

HPSG: Background

2.1 Introduction

In this chapter, we outline the basics of a particular version of HPSG. We begin with the lexicon and its organization, turning next to the important consequences of our decision to model phrases as feature structures. A multidimensional hierarchy of phrases is introduced, along with a sketch of how this allows cross-classifying generalizations about constructions to be expressed. We build up an account of simple finite clauses—both indicative and subjunctive, and then extend this account to subjectless infinitival clauses. Finally, we provide a few examples of lexical entries whose complement selection properties can be simplified, given the semantic types associated with the clausal constructions presented here. We leave until Chapter 5 a presentation of our treatment of filler-gap constructions, the inheritance of *wh*-specifications, and an account of quantifier scoping, all of which play a role in the analyses of interrogatives developed in subsequent chapters.

2.2 Feature Structures

Utterances in HPSG are modeled as feature structures of type *sign*. Since the features associated with structures of this type include PHONOLOGY and SYNSEM, the latter specifying both syntactic and semantic information, the constraints we impose on signs correspond to the general conventions governing the sound-syntax-meaning relation in a given language. A system of signs thus provides a finite specification of an infinite set of utterance kinds.

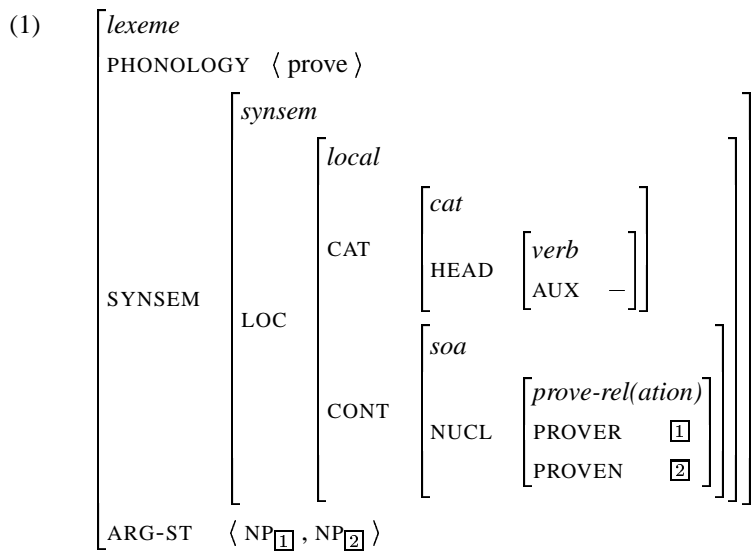
But linguistic information can be complex. Within the feature structures specified as values of SYNSEM, numerous grammatical distinctions must be made: *noun* vs. *verb* vs. other parts of speech; *nom* vs. *acc* case; *vowel* vs. *consonant*, and so forth. In order to make such distinctions, a grammar must posit many kinds of linguistic entities ‘smaller’ than the signs, and must provide an account of the specific properties of each such kind. The grammar of a language thus must include:

- an enumeration of the set of types (sometimes called ‘sorts’) that play a role in the grammar—a linguistic ontology,
- a statement of which features are appropriate for each type,
- a statement what type of value is appropriate for each such feature, and
- a specification of all constraints that instances of particular types must satisfy (usually referred to simply as ‘type constraints’).

The modeling assumptions of HPSG have provided a novel way of working with certain traditional notions of grammar (e.g. ‘lexical entry’ and ‘phrase structure rule’) that allows increased precision and analytic uniformity. Lexical entries are descriptions of (or constraints on) feature structures that belong to the type *word*; construction rules (or phrase structure rules, or ‘immediate dominance schemata’) are partial descriptions of feature structures of type *phrase*. These are the immediate subtypes of the type *sign*. Hence, the lexical entries and construction rules work together: the lexical entries define a set of words; the construction rules define a set of phrases built from words or phrases. A language can then be viewed in various ways, for example as the set of feature structures of type *sign* that satisfy the constraints of the grammar, the set of feature structures of type *phrase* that satisfy all relevant constraints, as the set of PHONOLOGY values of those signs, etc. Each such characterization may have its own utility.

2.3 Words

In the case of words, then, an HPSG grammar specifies an inventory of lexical types and the various constraints that instances of those types must obey. We follow the common practice of formulating lexical descriptions (constraints on objects of type *lexeme* or *word*) in the language of attribute-value matrices (AVMs):¹



Lexical descriptions like (1) specify complexes of phonological, syntactic and semantic information that are satisfied by a family of feature structures. These feature structures are structured according to a particular nonarbitrary feature geometry. For example, *synsem* objects (the syntactico-semantic complexes that serve as values of the feature SYNSEM) encapsulate precisely the information that heads can select for, and thus play a key role in the HPSG account of the locality of subcategorization. Similar considerations motivate the supposition of *local* objects (these encapsulate the information transmitted in raising and extraction dependencies), and the other embedded feature structures illustrated in (1).

¹We abbreviate feature names as needed, e.g. CAT (CATEGORY), CONT (CONTENT), ARG-ST (ARGUMENT-STRUCTURE), NUCL (NUCLEUS), LOC (LOCAL) and SS (SYNSEM). For a complete list of abbreviations used in this book, see the List of Abbreviations.

The feature geometry illustrated in (1) is a consequence of the linguistic ontology specified by the grammar. In particular, a grammar provides a complete specification of what types of feature structure exist and how those types are organized into a hierarchy, i.e. for each type, what its immediate supertypes (IST) are. The grammar also specifies which features are appropriate for each grammatical type, as well as what type of value is appropriate for each feature, as in (2):²

(2)

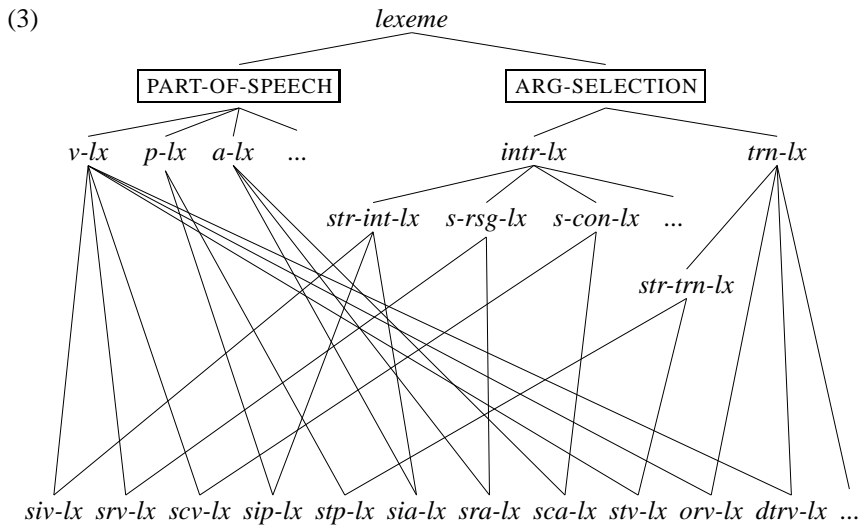
TYPE	FEATURES/TYPE OF VALUE	IST
<i>sign</i>	$\left[\begin{array}{ll} \text{PHONOLOGY} & \text{list}(\text{form}) \\ \text{SYNSEM} & \text{synsem} \\ \text{CONTEXT} & \text{conx-obj} \end{array} \right]$	<i>feat-struct</i>
<i>phrase</i>	...	<i>sign</i>
<i>lex-sign</i>	$\left[\text{ARG-ST} \quad \text{list}(\text{synsem}) \right]$	<i>sign</i>
<i>lexeme</i>		<i>lex-sign</i>
<i>word</i>		<i>lex-sign</i>
<i>synsem</i>	$\left[\begin{array}{ll} \text{LOCAL} & \text{local} \\ \text{SLASH} & \text{set}(\text{local}) \\ \text{WH} & \text{set}(\text{scope-obj}) \\ \text{BCKGRND} & \text{set}(\text{fact}) \end{array} \right]$	<i>feat-struct</i>
<i>local</i>	$\left[\begin{array}{ll} \text{CATEGORY} & \text{category} \\ \text{CONTENT} & \text{sem-object} \\ \text{STORE} & \text{set}(\text{scope-obj}) \end{array} \right]$	<i>feat-struct</i>
<i>category</i>	$\left[\begin{array}{ll} \text{HEAD} & \text{part-of-speech} \\ \text{SUBJ} & \text{list}(\text{synsem}) \\ \text{COMPS} & \text{list}(\text{synsem}) \\ \text{SPR} & \text{list}(\text{synsem}) \end{array} \right]$	<i>feat-struct</i>
...

We will add further details about types, features, and type constraints as we present particular analyses. The type system for the English grammar fragment developed in this book is summarized in Appendix A.

²The feature WH corresponds to the feature QUE of Pollard and Sag (1994). This feature has two related functions: (1) to distinguish interrogative and exclamative *wh*-words from all other words, and (2) to distinguish phrases containing such words from those that do not. WH and the feature SLASH—used to encode information about extracted elements in filler-gap constructions—are discussed in Chapter 5. The feature BACKGROUND (BCKGRD), which plays a significant role in the analyses of Chapters 7 and 8, is discussed in Chapters 3, 4 and 5. Finally, it should be noted that in the feature geometry we assume, the feature CONTEXT (CTXT) is declared at the level of SIGN, not LOCAL, as in previous proposals. Nothing essential hinges on this aspect of our analysis. *Conx-obj* abbreviates *contextual-object*.

This particular organization of linguistic information seeks to provide an account of the empirical fact that subcategorization (category selection in the familiar sense), case and role assignment, semantic selection, and head-valent agreement all operate in highly constrained local domains. Agreement with or selection for the complement of a complement, for example, is systematically precluded, as is case or role ‘assignment’ to a complement’s complement. By constraining head-valent constructions so that the head daughter’s value for valence features like SUBJ(ECT), COMPLEMENTS (COMPS), and SPECIFIER (SPR) is identified with the SYNSEM value of the relevant valent daughter(s), it follows that lexical heads have restricted access to information about the elements they combine with: they may select for only information that is encoded within a valent’s *synsem* object. The relevant locality effects are thus a consequence of the interaction of the geometry of *synsem* objects and the theory of head-valent constructions.³

Let us return to lexical entries. Note first that very little of the information in a lexical entry like (1) must be listed in the lexicon. This is true because lexical types, type inheritance, and the theory of linking⁴ allow complex lexical information like that shown in (1) to be derived, rather than stipulated. That is, much of this information is inferred via the ‘logic of the lexicon’. For example, inflectional rules build words from lexemes whose basic subcategorizational properties are diverse. These lexemes are in turn organized into families whose members share grammatically significant properties, stated as constraints on particular lexemic types (corresponding to particular lexical natural classes). The resulting lexical architecture is represented as a multiple inheritance hierarchy where, for example, ‘part of speech’ and ‘argument selection’ provide independent dimensions of classification and constraint, as shown in (3).



³The *synsem* architecture presented here is not yet entirely satisfactory in this respect, as it makes arbitrarily deep semantic structures available for local selection. For recent work that provides the basis of a solution to this problem, see Copestake et al. 2000.

⁴See Wechsler 1995, Davis 1996, 2001, Davis and Koenig 1999. It should be noted that these researchers utilize semantic roles of a coarser grain (e.g. ACTOR, UNDERGOER) than those assumed here. Nothing in this book turns on the issue of role granularity.

Here we use the following abbreviations for non-maximal types:

- (4) a. *v-lx*: *verb-lexeme*
 b. *p-lx*: *preposition-lexeme*
 c. *a-lx*: *adjective-lexeme*
 d. *intr-lx*: *intransitive-lexeme*
 e. *trn-lx*: *transitive-lexeme*
 f. *str-int-lx*: *strict-intransitive-lexeme*
 g. *s-rsg-lx*: *subject-raising-lexeme*
 h. *s-con-lx*: *subject-control-lexeme*
 i. *str-trn-lx*: *strict-transitive-lexeme*

And the maximal lexemic types (the lexemic ‘species’ in the sense of King (1989) at the bottom of this hierarchy can then be assumed to be those shown in (5).⁵

- (5) a. *siv-lx*: *strict-intransitive-verb-lexeme* (e.g. *die*)
 b. *srv-lx*: *subject-raising-verb-lexeme* (e.g. *seem*)
 c. *scv-lx*: *subject-control-verb-lexeme* (e.g. *try*)
 d. *sip-lx*: *strict-intransitive-preposition-lexeme* (e.g. *of*)
 e. *stp-lx*: *strict-transitive-preposition-lexeme* (e.g. *in*)
 f. *sia-lx*: *strict-intransitive-adjective-lexeme* (e.g. *big*)
 g. *sra-lx*: *subject-raising-adjective-lexeme* (e.g. *likely*)
 h. *sca-lx*: *subject-control-adjective-lexeme* (e.g. *eager*)
 i. *stv-lx*: *strict-transitive-verb-lexeme* (e.g. *prove*)
 j. *orv-lx*: *object-raising-verb-lexeme* (e.g. *believe*)
 k. *dtrv-lx*: *ditransitive-verb-lexeme* (e.g. *believe*)

Note that *of* is here classified as an *sip* because it takes only one argument—its object.⁶ The preposition *in*, by contrast, is transitive because it has two arguments. In modificational uses (e.g. *the nail in the bowl*), the first argument of *in* is the modified nominal (*nail*). In predicative uses (e.g. *The nail is in the bowl.*), the first argument is the unexpressed subject of *in*, which the copula identifies with its own subject (*the nail*).⁷

This mode of lexical analysis, pioneered by early work in HPSG, reflects the fundamental fact that lexical regularities are cross-cutting in a way that is elegantly modeled by lexical type hierarchies and constraint inheritance. Some of the particular constraints we assume for particular lexical types are illustrated in (6):⁸

⁵For convenience, we are here ignoring a number of lexical types that this system of classification is easily extended to accommodate.

⁶At least this is the most common classification. *Of* also leads a life as a transitive preposition, for example, in *This is of the utmost importance.*

⁷For further discussion of this distinction, see Sag and Wasow 1999, chap. 7.

⁸{ *XP* } designates a list containing exactly one phrasal element; { } designates the empty list. Note that the analysis of raising assumed here (unlike the one presented in Pollard and Sag 1994, for example) involves identifying just the LOCAL value of the raised argument and the unexpressed subject of a following argument. By sharing only LOCAL information in raising constructions, we allow certain discrepancies between the raised argument and the relevant unexpressed subject. As shown by Miller and Sag (1997) and Abeillé et al. (1998), non-local information distinguishing ‘clitic’ arguments (realized as pronominal affixes on the verb) must not be transmitted through raising dependencies. Similarly, in the analysis of extraction dependencies we present in Chapter 5, the extraction information encoded in the value of the nonlocal feature SLASH must not be transmitted in raising.

- (6) a. $v\text{-}lx \Rightarrow \left[\text{SS|LOC|CAT} \left[\begin{array}{l} \text{HEAD } v \\ \text{SPR } \langle \rangle \\ \text{SUBJ } \langle \text{XP} \rangle \end{array} \right] \right]$
- b. $p\text{-}lx \Rightarrow \left[\text{SS|LOC|CAT|HEAD } p \right]$
- c. $a\text{-}lx \Rightarrow \left[\text{SS|LOC|CAT|HEAD } a \right]$
- d. $tm\text{-}lx \Rightarrow \left[\text{ARG-ST } \langle \text{NP}, \text{NP}, \dots \rangle \right]$
- e. $str\text{-}intr\text{-}lx \Rightarrow \left[\text{ARG-ST } \langle \text{NP} \rangle \right]$
- f. $s\text{-}rsg\text{-}lx \Rightarrow \left[\text{ARG-ST } \langle [\text{LOC } \square], [\text{SUBJ } \langle [\text{LOC } \square] \rangle] \rangle \right]$
- g. $s\text{-}ctrl\text{-}lx \Rightarrow \left[\text{ARG-ST } \langle \text{NP}_i, [\text{SUBJ } \langle \text{NP}_i \rangle] \rangle \right]$
- h. $str\text{-}tm\text{-}lx \Rightarrow \left[\text{ARG-ST } \langle \text{NP}, \text{NP} \rangle \right]$
- i. $orv\text{-}lx \Rightarrow \left[\text{ARG-ST } \langle \text{NP}, [\text{LOC } \square], [\text{SUBJ } \langle [\text{LOC } \square] \rangle] \rangle \right]$
- j. $dtrv\text{-}lx \Rightarrow \left[\text{ARG-ST } \langle \text{NP}, \text{NP}, \text{NP} \rangle \right]$
- ...

