English Interrogative Constructions

Jonathan Ginzburg
Hebrew University

Ivan A. Sag
Stanford University

Draft of January, 2000
Comments Welcome

Contents

Chapter 1

Introduction

Chapter 2

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 modelling 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 constuctions, 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 modelled 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 constraints on the sound-meaning relation that are conventional 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 — having to do with everything from semantic predications, and generalized quantifiers to person and case — must be made. In service of this, a grammar must posit many kinds of linguistic entities 'smaller' than the signs, and must provide an account of their specific properties. The grammar of a language thus consists of (minimally) a specification of:

- the set of types that play a role in the grammar — a linguistic ontology,

- which features are appropriate for each type,
- what type of value is appropriate for each such feature, and
- all the constraints that must be true of instances of particular types. (These are usually referred to simply as 'type constraints'.)

The modelling assumptions of HPSG have provided a novel way of working with certain traditional notions of grammar, for example 'lexical entry' and 'phrase structure rule' that is allows increasing precision and analytic uniformity to be obtained. Lexical entries are descriptions of (or constraints on) feature structures of type *word*; phrase structure rules (or 'immediate dominance schemata') are partial descriptions of feature structures of type *phrase*. These are the two immediate subtypes of the type *sign*.

2.3 Words

In the case of words, then, an HPSG grammar must specify an inventory of lexical types and the various constraints that words obey. We will follow the common practice of formulating lexical descriptions (constraints on objects of type *lexeme* or *word*) in the language of attribute-value matrices (AVMs) like the one sketched in (1).[1]

$$
\text{(1)}\quad \left[\begin{array}{ll} \textit{lexeme} & \\ \text{PHONOLOGY} & \langle \text{ prove } \rangle \\ \text{SYNSEM} & \left[\begin{array}{ll} \textit{synsem} & \\ \text{LOCAL} & \left[\begin{array}{ll} \textit{local} & \\ \text{CATEGORY} & \left[\begin{array}{ll} \textit{category} & \\ \text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{AUX} & - \end{array}\right] \\ \text{ARG-ST} & \langle \text{ NP}_{\boxed{1}} , \text{NP}_{\boxed{2}} \rangle \end{array}\right] \\ \text{CONTENT} & \left[\begin{array}{ll} \textit{soa} & \\ \text{NUCL} & \left[\begin{array}{ll} \textit{prove-rel(ation)} & \\ \text{ACTOR} & \boxed{1} \\ \text{UNDERGOER} & \boxed{2} \end{array}\right] \end{array}\right] \end{array}\right] \\ \text{SLASH} & \ldots \\ \text{WH} & \ldots \end{array}\right] \end{array}\right]
$$

Lexical descriptions like these specify complexes of phonological, syntactic and semantic information that are satisfied by a family of feature structures. These feature structures are

[1]The feature WH correponds 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.

organized 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 theory of locality. This theory seeks to accont for the empirical fact that subcategorization (category selection in the familiar sense), case and role assignment, semantic selection, and head-dependent agreement all operate in highly constrained 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.[2] Similar considerations motivate the supposition of *local* objects (these encapsulate the information transmitted in raising and extraction dependencies), and the other embedded feature stuctures illustrated in (1).

The feature geometry illustrated in (1) is a consequence of the linguistic ontology specified by the grammar. In particular, a grammar must provide a complete specification of what types of feature structure exist and how they are organized into a hierarchy, i.e. for each type, what its immediate supertypes (IST) are. The grammar must further specify which features are appropiate for each type of feature structure and also what type of value is appropriate for each feature. Such specifications are illustrated in (2):

(2)

TYPE	FEATURES/TYPE OF VALUE	IST
sign	$\begin{bmatrix} \text{PHONOLOGY} & \text{list}(\textit{speech-sound}) \\ \text{SYNSEM} & \textit{synsem} \end{bmatrix}$	*feat-struc*
phrase	...	*sign*
word	...	*sign*
synsem	$\begin{bmatrix} \text{LOCAL} & \textit{local} \\ \text{SLASH} & \text{set}(\textit{local}) \\ \text{WH} & \textit{operator} \mid \textit{none} \\ \quad \ldots & \end{bmatrix}$	*feat-struc*
local	$\begin{bmatrix} \text{CATEGORY} & \textit{category} \\ \text{CONTENT} & \textit{sem-object} \end{bmatrix}$	*feat-struc*
...	...	...

We will add further details about types, appropriate features, and constraints of various kinds as we develop our theory.

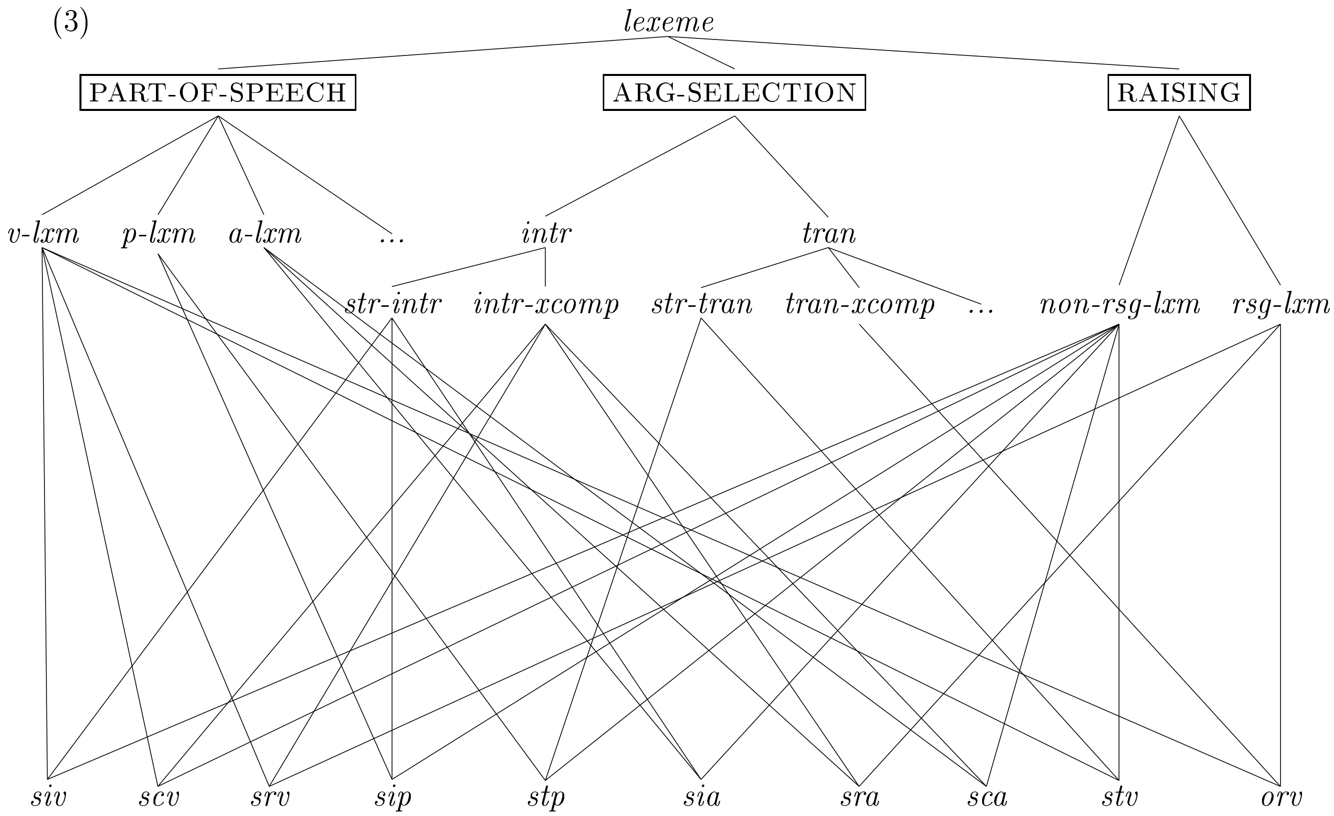
Note first that very little of the information in a lexical entry like (1) must be listed in the lexicon, as lexical types, type inheritance, and the theory of linking[3] allow complex

[2]The *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. to appear.

[3]See Wechsler 1995, Davis 1996, to appear, Koenig and Davis to appear.

lexical information such as that illustrated here to be be inferred, rather than stipulated — that is, much of this information is derived via the logic of the lexicon. For example, general (inflectional) rules relate individual lexemes to a family of inflected forms and a general organization of lexemes breaks them down into families whose members share grammatically significant properties, stated as constraints on lexical classes (lexical types). The resulting lexical architecture can be represented by a multiple inheritance hierarchy where, for example, 'part of speech' and 'argument selection' provide independent dimensions of constraint, as illustrated in (3).[4]

(3)



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 illustrated in (4).[5]

(4) a. *siv*: strict-intransitive-verb-lexeme (e.g. *die*)

b. *srv*: subject-raising-verb-lexeme (e.g. *seem*)

c. *scv*: subject-control-verb-lexeme (e.g. *try*)

[4]The RAISING partition will play a role in our analysis (Chapter 5) of how words amalgamate and pass up the SLASH values and the stored operators of their arguments. The generalization we require is that a verb does not amalgamate the SLASH and STORE values of its 'raised' arguments. In Chapter 5, we add one further informational dimension in order to distinguish argumental lexemes (those that take arguments) from non-argumental lexemes.

