Grammar 20

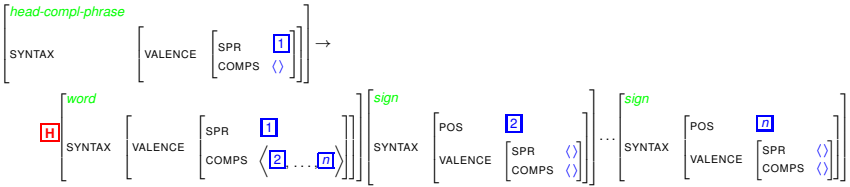
Gert Webelhuth

University of Frankfurt

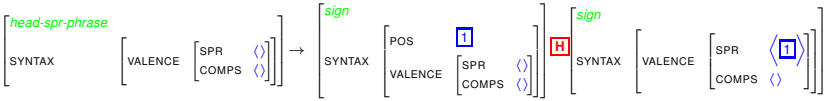
The grammar so far

There are only two grammar rules:

Head-Complement Rule:



Head-Specifier Rule:



The Head Feature Principle (HFP)

The value of HEAD of a phrase is also the value of HEAD of the phrase's head-daughter.

A typical lexical entry

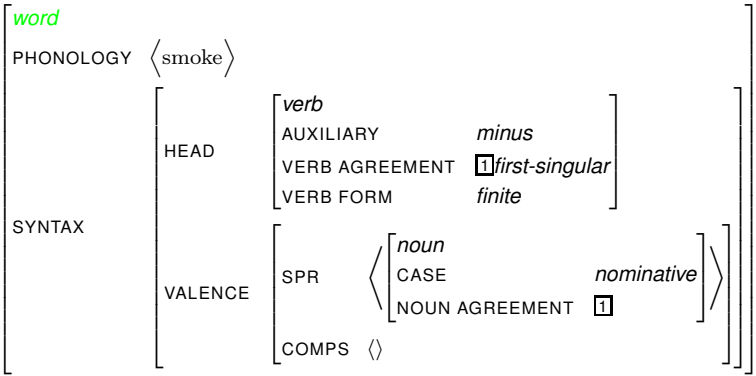


Illustration: the Head-Complement Rule

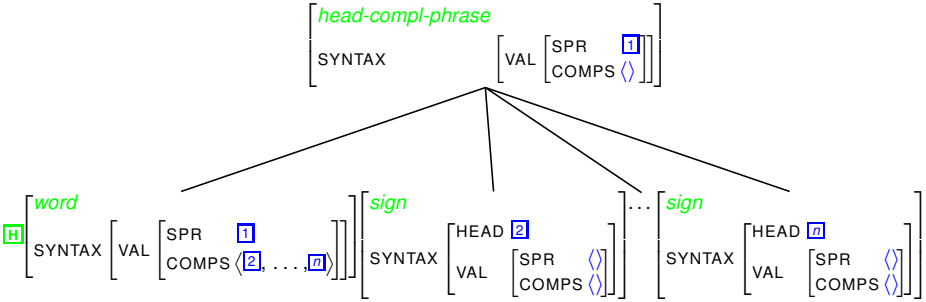
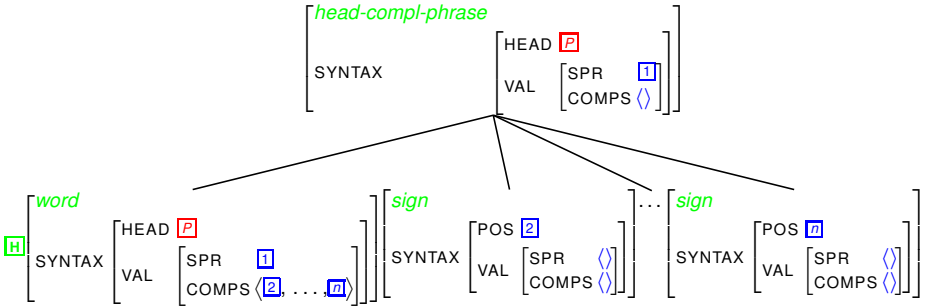