The type constraints in (6) should be understood as stating general properties (as particular feature-value specifications) of particular lexemic types. Individual lexemes assigned to an appropriate maximal lexemic type (a leaf type in a type hierarchy like (6)) inherit the constraints associated with that type and all its supertypes. Some of these constraints involve default specifications that may be overridden by conflicting constraints on subtypes or by idiosyncratic individual lexemes. All the defaults we employ here (based on the theory outlined in Lascarides and Copestake 1999) are non-persistent. That is, though the ‘initial description’ of the lexeme hierarchy may involve the use of default constraints, these defaults are either overridden or rigidified through the process of constraint inheritance. Thus, in each description of some instance of a maximal type (e.g. a full description of the lexeme *give*) there are only hard constraints.⁹

⁹Thus the lexical descriptions we posit could be formulated within a logic like SRL (King 1989) or RSRL (Richter 1999, 2000). However, the latter foundations provide no means for expressing default regularities of the sort that we claim constitute linguistically significant generalizations about lexemes, words, and constructions. By contrast, the foundations adopted here allow default generalizations to be expressed and are highly restrictive in addition. For example, relations, negation, and full quantification are all lacking, a fact that prevents certain kinds of HPSG analyses that are in the literature from being expressed *directly*. The trade-off for this, however, is computational tractability—the immediate possibility of developing precise, wide-coverage grammars that can be used in practical applications involving both parsing and generation. On this last point, see Flickinger et al. 2001.

Finally, consider the following principle (essential to our treatment of words) that relates ARG-ST lists to the valence features SUBJ, COMPS, and SPR:¹⁰

(7) Argument Realization Principle (ARP; preliminary version):

$$word \Rightarrow \left[\begin{array}{l} SS|LOC|CAT \left[\begin{array}{l} SUBJ \quad \boxed{A} \\ SPR \quad \boxed{B} \\ COMPS \quad \boxed{C} \end{array} \right] \\ ARG-ST \quad \boxed{A} \oplus \boxed{B} \oplus \boxed{C} \end{array} \right]$$

This formulation of the ARP simply ensures that all arguments are realized on the appropriate valence list—and hence are selected by a given word in a headed construction (see below). Note that if a word is specified as [SUBJ < >] and [SPR < >], it then follows from the ARP that all of that word’s arguments appear on its COMPS list. A word like *proves*, on the other hand, is an inflected form of the lexeme *prove* and hence, through the interaction of the constraint in (6a) and the ARP, it must include the following information:

$$(8) \left[\begin{array}{l} SS|LOC|CAT \left[\begin{array}{l} SUBJ \quad \langle \boxed{3} \rangle \\ SPR \quad \langle \rangle \\ COMPS \quad \langle \boxed{4} \rangle \end{array} \right] \\ ARG-ST \quad \langle \boxed{3}NP, \boxed{4}NP \rangle \end{array} \right]$$

The valence properties of the inflected form *proves*—that it must combine with an object complement and a subject, but no specifier—are thus appropriately related to the argument structure specified for the (*stv*) lexeme *prove* in (1).¹¹

Of course there is considerable cross-linguistic variation in argument realization. In the approach adopted here, the various well known patterns, e.g. the realization of unexpressed arguments as pronominal affixes in Romance languages (Miller and Sag 1997, Monachesi 1999) and so-called ‘pro drop’ phenomena, are treated as language-particular variations of the ARP. All these realizations are intuitively ‘subtractions’—that is, they involve *synsem* elements that belong to a word’s ARG-ST list, but which fail to be realized in any of its valence lists (SUBJ, COMPS, or SPR). In Chapter 5, we present a modification of (7) that treats extracted complements in terms of a discrepancy between a word’s ARG-ST list and its COMPS list.

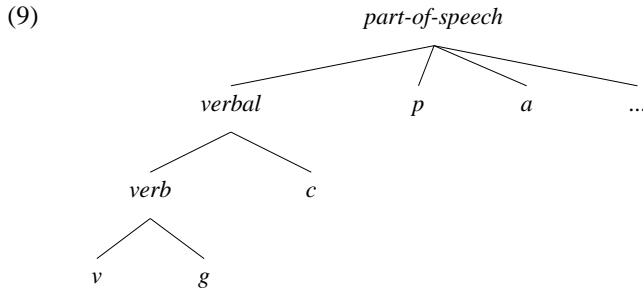
2.4 Features of Verbals

We follow Sag (1997) in assuming that the part of speech types (the values of the feature HEAD) associated with verbs (*v*) and complementizers (*c*) are subtypes of a common supertype called *verbal*. Taking into account the theory of gerunds developed by Malouf (1998, 2000), where

¹⁰Here and throughout we use capital letters to distinguish tags designating lists.

¹¹The relation between lexemes and words can be captured either via lexical rules or else by a type system where words are simultaneously classified in two dimensions: lexeme and inflection. For proposals along the latter lines, see Koenig 1999, Miller and Sag 1997, Abeillé et al. 1998, and Bouma et al. 2001. For convenience, we will adopt the lexical rule approach here. For more on lexical rules in HPSG, see Copestake 1992, Meurers and Minnen 1997, Copestake and Briscoe 1992, Meurers 2000, and (for a more elementary presentation) Sag and Wasow 1999. Bouma et al. (2000) provide an overview and comparison of various approaches to expressing lexical regularities in HPSG. See also Briscoe et al. 1994.

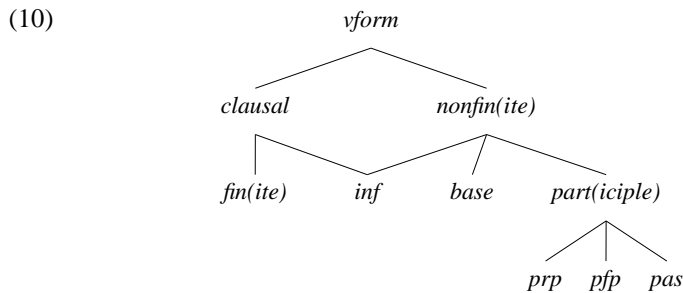
gerunds and verbs also share a common supertype—*verb*, we have the following part of speech hierarchy:¹²



The features VERBFORM (VFORM) and IC (INDEPENDENT-CLAUSE) are declared to be appropriate for the type *verbal*, i.e. for both verbs and complementizers. Other features, e.g. AUX(ILIARY), and INV(ERTED), are appropriate only for instances of type *verb*. Verbs and gerunds are distinguished in terms of their valence properties, as explained briefly in Chapter 5.

2.4.1 Distinguishing Verbal Forms

The values of the feature VFORM—*finite* (*fin*), *infinitive* (*inf*), *base*, *present-participle* (*prp*), *perfect-participle* (*pfp*), and *passive-participle* (*pas*)—are organized into a (multiple-inheritance) hierarchy as shown in (10):



These correspond to familiar distinctions among verb forms; the types in this hierarchy are motivated in part by subcategorization. For example, *pfp* is the VFORM value of perfect participles, e.g. those heading VPs selected by the auxiliary *have*. By contrast, modals select for a VP whose head daughter is [VFORM *base*]. In both cases, the VP complement's VFORM specification is the same as that of the VP's head daughter, in virtue of the (Generalized) Head Feature Principle discussed in the next section. The nonfinite auxiliary verb *to* is the only verb specified as [VFORM *inf*]. The complementizer *for* is also so specified, however, making *to*-phrases and *for-to* phrases a syntactic natural class.

Since the analysis sketched here differs in a variety of ways from previous work in the tradition of GPSG/HPSG, we will briefly consider the evidence for the grammatical distinctions we have made. The groupings in (10) are motivated by various criteria. First, the supertype *clausal* is used to distinguish the verb forms (those specified as [VFORM *fin*] and [VFORM *inf*]) that head

¹²Malouf also classifies the HEAD values of gerunds and nouns as having a common supertype *noun*, thus providing a treatment of gerunds as a 'mixed' category.

clausal constructions. All of the clauses we study here (declaratives, interrogatives, exclamatives, relatives and imperatives) are headed by verbs specified as [VFORM *clausal*].

Our second, partially overlapping, classification of verb forms has an independent motivation. The grammar of negation, for example, makes reference to the *fin/nonfin* distinction. Constituent negation of verbal phrases is possible in English only when the phrase modified is headed by a [VFORM *nonfin*] form, as illustrated by the contrasts in (11).

- (11) a. *Kim [not [walks]]. ([VFORM *fin*)
 b. I prefer to [not [be nominated]]. ([VFORM *base*)
 c. I prefer [not [to be nominated]]. ([VFORM *inf*)
 d. [Not speaking French] is a disadvantage. ([VFORM *prp*)
 e. I would have [not finished in time]. ([VFORM *ppp*)
 f. [Not given any awards at the banquet], Sandy went home disgruntled. ([VFORM *pas*)

Moreover, auxiliary verb forms that are [VFORM *fin*] (and indicative or conditional, in addition) are the only ones that can select *not* as a complement, so as to express sentential negation:¹³

- (12) a. Kim is not going.
 b. If Kim were not going to the party, then....
 c. *I prefer that Kim be not put in charge.
 d. *Be not overly concerned!
 e. *Being not a Republican is a disadvantage.
 f. *Pat has been not to Paris.
 g. *Sandy was visited not.

Note that imperative forms (e.g. *Go!*, *Eat!*) are [VFORM *fin*] in our analysis, predicting that they cannot be modified by constituent negation:

- (13) *[Not [go to the store]]!

The type *part* is motivated by the grammar of modification. The participles (but not [VFORM *base*] VPs, for example) all share the ability to modify nominals. The lexical rules that characterize such modifiers make reference to the specification [VFORM *part*]. The hierarchy of VFORM values thus serves the grammar of selection and modification in a variety of ways.

Finally, we should point out that cutting across the distinctions discussed in this section is the dichotomy between predicative and nonpredicative forms, which is crucial to a number of English constructions. We treat these in terms of the binary feature PRED, which distinguishes predicative from nonpredicative forms of verbs, nouns, prepositions, and adjectives. Among inflected verbal forms, only the passive and present participles are [PRED +] (See Sag and Wasow 1999, chap. 11).

¹³We thus assume that finite auxiliaries allow argument structures like the one shown in (i).

(i) [ARG-ST { NP, ADV_{neg}, VP }]

On this analysis of sentential negation, see Warner 1993, 2000, Kim and Sag 1995, In press, and Sag and Wasow 1999. We will not have much to say about counterfactual conditionals in this book, but it should be noted that we break with tradition in classifying verb forms like the one in (12b) as ‘conditional’, rather than ‘subjunctive’. Conditional forms in fact have little in common with subjunctive forms of the sort that appear embedded in examples like (ii).

(ii) I suggest that they *be* considered.

2.4.2 Distinguishing Verbal Meanings

The content of a verb specifies a state-of-affairs, or *soa*. *Soas* are the building blocks of the various kinds of message—*proposition*, *outcome*, *fact*, and *question*—that are described in detail in Chapter 3. But there is reason to draw a distinction between two kinds of *soa*. For example, the content of the sentence *Kim Sanderson left* is a proposition whose truth or falsity directly involves the real world. And the content of *whether Kim Sanderson left* is a question that is similarly realis—it is resolved according to whether the proposition that Kim Sanderson left is true or false (at a given real world spatio-temporal location). By contrast, the meaning of an imperative sentence like *Get out of here!* makes no direct reference to the real world; nor do subjunctive or infinitival clauses like *that Kim Sanderson leave* or *for Kim Sanderson to leave*. Intuitively, these clauses make reference to future outcomes involving Kim Sanderson’s leaving. Thus, the distinction between these two types of content, which we refer to as realis and irrealis, seems fundamental to the semantic interpretation of clauses.

In our analysis, this bifurcation of clausal meanings is reflected in terms of a bifurcation of the type *soa* into two subtypes, which we will call *realis-soa* (*r-soa*) and *irrealis-soa* (*i-soa*). The strategy is to impose a lexical restriction on finite indicative verb forms (*loves*, *went*, *is*, etc.) requiring that they have an *r-soa* as their content. Conversely, imperative and subjunctive verb forms, though finite, will have a CONTENT value of type *i-soa*. These lexical restrictions, taken together with type constraints guaranteeing that propositions must be constructed from *r-soas*, while outcomes must be constructed from *i-soas*, will modulate the kinds of meaning that can be associated with the phrasal constructions allowed by our grammar.¹⁴

Inflected forms of verbs are derived from verbal lexemes that specify only the nucleus of the *soa* in the verb’s content. Verbal lexemes specify their content type simply as *soa*, the immediate supertype of *r-soa* and *i-soa*. The lexical rules forming finite indicative verb forms will impose the further restriction that the CONTENT value be of type *r-soa*, as illustrated in (14)—the lexical description of the 3rd-singular present indicative form *proves*:¹⁵

¹⁴Fixing the CONTENT type of verbs as one or another kind of *soa* (rather than as, say, a proposition) also aids the treatment of preverbal adverbial modification, which is possible in all kinds of clauses:

- (i) Kim always wins.
- (ii) Does Kim always win?
- (iii) Always wear white!
- (iv) What a mess Kim always makes of things!

Because verbs and the VPs they project are uniformly treated as *soas*, there can be a uniform semantics for the combination of a VP and its modifier. This happens at a lower level of structure and gives rise to a more complex *soa* that is incorporated into the meaning (proposition, question, etc.) of the clause.

Similarly, quantifiers may be assigned scope within any kind of message:

- (v) Kim wrote three letters.
- (vi) Did Kim write three letters?
- (vii) Write three letters!
- (viii) How seldom Kim writes three letters!

This uniformity too is accounted for by the fact that all messages are constructed from *soas* (and the fact that quantifiers are scoped within *soas*).

¹⁵The features POL(ARIZED) and AUX(ILIARY) are discussed in the next section.

- (17) a. *Prove* that theorem!
 b. *Be* waiting for me!
- (18) a. I suggest that she *prove* that theorem right now.
 b. I suggest that you *be* waiting for me.

Note that here the subject argument's case must be nominative. We take this case assignment to be valid for all finite verb forms in English, a conclusion that is easy to justify for subjunctive uses:

- (19) I suggested (that) they/*them *be* made available.

In fact, the following kind of example may provide independent support for this case assignment, even for imperative uses:¹⁶

- (20) You and he *be* quiet!

In previous GPSG/HPSG proposals, it has sometimes been assumed that subjunctives are the same ([VFORM *base*]) forms that occur elsewhere, e.g. in the complements of raising verbs:

- (21) a. Kim expects to be nominated.
 b. I expect Merle to be nominated.

But there is reason to doubt this. The subject of a non-subjunctive infinitival form may be either nominative or accusative in case, depending on the nature of the 'raised' constituent with which it is identified:

- (22) a. *They* expect to be nominated.
 b. I expect *them* to be nominated.

Hence lexically, the subject (i.e. the first ARG-ST member) of this form bears no case restriction (or else is restricted to bear the non-maximal type *structural-case*).¹⁷ Since the subjunctive/imperative forms, as we just saw, assign nominative case to their subject, we conclude that subjunctive/imperative forms, though homophonous with base verb forms, should not be identified with them.