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

d. *sip*: strict-intransitive-preposition-lexeme (e.g. *of*)

e. *stp*: strict-transitive-preposition-lexeme (e.g. *in*)

f. *sia*: strict-intransitive-adjective-lexeme (e.g. *big*)

g. *sra*: subject-raising-adjective-lexeme (e.g. *likely*)

h. *sca*: subject-control-adjective-lexeme (e.g. *eager*)

i. *stv*: strict-transitive-verb-lexeme (e.g. *prove*)

j. *orv*: object-raising-verb-lexeme (e.g. *believe*)

Note that *of* is here classified as an *sip* because it generally takes only one argument — its object. 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, i.e. with *the nail*.[6]

This mode of lexical analysis, pioneered by early work in HPSG and closely related descriptive frameworks, reflects the fundamental fact that lexical classes share cross-cutting partial similarities, as reflected by constraints on various types in the hierarchy, for example the following:[7]

(5) a. $$v\text{-}lxm \Rightarrow \begin{bmatrix} \text{SS}|\text{LOC}|\text{CAT} & \begin{bmatrix} \text{HEAD} & verb \\ \text{SPR} & \langle\ \rangle \\ \text{SUBJ} & \langle\, [\ ]\, \rangle \end{bmatrix} \end{bmatrix}$$

b. $$p\text{-}lxm \Rightarrow \begin{bmatrix} \text{SS}|\text{LOC}|\text{CAT}|\text{HEAD} & prep \end{bmatrix}$$

c. $$str\text{-}intr \Rightarrow \begin{bmatrix} \text{SS}|\text{LOC}|\text{CAT}|\text{ARG-ST} & \langle\, \text{NP}\, \rangle \end{bmatrix}$$

d. $$orv \Rightarrow \begin{bmatrix} \text{SS}|\text{LOC}|\text{CAT}|\text{ARG-ST} & \left\langle \text{NP}\ ,\ \begin{bmatrix} \text{LOC} & \boxed{1} \end{bmatrix}\ ,\ \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix} \text{LOC} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

e. $$intr\text{-}xcomp \Rightarrow \begin{bmatrix} \text{SS}|\text{LOC}|\text{CAT}|\text{ARG-ST} & \left\langle \text{NP}\ ,\ \begin{bmatrix} \text{SUBJ} & \left\langle [\ ] \right\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

[6] For further discussion of this distinction, see Sag and Wasow (1999: chap. 7).

[7] ‘⟨[ ]⟩’ describes a list containing exactly one element; ‘⟨ ⟩’ describes 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. (1999), nonlocal 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.

f. $\textit{s-ctrl} \Rightarrow \left[\text{SS}|\text{LOC}|\text{CAT}|\text{ARG-ST} \quad \langle\ \text{NP}_i\ ,\left[\text{SUBJ}\ \langle \text{NP}_i\rangle\right]\rangle\right]$

g. $\textit{str-tran} \Rightarrow \left[\text{SS}|\text{LOC}|\text{CAT}|\text{ARG-ST} \quad \langle\ \text{NP}\ ,\ \text{NP}\ \rangle\right]$

h. ...

The type constraints in (5) should be understood as constraints that state 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 (5)) 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 on 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 use default constraints, in each description of an instance of a maximal type, e.g. a full description of the lexeme *give* according to our grammar, there are only hard constraints, as all the consistent default constraints on superordinate types become rigid through inheritance.

Finally, consider the following principle, essential to our treatment of words, that relates ARG-ST lists to the valence features SUBJ, COMPS, and SPR:[8]

(6) Argument Realization Principle (ARP; Preliminary Formulation):

$$\textit{word} \Rightarrow \left[\text{SS}|\text{LOC}|\text{CAT} \begin{bmatrix} \text{ARG-ST} & \boxed{A} \oplus \boxed{B} \oplus \boxed{C} \\ \text{SUBJ} & \boxed{A} \\ \text{SPR} & \boxed{B} \\ \text{COMPS} & \boxed{C} \end{bmatrix}\right]$$

In this preliminary formulation, 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 $\langle\ \rangle$] and [SPR $\langle\ \rangle$], 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 (5a) and the ARP, it must include the following information:

(7)
$$\left[\text{SS}|\text{LOC}|\text{CAT} \begin{bmatrix} \text{ARG-ST} & \langle \boxed{3}\ ,\ \boxed{4} \rangle \\ \text{SUBJ} & \langle \boxed{3} \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle \boxed{4} \rangle \end{bmatrix}\right]$$

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 (*stv*) lexeme *prove* in (1).[9]

[8]Here and throughout we use capital letters to distinguish tags ranging over lists.

[9]The relation between lexemes and words can be captured either via lexical rules (see Sag and Wasow 1999 for an explicit proposal) or else by providing a type system where words are simultaneously classified

There is of course considerable cross-linguistic variation in argument realization. In the approach we adopt 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 in the formulation of the ARP. All of these variations, are intuitively 'subtractions' — that is, they involve *synsem* elements that belong to a word's ARG-ST list, but which fail to be realized on any of its valence lists (SUBJ, COMPS, or SPR). In Chapter 5, we present a modification of (6) that allows us to treat extracted arguments in terms of a discrepancy between a word's ARG-ST list and its COMPS list.

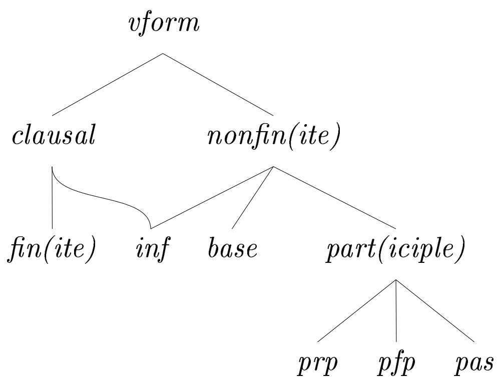
2.4 Features of Verbals

We follow Sag (1997) in assuming that the part of speech types (the values of the feature HEAD) associated with verbs (*verb*) and complementizers (*comp*) are subtypes of a common supertype called *verbal*. The features VERBFORM (VFORM) and IC (INDEPENDENT-CLAUSE), *inter alia*, are declared to be appropriate for all instances of this type, i.e. for both verbs and complementizers. Other features, e.g. AUX(ILIARY), and CINV (CLAUSE-INITIAL-VERB), are appropriate only for verbs.

2.4.1 Distinguishing Verbal Forms

The values of VFORM (*fin(ite)*, *inf(initive)*, *base*, *present-participle*(*prp*), *perfect-participle* (*pfp*), and *passive-participle* (*pas*)) are organized into a (multiple-inheritance) hierarchy as shown in (8):

(8)



These correspond to familiar distinctions among verb forms; the types in this hierarchy are motivated in part by lexical selection (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 VPs whos heads are [VFORM *base*]. In both cases, the VP complement's

in two dimensions: lexeme and inflection. For proposals along these lines, see Koenig 1998, Miller and Sag 1997, and Abeillé et al. 1998. For convenience, we will adopt the former approach here.

VFORM specification is the same as that of the VP's verbal head, in virtue of the (Generalized) Head Feature Principle discussed in the next section. The nonfinite auxiliary verb *to* is the only verb that is 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 touch on the evidence supporting the grammatical distinctions we have made. The groupings in (8) 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 (typically, at least) head clausal constructions. All of the clauses we study in this monograph (declaratives, interrogatives, exclamatives and imperatives) are headed by a verb that falls into this class, i.e. the class specifiable 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 (9).

(9) 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 *pfp*])

f. [Not given any awards at the banquet], Sandy went home disgruntled. ([VFORM *pas*])

Moreover, auxiliary verb forms that are [VFORM *fin*] (and are indicative or conditional, in addition) are the only ones that can select *not* as a complement when sentential negation is expressed:[10]

(10) 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 disdvantage.

[10]On this analysis of sentential negation, see Warner 1993, 1999; Kim and Sag 1995, to appear; 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 (10b) 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 (i).

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

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:

(11) *[Not [go to the store]]!

The type *part* is motivated by the grammar of modification. The participles (but not base form VPs, for example) all share the ability to modify nominals and the lexical rules or lexical types 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 cross-cutting 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, chapter 11).

2.4.2 Distinguishing Verbal Meanings

The content of a verb specifies a state-of-affairs, or *soa*. These *soa*s are the building blocks of the various kinds of message, e.g. *propositions*, *directives*, *questions*, that are described in detail in the next chapter. Certain types of message have a realis interpretation, while others do not. 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, as it is resolved according to whether the proposition that Kim Sanderson left is true or false (at the time and place of the utterance). By contrast, the meaning of an imperative sentence like *Get out of here!* does not involve the real world in the same way as a declarative or interrogative sentence. Nor do subjunctive or infinitival clauses (e.g. *I insist that Kim Sanderson leave.* or *It's hard for Kim Sanderson to leave.*) involve realis meanings. Intuitively, these involve no consideration of whether Kim Sanderson has left or is now actually leaving. Rather, all three kinds of clause make reference to future outcomes involving Kim Sanderson's leaving. The distinction between these two types of content — realis and irrealis — seems fundamental.

We propose that this bifurcation of clausal meanings be reflected in terms of a bifurcation of the type *soa* into two subtypes which we will refer to as *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 general constraints guaranteeing, for example, that propositions can be constructed only from *r-soa*s, while directives and the like are constructed only from *i-soa*s, will modulate the kinds of meaning that can be associated with the phrasal constructions allowed by our grammar.[11]

Inflected forms of verbs are derived from verbal lexemes which 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 require that their output be further restricted to have content type *r-soa*, as illustrated in (12), the lexical description of the 3rd-singular present indicative form *proves*.

(12)
$$
\left[\begin{array}{ll}
\textit{word} & \\
\text{PHONOLOGY} & \langle \text{ proves } \rangle \\
\text{SYNSEM} & \left[\begin{array}{ll}
\textit{synsem} & \\
\text{LOCAL} & \left[\begin{array}{ll}
\textit{local} & \\
\text{CATEGORY} & \left[\begin{array}{ll}
\textit{category} & \\
\text{HEAD} & \left[\begin{array}{ll} \textit{verb} & \\ \text{AUX} & - \\ \text{NEG} & - \\ \text{VFORM} & \textit{fin} \end{array}\right] \\
\text{ARG-ST} & \left\langle \text{NP}[\textit{nom},\textit{3sing}]_{\boxed{1}} \text{ , } \text{NP}_{\boxed{2}} \right\rangle
\end{array}\right] \\
\text{CONTENT} & \left[\begin{array}{ll}
\textit{r-soa} & \\
\text{NUCL} & \left[\begin{array}{ll} \textit{prove-rel} & \\ \text{ACTOR} & \boxed{1} \\ \text{UNDERGOER} & \boxed{2} \end{array}\right]
\end{array}\right]
\end{array}\right] \\
\text{SLASH} & \ldots \\
\text{WH} & \ldots
\end{array}\right]
\end{array}\right]
$$

All inflectional rules forming nonfinite forms, e.g. participles or *base* verbal forms, say nothing about the content type, and hence preserve the lexeme's semantic indeterminacy. This allows the content of a participle, for example, to be the *r-soa* required by a proposition, as in (13a) or the*i-soa* required by a directive, as in (13b).