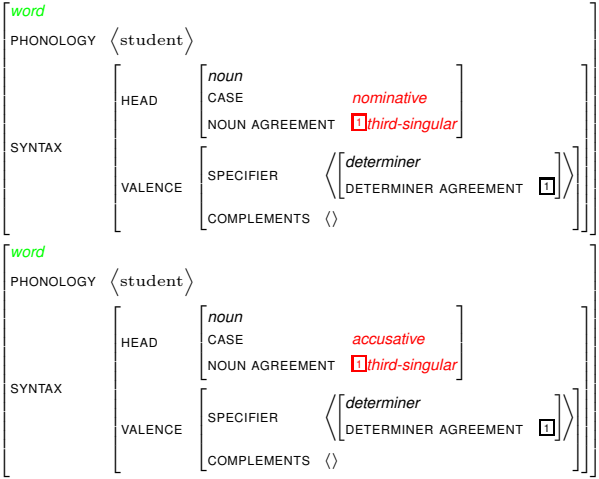
Illustration: the Head-Complement Rule + HFP



- The information in blue and green is a consequence of the Head-Complement Rule.
- The information in red is a consequence of the Head Feature Principle.

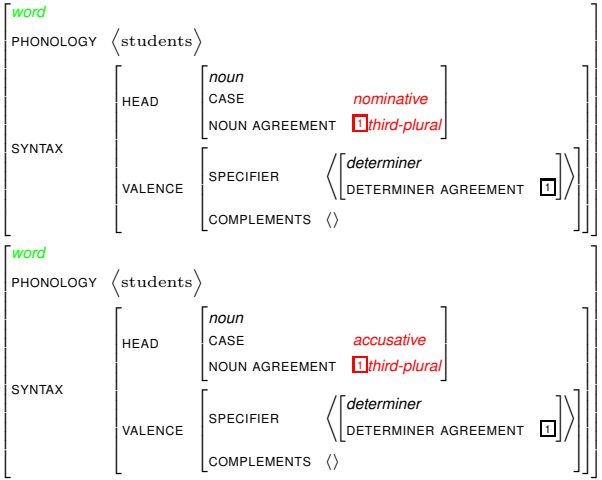
Our grammar contains 4 lexical entries for most common nouns

Example: *student* and *students*



Our grammar contains 4 lexical entries for most common nouns

Example: *student* and *students*



Dictionaries have only a single lexical entry for most common nouns

- Question: How can dictionaries get away with listing each common noun only once?
- Answer: They only list the citation form for the noun and assume that the user can create all the word forms belonging to the word family of the citation form!

From here on, we want to do the same, as it greatly simplifies the grammar if we do not list words whose properties can be derived from citation forms. To that end, we must accomplish two things:

1. We must formulate a citation form for each common noun.
2. We must develop a mechanism that creates all the words that the citation form represents.

Step 1: a citation form for the word family *student*

$$\begin{bmatrix} \textit{noun-lexeme} \\ \text{PHONOLOGY} & \langle \text{student} \rangle \\ \text{SYNTAX} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VALENCE} & \begin{bmatrix} \text{SPECIFIER} & \left\langle \left[\textit{determiner} \right] \right\rangle \\ \text{COMPLEMENTS} & \langle \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Note:

- All citation forms are *lexemes*.
- For each inflectable part of speech, there is an individual lexeme type: *noun-lexeme*, *verb-lexeme*, etc.
- Lexemes are neutral between the various inflected word forms of the word family.
- Hence, noun lexemes do not specify any case or agreement, since different words of the word family can have different values for those features.