2.4.3 Some Auxiliary Issues

There is a further curiosity about imperatives and subjunctives that remains to be discussed. This has to do with sentential negation, which appears to be expressed in subjunctive clauses via preverbal *not*:

- (23) a. I urged that they not attend the reception.
 b. *I urged that they attend not the reception.

¹⁶The difficulty with this argument is the variation of case marking in NP conjuncts, possibly tied to shifts of register. Thus (i) is also acceptable, but apparently only in an informal register where (ii) is also possible.

(i) You and him *be* quiet!

(ii) You and him should *be* quiet.

Note, however, that many languages have full imperative paradigms and that it is a general property of Indo-European languages that imperative verb forms are finite, assigning nominative case to their subject argument.

¹⁷HPSG Analyses of case based on this distinction between *structural-case* and *lexical-case* can be found in Pollard 1994, Heinz and Matiaszek 1994, and Przepiórkowski 1999.

However, as Potsdam (1996) notes, the *not* that appears in subjunctive clauses allows VP Ellipsis:

- (24) I don't really want you to use my name; in fact I must insist that you not __, because I have concerns about my family.

Since VP Ellipsis is licensed only by auxiliary verbs (Bresnan 1976, Sag 1976), the possibility of (24) suggests that there is a homophonous *not* that functions as a subjunctive auxiliary verb.¹⁸ We will assume this analysis, which is similar to Potsdam's (and quite like the analysis of Italian negation suggested on independent grounds by Kim (1995)). The result is a simple account of sentential negation in subjunctive clauses: the auxiliary *not* has a negative *i-soa* as its CONTENT value. We block analogous examples involving matrix imperatives, e.g. (25), by restricting this auxiliary to embedded clauses, i.e. by an [IC -] lexical specification.

- (25) *Not go to the store!

(The feature IC is discussed in more detail in section 2.7 below.)

Our analysis of the English auxiliary system builds on previous GPSG/HPSG work that has posited the feature INV(ERTED) (Gazdar et al. 1982), which is used to distinguish auxiliary verbs heading inverted phrases from all other verbs. The lexical encoding of this information allows us to accommodate lexical exceptions to inversion—both positive and negative.¹⁹ However, the present analysis diverges in one crucial way from previous work in the PSG tradition. Following Sag (2000), we utilize the specification [AUX +] not to distinguish auxiliary verbs from other verbs, but rather to identify auxiliary constructions. Thus inversion constructions, instances of sentential VP Ellipsis, and finite verbs that are positively or negatively 'polarized' will be specified as [AUX +] and auxiliary verbs—lexically unspecified for AUX—will be the only verbs that are compatible with these constructions.²⁰

This shift in the interpretation of the feature AUX provides a simple lexical account of the restricted distribution of unfocussed *do*: *do* is lexically specified as [AUX +] and hence is compatible ONLY with the auxiliary constructions. This excludes examples like (26), as explained in section 2.6 below.

- (26) *Kim did leave.

2.5 Phrases as Feature Structures

Phrases too can be modeled as typed feature structures, as they have been within HPSG. Features such as HEAD-DAUGHTER (HD-DTR; whose value is a sign, i.e. a word or phrase) and DAUGHTERS (DTRS; whose value is a list of signs) encode roughly the same information that branches do in conventional phrase structure trees.²¹ The figure in (27) presents a simplified analysis in

¹⁸It is interesting to note that in Modern Greek there is a negative particle *mi* that is confined to subjunctive and imperative clauses. Thus something quite like the distinction we posit here is lexicalized in at least one other language.

¹⁹For discussion, see Gazdar et al. 1982, Gazdar et al. 1985, Green and Morgan 1996, Warner 2000, Kim and Sag 1995, In press, and Sag 2000.

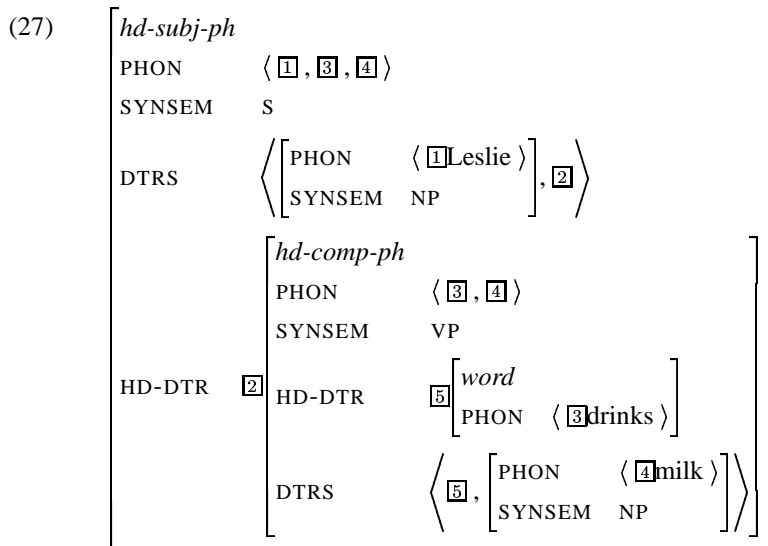
²⁰In Sag's analysis, there are three kinds of polarized auxiliary forms:

- Not-contracted forms: *haven't*, *won't*, etc.
- Forms selecting a polarized adverbial (*not*, *so*, or *too*) as a complement, e.g. *Kim will not/so/too do that*.
- Positively polarized forms, i.e. focussed finite auxiliaries, e.g. *Kim WILL/DID go to the store*.

All [POL +] verbs belong to one of these three classes.

²¹This application of feature structures remains less familiar within linguistics, largely for historical reasons having to do with the ubiquity of rewrite rules, tree structure derivations, and other foundational tools adapted to natural language in the 1950s.

feature structure terms of the sentence *Leslie drinks milk*:



It may not be obvious that there is any significant difference between the sign-based (feature structure) representation of this phrase and the corresponding, more familiar tree structure. However, there are several advantages of this new analytic perspective, as pointed out in Sag (1997). The most important consequences of the sign-based theory of phrases is that it allows us to address such questions as the following, that were touched on in Chapter 1:

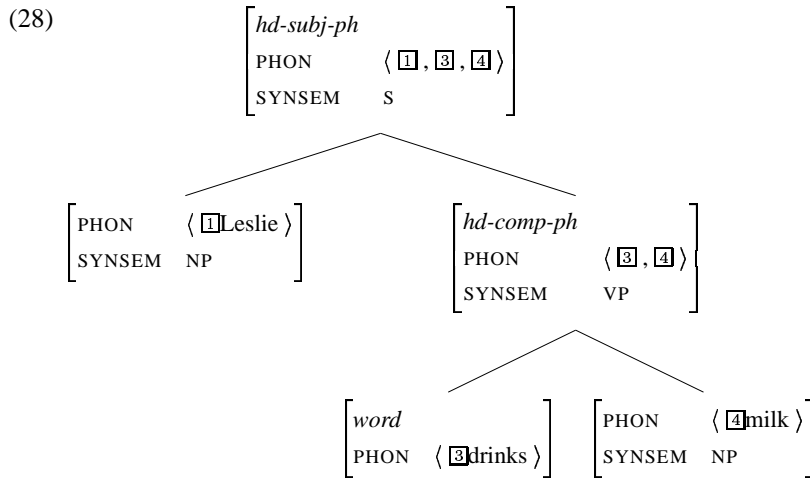
- How are specific constructions related to one another?
- How can cross-constructional generalizations be expressed?
- How can constructional idiosyncrasy be accounted for?

Once phrases are modeled as typed features structures, the types of phrase can be organized into a hierarchy, in much the same way as the lexical types discussed earlier in this chapter. By introducing appropriate supertypes into the hierarchy, one can model the fact that phrase types come in families—i.e. that there are family resemblances cutting across the various types of phrase. Moreover, a type-based analysis can express high-level generalizations (e.g. as a constraint on the type *phrase*), idiosyncratic constructional properties (as a constraint on some maximal (leaf) phrasal type) or generalizations involving ‘family resemblances’ (via constraints on some intermediate-level phrasal type, e.g. *headed-phrase*, *declarative-clause*, or *head-complement-phrase*). As noted in the previous chapter, there is considerable evidence that this is precisely the way natural languages are organized. Natural language grammars are systems of constraints of varying grain.

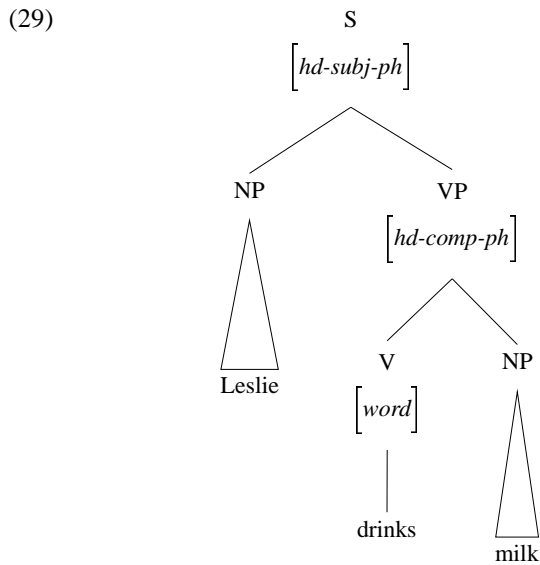
The sign-based approach plays a particularly useful role in the theory of clause types we develop below. For example, it provides a unified framework for describing the ‘correspondence rules’ that constructions encode. This is because such correspondences, as we will show in detail, involve syntactic, semantic, phonological and even contextual information. Our approach also allows us to extend the application of general grammatical constraints to more and more ‘peripheral’ constructions, without having to posit a bifurcation (otherwise unmotivated, as far

as we are aware) between the ‘core’ and ‘periphery’ of language.²²

Although we believe the shift in analytic perspective embodied in our analysis is highly significant, the fact remains that most linguists are accustomed to thinking of grammatical structure in terms of trees. For this reason, we will often describe phrases in terms of more familiar tree diagrams. A feature structure like (27), for example, can be represented in tree-based terms, as shown in (28).



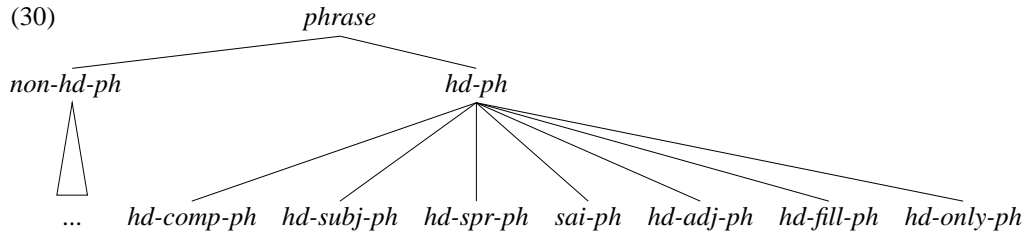
In fact, whenever possible, we will abbreviate tree diagrams like (28) in an even more familiar format, e.g. (29).



²²See Chapter 1. Also relevant is the discussion of Fillmore and Kay 1999, Fillmore et al. to appear, and Bender and Flickinger 1999.

In addition, since all our constraints on phrasal signs are local (making reference only to a mother and its daughters), we will write them in the familiar notation of context-free rewrite rules.

We claim that all phrases found in natural languages are classified according to the following hierarchy of phrasal types:²³



That is, phrases are classified as either *headed-phrase* (*hd-ph*) or *non-headed-phrase* (*non-hd-ph*), with each type exhibiting various subtypes. Among the headed-phrases, seven subtypes are recognized, including: *head-adjunct-phrase* (*hd-adj-ph*), *head-filler-phrase* (*hd-fill-ph*), *head-only-phrase* (*hd-only-ph*), *head-subject-phrase* (*hd-subj-ph*), *head-complement-phrase* (*hd-comp-ph*), *head-specifier-phrase* (*hd-spr-ph*), and *subject-auxiliary-inversion-phrase* (*sai-ph*), as indicated.²⁴ We will illustrate each of these types in due course.

Phrases are governed by the feature declarations shown in (31):²⁵

(31)

TYPE	FEATURES/TYPE OF VALUE	IST
<i>sign</i>	[PHONOLOGY <i>list(speech-sound)</i> SYNSEM <i>canon-ss</i> CONTEXT <i>conx-obj</i>]	<i>feat-struct</i>
<i>phrase</i>	[DTRS <i>nelist(sign)</i>]	<i>sign</i>
<i>hd-ph</i>	[HD-DTR <i>sign</i>]	<i>phrase</i>

Each headed phrase must have some value (possibly the empty list) for DTRS because it is also of type *phrase*, for which that feature is appropriate and necessary. Similarly, all phrases must have some value for PHONOLOGY and SYNSEM, because they are also of type *sign*, for which those features are both appropriate and necessary.

²³At this more abstract level of contemplation, *subject-auxiliary-inversion-phrase* is no doubt too specific a rubric and should be renamed as *head-initial-phrase* or *head-subject-complement-phrase*. For our present, English-specific purposes, however, we will keep the present name. Our analysis of *sai-ph* is based on the SAI construction of Fillmore 1999.

²⁴Note that we make no use here of the type *hd-marker-ph*, although this may in fact need to be recognized for the treatment of English conjunctions.

²⁵*Nelist* stands for *non-empty-list*.

Phrases are subject to the following general constraint:²⁶

(32) Empty COMPS Constraint (ECC)

$$\textit{phrase:} \\ \left[\text{CAT} \left[\text{COMPS} \langle \rangle \right] \right] \rightarrow \dots$$

By requiring that all phrases be $[\text{COMPS} \langle \rangle]$, (32) guarantees that within any phrase, the complements have already been ‘consumed’ by the phrase’s lexical head. Complements are introduced as sisters of the lexical head (as guaranteed by the constraint introduced below on head-complement phrases) and hence they must be more deeply embedded than specifiers, subjects, or fillers, all of which combine with head daughters specified as $[\text{COMPS} \langle \rangle]$. (This follows without further stipulation, because the head daughter of a head-complement phrase can never be phrasal.)

Just as in the case of the lexicon, certain kinds of phrase obey type-specific constraints. For example, our generalization of the Head Feature Principle (analogous to the ‘X’ identity condition of ‘ \bar{X} Theory’) can be formulated as a constraint on phrases of type *hd-ph* (The symbol ‘H’ here is used to indicate the head daughter of a given phrase.):

(33) Generalized Head Feature Principle (GHFP)

$$\textit{hd-ph:} \\ \left[\text{SYNSEM} / \boxed{\text{I}} \right] \rightarrow \dots \mathbf{H} \left[\text{SYNSEM} / \boxed{\text{I}} \right] \dots$$

The ‘/’ notation (Lascarides and Copestake 1999) here indicates a default constraint—specifically one requiring that the *SYNSEM* value of the mother of a headed phrase and that of its head daughter be identical by default. Specific subtypes of *hd-ph* may override the GHFP, but by formulating (33) in defeasible terms, it will require that the features of a head daughter and those of its mother have identical values, except in cases where this is explicitly contradicted by constraints on those subtypes.²⁷

The GHFP allows considerable simplification in our grammar. Earlier work in HPSG posited principles such as the (nondefault) Head Feature Principle and the Valence Principle (P&S–94, chap. 9)), which required the head daughter’s *SUBJ*, *COMPS*, and *SPR* specifications either to be ‘cancelled off’ (analogous to function application in categorial grammar) or else to be inherited by the mother in a headed phrase. Since these valence features are part of the *SYNSEM* value, the GHFP ensures that the head daughter’s values for all these features will be inherited by the mother, unless the phrase in question is subject to a particular constraint requiring that there be a discrepancy between head daughter and mother (a cancellation) for the value of some particular

²⁶Henceforth, abbreviations of an obvious sort will be used, where no confusion should arise. For example, $[\text{COMPS} \langle \rangle]$ in (32) abbreviates $[\text{SS}|\text{LOC}|\text{CAT}|\text{COMPS} \langle \rangle]$.