---

[11]Fixing 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 *soa*s, there can be a uniform semantics for the combination of the VP with the modifier. This happens at a lower level of structure — before the *soa* is incorporated into the meaning of the clause.

(13) a. Kim Sanderson is *proving* an important theorem at this very moment.

b. Be *proving* a theorem when your math teacher walks in!

We may now assume one more inflectional rule creating (uninflected) finite forms with content of type *i-soa*. These finite forms, like the one illustrated in (14), will appear in imperative clauses like (15a) or in subjunctive clauses like (15b).

(14)
$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}} \\
\text{PHONOLOGY} & \langle \text{ prove } \rangle \\
\text{SYNSEM} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{synsem}} \\
\text{LOCAL} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{local}} \\
\text{CATEGORY} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}} \\
\text{HEAD} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{verb}} \\
\text{AUX} & - \\
\text{NEG} & - \\
\text{VFORM} & \textit{fin}
\end{array}\right] \\
\text{ARG-ST} & \left\langle \text{NP}[\textit{nom}]_{\boxed{1}}\text{ , NP}_{\boxed{2}} \right\rangle
\end{array}\right] \\
\text{CONTENT} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{i-soa}} \\
\text{NUCL} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{prove-rel}} \\
\text{ACTOR} & \boxed{1} \\
\text{UNDERGOER} & \boxed{2}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\text{SLASH} & \ldots \\
\text{WH} & \ldots
\end{array}\right]
\end{array}\right]$$

(15) a. *Be* waiting for me!

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:

(16) a. I suggested (that) they *be* made available.

b. *I suggested (that) them *be* made available.

In fact, the following kind of example may provide independent support for this case assignment, even for imperative uses:[12]

(17) You and he *be* quiet!

[12]The 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.

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:

(18) 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:

(19) 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*, as proposed for French by Calcagno and Pollard (1997)). 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*:

(20) a. I urged that they not attend the reception.

b. *I urged that they attend not the reception.

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

(21) 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 (21) suggests that there is a homophonous *not* that functions as a subjunctive auxiliary verb. 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 verb *not* has a negative *i-soa* as its CONTENT value. We block analogous examples involving matrix imperatives, e.g. (22), by restricting this auxiliary to embedded clauses, i.e. by an [INDEPENDENT-CLAUSE −] ([IC −]) lexical specification.

(i) You and him *be* quiet!

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

Note, however, that many languages have imperative inflectional 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.

(22) *Not go to the store!

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

Our assumptions about the English auxiliary system diverge in two significant ways from previous work in the PSG tradition. First, following Sag (1999), we utilize the specification [AUX +] not to distinguish auxiliary verbs from other verbs, but rather to identify auxiliary constructions. Thus the inversion construction, instances of sentential negation, and so forth will be specified as [AUX +] and auxiliary verbs — lexically unspecified for AUX — will be the only verbs that are compatible with these. This enables a simple lexical account of the restricted distribution of unfocussed *do*: this *do* is lexically specified as [AUX +] and hence is compatible ONLY with the auxiliary constructions. This allows an account of (23), as discussed in section 2.6 below.

(23) *Kim dĭd leave.

A second point of divergence with previous PSG work on auxiliaries involves the feature CINV (CLAUSE-INITIAL-VERB). This is analogous to the binary feature INV posited by Gazdar et al. (1982) — it is used to distinguish auxiliary verbs heading inverted phrases from others.[13] Our analysis is distinctive, however, in that finite verbs (whether they are auxiliaries or nonauxiliaries) have no lexically specified CINV value. This allows them to be [CINV +] in appropriate environments. As discussed in the next section and in Chapter 6, this provides a uniform account of inversion in matrix *wh*-interrogatives, where the verbs in (24) are all treated as [CINV +].

(24) a. Who *will* they visit?

b. Who *will* visit them?

c. Who *visited* them?

2.5 Phrases as Feature Structures

Phrases too can be modelled as typed feature structures, as they have been since the earliest work in HPSG. Features such as HD-DTR (whose value is a sign (i.e. a word or phrase) and NON-HD-DTRS (whose value is a list of signs) encode roughly the same information that branches do in conventional phrase structure trees.[14] The figure in (25) presents a simplified analysis in feature structure terms of the sentence *Leslie drinks milk*.

[13] The lexical encoding of this information allows one to accommodate lexical exceptions to inversion — both positive and negative. For discussion, see Gazdar et al. 1982, Gazdar et al. 1985, Green and Morgan 1996, Kim and Sag 1995, to appear, and Warner 1999.

[14] This 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.

(25)
$$
\begin{bmatrix}
\textit{hd-subj-ph} & \\
\text{PHON} & \langle \boxed{1}, \boxed{3}, \boxed{4} \rangle \\
\text{SYNSEM} & \text{S} \\
\text{NON-HD-DTRS} & \left\langle \begin{bmatrix} \text{PHON} & \langle \boxed{1}\text{Leslie} \rangle \\ \text{SYNSEM} & \text{NP} \end{bmatrix} \right\rangle \\
\text{HD-DTR} & \begin{bmatrix}
\textit{hd-comp-ph} & \\
\text{PHON} & \langle \boxed{3}, \boxed{4} \rangle \\
\text{SYNSEM} & \text{VP} \\
\text{HD-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \boxed{3}\text{drinks} \rangle \end{bmatrix} \\
\text{NON-HD-DTRS} & \left\langle \begin{bmatrix} \text{PHON} & \langle \boxed{4}\text{milk} \rangle \\ \text{SYNSEM} & \text{NP} \end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix}
$$

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 diagram. 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 (see Chapter 1, sec. ??):

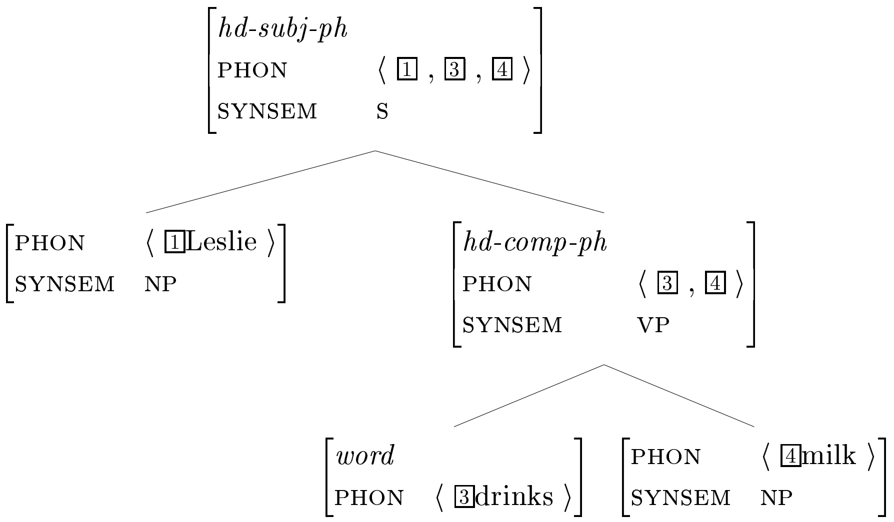
- How are specific constructions related to one another?
- How can cross-constructional generalizations be expressed?
- How can constructional idiosyncrasy be accounted for?

Put simply, our approach allows cross-classifying generalizations about constructions to be expressed, explaining 'family resemblance' across constructions. At the same time, it allows construction-particular idiosyncrasy. The sign-based approach plays a particularly useful role in the theory of clause types we develop below. In addition, it allows us to extend the application of general grammatical constraints to more and more 'peripheral' constructions, without having to posit an otherwise unmotivated bifurcation between the 'core' and 'periphery' of language.[15]

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 represent our feature structure based theory of phrases in terms of more familiar tree diagrams. A tree representation for (25), for example, is shown in (26).

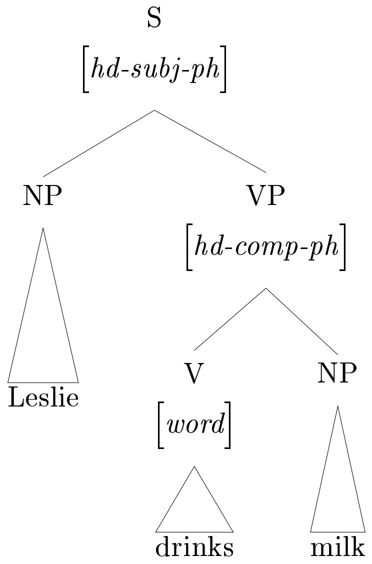
[15]See, for example, the discussion of Fillmore and Kay (1999, to appear) and Bender and Flickinger 1999.

(26)



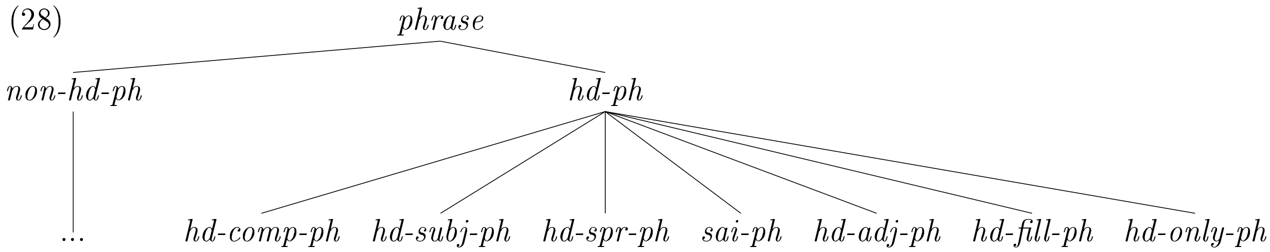
In fact, whenever possible, we will abbreviate tree diagrams like (26) in an even more familiar format, e.g. (27).

