2.48 accomplish this by taking advantage of the fact that the CAT values of words and phrases are already distinguished. Since ARG-ST is only appropriate for words,[74] we need the subhierarchy in Figure 2.49. Since *phrasal-cat* and *lexical-cat* are sister types in this hierarchy, they are incompatible. *Not*, as a word, will have a CAT value of type *lexical-cat* and its *synsem* won't be able to unify with the type *elided*.

[74]This is crucial to the theory of locality of selection: ARG-ST values, unlike SUBJ/SPR/COMPS values don't get canceled off. If they were to project up the tree, words could select for properties such as the case of the complement of a complement.

$$\begin{bmatrix} \textit{elided} & \\ \textsc{local} & \begin{bmatrix} \textsc{cat} & \textit{phrasal-cat} \\ \textsc{cont} & \ldots \end{bmatrix} \end{bmatrix}$$

Figure 2.48: Constraints on the type *elided*

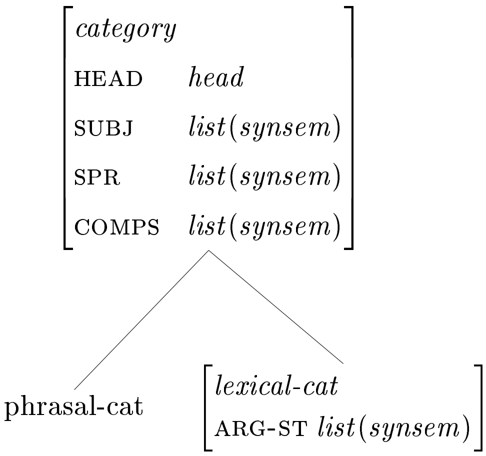

Figure 2.49: Subhierarchy of *category* types

There will also be semantic constraints, on the type *elided* (' ... ' in Figure 2.48) and/or on the type *ellipsis-phrase*, that model the interpretation of the elided phrases and their dependence on preceding linguistic context. Finally, the type *elided* is taken to be a subtype of *non-canonical* synsem, similar to the type *gap* (see Figure 2.50).

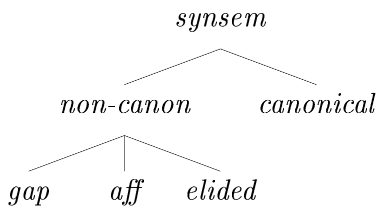

Figure 2.50: Subhierarchy of *synsem* types

The type *ellipsis-phrase* only produces ellipsis in non-inverted clauses. However, ellipsis is also possible in inverted clauses, as shown in (61).

(61) Everyone else has pledged their support. Will you?

As with non-inverted clauses, ellipsis affects a right-most sublist of the complements of the auxiliary:

(62) There aren't many second-year students angry about their grades. Are there any first-years?

It turns out that these can be modeled by making a change to the existing *sai-phrase*. As shown in Figure 2.51, all that is required is that the COMPS value of the head daughter be similar to that of the head daughter in the *ellipsis-phrase*. They differ, however, in that the list of elided synsems in the *sai-phrase* may be the empty list. This allows this single *sai-phrase* to model both elliptical and non-elliptical inverted clauses.

The ability to model both elliptical and non-elliptical inverted clauses with one type is especially desirable given that ellipsis is possible in more than one subtype of *sai-phrase*.[75] (For more on the subtypes of *sai-phrase*, see Fillmore 1999.)

[75]If the semantic constraints on ellipsis involve the whole phrase and not just the type *elided*, this *sai-phrase* may in fact need to be separated into two, one for ellipsis and one for other inversion constructions. However, with lexical amalgamation of relations, it should be possible to state all of the necessary conditions on the type *elided*.

$$\begin{bmatrix} \textit{sai-phrase} & \\ \text{SS} \mid \text{LOC} \mid \text{CAT} & \begin{bmatrix}\text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle\end{bmatrix} \\ \text{HD-DTR} \mid \text{SS} \mid \text{LOC} \mid \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix}\textit{verb} & \\ \text{INV} & + \\ \text{FORM} & \textit{fin}\end{bmatrix} \\ \text{SUBJ} & \langle \boxed{2} \rangle \\ \text{COMPS} & \textbf{sts}(\boxed{A}) \oplus \textit{list}(\textit{elided}) \end{bmatrix} \\ \text{NON-HD-DTRS} & \langle \boxed{2} \rangle \oplus \boxed{A} \end{bmatrix}$$

Figure 2.51: Subject-auxiliary inversion construction (final version)

(63) a. Had we known what was in it, we wouldn't have opened it.
b. —There was no way to know what was in it.
—Yes, but had we, we wouldn't have opened it.

2.5.4 Tag questions

Bender and Flickinger (1999) analyze tag questions by means of a lexical rule that produces the auxiliaries that head tags. The constructional analysis of ellipsis and inversion developed here allows for an analysis of tags as produced by a construction, and in particular, a subtype of *sai-phrase*.

It may appear that tag questions such as (64) are just special uses of elliptical inverted sentences.