²⁷Note that one could replace the GHFP with a set of nondefault constraints, each of which specifies the relevant identities on particular subtypes of *hd-ph*. Our use of defaults could thus be regarded as abbreviatory. However, our system of constraints is conceptually quite different from one cast in a pure monotonic system. By using defeasible constraints, we express generalizations about construction types that appear to be beyond the reach of monotonic logics like (R)SRL. We thus achieve a significant gain in descriptive simplicity which, as noted by Lascarides and Copestake (1999), is typical of systems using default constraints.

valence feature.²⁸

One such cancellation affects instances of the type *hd-comp-ph*, which permits a lexical head to combine with exactly as many complements as it selects via the COMPS feature:²⁹

$$(34) \quad \textit{hd-comp-ph}: \\ [] \rightarrow \mathbf{H} \left[\begin{array}{l} \textit{word} \\ \textit{COMPS} \textit{ nelist}(\underline{A} \oplus \textit{list}) \end{array} \right], \underline{A}$$

The constraint in (34) factors out only the information that is specific to phrases of this type. The synslems of the complement daughters are identified with an initial sublist of the head daughter's COMPS list. Moreover, by default, there are no further elements on that COMPS list. Thus, if no subordinate type says otherwise,³⁰ the synslems of the non-head daughters of a head-complement phrase will correspond exactly to the list of complements selected by the head daughter. Finally, observe that because the list of non-head daughters must be nonempty, a strictly intransitive lexical head—one whose COMPS list is empty—cannot head a *hd-comp-ph*. Our phrase structure is thus 'bare', in the sense of Chomsky 1995a.

The following two constraints factor out what is specific to phrases of the type *hd-subj-ph* and *hd-spr-ph*.

$$(35) \quad \textit{hd-subj-ph}: \\ \left[\textit{SUBJ} \langle \rangle \right] \rightarrow \left[\textit{SS} \underline{\square} \right], \mathbf{H} \left[\begin{array}{l} \textit{SPR} \langle \rangle \\ \textit{SUBJ} \langle \underline{\square} \rangle \end{array} \right]$$

$$(36) \quad \textit{hd-spr-ph}: \\ \left[\textit{SPR} \langle \rangle \right] \rightarrow \left[\textit{SS} \underline{\square} \right], \mathbf{H} \left[\textit{SPR} \langle \underline{\square} \rangle \right]$$

In both cases, only one non-head daughter is allowed and this daughter's SYNSEM value is identified with the value of the appropriate valence feature (SUBJ or SPR) of the head daughter that selects it. The phrase itself has an empty value for the corresponding feature.

The GHFP interacts with these last three type constraints to specify appropriate values for all valence features. All valence features not mentioned in the individual constraints will have identical values on the mother and head daughter, as shown in (37a-c). These represent the unification of the GHFP and the ECC with the particular constraints in (34)–(36).³¹

²⁸Our account thus builds on the insights of Borsley (1993), who argues that the HFP should be viewed in default terms. See also Gazdar et al. 1985.

²⁹The notation in (34) is intended to be equivalent to the following:

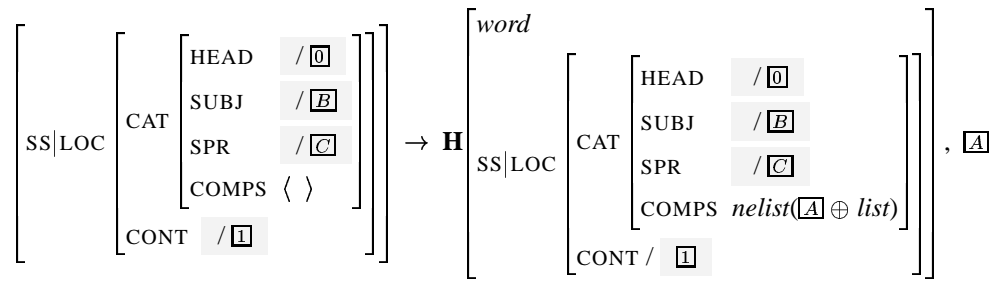
$$(i) \quad \textit{hd-comp-ph} \Rightarrow \left[\begin{array}{l} \textit{HD-DTR} \underline{\square} \left[\begin{array}{l} \textit{word} \\ \textit{COMPS} \textit{ signs-to-synslems}(\textit{nelist}(\underline{A} \oplus \textit{list})) \end{array} \right] \\ \textit{DTRS} \langle \underline{\square} \rangle \oplus \underline{A} \end{array} \right]$$

Here the *signs-to-synslems* function relates a list of signs to the corresponding list of the synslems of those signs. In the revised construction theory of Sag 2001 (which draws a fundamental distinction between constructions and phrases), the function *signs-to-synslems* is eliminated.

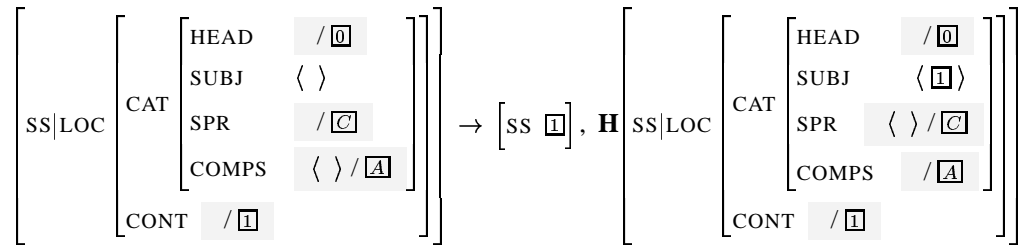
³⁰For example, the *elliptical-vp* phrase Sag (2000) posits for the treatment of VP Ellipsis. See below and also Bender 2000.

³¹For the moment, we ignore the features SLASH, WH, STORE, and BACKGROUND whose values the mother and head daughter will agree on in all these constructions. See Appendix A.

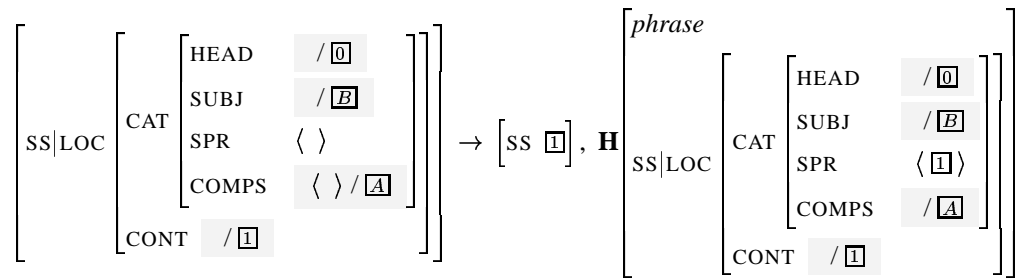
(37) a. GHFP & ECC & *hd-comp-ph*:



b. GHFP & ECC & *hd-subj-ph*:



c. GHFP & ECC & *hd-spr-ph*:



Here the head daughter and mother have identical values for every feature within SYNSEM that can be identified in a way that is consistent with the nondefeasible constraints associated with the particular construction. The complex pattern of identities (indicated here by shading) is a direct consequence of the GHFP (given the default logic of Lascarides and Copestake 1999), without which all the identities of (37a–c) would have to be stipulated piecemeal in unrelated constraints, i.e. in a completely ad hoc manner. Because none of the types illustrated here are maximal, all the shading in (37) involves default identities, that will possibly be overridden by hard constraints on subordinate types. Finally, note that ARG-ST is a feature that lives ‘outside’ of SYNSEM. Hence, although the mother of a head-complement phrase has no ARG-ST specification (it is a phrase) while the head daughter must have some non-empty ARG-ST list, this is in no way inconsistent with the GHFP.

Let us now turn to the type *sai-ph*, all instances of which are subject to the following constraint:³²

$$(38) \quad \textit{sai-ph}: \quad \left[\text{SUBJ } \langle \rangle \right] \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{INV} \quad + \\ \text{AUX} \quad + \\ \text{SUBJ} \quad \langle \boxed{0} \rangle \\ \text{COMPS} \quad \boxed{A} \end{bmatrix}, \boxed{0}, \boxed{A}$$

Thus in this kind of phrase, which must be headed by an inverted ([INV +]) auxiliary verb, elements are ‘cancelled’ from both head daughter’s SUBJ list and its COMPS list. Again, the further constraints on such phrases illustrated in (39) are a consequence of the GHFP and ECC and need not be stipulated.

$$(39) \quad \text{GHFP} \ \& \ \text{ECC} \ \& \ \textit{sai-ph}: \quad \left[\begin{array}{c} \left[\begin{array}{c} \left[\begin{array}{c} \text{HEAD} \quad / \boxed{0} \\ \text{SUBJ} \quad \langle \rangle \\ \text{SPR} \quad / \boxed{C} \\ \text{COMPS} \langle \rangle \end{array} \right] \\ \text{CAT} \\ \text{CONT} \quad / \boxed{1} \end{array} \right] \\ \text{SS|LOC} \end{array} \right] \rightarrow \mathbf{H} \left[\begin{array}{c} \textit{word} \\ \left[\begin{array}{c} \left[\begin{array}{c} \text{HEAD} \quad \left[\begin{array}{c} \text{INV} \quad + \\ \text{AUX} \quad + \end{array} \right] / \boxed{0} \\ \text{SUBJ} \quad \langle \boxed{2} \rangle \\ \text{SPR} \quad / \boxed{C} \\ \text{COMPS} \quad \boxed{A} \end{array} \right] \\ \text{CAT} \\ \text{CONT} \quad / \boxed{1} \end{array} \right] \\ \text{SS|LOC} \end{array} \right], \boxed{2}, \boxed{A}$$

The identity of CONTENT values illustrated in (39) will in fact be overridden by the particular inverted constructions we discuss below.

The types just illustrated serve to classify the maximal phrasal types of our analysis. Finite verb phrases, for example, are analyzed by a kind of head-complement phrase we call *finite verb-phrase* (*fin-vp*). This type of phrase, which allows us to incorporate the treatment of auxiliaries developed by Sag (2000), is subject to the following constraint.

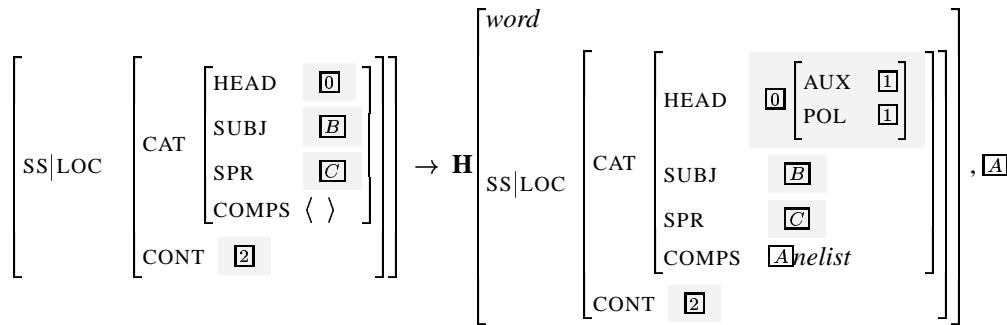
$$(40) \quad \textit{fin-vp}: \quad [] \rightarrow \mathbf{H} \begin{bmatrix} \text{HEAD} \quad \left[\begin{array}{c} \text{AUX} \quad \boxed{1} \\ \text{POL} \quad \boxed{1} \end{array} \right] \\ \text{COMPS} \quad \boxed{A} \end{bmatrix}, \boxed{A}$$

This construction inherits the constraints shown in (41). Shaded values again mark the effects of the GHFP, i.e. those identities that are not defeated by hard constraints on subordinate types.

³²If we allow ellipsis in this construction (in order to handle examples like *Is Kim?*), then the constraint in (38) must be subtly modified to avoid unwanted interaction with the GHFP. Such a revision of (38) could take the following form:

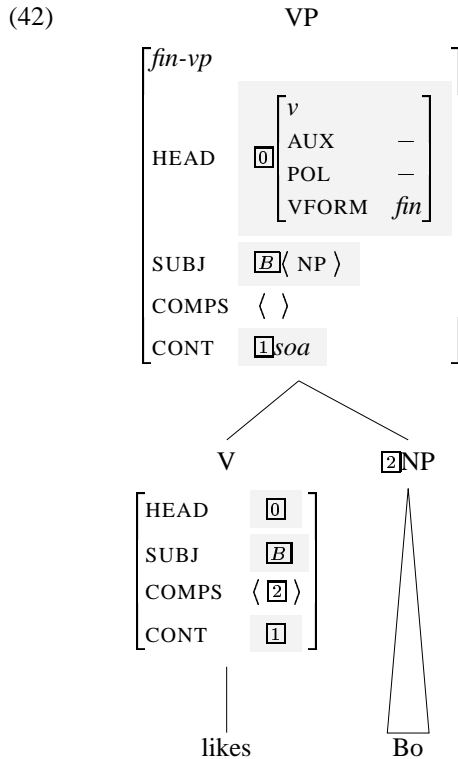
$$(i) \quad \textit{sai-ph}: \quad \left[\text{SUBJ } \langle \rangle \right] \rightarrow \mathbf{H} \begin{bmatrix} \textit{word} \\ \text{INV} \quad + \\ \text{AUX} \quad + \\ \text{SUBJ} \quad \langle \boxed{0} \rangle \\ \text{COMPS} \quad \textit{nelist}(\boxed{A} \oplus \boxed{B}) \end{bmatrix}, \boxed{0}, \boxed{A}$$

(41) GHFP & ECC & *hd-comp-ph* & *fin-vp*:



Note that since *fin-vp* is a maximal type in our analysis, there is no more specific type that will override the identities enforced by the GHFP. Hence these are all hard constraints, as indicated in (41).

The type *fin-vp* allows a treatment of phrases like the following.



The identification of AUX and POL values here may seem puzzling at first. Intuitively, this is just a way of collapsing what would otherwise be two constructions (polarized and non-polarized finite VPs) into one. Recall that *not*-contracted auxiliaries are all [POL +], as are focussed finite auxiliaries and those that select for polar adverbs (see note 20). All other verb forms are lexically specified as [POL −]. Given this, and the fact that only auxiliary verbs are unspecified for the feature AUX, it follows that an instance of the *fin-vp* construction will be [AUX +] just in case it is polarized and is headed by an auxiliary verb, as in (43).

- (43) a. Kim won't go to the store.
 b. Kim has not/TOO/SO gone to the store.
 c. Kim HAS gone to the store.

Any *fin-vp* not headed by a polarized verb form will be [AUX –], even if it is headed by an auxiliary verb, as in (44b):

- (44) a. Kim went to the store.
 b. Kim has gone to the store.

This reinterpretation of positive AUX specifications as distinguishing auxiliary constructions (instead of distinguishing the auxiliary verbs, as in previous analyses) provides the key to analyzing the much discussed distribution of unfocussed *do*. As noted earlier, the auxiliary verb lexeme *do* is exceptionally specified as [AUX +]. Given what we have said about VPs so far, this predicts that *do* can head a *fin-vp* construction only when it is polarized, accounting for familiar contrasts like the following:

- (45) a. Kim didn't go to the store.
 b. Kim did not/TOO/SO go to the store.
 c. *Kim did go to the store.

This is not the whole story of *do*, of course. Following Sag (2000), we treat VP Ellipsis via a separate construction (*elliptical-vp*—also a subtype of *hd-comp-ph*) that allows an [AUX +] verb to appear with a proper subset of its complements. Unnegated, unfocussed *do* may head such a construction and is thus predicted to occur in elliptical examples like (46):

- (46) Kim *dīd* __ .