(27)



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

(28)



Phrases are thus classified as either *headed-phrase* (*hd-ph*) or *non-headed-phrase* (*non-hd-ph*), with each type exhibiting a variety of 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.[16] We will illustrate each of these types in due course.

Phrases are governed by the feature declarations shown in (29):

(29)

TYPE	FEATURES/TYPE OF VALUE	IST
sign	[PHONOLOGY list(*speech-sound*) SYNSEM *canon-ss*]	*feat-struc*
phrase	[NON-HD-DTRS list(*sign*)]	*sign*
hd-ph	[HD-DTR *sign*]	*phrase*

Each headed phrase must have some value (possibly the empty list) for NON-HD-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.

Phrases are subject to the following general constraint:[17]

(30) Empty COMPS Constraint (ECC):

phrase $\Rightarrow$ [COMPS $\langle\ \rangle$]

(30) guarantees that within any phrase, the complements have been 'already 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 are more deeply embedded than specifiers, subjects, or fillers, all of which combine with head daughters that must be phrasal, and hence [COMPS $\langle\ \rangle$], according to (30).

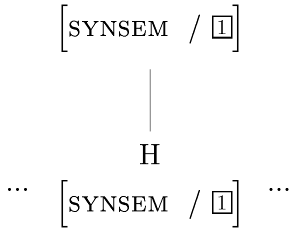
Just as in the case of the lexicon, certain kinds of phrase obey type-specific constraints. For example, our generalization of the Head Feature Principle (analogous to the 'X' part of '$\bar{\text{X}}$ Theory') can be formulated as a constraint on phrases of the type *hd-ph*: (The symbol 'H' here is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

[16] The phrasal type *sai-ph* is based on the SAI construction of Fillmore 1999. Note that we make no use here of the type *hd-marker-ph*, although this may in fact be needed for the treatment of conjunctions.

[17] Henceforth, abbreviations of an obvious sort will be used throughout. For example, [COMPS $\langle\ \rangle$] in (30) abbreviates [SS|LOC|CAT|COMPS $\langle\ \rangle$].

(31) Generalized Head Feature Principle (GHFP)

hd-ph $\Rightarrow$

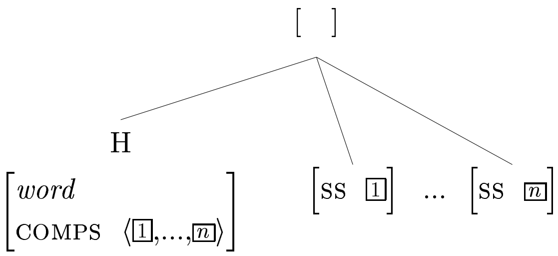


Here the '/' notation (Lascarides and Copestake 1999) indicates a default constraint, specifically one requiring that the SYNSEM value of the mother of a headed phrase and that of its head daughter are, by default, identical. Specific subtypes of *hd-ph* may override the GHFP, but by formulating (31) in defeasible terms, we need only state interacting constraints on subtypes when we want to circumvent inheritance from the head daughter.[18]

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 (Pollard and Sag (1994: 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 head-mother discrepancy (a cancellation) for the value of some particular valence feature.[19]

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 (including zero):

(32) *hd-comp-ph* $\Rightarrow$



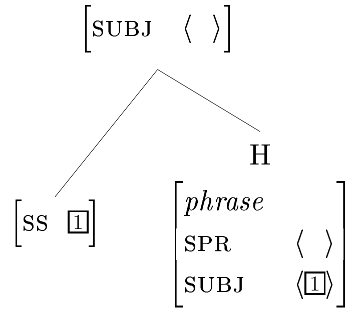
[18]Let us reemphasize that our use of defaults here is essentially abbreviatory in nature: one could replace the GHFP with a set of nondefault constraints, each of which specified the relevant identities on particular subtypes of *hd-ph*. By not doing so, however, we achieve a gain in descriptive simplicity which, as noted by Lascarides and Copestake (1999), is typical of systems using default constraints.

[19]Our 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.

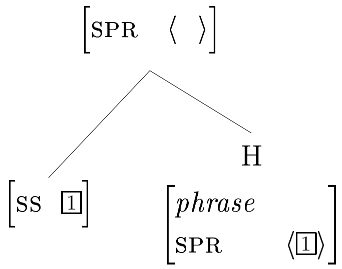
The constraint in (32) factors out only that information which is specific to phrases of this type.

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

(33) *hd-subj-ph* $\Rightarrow$



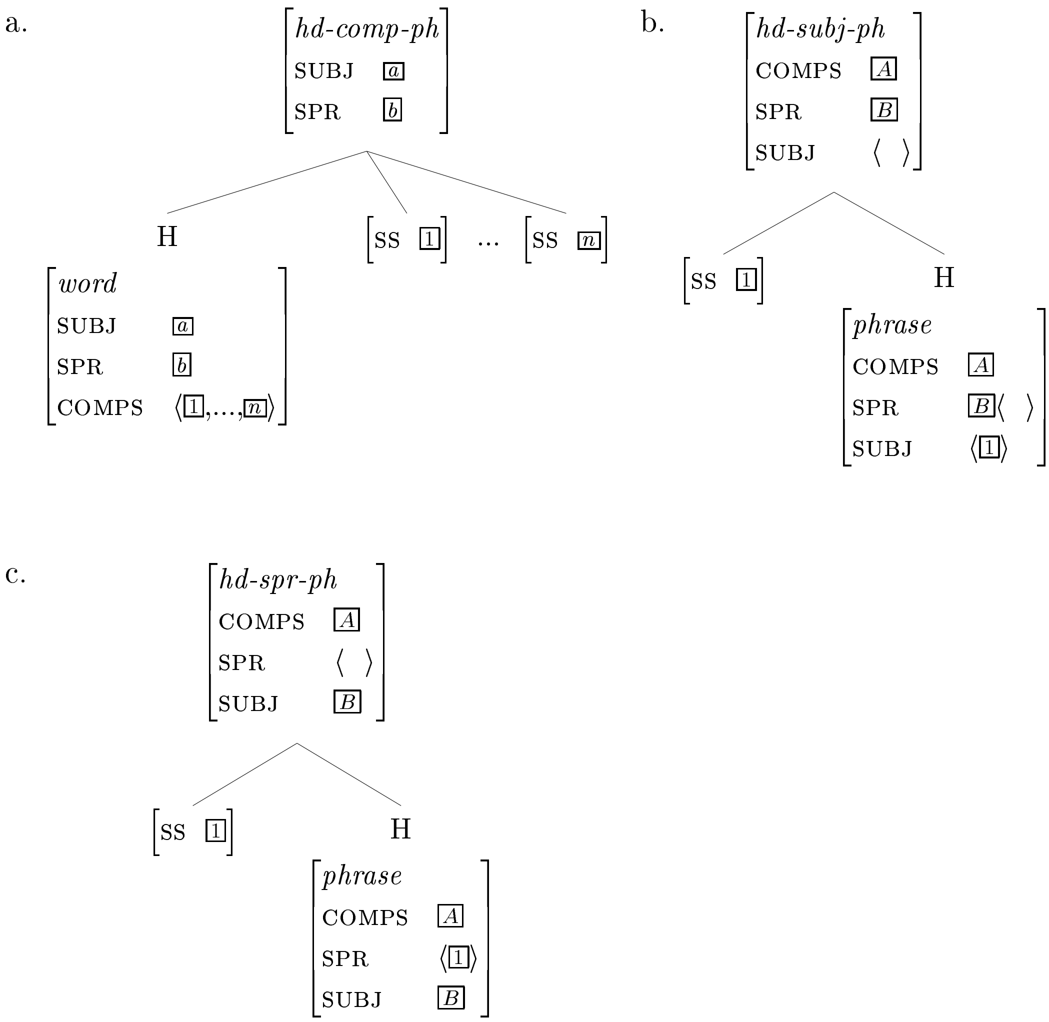
(34) *hd-spr-ph* $\Rightarrow$



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 (35a-c), the unification of the GHFP with the (32)–(34):

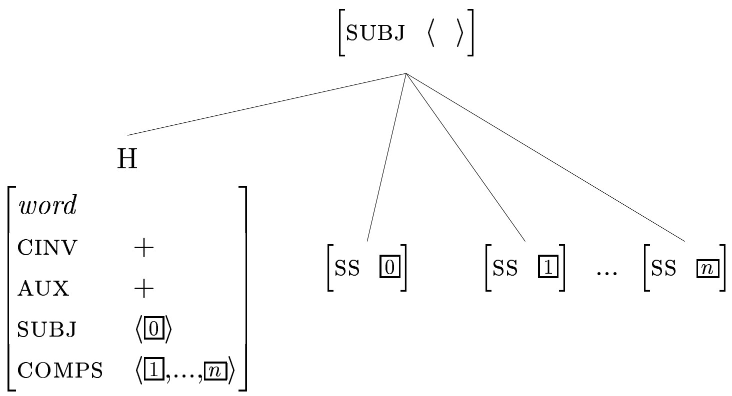
(35)



And all of the phrases in (35a–c) are in addition [COMPS ⟨ ⟩], courtesy of the ECC.