(64) Bob left early today, didn't he?

However, there are at least two kinds of evidence to support a separate tag construction. First, the subject of a tag must be a pronoun. Another referentially dependent expression, such as an epithet, won't do:

(65)*Bob left early, didn't the slacker?

Second, the subject of the tag is always coindexed with the subject of the clause it modifies, and not any other, potentially more salient, NP:

(66) a. There were three people there, weren't there?
b.*There were three people there, weren't they?

Figure 2.52 shows the constraints on the type *tag-phrase*. As a subtype of *sai-phrase*, it inherits all of the constraints in Figure 2.51.

$$
\begin{bmatrix}
\textit{tag-phrase} & \\
\text{SS} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} \mid \text{MOD} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AGR} & \boxed{5} \end{bmatrix} \\ \text{SUBJ} & \langle\,\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{MSG} & \begin{bmatrix} \textit{propositional} & \\ \text{SOA} & \boxed{6} \end{bmatrix} \\ \text{KEY} & \boxed{2} \\ \text{TENSE} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{MSG} & \begin{bmatrix} \textit{tag\_rel} & \\ \text{SOA} & \boxed{6} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\
\text{HD-DTR} \mid \text{SS} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \\ \text{SUBJ} & \left\langle \boxed{4} \begin{bmatrix} \textit{pron-synsem} & \\ \text{LOC} & \begin{bmatrix} \text{CAT} \mid \text{HEAD} \mid \text{AGR} & \boxed{1} \\ \text{CONT} \mid \text{INDEX} & \boxed{5} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{KEY} & \boxed{2} \\ \text{TENSE} & \boxed{3} \end{bmatrix} \end{bmatrix} \\
\text{NON-HD-DTRS} & \langle \boxed{4} \rangle \oplus \textit{list}(\text{Adv}[\textit{not}])
\end{bmatrix}
$$

Figure 2.52: Constraints on the type *tag-phrase*

The first thing to note is that, in addition to the constraints relating the COMPS value of the head daughter to the NON-HD-DTRS value inherited from *sai-phrase*, the *tag-phrase* further stipulates that only complements that are the word *not* may be realized. This allows the somewhat stilted sentences in (67) while correctly ruling out (68).

(67) a. Bob left, did he not?
b. There are several people angry about this, are there not?

(68) a.*Bob left, didn't he leave?
b.*There are several people angry about this, are there not several people?

Thus this construction solves a problem left unresolved by the analysis of Bender and Flickinger (1999): namely, the possibility of retaining complement *not* in tags. The other open question, how to model the polarity effects shown in (69) still awaits a theory of polarity phenomena for its resolution.

(69) a. Sara slept, didn't she?
b. Sara didn't sleep, did she?
c. Sara slept, did she? (challenge tag)
d.*Sara didn't sleep, didn't she?

The rest of the constraints on this type are analogous to those posited in Bender and Flickinger 1999 and motivated there. They will be only briefly reviewed here.

The MOD specification indicates that tag phrases modify declarative sentences. This accounts for their distribution, and also serves as a means to constrain the tag to agree with the sentence it modifies in several ways.[76] In particular, the auxiliary in the tag must match the auxiliary in the sentence, or, if the sentence is headed by a main verb, be a form of *do*. On Bender and Flickinger's analysis, this is done by matching the KEY values of the tag auxiliary and the modified sentence.[77] Tags also agree with their sentences in tense, modeled here by a feature TENSE.[78]

[76]The feature MSG (message) is a representation of illocutionary force. A verb can select for a declarative clause by constraining its complement to be [MSG *propositional*]. The semantic contribution of the tag is also represented as a message, or type of illocutionary force. For more on clause types and their semantics, see Ginzburg and Sag to appear.

[77]To make this work, Bender and Flickinger assume that forms of *be* are not semantically vacuous, but do contribute some relation, that all modals have unique KEY values, and that *do* bears a KEY value that unifies with the KEY value of any main verb, but no auxiliaries.

[78]Another example of a construction that constrains tense is the set of 'introducers' *it's time*, *it's about time* and *it's high time*, which generally require past tense in the following indicative clause (Fillmore 1988:51).

(i) It's high time you started thinking about your future.

The third way in which the tag must agree with its sentence is that the subject of the tag must be coindexed with the subject of the sentence. However, as tags are sentence modifiers (as opposed to VP modifiers), information about the subject of the sentence is not directly available through the MOD value. Following Kathol (1999), Bender and Flickinger use the agreement facts to motivate a head feature of nouns and verbs called AGR, that makes this information available at the sentence level.

While most nouns identify their AGR value with their semantic index, the singular *they* series of pronouns do not. The values of AGR and INDEX in Figure 2.52 are designed to account for sentences like (70).

(70) Everyone wins, don't they?

2.6 Summary

This chapter has presented an overview of the framework of Construction-HPSG, with particular attention to analyses that obviate the need for empty categories and the treatment of English auxiliaries. With this as background, we are ready to turn to the analysis of copula absence in AAVE.