This construction requires that some complements selected by the head remain unexpressed.

Nonfinite VPs are to be handled in terms of a distinct construction that we will refer to simply as *nonfinite-head-complement-phrase* (*nf-hc-ph*).³³ As discussed by Sag (2000), when nonfinite auxiliaries head such phrases, they are always [AUX +].

Finally, it should be noted that we assume a finer grained analysis for the type *hd-subj-ph*. Declarative instances of this type can be distinguished from those head-subject phrases that involve accusative subjects (so-called 'Mad Magazine' sentences, e.g. *What, [me worry?]* (Akmajian 1984, Zhang 1990, Lambrecht 1990), or absolute constructions, like [*My kids in college now*], *I'm going to have lots of free time*). We will not present an analysis of these constructions here.³⁴

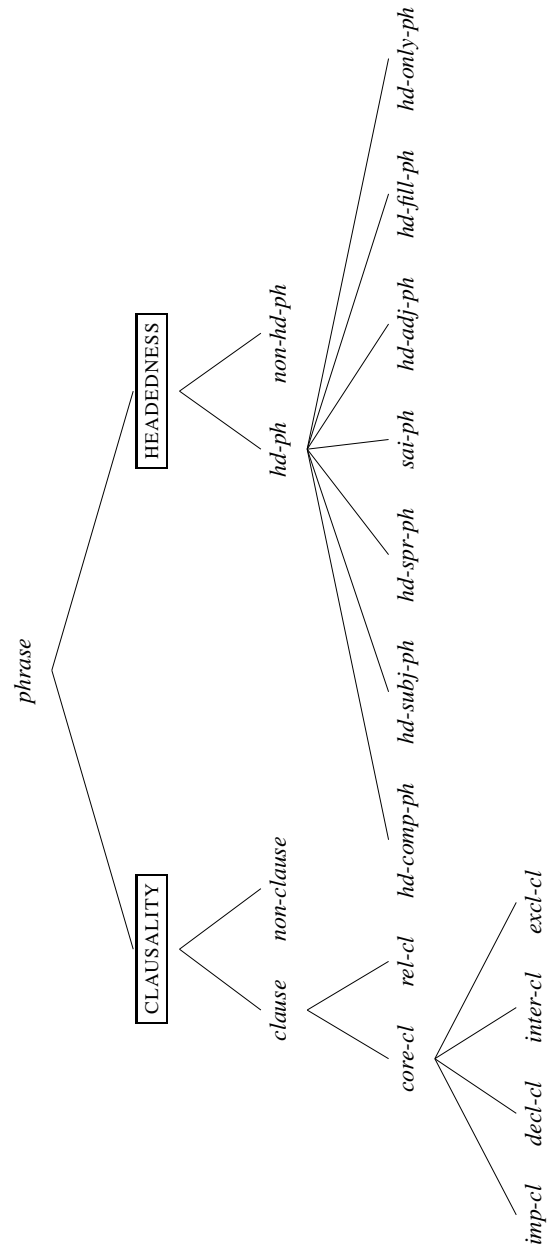
2.6 Clause Types

To express generalizations about the shared properties of diverse construction types, we propose (following Sag 1997) to classify phrases not only in terms of their 'X̄' type (e.g. whether they are headed or not; if they are headed, what kind of daughters are involved, etc.), but also relative to an independent dimension of 'clausality'. On our theory, each type of phrase is cross-classified: each maximal phrasal type inherits both from a CLAUSALITY type and from a HEADEDNESS type.

³³If certain refinements are made, it is possible to modify this construction so that it encompasses most other unsaturated ([SUBJ ⟨YP⟩]) XPs, as well as VPs.

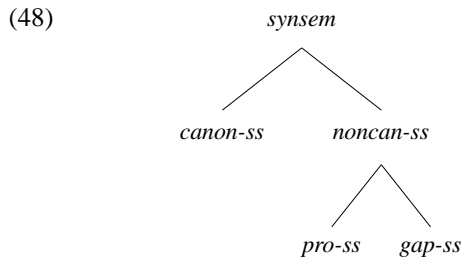
³⁴We have not said anything about the principles of linear ordering that guarantee the proper sequencing of PHON values in these signs. This is an issue too far afield from the present study. Though we will identify the head daughter with a particular member of the DTRS list and will continue to represent constructions in terms of ordered trees, the reader should bear in mind that the linear order of constituents in our theory should in fact be determined by linearization constraints of greater generality. For relevant further discussion, see Pollard and Sag 1987, Chap. 7, Reape 1994, and Kathol 1995, 2000.

(47)



This classification recognizes a distinction between clauses and nonclauses, and also identifies at least the following subtypes of *clause*: *decl(arative)-cl(ause)*, *inter(rogative)-cl(ause)*, *imp(erative)-cl(ause)*, *excl(amative)-cl(ause)*, *core-cl(ause)* and *rel(ative)-cl(ause)*.³⁵ Our analysis lets us express generalizations about phrases with the same simplicity and precision that is standard in work on hierarchical lexicons. With the phrasal multiple inheritance hierarchy, we also have no need to posit phantom formatives—the inaudible functional heads that are routinely assumed in many competing analyses of clausal structure. The work done by these elements is replaced by constraints associated with the various types of clause.

The subtypes of *clause* are the locus of constraints that will be highly relevant to our treatment of interrogatives. Before proceeding, however, we must clarify our assumptions about the subtypes of *synsem*. Both the analysis of unexpressed controlled subjects and our account of filler-gap dependencies lead us to distinguish various subtypes of *synsem*. We follow familiar terminology and refer to the type of unexpressed controlled subjects (of nonfinite phrases) as *pro-synsem* (*pro-ss*)³⁶ and to the type of the ‘gap’ argument in an extraction construction as *gap-synsem* (*gap-ss*).³⁷ We assume that instances of all such types exhibit exceptional properties (for example, they cannot be locally realized through simple combination of a head with its subject, complement, or specifier) and are hence ‘noncanonical’. To reflect this, we posit the hierarchy of *synsem* subtypes shown in (48). (Here *canon-ss* abbreviates *canonical-synsem* and *noncan-ss* abbreviates *noncanonical-synsem*.)



All signs (both words and phrases) are governed by the following principle, already built into the type declaration for *sign* in (31) above:

- (49) Principle of Canonicity:
 $sign \Rightarrow \left[\text{SYNSEM } canon-ss \right]$

The Principle of Canonicity ensures that every overt linguistic expression has a SYNSEM value of type *canon-ss*. Since overt complements, subjects, and specifiers must have a *canon-ss* SYNSEM value, the constraints already illustrated for head-complement, head-subject, and head-specifier phrases guarantees that whenever some head selects for a *noncan-ss* argument, that argument cannot be a daughter of any such phrase.

In our grammar, clauses are the constructions that are used exclusively to build phrases whose content is communicatively complete, i.e. whose CONTENT value is some subtype of *message: proposition, question, outcome, or fact*. The following constraint ensures this:

³⁵The hierarchy assumed here may well require revision as the analysis is extended, e.g. to purpose, rationale, absolute, gerund, and conditional clauses.

³⁶The type *pro-ss* is analogous to the type PRO proposed by Pollard (1989) and shares its properties, in the main.

³⁷For languages that allow missing arguments with distinctive properties, further subtypes of *synsem* may be necessary.

(50) *clause*:

$$\left[\text{CONT } message \right] \rightarrow \dots$$

(50) guarantees, for example, that no clause can have a *soa* as its content.

The intuition our analysis encodes is that verbs and the verb phrases they project have *soa* content, i.e. their content is not yet a proposition (or any other kind of message). Hence neither verbs nor the VPs they project can function as independent utterances; nor can they serve as complements of verbs like *believe* or *think*, which select propositional arguments. In order to build a phrase whose content *is a message*, it is necessary to embed a VP within a clausal construction. As noted in Chapter 1, the clausal constructions thus serve to ground a recursive process that generates message-denoting signs.

Let us now turn to the constraints governing the various subtypes of *clause*. Core clauses (declaratives, interrogatives, exclamatives, and imperatives) cannot be modifiers, as is guaranteed by the following constraint:³⁸

(51) *core-cl*:

$$\left[\text{HEAD } \left[\begin{array}{ll} verbal & \\ \text{VFORM } & clausal \\ \text{MOD } & none \end{array} \right] \right] \rightarrow \dots$$

(51) also guarantees that core-clauses are headed by verbs whose VFORM value is (a subtype of) *clausal*, i.e. by finite verbal forms or the auxiliary *to*.³⁹

Of the clause types considered here, only relative clauses may be used as modifiers. The constraint in (52) allows for this possibility.

(52) *rel-cl*:

$$\left[\begin{array}{ll} \text{HEAD } & \left[\begin{array}{ll} \text{IC} & - \\ \text{INV} & - \\ \text{MOD} & [noun] \end{array} \right] \\ \text{CONT} & fact \end{array} \right] \rightarrow \dots$$

Relative clauses have *fact* as the type of their CONTENT value, yet the content of a relative clause always contains an index that is identified with the index of the nominal that the clause modifies.

The feature IC (INDEPENDENT-CLAUSE—see section 2.7 below) is a variant of Uszkoreit's (1987) MAIN-CLAUSE feature; the [IC -] constraint in (52) ensures that relative clauses cannot serve as independent clauses, and hence that they have no status as (non-elliptical) independent utterances. The specification [INV -] in (52) guarantees not only that post-auxiliary subjects are in general impossible in relative clauses, but also that forms like first-person singular *aren't*, lexically specified as [INV +], can never head a relative clause:

- (53) a. *The person [(that/who) are they visiting] is Sandy.
 b. *The person [(that/who) I aren't visiting] is Sandy.

³⁸For discussion of modifier analyses based on the feature MOD, see Pollard and Sag 1994.

³⁹Note that core clauses may also be headed by complementizers, as their part-of-speech is also a subtype of *verbal*. Thus, as sketched below, we will treat diverse kinds of CP as instances of type *core-cl*.

We now introduce further constraints on the immediate subtypes of *core-cl*:

$$(54) \quad \textit{decl-cl}: \left[\text{CONT} \begin{array}{l} \textit{austinian} \\ \text{SOA} / \mathbb{I} \end{array} \right] \rightarrow \dots \mathbf{H}[\text{CONT} / \mathbb{I}] \dots$$

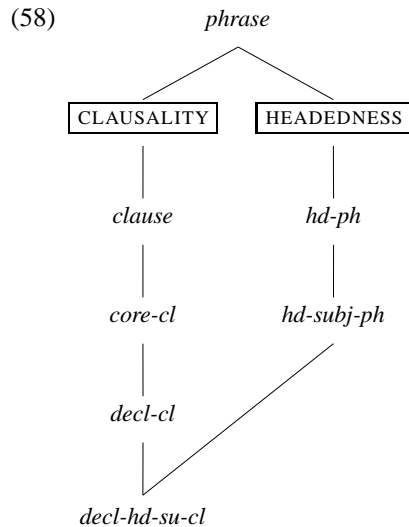
$$(55) \quad \textit{inter-cl}: [\text{CONT} \textit{question}] \rightarrow \dots$$

$$(56) \quad \textit{imp-cl}: [\text{CONT} \textit{outcome}] \rightarrow \dots$$

$$(57) \quad \textit{excl-cl}: [\text{CONT} \textit{fact}] \rightarrow \dots$$

The effect of these constraints is to establish a correlation between clausal construction types and types of meaning. As explained earlier, the semantic type *austinian* mentioned in (54) has two subtypes: *proposition* and *outcome*. Indicative declarative clauses (see Chapter 3) denote propositions,⁴⁰ while subjunctive clauses (analyzed here as a kind of *decl-cl*) denote outcomes. Note that the content of the head daughter of a declarative clause (a *soa*) is embedded as the mother's SOA value by default.⁴¹ The treatment of exclamative clauses in terms of facts is explained in Chapters 3 and 6.

We are now ready to analyze more fully our previous example *Leslie likes Bo*. The more embedded phrase *likes Bo* is an instance of the *fin-vp* construction illustrated in the previous section. However, in order to combine a finite VP with the subject NP *Leslie* we need a new type of phrase, which we will call *declarative-head-subject-clause* (*decl-hd-su-cl*). This is a subtype of both *decl-cl* and *hd-subj-ph*:



The only constraint particular to *decl-hd-su-cl* is shown in (59):

⁴⁰In fact, as motivated in Chapter 3, we allow indicative declarative clauses to denote either a proposition or a FACT. We introduce a construction ‘promoting’ proposition-denoting signs to fact-denoting clauses in section 2.10 below.

⁴¹The reason this is a default is that the analysis of elliptical declarative clauses developed in Chapter 8 will construct a SOA value from contextual information, rather than the head daughter’s CONTENT.

(59) *decl-hd-su-cl*:

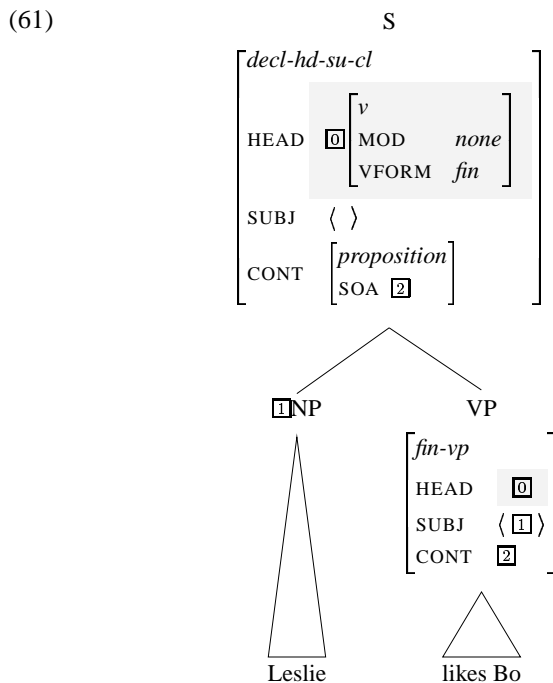
$$[] \rightarrow [], \left[\text{HEAD} \begin{bmatrix} \text{VFORM} & \textit{fin} \\ \text{INV} & - \end{bmatrix} \right]$$

Simply declaring the existence of the type *decl-hd-su-cl* and indicating its place in the hierarchy of phrasal types is sufficient to predict (through constraint inheritance) that all instances of this type have the properties shown in (60):⁴²

(60) ECC & GHFP & *hd-subj-ph* & *core-cl* & *decl-cl* & *decl-hd-su-cl*:

$$\left[\text{SS|LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{MOD} & \textit{none} \\ \text{VFORM} & \textit{clausal} \end{bmatrix} \\ \text{SUBJ} & \langle \rangle \\ \text{SPR} & \text{C} \\ \text{COMPS} & \text{A} \langle \rangle \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \textit{austinian} \\ \text{SOA} & \text{2} \end{bmatrix} \end{bmatrix} \right] \rightarrow \left[\text{SS} \text{1} \right], \mathbf{H} \left[\text{SS|LOC} \begin{bmatrix} \text{CAT} \begin{bmatrix} \text{HEAD} & \text{0} \\ \text{SUBJ} & \langle \text{1} \rangle \\ \text{SPR} & \text{C} \langle \rangle \\ \text{COMPS} & \text{A} \end{bmatrix} \\ \text{CONT} & \text{2} \end{bmatrix} \right]$$

And these constraints serve to define the existence of phrases like the following:



⁴²Perhaps the [INV -] constraint assumed here should apply to *hd-subj-ph*, rather than directly to *decl-hd-su-cl*. But the constraint would then have to be a default, assuming some instances of *hd-subj-ph* are nonverbal (and hence incompatible with the feature INV).