Instances of the type *sai-ph* are subject to the following constraint:

(36) *sai-ph* ⇒



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

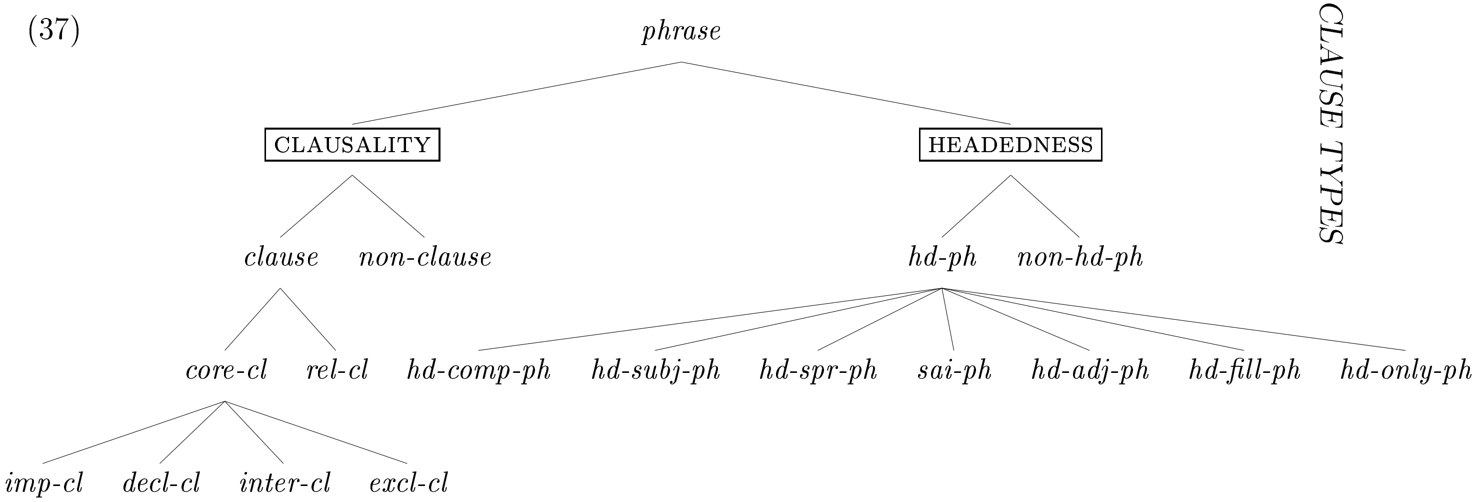
It should be noted that we assume a finer grained analysis for the type *hd-subj-ph* (and possibly for other types of phrase). Declarative instances of this type can be distinguished from those head-subject phrases that involve accusative subjects (so-called 'Mad Magazine' sentences, e.g. *What,* [*me 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 a treatment of these constructions here.

Finally, we have not said anything about the principles of linear ordering that guarantee the proper sequencing of PHON values in these signs; nor will we. Though we will continue to represent constructions in terms of ordered trees, the reader should bear in mind that the linear order of consituents in our theory is in fact determined by linearization constraints of considerable generality. For relevant further discussion, see Pollard and Sag 1987 (Chap. 7), Reape 1994, and Kathol 1995.

2.6 Clause Types

To express generalizations about the shared properties of diverse types of phrases, we propose (following Sag (1997)) to classify phrases not only in terms of their '$\bar{\mathrm{X}}$' 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 informational 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.

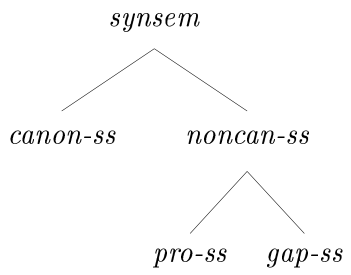
(37)


phrase
CLAUSALITY
HEADEDNESS
clause
non-clause
hd-ph
non-hd-ph
core-cl
rel-cl
hd-comp-ph
hd-subj-ph
hd-spr-ph
sai-ph
hd-adj-ph
hd-fill-ph
hd-only-ph
imp-cl
decl-cl
inter-cl
excl-cl

This classification allows us not only to recognize a distinction between clauses and non-clauses, but also to identify at least the following subtypes of the type *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)*.[20] This analysis lets us express generalizations about phrases with the same simplicity and rigor that has been customary in research on hierarchical lexicons. With the phrasal multiple inheritance hierarchy, we will also have no need for 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 first clarify our assumptions about the subtypes of *synsem*. Both the problem of unexpressed controlled subjects and that of filler-gap dependencies lead us to distinguish among various subtypes of the type *synsem*. We will follow familiar terminology and refer to the type of unexpressed controlled subjects (of nonfinite phrases) as *pro-synsem* (*pro-ss*)[21] and to the type of the 'gap' argument in an extraction construction as *gap-synsem* (*gap-ss*). For languages that allow missing arguments with distinct properties, further subtypes may be necessary, but 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 hierarchical organization of *synsem* types shown in (38). (Here *canon-ss* abbreviates *canonical-synsem* and *noncan-ss* abbreviates *noncanonical-synsem*.)

(38)



All signs (both words and phrases) are subject to the following principle, already built into the type declaration for *sign* illustrated in (29) above:

(39) Principle of Canonicality:

$$\mathit{sign} \;\Rightarrow\; \Big[\text{SYNSEM} \quad \mathit{canon\text{-}ss}\Big]$$

[20] The notion of 'core' clause assumed here is somewhat artificial, as we have not attempted to fit our account of interrogative, declarative, imperative and relative clauses into a broader scheme that would include other clausal types in English, e.g. purpose, rationale, absolute, gerund, or conditional clauses. The analyses we present here are consistent with a wide range of conceptions of how clausal types are organized.

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

The Principle of Canonicality ensures that every overt linguistic expression has a SYNSEM value of type *canon-ss*. The interaction of canonicality with the constraints we have already illustrated for head-complement, head-subject, and head-specifier phrases further guarantees that whenever some head selects for a *noncan-ss* argument, that argument can never be overtly realized in the local syntactic context.

One of the reasons for having clauses in our system of constructions is to have a way of building phrases whose content is some kind of message, i.e. phrases that are in some sense communicatively complete. The following constraint ensures that all clauses have as their CONTENT value a feature structure of type *proposition*, *question*, *outcome*, *fact*, or *directive*, i.e. some subtye of the type *message*.

(40) $$\textit{clause} \Rightarrow \left[\textsc{ss}|\textsc{loc}|\textsc{cont}\ \textit{message}\right]$$

This constraint ensures, for example, that no clause can have a *soa* as its content.

The intuition our analysis encodes is that verbs, which have *soa* content lexically, cannot convey a message; nor can the VPs they give rise to. Hence neither verbs nor the VPs they project can function as stand-alone utterances; nor can they serve as complements of verbs like *believe* or *think*, which select propositional arguments. It is only by embedding a VP within a clausal construction that we build a phrase whose content is a *message*, making that phrase selectable as a complement or utterable on its own (if the clause is also [IC +]). The clausal constructions thus serve to ground the message recursion, as there are other kinds of phrase (e.g. coordinate phrases) that can be used to build more complex messages from those that are clausally constructed.

Let us now turn to the constraints governing the various subtypes of *clause*. Core clauses must not be modifiers, as is guaranteed by the following constraint:

(41) $$\textit{core-cl} \Rightarrow \left[\textsc{head}\ \begin{bmatrix}\textit{verb} & \\ \textsc{vform} & \textit{clausal} \\ \textsc{mod} & \textit{none}\end{bmatrix}\right]$$

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

Of those clause types considered here, relative clauses are the only ones that may be used as modifiers. The constraint in (42) allows for this possibility.

(42)
$$\underset{}{\textit{rel-cl}} \;\Rightarrow\; \underset{\triangle}{\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{IC} & - \\ \text{CINV} & - \\ \text{MOD} & \begin{bmatrix} \text{HEAD} & \textit{noun} \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \textit{proposition} \end{bmatrix}}$$

Relative clauses have a *proposition* as 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. Our semantic analysis of relatives is thus more akin to a propositional function than a property.

The feature IC (INDEPENDENT-CLAUSE — see section 2.7 below) is a variant of Uszkoreit's (1987) MAIN-CLAUSE feature; the [IC $-$] constraint ensures that relative clauses cannot serve as independent clauses, and hence have no status as (non-elliptical) independent utterances. The specification [CINV $-$] in (42) ensures 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 [CINV $+$], can never head a relative clause:

(43) a. *The person [(that/who) are they visiting] is Sandy.

b. *The person [(that/who) I aren't visiting] is Sandy.

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

(44)
$$\textit{decl-cl} \;\Rightarrow\; \begin{array}{c} \begin{bmatrix} \text{CONT} & \begin{bmatrix} \textit{propositional} \\ \text{SOA} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ | \\ \text{H} \\ \ldots \begin{bmatrix} \text{CONT} & \boxed{1} \end{bmatrix} \ldots \end{array}$$

(45)
$$\textit{inter-cl} \;\Rightarrow\; \underset{\triangle}{\begin{bmatrix} \text{CONT} & \textit{question} \end{bmatrix}}$$

(46) $$\textit{imp-cl} \;\Rightarrow\; \left[\textsc{cont}\quad \textit{directive}\right]$$

(47) $$\textit{excl-cl} \;\Rightarrow\; \left[\textsc{cont}\quad \textit{fact}\right]$$

The effect of these constraints is to establish a correlation, plausibly universal in nature, between clausal construction types and types of meaning. The semantic type *propositional* mentioned in (44) has three subtypes: *proposition*, *fact*, and *outcome*. Finite declarative clauses, as explained in the next chapter, are systematically ambiguous between propositions and facts, while a subjunctive clause (analyzed here as a kind of *decl-cl*) is treated semantically as an outcome. Note that in all English declarative clauses, the head daughter's content (a *soa*) is embedded as the SOA value of the clause's propositional content. The treatment of exclamative clauses in terms of facts is also explained in the next Chapter.

Before proceeding, we must digress briefly into the grammar of non-clauses. We assume a number of subtypes of *non-clause*, including *coordinate-phrase* and certain phrasal types that will also be subtypes of *hd-comp-ph*, *hd-spr-ph*, or *hd-only-ph*. Our treatment of auxiliaries, following Sag (1999), includes a construction we call *verb-phrase* (*vb-ph*). This type of phrase, also a kind of *hd-comp-ph*, is subject to the following constraint.

(48) $$\textit{vb-ph} \;\Rightarrow\; \left[\textsc{head}\quad \begin{bmatrix}\textsc{aux} & \boxed{1}\\ \textsc{neg} & \boxed{1}\end{bmatrix}\right]$$

The identification of AUX and NEG values here may seem puzzling at first glance. However, contracted auxiliaries are [NEG +], as are finite auxiliaries that select for *not*. All other verb forms are lexically specified as [NEG −]. Given this, and the fact that only auxiliary verbs are lexically unspecified for the feature AUX, it follows that an instance of the *vb-ph* construction will be [AUX +] just in case it is expresses negation and is headed by an auxiliary verb, as in (49).

