Assumptions (Week 3)

1 The Lexicon

Lexical entries are complex feature structures containing (at least) syntactic, semantic and phonological information about each morpheme in the language. The syntactic features are organized as in (1).

(1) $$\text{LI} = \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{TYPE} & \{\text{N, V, WH, etc.}\} \\ \text{INFL} & [x\text{FORM} \quad val] \end{bmatrix} \\ \text{INFL} & \begin{bmatrix} y\text{FORM} & val \\ z\text{FORM} & \_ \end{bmatrix} \\ \text{SEL} & \begin{bmatrix} \text{COMP} & \text{CAT}_1 \\ \text{SPEC} & \text{CAT}_2 \end{bmatrix} \end{bmatrix}$$

The CAT feature complex encodes information about the basic syntactic properties of the object: its basic 'type' (i.e., part of speech), and and the kind of agreement it imposes on other objects. The latter is encoded as an INFL subfeature; the representation in (1) indicates that LI licenses valuation of another object's xFORM feature as *val*. (Here xFORM is just a placeholder for some other feature — VFORM, CASE, whatever.) In writing these out, we omit TYPE and just write in the part of speech information, which may be a set (e.g., {N, WH}).

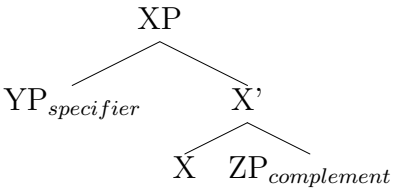
The INFL feature complex controls the (syntactically determined) surface shape of the lexical item. This information may be explicitly specified in a lexical entry, as with the yFORM value in (1), which has the value *val*, or it may be 'unvalued', indicated with an underline, as with the zFORM feature in (1). Unvalued features may become valued through the AGREE relation, defined below. We have left open the question of whether derivations that include unvalued INFL features are ungrammatical (because there are insufficient instructions about the shape that the lexical item should take), or whether they can be filled by some sort of default rule.

The SEL feature complex specifies the selectional requirements of a lexical item. It is further subdivided into a COMPlement feature and a SPECifier feature, each of which have categories (or possibly ordered lists of categories, though I have not shown this in (1)) as their values. The difference between specifiers and complements concerns order and position within the phrasal projection of a head. Specifically:

1. Heads first combine with complements, then with specifiers. That is, all COMP features must be discharged first.
2. Languages may specify the relative ordering of head, complement, and specifier.

In English, complements are ordered to the left of the head, and specifiers are ordered to the right of the head (and any complements), giving us structures like (2).

(2)



Note that nothing rules out the possibility that different heads impose different ordering requirements on the things they select — e.g., that a head X wants its complements on the right, and a head Y wants them on the left. If we discover that this is the case, we will need to figure out some way of encoding this in the lexical entries of the relevant heads.

2 The generative component

Our generative component contains three (or maybe two) rules: AGREE, MERGE and MOVE. MOVE is defined in terms of MERGE, so is arguably not a separate principle, but to keep things as clear as possible I will define it separately here. The details of the rules as they currently stand are as stated in (3)-(5). (Note that the definition of MERGE is worded a little bit differently than the one we settled on in class, but this is only because I spelled out the definition of 'select'; the overall requirements of the rule are identical.)

(3) AGREE
If syntactic object X has an unvalued INFL feature F_1 and syntactic object Y has a matching valued CAT feature F_2, let the value of F_1 = the value of F_2.

(4) MERGE
If X is a syntactic object with undischarged SEL feature α and Y is a syntactic object with CAT feature α:
i. Form Z such that Z immediately dominates X and Y.
ii. Discharge (delete) α on X.
iii. Let the features of Z = the features of X.

(5) MOVE
If X is a syntactic object with undischarged SEL feature α and Y is a syntactic object with CAT feature α and X contains Y:
i. Form Z such that Z immediately dominates X and a copy of Y.
ii. Discharge (delete) α on X.
iii. Let the features of Z = the features of X.
iv. Delete the original occurrence of Y.

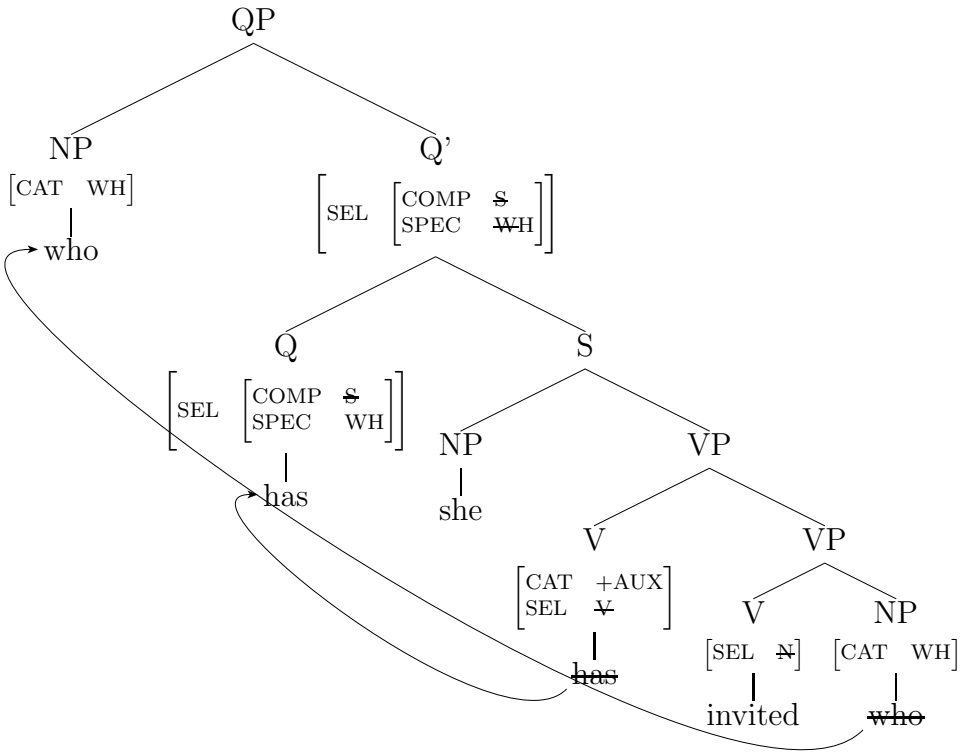
The two significant differences between MERGE and MOVE is that the latter states explicitly that the selected object Y is part of the structure that is already built (this is the 'contained within X' clause), that what gets merged to X is a copy of Y, and that the original gets deleted. It may very well be that none of these things need to be stipulated at the end of the day — that MOVE really can be viewed as a special case of MERGE where the selected object comes from structure that is

already built ('internal merge') rather than from the lexicon ('external merge'), and that the deletion requirement follows from general principles governing the mapping from syntax to phonology. For the moment, though, we can treat them as two separate (though related) principles, just to keep things as clear and explicit as possible.

Finally, MOVE as currently stated is triggered by selection. While this looks like a good way of handling phrasal movements like *wh*-movement, it is not clear that it will work for head movements, like auxiliary fronting in yes-no (and *wh*-) questions. For now you can simply assume that there is some principle that licenses movements from one head position to another, so what is going on in an example like (6a) are two instances of movement: phrasal movement of *who* to the specifier position of Q, and head movement of *has* to Q, as shown in (6b).

(6) a. Who has she invited.

b.



Note that this tree leaves out INFL information, and CAT information is conveyed by node labels plus features if necessary. For example, *invited* is CAT V, as indicated by its node label, but *who* has two category features: N and WH. The former is represented by the node label; the later is listed in the feature structure.