Here the *VFORM* value *clausal* has been resolved to *fin* and the *CONTENT* value *austinian* has been resolved to *proposition*. These resolutions are a consequence of the lexical properties of the indicative verb *likes* and their interaction with various of the type constraints we have discussed. In particular, a finite verb has either an *r-soa* (indicatives) or an *i-soa* (subjunctives) as its content. The type *austinian* has the subtypes *proposition* and *outcome*. The former type is constructed from (has as its *SOA* value) an *r-soa*; the latter only can be constructed only from an *i-soa*. Hence, it follows that instances of the *decl-hd-su-cl* construction like (61), i.e. those that are headed by an indicative verb form, can only denote a *proposition*.

Conversely, when a *decl-hd-su-cl* is headed by a subjunctive/imperative verb, the content can only be of type *outcome*. This construction therefore gives rise to (embedded) subjunctive clauses like (62) and (independent) imperative utterances with subjects, like those in (63):

- (62) a. they be careful
 b. he go home
- (63) a. Everyone get out of here!
 b. Nobody move!
 c. Somebody help me!
 d. Kim take the high road! (The rest of you are with me.)

Although imperative sentences like (63) have many mysterious properties that we cannot try to unravel here, it is a pleasing consequence of our approach that it generalizes to these, as well as to indicative clauses.

Finally, it should be noted that, in consequence of the constraints in (59), all instances of *decl-hd-su-cl* are headed by a noninverted, finite verb or VP.⁴³ This means, among other things, that infinitival clauses like (64) are systematically excluded:

- (64) * [Sandy [to go to the store]]

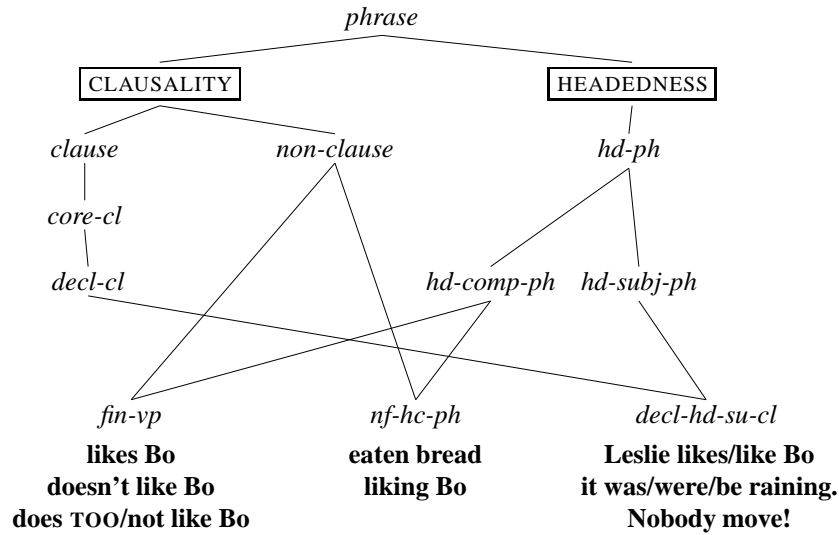
This discussion has provided only a preliminary glimpse of how our constructional analysis will assign meanings to phrases. The phrasal types introduced so far are summarized in (65). Among the further details yet to be presented is an account of quantification, a matter we take up in Chapter 3 and in section 5.3.⁴⁴

⁴³When the finite verb is strictly intransitive, it cannot serve as the head daughter of any head-complement phrase (the head daughter of such a phrase must have a nonempty *COMPS* list). In this case, the verb itself is the head of the *decl-hd-su-cl* (e.g. *Kim left*). When a finite, noninverted verb selects complements, it first combines with those complements by the *fin-vp* construction.

⁴⁴We will not discuss imperative clauses in this book. However, a first cut at formulating the constraint governing the subjectless imperative construction (*no-subject-imperative-clause*—a subtype of *imp-cl* and *hd-only-ph*) might be (i):

- (i) *ns-imp-cl*:
- $$[\text{CONT}|\text{SOA } \boxed{\text{I}}] \rightarrow \left[\begin{array}{l} \text{VFORM } \textit{fin} \\ \text{SUBJ } \langle \text{NP}[2\text{nd}] \rangle \\ \text{CONT } \boxed{\text{I}} \end{array} \right]$$

(65)



2.7 Main and Embedded Clauses

A grammar must specify what kinds of phrase can be used as an independent utterance. In standard presentations of context-free grammar, for example, this is done by designating ‘S’ as the ‘start symbol’. In a system of feature structures like the one developed here, this is often done by positing a distinguished type—*root*—whose constraints must be satisfied by any ‘stand-alone’ utterance. Fragments present a special case, of course, one that we return to in Chapter 8. Here we present a preliminary statement (one that we will revise in Chapter 7) of the constraints to be associated with the type *root* so as to predict which phrases can function as independent utterances.

In earlier discussions, we introduced the feature IC to distinguish between independent clauses and others. If a clausal construction is specified as [IC −], then it cannot function independently (modulo elliptical fragments); rather it must be an embedded clause. However, in light of the fact that independent clause phenomena sometimes appear in subordinate clauses (Hooper and Thompson 1973, Green 1976), we must keep the notions ‘independent clause’ and ‘main clause’ distinct. Simplifying what is a complex issue well beyond the scope of this monograph (for discussion, see Green 1996), a clause can appear independently only if it is [IC +]. However, certain embedded environments, as we will see in the next section, also allow [IC +] phrases.

To guarantee this effect, we build an [IC +] condition into the *root* type constraint, which we formulate as follows:

$$(66) \quad \textit{root} \Rightarrow \left[\begin{array}{l} \text{HEAD} \quad \left[\begin{array}{l} \textit{verbal} \\ \text{IC} \quad + \\ \text{VFORM} \quad \textit{fin} \end{array} \right] \\ \text{CONT} \quad \textit{message} \\ \text{STORE} \quad \{ \} \\ \text{SLASH} \quad \{ \} \\ \text{WH} \quad \{ \} \end{array} \right]$$

(66) says that a root phrase must be an [IC +], finite verbal projection (i.e. one whose HEAD value is either of type *v*, *g* or *c*). This of course allows an instance of *decl-hd-su-cl* like (61) above to serve as an independent utterance. Gerund phrases (e.g. *Les liking Bo*) will be excluded

as independent utterances because gerunds are lexically specified as [IC –], and hence are incompatible with *root*.

Our decision to in addition allow CPs (discussed below) as independent utterances is somewhat speculative. The treatment of *that*-clauses and *for-to*-clauses that we will outline requires them to be [IC –], and hence incompatible with *root*. However, we leave open the possibility that other expressions, e.g. *how come*, are [IC +] complementizers, and hence that the clauses they project are [IC +] CPs, as proposed in Chapter 6.⁴⁵

The constraint in (66) also requires that the CONTENT value be some subtype of *message*. This constraint allows for *root* phrases whose content is of type *proposition*, *fact*, *question*, or *outcome*, but it rules out phrases whose content is of type *soa*.⁴⁶ Finally, (66) requires in addition that the values for STORE, SLASH and WH must be empty. These features are all discussed in Chapter 5.

How then do embedded clauses acquire the specification [IC –]? The answer to this is partly lexical. We assume that there are lexemic types specifying ARG-ST lists that include sentential complements, i.e. verbal complements whose CONTENT value is some subtype of *message*. The relevant lexemic types are constrained so that each such argument must be [IC –] as well.⁴⁷ Sentential complements of all kinds⁴⁸ will be specified as [IC –].

2.8 Complementizers and *To*

As noted above, the part of speech types associated with verbs and complementizers share a common supertype *verbal*, for which the features VFORM and IC are both appropriate. Given these assumptions, we may formulate the lexical entry for the complementizer *that* as shown in (67):

$$(67) \left[\begin{array}{l} \text{PHON} \quad \langle \text{that} \rangle \\ \text{SS|LOC} \quad \left[\begin{array}{l} \text{CAT} \quad \left[\begin{array}{l} \text{HEAD} \quad \left[\begin{array}{l} c \\ \text{IC} \quad - \\ \text{VFORM} \quad \textit{fin} \end{array} \right] \\ \text{SUBJ} \quad \langle \rangle \end{array} \right] \\ \text{CONT} \quad \underline{\text{I}}\textit{austinian} \end{array} \right] \end{array} \right] \\ \text{ARG-ST} \quad \left\langle \begin{array}{l} \text{S} \\ \left[\begin{array}{l} \text{INV} \quad - \\ \text{VFORM} \quad \textit{fin} \\ \text{SUBJ} \quad \langle \rangle \\ \text{CONT} \quad \underline{\text{I}} \end{array} \right] \end{array} \right\rangle \end{array} \right]$$

⁴⁵The decision to let CPs be compatible with *root* is of course independently motivated in many languages, including French (the exclamative complementizer *comme*), Gascon (the affirmative complementizer *que*), Hindi (the interrogative complementizer *kyaa*), and Arabic (the intensifying complementizer *inna*). This analysis might also be independently motivated for English by examples like (i), pointed out to us by Carl Pollard:

(i) That it should come to this!

This motivation would turn on showing that such clauses are not better analyzed in terms of ellipsis.

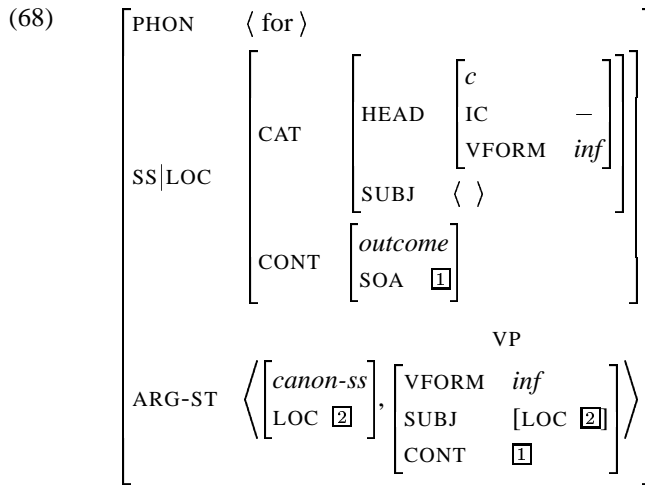
⁴⁶This semantic condition may be an independent factor ruling out gerund phrases as independent utterances.

⁴⁷An alternative would be to introduce a further classification of phrases, placing an [IC –] condition on any sentential daughter. This would have the virtue of providing a unified treatment of the ‘embeddedness’ of complements and other subordinate clauses, if indeed a uniform treatment is warranted.

⁴⁸Except perhaps complements of direct quotation, depending on how these are analyzed.

As just noted, *that* is [IC –]; and hence *that*-clauses cannot function as independent utterances. There is a further effect of the constraints in (67): because (1) the VFORM and CONTENT values of *that* are shared with those of its sentential complement and (2) that shared content is of type *austinian*, *that* allows both indicative and subjunctive complements. These can be simple declarative clauses (e.g. instances of the type *decl-hd-su-cl*), topicalized clauses (in certain restricted circumstances discussed below), or coordinations of such.⁴⁹

The lexical entry for the complementizer *for* is similar to (67).⁵⁰ We assume that *for* differs from *that* in taking two syntactic arguments, instead of one. In addition, the LOCAL value of the first argument is identified with that of the unexpressed subject of the second argument:



As noted by Sag (1997), if we assume that *for-to* clauses, but not *that*-clauses, project a flat (ternary) structure, we obtain an immediate account for contrasts in (69), first noted by Emonds (1976: 196).

- (69) a. Mary asked me if, in St. Louis, John could rent a house cheap.
 b. He doesn't intend that, in these circumstances, we be rehired.
 c. *Mary arranged for, in St. Louis, John to rent a house cheap.
 d. *He doesn't intend for, in these circumstances, us to be rehired.

This argument assumes that these sentential modifiers must have a sentential constituent to modify. On our analysis, there is a clause for the adverbial to modify only in the case of *that*-clauses like (69a,b), not in the case of *for/to*-clauses like (69c,d). And no other analysis of (69c,d) is possible because any such analysis would require that two complements be allowed to right-shift over an adverb. However, as (70) shows, this possibility must be disallowed on independent grounds:

- (70) a. *Kim persuaded in St. Louis Sandy to rent a house cheap.
 b. *Lee believed in these circumstances Sandy to be in the right.

⁴⁹*Coordinate-phrase* is not a subtype of clause, though (as noted above) the semantics we assume allows such non-clauses to have propositional content.

⁵⁰It no doubt makes sense to organize the complementizers via a lexical type, factoring out all common properties as constraints on that type. However, we will not concern ourselves with such matters here, simply presenting the individual lexical items that would result from such an analysis.

Observe that the object of *for* is required to be of type *canon-ss*. This is inconsistent with that object being a *gap-ss* and hence correctly predicts that the object of *for* cannot be extracted:

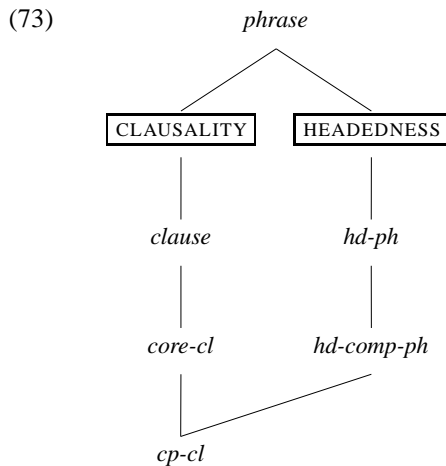
(71) *Who did you prefer for to get the job?

Finally, the semantics of *for* is an outcome formed from the *soa* of the infinitival VP complement. Like subjunctive clauses, the meaning of a *for-to* phrase is irrealis in nature.

Since these complementizers have a lexically assigned content of type *austinian* (*that*) or *outcome* (*for*),⁵¹ in order to provide *that*-clauses and *for-to*-clauses with the right semantics, we identify the complementizer's content with that of the CP it projects. To this end, we posit the clause type *CP-clause* (*cp-cl*), which is subject to the following constraint:

(72) *cp-cl*:
[] → **H**[HEAD *c*], ...

Nothing more needs to be said about this clause type except that it is a subtype of both *core-cl* and *hd-comp-ph*:



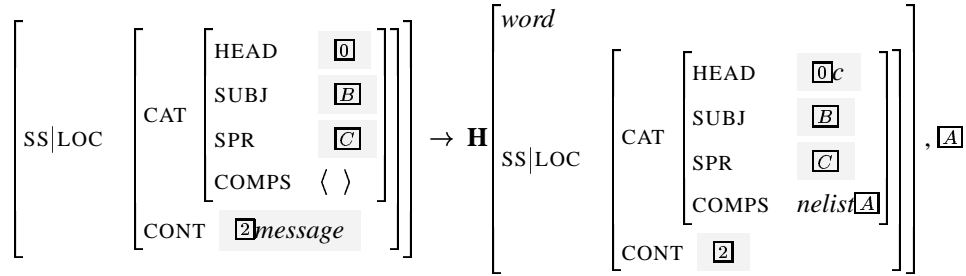
The inheritance of content and all other feature specifications from the head daughter to the CP is guaranteed by the GHFP:

⁵¹Or *question*, as is the case for the complementizers *whether* and *if*, discussed in Chapter 6. Similarly, French exclamative clauses like (i) are likely best analyzed as an instance of this type, with the lexical entry for *comme* specifying a CONTENT value of type *fact*.