(49) a. Kim won't go to the store.

b. Kim has not gone to the store.

When no negation is present, the *vb-ph* will be [AUX −], even if it is headed by an auxiliary verb as in (50b).

(50) a. Kim went to the store.

b. Kim has gone to the store.

This realignment of positive AUX specifications with the auxiliary constructions provides the key to analyzing the distribution of *do*. The unfocussed *do* has an exceptional status — it is the only auxiliary verb lexically specified as [AUX +]. Given what we have said about VPs so far, this will predict that unfocussed *do* can head a *vb-ph* construction only when it is negated, accounting for familiar contrasts like the following:

(51) a. Kim didn't go to the store.

b. Kim did not go to the store.

c. *Kim dĭd go to the store.

This is not the whole story of *do*, of course. First, focussed *do*, like other auxiliary verbs, is lexically unspecified for AUX and thus has no relevant distributional restrictions:

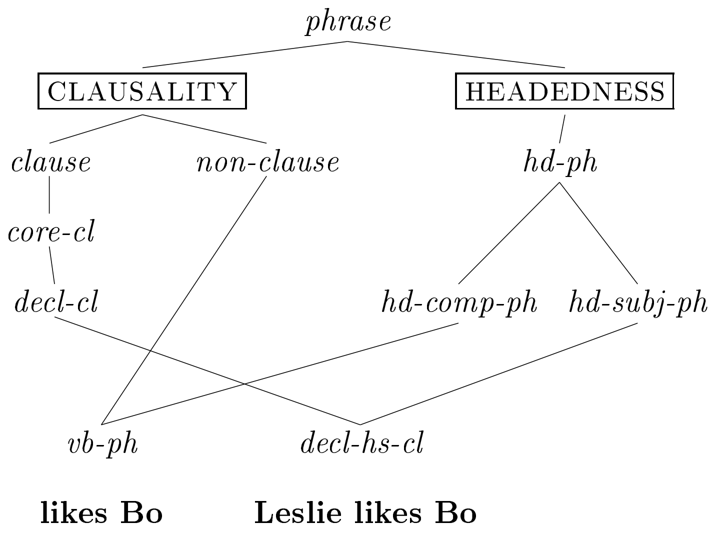
(52) Kim DID go to the store.

Second, following Sag (1999), we treat VP Ellipsis via a derivational rule that both reduces the ARG-ST list of auxiliary verbs and renders them unspecified for the feature AUX. Hence, once unfocussed *do* undergoes this rule, it is also compatible with the [AUX −] specification required of nonnegative instances of the *vb-ph* construction. Unnegated, unfocussed *do* is thus predicted to occur in elliptical examples like (53).

(53) Kim *dĭd* __ .

We are now ready to analyze more fully our previous example *Leslie likes Bo*, which requires two phrasal types. The more embedded phrase *likes Bo* is an instance of the *vb-ph* type just illustrated. To combine this VP with the subject NP *Leslie* we need a new type of phrase — we will call it *declarative-head-subject-cl* (*decl-hs-cl*). This is a subtype of both *decl-cl* and *hd-subj-ph*. The relevant piece of the hierarchy of phrasal types is sketched in (54).

(54)



Simply declaring the existence of the type *decl-hs-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 (55).

(55)

$$\left[\begin{array}{ll}
\textit{decl-hs-cl} & \\
\text{HEAD} & \boxed{3}\left[\begin{array}{ll}\textit{verb} & \\ \text{MOD} & \textit{none} \\ \text{VFORM} & \textit{clausal}\end{array}\right] \\
\text{COMPS} & \boxed{a}\langle\ \rangle \\
\text{SPR} & \boxed{b}\langle\ \rangle \\
\text{SUBJ} & \langle\ \rangle \\
\text{CONTENT} & \left[\begin{array}{l}\textit{propositional} \\ \text{SOA}\ \boxed{2}\end{array}\right]
\end{array}\right]$$

$$\left[\text{SS}\ \boxed{1}\right] \qquad \begin{array}{c}\text{H} \\ \left[\begin{array}{ll}\textit{phrase} & \\ \text{HEAD} & \boxed{3} \\ \text{COMPS} & \boxed{a} \\ \text{SPR} & \boxed{b} \\ \text{SUBJ} & \langle\boxed{1}\rangle \\ \text{CONT} & \boxed{2}\end{array}\right]\end{array}$$

This is just the unification of the constraints associated with the supertypes of *decl-hs-cl*. (55) must be supplemented by the following constraint, particular to *decl-hs-cl*:

(56)

$$\textit{decl-hs-cl} \Rightarrow \left[\text{HEAD}\ \left[\begin{array}{ll}\text{VFORM} & \textit{fin} \\ \text{CINV} & -\end{array}\right]\right]$$

In consequence of (56), all instances of *decl-hs-cl* are headed by a VP whose lexical head is (via the GHFP) a noninverted, finite verb. Note in particular that infinitival clauses like (57) are systematically prohibited:

(57) *[Sandy [to go to the store]]

Finite verbs have either *r-soa*s (indicatives) or *i-soa*s (subjunctives) as their content. The type *propositional* has the three subtypes *proposition*, *fact*, and *outcome*. The first two of these are constructed from (have as their SOA value) an *r-soa*; the last only from an *i-soa*. Hence it follows that instances of the *decl-hs-cl* construction that are headed by an indicative verb form are ambiguous — they denote *proposition*s or *fact*s. When such a clause is headed by a subjunctive verb, however, the only possible content is of type *outcome*.

This brief presentation has provided a glimpse of how our constructional analysis will assign meanings to phrases. Among the further details yet to be presented is our account of quantification, a matter which we take up in section **??** of Chapter 5.

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 done by positing a distinguished type — *root* — whose constraints must be satisfied by any 'stand-alone' utterance. Ellipsis presents a special case, of course,[22] and we will not attempt to extend our account to deal with elliptical utterances. The type *root* will be constrained so as to predict which nonelliptical clauses 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 ellipsis); 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 do not equate the notions of 'independent' clause and 'main' clause. Simplifying considerably what is a complex issue well beyond the scope of this monograph (for discussion, see Green 1996), a nonelliptical clause can appear independently only if it is [IC +], but 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 constraint on the type *root*, which we may formulate as in (58).

$$(58)\quad \textit{root} \Rightarrow \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{IC} & + \\ \text{VFORM} & \textit{fin} \end{bmatrix} \\ \text{CONT} & \textit{message} \\ \text{STORE} & \{\ \} \\ \text{SLASH} & \{\ \} \\ \text{WH} & \textit{none} \end{bmatrix}$$

(58) says that a root phrase must be verbal (e.g. not a CP), must be a finite, independent clause, and must express some kind of message. This will allow phrases whose content is of type *proposition*, *fact*, *question*, *outcome*, or *directive*, but will disallow those whose content

[22]One that might be dealt with in context-free grammar simply by adding other possible start symbols.

is a *soa* as (nonelliptical) stand-alone utterances. (58) requires in addition that the values for STORE, SLASH and WH (all explained in Chapter 5) must be appropriately empty.[23]

How then do embedded clauses acquire the specification [IC $-$]? The answer to this could be partly lexical. We might 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 would be constrained so as to require that such arguments be [IC $-$] as well. Alternatively, we might 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 they deserve a uniform treatment. Whichever analysis of embedded clauses is adopted, sentential complements of all kinds[24] will be specified as [IC $-$].

2.8 Complementizers and *To*

As noted above, we assume that the part of speech types associated with verbs and complementizers are related as two subtypes of the common supertype *verbal*, for which the features VFORM and IC are all appropriate.

Given these assumptions, we may formulate the lexical entry for the complementizer *that* as shown in (59):

(59) a.

$$\left[\begin{array}{ll} \textsc{phon} & \langle \text{ that } \rangle \\ \textsc{ss}|\textsc{loc} & \left[\begin{array}{ll} \textsc{cat} & \left[\begin{array}{ll} \textsc{head} & \left[\begin{array}{ll} \textit{comp} & \\ \textsc{ic} & - \\ \textsc{vform} & \textit{fin} \end{array}\right] \\ \textsc{subj} & \langle\ \rangle \\ \textsc{arg-st} & \left\langle \begin{array}{c} \text{S} \\ \left[\begin{array}{ll} \textsc{cinv} & - \\ \textsc{vform} & \textit{fin} \\ \textsc{subj} & \langle\ \rangle \\ \textsc{cont} & \boxed{1} \end{array}\right] \end{array} \right\rangle \end{array}\right] \\ \textsc{cont} & \boxed{1}\textit{propositional} \end{array}\right] \end{array}\right]$$

Note that the semantic type of *that* (unlike that of the verbs illustrated above) is *propositional*, as determined by the sentential phrase that serves as its complement. It should also

[23]By requiring root phrases to be verb projections, we disallow CPs as stand-alone expressions. This could be too strong, in light of examples like (i), pointed out to us by Carl Pollard.

(i) That it should come to this!

If utterances like (i) are to be analayzed as root clauses, rather than as some kind of ellipsis, then (58) is in need of minor revision, as is the analysis of complementizers presented in section 2.8 below.

[24]Except perhaps complements of direct quotation, depending on how these are analyzed.

be noted that *that*, like all complementizers in English, is [IC −], restricting its distribution to embedded clauses. Finally, 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 *propositional*, we allow both indicative and subjunctive complements. These can either be simple declarative clauses (e.g. instances of the type *decl-hs-cl*), topicalized clauses (in certain restricted circumstances discussed below), or coordinations of such.[25]

The lexical entry for the complementizer *for* is similar to the one for *that*.[26] We assume that *for* differs from *that* in taking two arguments, identifying the LOCAL value of unexpressed subject of the second argument with that of the first:

(60)
$$\begin{bmatrix} \text{PHON} & \langle \text{for} \rangle \\ \text{SS}|\text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{comp} & \\ \text{IC} & - \\ \text{VFORM} & \textit{inf} \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{ARG-ST} & \left\langle \begin{bmatrix}\textit{canon-ss} & \\ \text{LOC} & \boxed{2}\end{bmatrix}, \overset{\text{VP}}{\begin{bmatrix} \text{VFORM} & \textit{inf} \\ \text{SUBJ} & \left\langle\begin{bmatrix}\text{LOC} & \boxed{2}\end{bmatrix}\right\rangle \\ \text{CONT} & \boxed{1}\end{bmatrix}} \right\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix}\textit{outcome} & \\ \text{SOA} & \boxed{1}\end{bmatrix} \end{bmatrix} \end{bmatrix}$$

As noted by Sag (1997), by assuming that *for-to* clauses, unlike *that*-clauses, project a flat (ternary) structure, we obtain an immediate account for the contrasts in (61), noted by Emonds (1976: 196).

(61) 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.

Adverbials that should be able to introduce a sentential constituent cannot introduce the putative sentential constituent that follows *for*. On our analysis, there is a clause for the

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

[26] It no doubt makes sense to organize the complementizers into 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.

adverbial to modify only in the case of *that*-clauses like (61a,b), not in the case of *for/to*-clauses like (61c,d). These last two cases are ill-formed because two complements appear right-shifted over an adverb, as in, say, **persuaded in St. Louis John to rent a house cheap.*

Note 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 prevents the object of *for* from being extracted:

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

Second, the semantics of *for* is an outcome formed from the *soa* of the infinitival VP complement. Like the propositions that subjunctive clauses denote, the meanings of *for-to* phrases are irrealis in nature.

Since these complementizers have a lexically assigned content of type *propositional* (*that*) or *outcome* (*for*),[27] in order to provide *that*-clauses and *for-to*-clauses with the right semantics, we must 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:

(63) $\textit{cp-cl} \Rightarrow \begin{bmatrix}\text{HEAD} & \textit{comp}\end{bmatrix}$

Nothing more needs to be said about this clause type except that it is a subtype of both *clause* 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.

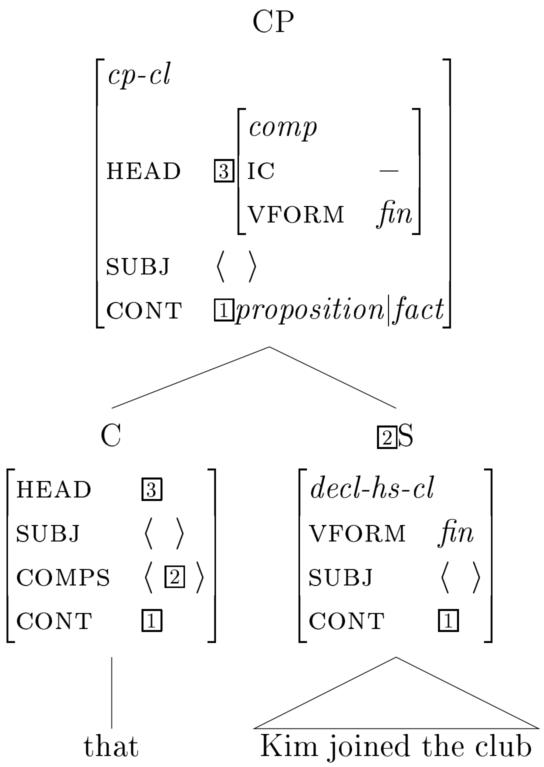
This clause type accounts for clauses like (64a,b).

[27] Or *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*.

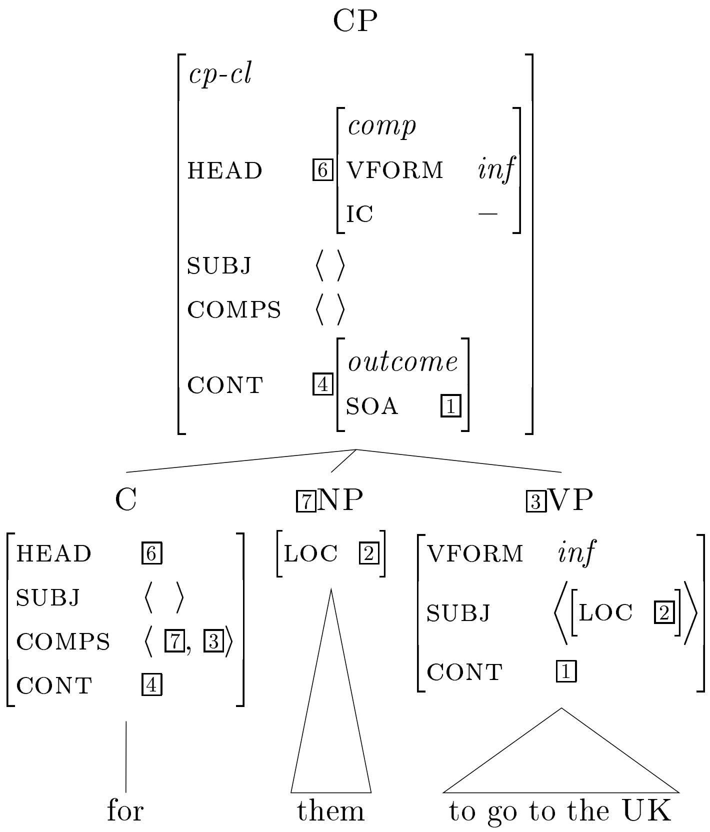
(i) Comme il fait beau!
How it makes beautiful
'How beautiful it is!'

For discussion, see Desmets in preparation.

(64) a.



b.



In (64a), the CONTENT value can be either a *proposition* or a *fact* — the two subtypes of *propositional* that allow a *soa* of type *r-soa*, which is the type assigned lexically for all indicative verb forms, e.g. *joined*. The CONTENT value of *to* is lexically underspecified (as being of type *soa*), but in contexts like (64b), it is resolved to type *i-soa*. Elsewhere, as detailed below, a *to*-phrase can denote a *proposition* or a *fact*, both of which require that *to*'s lexical content be resolved as *r-soa*.

As a final observation, note that the complementizer *that* is specified as [IC −], yet the IC value of its complement is left unspecified. This has the effect of allowing 'main clause phenomena'[28] in certain embedded ([IC −]) environments, but only if the complementizer *that* is present. This appears to be a correct prediction:

(65) 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].

(66) a. They believed (that) they were oppressed.

b. They believed *(that) [never again would they have to do housework].

(67) 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].

We make no attempt here to explain when independent clauses can appear in embedded environments.[29] 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. However, our (admittedly partial) account of the matter is strikingly simple.

To-phrases are more complex. Let us begin with the lexical entry for *to*, shown in (68).

(68)
$$\begin{bmatrix} \text{PHON} & \langle \text{ to } \rangle \\ \text{SS}|\text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{inf} \end{bmatrix} \\ \text{SUBJ} & \left\langle \boxed{7}\begin{bmatrix}\text{LOC} & \boxed{2} \\ \text{SLASH} & \{\ \}\end{bmatrix} \right\rangle \\ \text{ARG-ST} & \left\langle \boxed{7}\ , \overset{\text{VP}}{\begin{bmatrix} \text{VFORM} & \textit{base} \\ \text{SUBJ} & \left\langle \begin{bmatrix}\text{LOC} & \boxed{2}\end{bmatrix} \right\rangle \\ \text{CONT} & \boxed{1}\end{bmatrix}} \right\rangle \end{bmatrix} \\ \text{CONT} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

[28]We mean to include here most of the phenomena discussed by Emonds (1976) under the rubric of 'root transformations.'

[29]For some discussion, see Hooper and Thomspon 1973, Green 1976, and Green 1996.

As shown by Pullum (1982), the word *to* is profitably analyzed as a defective (i.e. paradigm-less), 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 (69).

(69) 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:[30]

(70) a. *Pat preferred for Sandy to get the job, and we preferred for __ , too.

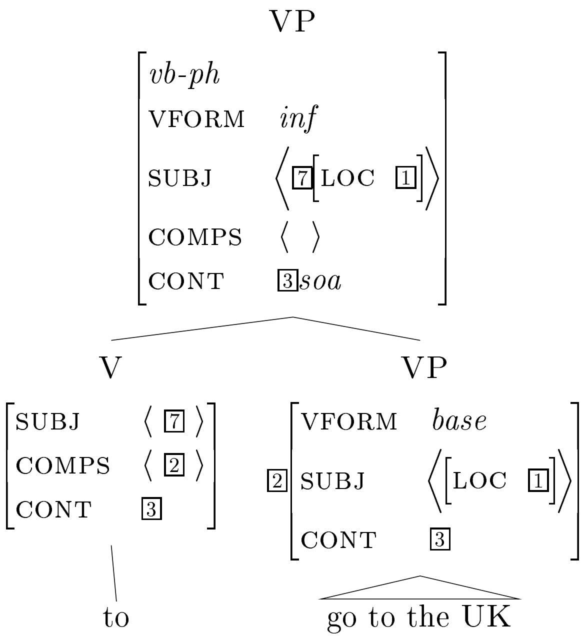
b. *Kim ordered us to leave, and Sandy ordered us __ .

c. *They ordered us to leave, and we want __ .

The lexical entry in (68) 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*] and hence (since *inf* is a subtype of *clausal*) *to* may project 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 *vb-ph*, as shown in (71).

(71)



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

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

Note that the content of this phrase, like that of *to* and *to*'s *base*-form complement, is specified as a *soa*. Hence a phrase like (71) may be resolved as either an *r-soa* or an *i-soa*, in the appropriate context. *Soa*-denoting phrases like (71) serve as the complement of a raising verb (or as complement of the complementizer *for*).