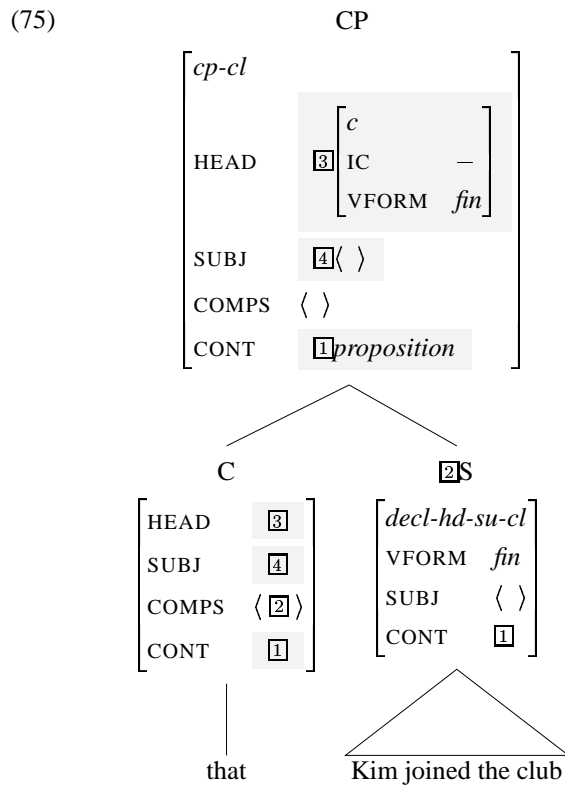
- (i) Comme il fait beau!
How it makes beautiful
How beautiful it is!

For discussion, see Desmets in preparation.

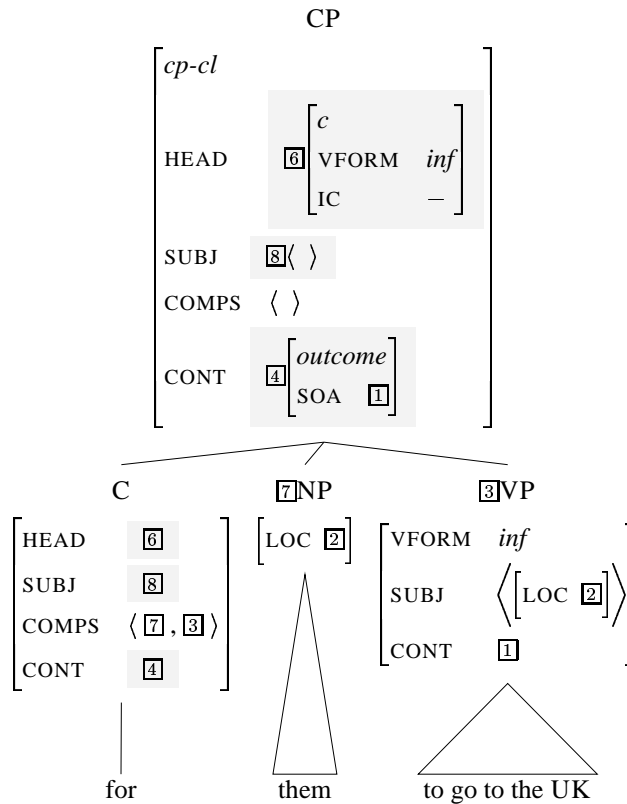
(74) GHFP & ECC & hd-comp-ph & clause & core-cl & cp-cl:



This clause type accounts for clauses like (75) and (76).



(76)



In (75), the CONTENT value must resolve to *proposition*—the only subtype of *austinian* constructible from the *r-soa* provided by the indicative verb *joined*. The CONTENT value of *to* is lexically underspecified (as type *soa*), but in contexts like (76) (where *for* introduces a content of type *outcome*), it is resolved to type *i-soa*. Elsewhere, as detailed below, a *to*-phrase can denote a *proposition*, requiring that *to*'s CONTENT value be resolved as an *r-soa*.

A final observation: although the complementizer *that* is specified as [IC –], the IC value of its complement is left unspecified. This has the effect of allowing ‘main clause phenomena’⁵² in certain embedded ([IC –]) environments, but only if the complementizer *that* is present. This appears to be a correct prediction:

- (77) a. She subtly suggested (that) we had to visit France.
 b. She subtly suggested *(that) [problems of this sort, our analysis would never account for].
- (78) a. They believed (that) they were oppressed.
 b. They believed *(that) [never again would they have to do housework].
- (79) a. The kids were under the impression (that) they had to leave.
 b. The kids were under the impression *(that) [out from under the bush would appear a small animal].

⁵²We mean to include here most of the phenomena discussed by Emonds (1976) under the rubric of ‘root transformations’.

We make no attempt here to explain the pragmatic circumstances governing when independent clauses can appear in embedded environments.⁵³ However, the fact that a complementizer, an unambiguous marker of syntactic embedding, is required in order for independent clauses to be embedded is both surprising and descriptively challenging. Our (admittedly partial) account of the matter is strikingly simple.

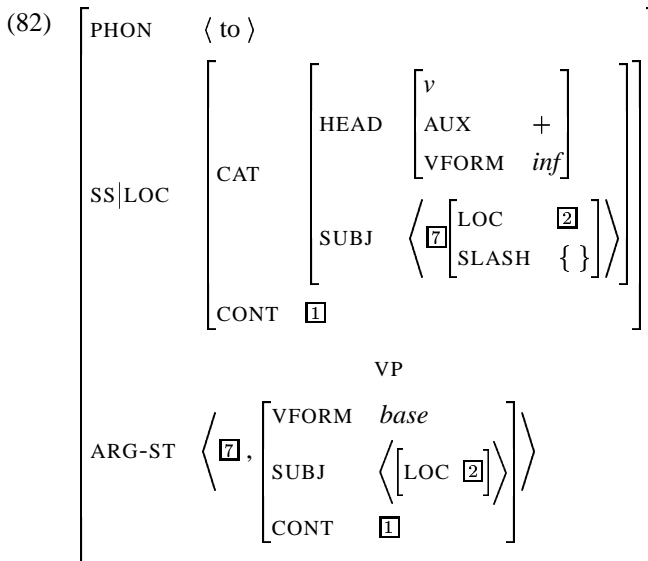
To-phrases are more complex. As shown by Pullum (1982), the word *to* is profitably analyzed as a defective (i.e. paradigmless), nonfinite auxiliary verb—that is, a verb that is lexically unspecified for the feature AUX. This treatment, for example, provides an account of why *to* ‘licenses’ VP Ellipsis, as in (80).

- (80) a. They ordered us to leave, and we *will, should, are* __.
- b. They ordered us to leave, and we want *to* __.

Crucially, this is a property not shared by complementizers or by nonauxiliary verbs, as the impossibility of ellipses like the following show.⁵⁴

- (81) a. *Pat preferred for Sandy to get the job, and we preferred for __, also.
- b. *Kim ordered us to leave, and Sandy ordered us __.
- c. *They ordered us to leave, and we want __.

We therefore formulate the lexical entry for *to* as shown in (82).



This is quite similar to the entries required for other auxiliary verb lexemes. It is distinctive, however, in that it is the only element specified as [VFORM *inf*]. Hence (since is a subtype of *clausal*; see (10) above), *to* is distinctive in being the only nonfinite verb form that can project

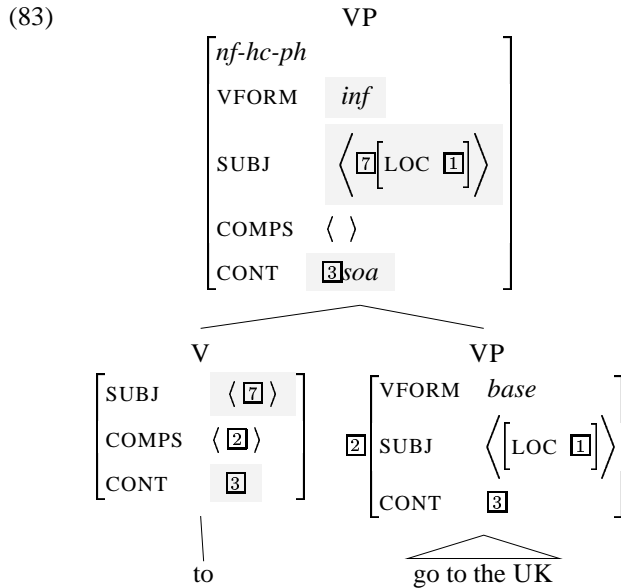
⁵³For some discussion, see Hooper and Thompson 1973, and Green 1976, 1996.

⁵⁴For arguments that apparent counterexamples, e.g. (i), are a phenomenon distinct from VP Ellipsis, see Hankamer and Sag 1976, Hankamer 1978 and Pullum 2000.

(i) Kim ordered us to leave, and we agreed.

a clause. In addition, note that the element on its SUBJ list is specified as [SLASH { }]. The consequences of the latter constraint will be explained in a moment.

The auxiliary *to* projects a head-complement phrase as an instance of the type *nf-hc-ph* introduced in section 2.6 above. A non-clausal *to*-phrase is illustrated in (83):



Note that the content of this phrase, like that of *to* and *to*'s VP[*base*] complement, is constrained only to be a *soa*. Hence, a phrase like (83) may be resolved as either an *r-soa* or an *i-soa* in the appropriate context. A *soa*-denoting phrases like (83) can be selected as a complement by a raising verb (in which case its content is generally resolved to type *r-soa*) or by the complementizer *for* (in which case its content is resolved to type *i-soa*).

However, in other contexts, *to*-phrases convey more than just a *soa*. For example, as a controlled complement,⁵⁵ a *to*-phrase denotes an *outcome* and many uses of *to* phrases appear to denote propositions with a so-called 'arbitrary' interpretation of the unexpressed subject. Thus we claim that both kinds of meaning are possible for *to*-clauses:

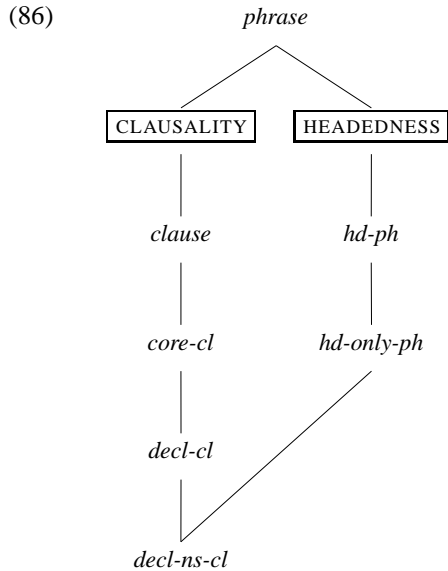
- (84) a. Lee wants [to be happy]. (*outcome*)
 b. They claimed [to know the answer to that question]. (*proposition*)

And when *to*-phrases stand alone as elliptical utterances or as short answers, they also acquire the force of a message.

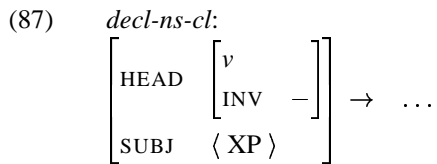
- (85) a. A: What do you want?
 B: *To go home.* (*outcome*)
 b. A: What are they claiming now?
 B: *To be able to read Hangul.* (*proposition*)

⁵⁵For an overview of the properties that distinguish control and raising, see Soames and Perlmutter 1979 or Sag and Wasow 1999.

The auxiliary *to* can give rise to such clauses in virtue of a further clause type: *declarative-nonsubject-clause* (*decl-ns-cl*), a subtype of *hd-only-ph* and *decl-cl*:

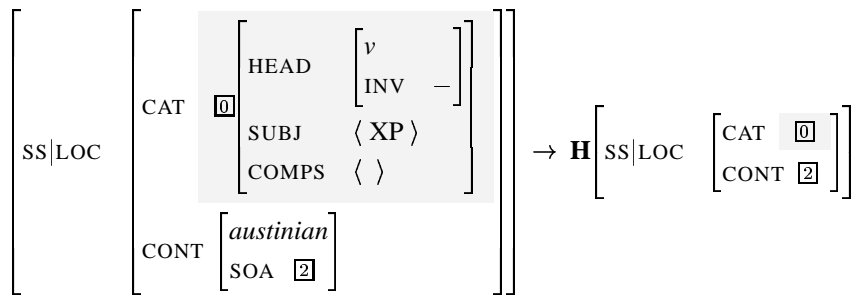


Instances of this type are constrained as follows:



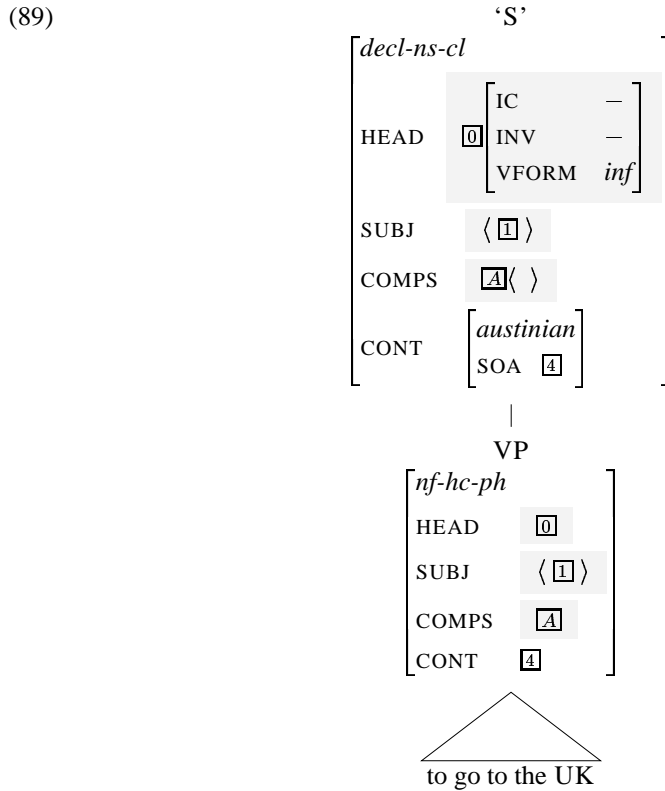
Constraint inheritance of a now familiar kind gives us the information in (88).⁵⁶

(88) GHFP & ECC & *decl-cl* & *hd-ns-cl*:



⁵⁶Our rule notation does not provide a perspicuous presentation of the information contributed by the type *hd-only-ph*, namely that the REST of the DTRS list is the empty list. See Appendix A.

This allows for *to*-clauses like (89) to be built from *soa*-denoting *to*-phrases like (83), once the effects of the GHFP and the constraint on the type *decl-cl* are taken into account:⁵⁷



Recall that there is a constraint on type *decl-cl* (see (54) above) requiring that declarative clauses have *austinian* content. Thus, because (87) says nothing about CONTENT, instances of this clausal type may have either subtype of *austinian*—*proposition* or *outcome*—as their content. As we will see in Chapter 6, it is this potential for *to*-phrases to denote propositions, as well as outcomes, that allows them to appear in *wh*-interrogative constructions (e.g. *who to visit*).

Crucially, other *soa*-denoting VPs, e.g. *going to the UK*, *gone to the UK* cannot serve as the head daughter of the *decl-ns-cl* construction. Such VPs are specified as [VFORM *prp*] or [VFORM *ppp*], both of which are incompatible (see (10) above) with the [VFORM *clausal*] constraint on this kind of phrase that is inherited from its supertype *core-cl* (see (51) above). That this constraint must be true of both mother and head daughter follows from the GHFP. Our claim, then, is that *to*-phrases are unique in being ambiguous between a nonclause whose content is a *soa* and a clause whose content is either a proposition or an outcome.

When a *to*-phrase has *soa* content, e.g. in raising contexts, no restrictions are placed on the unexpressed subject of that phrase (the element on its SUBJ list). That subject, whose LOCAL value is identified with the raised NP, can be any kind of nonreferential element, for example, as long as the verb heading the complement of *to* selects for a nonreferential NP of that particular kind:

⁵⁷We abbreviate such clauses as ‘S’, rather than as S or VP.