However in other contexts, *to*-phrases convey more than just a *soa*. As a controlled complement,[31] a *to*-phrase denotes an *outcome* and many uses of *to* phrases with a so-called 'arbitrary' interpretation of the unexpressed subject appear to denote facts. Thus we claim that both kinds of meaning are possible for *to*-clauses:

(72) a. Lee wants *to be happy.* (*outcome*)

b. Pat was mortified. *To be named salutatorian* was embarrassing enough. (*fact*)

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

(73) a. A: What do you want?
B: *To go home.* (*outcome*)

b. A: What did you find mortifying?
B: *To be named salutatorian.* (*fact*)

The auxiliary *to* can give rise to such clauses in virtue of a further clause type: *declarative-nonsubject-clause* (*decl-ns-cl*). Instances of this type — a subtype of *hd-only-ph* and *decl-cl* — are constrained as follows:

(74) $$\textit{decl-ns-cl} \;\Rightarrow\; \left[\textsc{subj}\quad \langle\,[\;]\,\rangle\right]$$

This allows for *to*-clauses like (75) to be built from *soa*-denoting *to*-phrases like (71), once the effects of the GHFP and the constraint on the type *decl-cl* are taken into account:

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

(75)

$$
\text{VP}\\
\begin{bmatrix}
\textit{decl-ns-cl} & \\
\text{HEAD} & \boxed{0}\begin{bmatrix}\text{IC} & - \\ \text{CINV} & - \\ \text{VFORM} & \textit{inf}\end{bmatrix} \\
\text{SUBJ} & \langle \boxed{1} \rangle \\
\text{COMPS} & \langle\ \rangle \\
\text{CONT} & \begin{bmatrix}\textit{propositional} & \\ \text{SOA} & \boxed{4}\end{bmatrix}
\end{bmatrix}
$$

$$
\text{VP}\\
\begin{bmatrix}
\textit{vb-ph} & \\
\text{HEAD} & \boxed{0} \\
\text{SUBJ} & \langle \boxed{1} \rangle \\
\text{COMPS} & \langle\ \rangle \\
\text{CONT} & \boxed{4}
\end{bmatrix}
$$

to go to the UK

Recall that there is a constraint on type *decl-cl* (see (44) above) requiring that declarative clauses have *propositional* content. Thus, by saying nothing about CONTENT in (74), instances of this clause type may have any subtype of *propositional*: *proposition*, *fact*, or *outcome* as their content. As we will see in Chapter 6, it is this potential for *to*-phrases to denote *proposition*s, as well as *fact*s or *outcome*s, that allows them to appear in *wh*-interrogative constructions (e.g. *who to visit*).

Note in addition that 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 *pfp*], both of which are incompatible[32] with the [VFORM *clausal*] constraint on this kind of phrase that is inherited from its supertype *core-cl* (see (41) 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 some subtype of *propositional*.

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:

(76) a. I believe there [to be a problem here]/*[to like Sandy].

[32]See the hierarchy of *vform* values in (8) above.

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 *to*-phrases appear in non-raising contexts, e.g. those in (77), the unexpressed subject must be referential.

(77) 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:

(78) 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 the unexpressed subject 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:

(79) 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]])

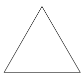
c. *Bother you(rself) that Kim left!

d. *[Raining] would bother them.

In our analysis, these facts are all accounted for by the following constraint, applicable to all clauses:

(80) Clause Constraint:

$$\textit{clause} \Rightarrow \begin{bmatrix}\textrm{SUBJ} & \textrm{list}(\textit{noncan-ss})\end{bmatrix}$$



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 (80) is to guarantee that a clause's SUBJ value is one of the following:

(81) 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 a subtype of any list type.

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

$$(82)\quad \textit{pro-ss} \;\Rightarrow\; \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{acc} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \textit{reflexive} & \\ \text{INDEX} & \textit{ref} \end{bmatrix} \end{bmatrix}$$

Since a *pro-ss* must have a referential index, it is inconsistent with the lexical specifications 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*) — 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 exactly as discussed in P&S-94, chaps. 6–7 to explain 'Visser's Generalization' (Bresnan 1982) in its full generality.[33]

The Clause Constraint plays an important role in the theory presented in this monograph, as will be explained in detail in Chapter 6. It interacts with 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, (2) have an unexpressed 'PRO' subject, or else (3) have a subject that is extracted, corresponding to the three options shown in (81). The Clause Constraint will in fact guarantee that the only option for *to*-clauses is to have a singleton SUBJ list containing a *pro-ss*. This follows because (1) the lexical entry for *to* ((68)) bars 'slashed' subjects,[34] while *gap-ss* elements are always slashed (see Chapter 5) and (2) the type *decl-hs-cl* discussed earlier allows only finite instances — thus there is no way to build up a clause like *Sandy to leave*.

[33] And, 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 (81), the Clause Constraint, and the lexical constraint requiring that finite verbs have nominative subjects.

[34] This 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.

Note that since there is no way to construct a phrase like *Kim 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 infinitival phrases as a natural class: [VFORM *inf*]. This is required for a number of constructions, e.g. purpose clauses and infinitival relatives:[35]

(83) 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].

(84) 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.

Of course we can still distinguish between *to*-phrases and *for*-phrases via the SUBJ value of a [VFORM *inf*] phrase or else by specifying the HEAD value as *verb* vs. *comp*. This enables us to write a lexical entry for a verb like *try*, which bars *for-to*-clauses as complement:

(85) 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 (71) (*to go to the* UK) can serve as the complement of most raising verbs. For example, the lexeme *believe* will include the information in (86), some of which is derived via constraint inheritance.

[35]See Green 1991 and Johnston in press.

(86)
$$
\begin{bmatrix} \textit{orv-lxm} & \\ \text{PHON} & \langle \text{believe} \rangle \\ \text{SS}|\text{LOC} & \begin{bmatrix} \text{ARG-ST} & \left\langle \text{NP}_i\ ,\ \begin{bmatrix}\text{LOC} & \boxed{2}\end{bmatrix},\ \overset{\text{VP}}{\begin{bmatrix}\text{VFORM} & \textit{inf}\\ \text{SUBJ} & \left\langle\begin{bmatrix}\text{LOC} & \boxed{2}\end{bmatrix}\right\rangle \\ \text{CONT} & \boxed{1}\end{bmatrix}} \right\rangle \\ \text{CONT} & \begin{bmatrix} \textit{soa} & \\ \text{NUCL} & \begin{bmatrix}\textit{believe-rel} & \\ \text{EXPERIENCER} & i \\ \text{PROP-ARG} & \begin{bmatrix}\textit{proposition} & \\ \text{SIT} & s \\ \text{SOA} & \boxed{1}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}
$$

Because the *to*-phrase here is not a clause, the Clause Constraint is inapplicable. Therefore the unexpressed subject of the VP[*inf*] argument (and hence the object of *believe*) is free to be of any type. This allows for nonreferential objects in examples like (87).

(87) 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, *soa*-denoting *to*-phrases may serve as complements of the complementizer *for*, as shown in (64b) above. But a *pro-ss* is ruled out in both (64b) and (87), as the object of *believe* or *for* must be an overt element, i.e. a *sign*,[36] and hence [SYNSEM *canon*]. These predictions follow from the interaction of the Principle of Canonicality in (39) and the constraints stated directly on the type *hd-comp-ph* ((32) above).

Next consider a proposition-embedding verb like *think*. Its lexical entry includes the information in (88). (*vfc-lxm* abbreviates *verb-finite-complement-lexeme*.)

[36]Note, however, that extraction of 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, not SYNSEM values. When a there is extraction of a raised object, the object argument of the raising verb will be of type *gap-ss* (see section ?? of Chapter 5) and hence slashed. But this *synsem* is distinct from the subject of the infinitive, which is unslashed.

(88)
$$\left[\begin{array}{ll} \textit{vfc-lxm} & \\ \text{PHON} & \langle \text{ think } \rangle \\ \text{SS}|\text{LOC} & \left[\begin{array}{ll} \text{ARG-ST} & \left\langle \text{NP}_i \, , \left[\begin{array}{ll} \text{VFORM} & \textit{fin} \\ \text{CONT} & \boxed{1} \end{array}\right] \right\rangle \\ \text{CONT} & \left[\begin{array}{ll} \textit{soa} & \\ \text{NUCL} & \left[\begin{array}{ll} \textit{think-rel} & \\ \text{EXPERIENCER} & i \\ \text{PROP-ARG} & \boxed{1}\textit{proposition} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right]$$

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 instances of the type *decl-hs-cl*. The predictions are the following (an asterisk here indicates information of the complement that is incompatible with the properties selected by *think*):

(89) a. I think [that Leslie is winning]. (CP[*fin*]:*proposition*)

b. I think [Leslie is winning]. (S[*fin*]:*proposition*)

c. *I think [Leslie be winning]. (S[*fin*]:**outcome*)

d. *I think [that Leslie be winning]. (CP[*fin*]:**outcome*)

e. *I think [Leslie to be winning]. (no such clause)

f. *I think [to be winning]. (VP[**inf*]:*proposition*)

g. *I think [for her to be winning]. (CP[**inf*]:**outcome*)

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

(90)
$$\left[\begin{array}{ll} \textit{v-lxm} & \\ \text{PHON} & \langle \text{ insist } \rangle \\ \text{SS}|\text{LOC} & \left[\begin{array}{ll} \text{ARG-ST} & \left\langle \text{NP}_i \, , \left[\begin{array}{ll} \text{VFORM} & \textit{fin} \\ \text{CONT} & \boxed{1} \end{array}\right] \right\rangle \\ \text{CONT} & \left[\begin{array}{ll} \textit{soa} & \\ \text{NUCL} & \left[\begin{array}{ll} \textit{insist-rel} & \\ \text{ACTOR} & i \\ \text{OUTCM-ARG} & \boxed{1}\textit{outcome} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right]$$

Our system allows exactly two kinds of clauses as the complement of verbs like *insist*, as shown in (91).

(91) a. I insist [he stay outside]. (S[*fin*]:*outcome*)

b. I insist [that he stay outside]. (CP[*fin*]:*outcome*)

c. *I insist [Leslie to stay outside]. (no such clause)

d. *I insist [to stay outside]. (VP[**inf*]:*outcome*)

e. *I insist [for him to stay outside]. (CP[**inf*]:*outcome*)

f. I insist [that they're incompetent]. (CP[*fin*]:**proposition*|**fact*)

g. I 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 Summary

This chapter has built up an analysis of the basic declarative clause constructions in English. The system of clausal features and types presented in this chapter makes it possible to specify natural classes for purposes of lexical selection. Following the approach articulated by Grimshaw (1978) 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 (92).

(92)