- (90) a. I believe there [to be a problem here]/*[to like Sandy].
 b. I believe it [to be raining]/*[to like Sandy].
 c. It's unlikely [for there [to be a solution here]/*[to like Sandy]].
 d. Close tabs seem [to have been kept on Kim]/*[to bother them].

Yet when a *to*-phrase appears in a non-raising context (e.g. those in (91)), the unexpressed subject of the infinitival phrase (always a clause on our analysis) must be referential.

- (91) a. *To run* would be wonderful.
 b. I want *to be running*.
 c. A: What do you want now?
 B: *To run*.
 B': *To bother them*.

Thus when the verb heading the complement of *to* selects for a nonreferential subject, such *to*-phrases are systematically ill-formed:

- (92) a. **To rain* would be wonderful. (cf. *For it to rain* would be wonderful.)
 b. *I want *to be raining*. (cf. I want it to be raining.)
 c. A: What do you want now?
 B: **To rain*.
 B': **To bother them that I'm not there*.

Moreover, the requirement that an unexpressed subject must be referential applies to all clauses in English, even those embedded within *wh*-questions (e.g. to all *to*-phrases other than raising complements), as the following additional examples illustrate:

- (93) a. *a yard [in which [to be a party]]. (cf. a yard [in which [to have a party]])
 b. *I wonder [where [to be a riot]] (cf. I wonder [where [there is a riot]])
 c. *Bother you(rself) that Kim left!
 d. *[Raining] would bother them.

In our analysis, these facts are all accounted for by a constraint governing all clauses:

- (94) Clause Constraint:
clause:
 $\left[\text{SUBJ } list(noncan-ss) \right] \rightarrow \dots$

The Clause Constraint says that the SUBJ value of all clauses must be a list each of whose members is of type (i.e. belongs to some subtype of) *noncan-ss*. Given that a SUBJ list has at most one member, the effect of (94) is to guarantee that a clause's SUBJ value is one of the following:

- (95) a. $\langle \rangle$ b. $\langle pro-ss \rangle$ c. $\langle gap-ss \rangle$

This follows because *pro-ss* and *gap-ss* are the only two subtypes of *noncan-ss* and the empty list is also a subtype of *list(noncan-ss)*.

Both a *gap-ss* element (corresponding to an extracted element, which can never be an expletive pronoun) and an unexpressed pronominal are referential in our analysis. In fact, elements of type *pro-ss* are always reflexive and *accusative*. The following constraint encodes these properties of unexpressed pronominals.

$$(96) \quad pro\text{-}ss \Rightarrow \left[\begin{array}{l} \text{HEAD} | \text{CASE } acc \\ \text{CONT} \left[\begin{array}{l} reflexive \\ \text{INDEX } ref \end{array} \right] \end{array} \right]$$

Since a *pro-ss* must have a referential index, it is inconsistent with the subject specification of any dummy-selecting predicate. Hence no infinitival clause can have a head daughter of the sort that would normally combine with an expletive subject (e.g. *to be raining*). This is a correct consequence, as we have seen. Moreover, the requirement that the CONTENT value of *pro-ss* be of type *reflexive* guarantees that the binding and control assignment behavior of *pro-ss* interact to explain ‘Visser’s Generalization’ (Bresnan (1982a)) in its full generality, as discussed in P&S–94, chaps. 6–7.⁵⁸

The Clause Constraint plays an important role in our theory. It interacts with other, independently motivated constraints to limit the possible SUBJ values of the particular clausal constructions that we analyze. Basically, the Clause Constraint guarantees that clauses either (1) are ‘subject-saturated’ (i.e. have an empty SUBJ list), (2) have an unexpressed ‘PRO’ subject, or else (3) have a subject that is extracted, corresponding to the three options shown in (95). The Clause Constraint will in fact guarantee that the only option for clausal *to*-phrases is to have a singleton SUBJ list containing a *pro-ss*. This follows because (1) the lexical entry for *to* ((82)) bars ‘slashed’ subjects,⁵⁹ while *gap-ss* elements are always slashed (see Chapter 5) and (2) the type *decl-hd-su-cl* discussed earlier allows only finite instances—thus there is no way in our grammar to build up a clause like *Sandy to leave*.

Since there is no way to construct a phrase like *Sandy to leave* in our analysis, there is no way that such a sequence can satisfy the specification [VFORM *inf*, SUBJ ⟨ ⟩]. This is a desired result, enabling us to specify ‘(for NP) to VP’ as a natural class: [VFORM *inf*]. This class, excluding ‘NP to VP’ sequences, is relevant for the grammar of such constructions as purpose clauses and infinitival relatives:⁶⁰

- (97) a. I bought it [to go to California in].
 b. I bought it [for us to go to California in].
 c. *I bought it [us to go to California in].
- (98) a. The car [to go to California in] is a Lincoln.
 b. The car [for us to go to California in] is a Lincoln.
 c. *The car [us to go to California in] is a Lincoln.

⁵⁸And, because *pro-ss* elements must be accusative, it follows that there are no unexpressed subject clauses headed by finite verbs (which require nominative subjects), e.g. no clauses like (i).

(i) *Kim wanted/tried/... goes to the store.

This familiar correlation follows from the interaction of (95), the Clause Constraint, and the lexical constraint requiring that finite verbs have nominative subjects.

⁵⁹This is actually a more general constraint, we believe, applying to all nonfinite verbal forms, and is independently motivated by the interaction of raising and extraction, as discussed briefly in Chapter 5. This analysis of raised subjects might benefit from the introduction of a new synsem type—*raised-synsem*, which is specified as [SLASH { }] and [WH { }], but we will not explore this refinement here.

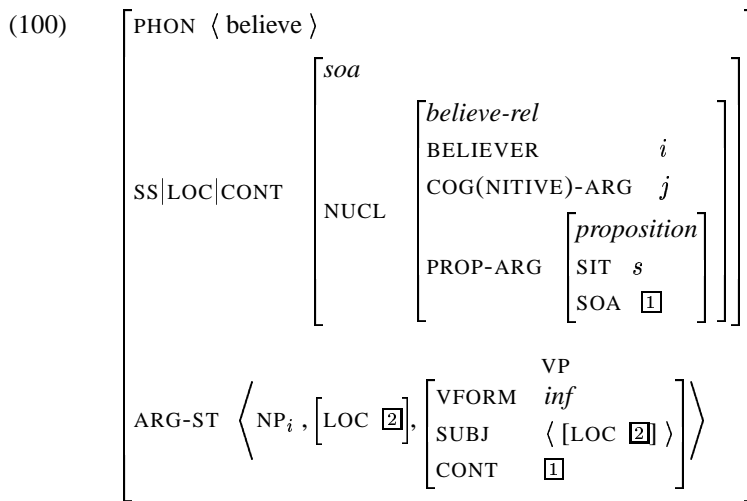
⁶⁰See Green 1991 and Johnston 1999.

Of course, we can still distinguish between *to*-phrases and *for*-phrases via the SUBJ value of a [VFORM *inf*] phrase or by specifying the HEAD value as *v* vs. *c*. Thus it is straightforward to write a lexical entry for a verb like *try*, which cannot take a *for-to*-clause as complement:

- (99) a. I tried [to go to California].
 b. *I tried [for us to go to California].
 c. *I tried [us to go to California].

2.9 Proposition-Embedding Verbs

A phrase like (83) (*to go to the UK*) can serve as the complement of most raising verbs. For example, the lexeme *believe* will include the information in (100), some of which is derived via constraint inheritance.⁶¹



Because the *to*-phrase here is not a clause, the Clause Constraint is inapplicable. Therefore the unexpressed subject of the VP[*inf*] argument and the object of *believe* are free to be of any type. This allows for nonreferential objects in examples like (101).

- (101) a. We believe [it] [to be obvious that Brooke is the one].
 b. Fergie believed [there] [to be no solution to this problem].
 c. Jan believes [it] [to be snowing now].

Similarly, a *soa*-denoting *to*-phrase may serve as complement of the complementizer *for*, as shown in (76) above. But a *pro-ss* is ruled out in both (76) (*for them to go to the UK*) and (101), as the object of *for* or *believe* must be an overt element, i.e. a *sign*,⁶² and hence [SYNSEM *canon*].

⁶¹We follow Cooper and Ginzburg (1996), who assume that attitudinal predicates carry an argument, labeled here as COGNITIVE-ARG, filled by a contextually supplied mental situation of the reported agent. For further discussion, see Chapter 8, note 35.

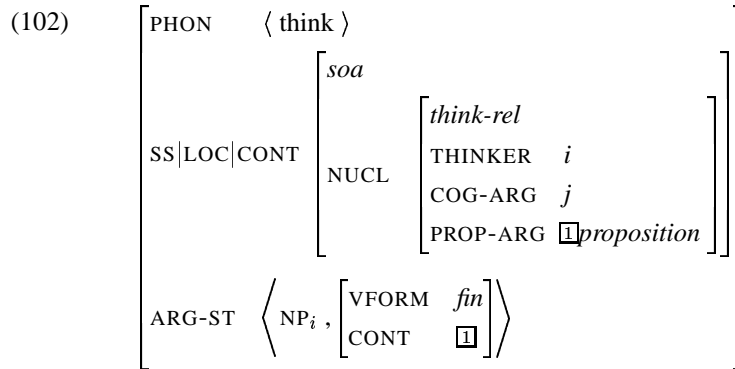
⁶²Note, however, that a raised element whose LOCAL value is shared with that of the unexpressed subject of a *to*-phrase can be extracted, as shown in (i):

- (i) Who did you believe to be the best candidate?

This follows, given our treatment of raising as sharing of LOCAL values, (NB: not SYNSEM values). When there is extraction of a raised object, the object argument of the raising verb will be of type *gap-ss* (see section 5.1), and hence slashed. But this *synsem* is distinct from the subject of the infinitive, which is unslashed.

These predictions follow from the interaction of the Principle of Canonicity in (49) and the constraints stated directly on the type *hd-comp-ph* ((34) above).

Next consider a proposition-embedding verb like *think*. Its lexical entry includes the information in (102).



In the system developed in this chapter, there are only two kinds of phrase that potentially satisfy the selectional requirements of this verb: *that*-clauses and indicative instances of the type *decl-hd-su-cl*.⁶³ The predictions are the following (an asterisk here indicates information of the complement that is incompatible with the properties selected by *think*):

- (103)
- | | | |
|----|---------------------------------------|--|
| a. | They think [that Leslie is winning]. | (CP[<i>fin</i>]: <i>proposition</i>) |
| b. | They think [Leslie is winning]. | (S[<i>fin</i>]: <i>proposition</i>) |
| c. | *They think [Leslie be winning]. | (S[<i>fin</i>]:* <i>outcome</i>) |
| d. | *They think [that Leslie be winning]. | (CP[<i>fin</i>]:* <i>outcome</i>) |
| e. | *They think [Leslie to be winning]. | (no such clause) |
| f. | *They think [to be winning]. | (VP[* <i>inf</i>]: <i>proposition</i>) |
| g. | *They think [for her to be winning]. | (CP[* <i>inf</i>]:* <i>outcome</i>) |

The lexical entry for a proposition-embedding verb like *claim*, however, differs from that in (102) in that it lacks the [VFORM *fin*] restriction on its complement. This produces the following paradigm for *claim*:

- (104)
- | | | |
|----|---------------------------------------|---|
| a. | They claim [that Leslie is winning]. | (CP[<i>fin</i>]: <i>proposition</i>) |
| b. | They claim [Leslie is winning]. | (S[<i>fin</i>]: <i>proposition</i>) |
| c. | *They claim [Leslie be winning]. | (S[<i>fin</i>]:* <i>outcome</i>) |
| d. | *They claim [that Leslie be winning]. | (CP[<i>fin</i>]:* <i>outcome</i>) |
| e. | *They claim [Leslie to be winning]. | (no such clause) |
| f. | They claim [to be winning]. | (VP[<i>inf</i>]: <i>proposition</i>) |
| g. | *They claim [for her to be winning]. | (CP[<i>inf</i>]:* <i>outcome</i>) |

⁶³For the moment, we ignore sentences like (i), which involve extraction of an embedded subject:

(i) Who did you think [_ was winning]?

These examples are discussed in section 5.1.

Again, these contrasts seem precisely in accord with intuition.

Finally, consider the lexical entry for a verb like *insist* in its ‘demand’ sense:

$$(105) \left[\begin{array}{l} \text{PHON} \quad \langle \text{insist} \rangle \\ \text{SS|LOC|CONT} \quad \left[\begin{array}{l} \text{soa} \\ \text{NUCL} \quad \left[\begin{array}{l} \text{insist-rel} \\ \text{INSISTOR} \quad i \\ \text{COG-ARG} \quad j \\ \text{OUTCM-ARG} \quad \boxed{1} \text{outcome} \end{array} \right] \end{array} \right] \\ \text{ARG-ST} \quad \left\langle \text{NP}_i, \left[\begin{array}{l} \text{VFORM} \quad \text{fin} \\ \text{CONT} \quad \boxed{1} \end{array} \right] \right\rangle \end{array} \right]$$

Our system allows exactly two kinds of clause as the complement of verbs like (105), as shown in (106)a,b:

- (106) a. They insist [he stay outside]. (S[fin]:outcome)
- b. They insist [that he stay outside]. (CP[fin]:outcome)
- c. *They insist [Leslie to stay outside]. (no such clause)
- d. *They insist [to stay outside]. (VP[*inf]:outcome)
- e. *They insist [for him to stay outside]. (CP[*inf]:outcome)
- f. They insist [that they’re incompetent]. (CP[fin]:*proposition|*fact)
- g. They insist [they’re incompetent]. (S[fin]:*proposition|*fact)

The last two examples are not unacceptable, of course. Rather they illustrate a different sense of the verb *insist*—one we may paraphrase as ‘maintain the truth of’. In this sense, the verb takes a proposition-denoting complement.

2.10 Fact-Denoting Declarative Clauses

In the next chapter, we will provide evidence that finite declarative clauses lead a double semantic life as *facts*, a type of message distinct from *proposition* and *outcome*. To accommodate this evidence, we introduce a new clausal type—a subtype of both *core-cl* and *hd-only-ph*—which ‘promotes’ a proposition-denoting sentence to one that denotes a fact. The constraints idiosyncratic to this construction are illustrated in (107).

$$(107) \quad \text{factive-cl:} \quad \left[\begin{array}{l} \text{HEAD} \quad \left[\begin{array}{l} \text{verbal} \\ \text{VFORM} \quad \text{fin} \end{array} \right] \\ \text{CONT} \quad \left[\begin{array}{l} \text{fact} \\ \text{PROP} \quad \boxed{1} \end{array} \right] \end{array} \right] \rightarrow \mathbf{H}[\text{CONT} \quad \boxed{1}]$$

This construction applies to both CPs and Ss (both main and embedded), allowing us to assign a *fact*-type meaning to all finite declarative clauses.

In providing a second semantic life for declarative clauses as factives, our analysis leads us to the view that such sentences have a semantic overlap with exclamatives, which we treat uniformly as having a *fact*-type meaning. This is an intuitive result, as exclamatives always seem to have paraphrases that have the outward appearance of declaratives:

- (108) a. Is Kim clever!
b. Kim IS CLEVER.
- (109) a. How clever Kim is!
b. Kim is truly clever.
c. Mmm. Mmm. KIM...IS...CLEVER.

These effects rely on context, and sometimes in addition on the presence of emphatic modifiers or the pragmatic help of intonation.

2.11 Summary

This chapter has built up an analysis of the basic declarative clause constructions in English. The system of clausal features and types presented here makes it possible to specify natural classes for purposes of lexical selection. Following the approach articulated by Grimshaw (1979) and others, this selection is partly semantic and partly syntactic. Semantic types and VFORM distinctions play a significant role in our account of this, as does our theory of declarative clause types.

The various phrasal types we have proposed are summarized in (110):