Step 2: deriving words from lexemes

Our system of noun lexemes and noun words now is designed as follows:

- For every family of 4 related common noun words like *student* above, the lexicon does not list any of the four words, but only the lexeme (= the citation form).
- We create four lexical rules that each take the noun lexeme as input and create one of the four inflected words of the word family.
- These rules are so general that they not only apply to the noun lexeme *student*, but also to *apple*, *house*, and all other noun lexemes that are the citation form of a four word common noun word family!

Cost-benefit calculation of this design:

1. Cost: 4 rules.
2. Benefit: 3 fewer lexical entries for each common noun family (old system: 4 listed words; new system: 1 listed lexeme).
3. Given that the lexicon contains several 10.000 common noun word families, this is a significant saving!

Our first lexical rule

The nominative singular common noun lexical rule:

$$\left[\textit{noun-lexeme}\right] \mapsto \left[\begin{array}{l}\textit{word} \\ \text{SYNTAX} \left[\text{HEAD} \left[\begin{array}{ll}\text{CASE} & \textit{nominative} \\ \text{NOUN AGREEMENT} & \boxed{1}\textit{third-singular}\end{array}\right]\right]\end{array}\right]$$

Things to note:

1. The arrow in lexical rules is $\mapsto$, which is different from the arrow in phrase structure rules, which is $\rightarrow$.
2. A lexical rule has the following meaning: for every object in the grammar that satisfies the description of the input of the rule, there is a well formed object in the grammar with the following properties:
   1. The new object has all the properties described for the output of the rule **and**
   2. all the properties of the input that do not conflict with the description for the output!

Applying the nominative singular noun lexical rule to the noun lexeme *student*

The input lexeme:

$$\begin{bmatrix} \textit{noun-lexeme} \\ \text{PHONOLOGY} & \langle \text{student} \rangle \\ \text{SYNTAX} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VALENCE} & \begin{bmatrix} \text{SPECIFIER} & \left\langle \left[\textit{determiner} \right] \right\rangle \\ \text{COMPLEMENTS} & \langle\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The nominative singular common noun lexical rule:

$$\left[\textit{noun-lexeme} \right] \mapsto \begin{bmatrix} \textit{word} \\ \text{SYNTAX} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CASE} & \textit{nominative} \\ \text{NOUN AGREEMENT} & \textit{third-singular} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

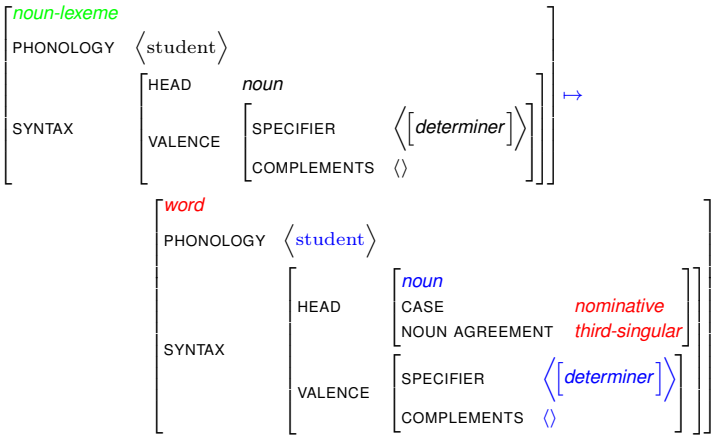
Step 1: inserting the lexeme into the input of the rule

The nominative singular common noun lexical rule with input *student*:

$$\begin{bmatrix} \textit{noun-lexeme} \\ \text{PHONOLOGY} \quad \langle \text{student} \rangle \\ \text{SYNTAX} \quad \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VALENCE} & \begin{bmatrix} \text{SPECIFIER} & \left\langle \left[\textit{determiner} \right] \right\rangle \\ \text{COMPLEMENTS} & \langle \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \mapsto \begin{bmatrix} \textit{word} \\ \text{SYNTAX} \quad \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CASE} & \textit{nominative} \\ \text{NOUN AGREEMENT} & \textit{third-singular} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Step 2: copying all the properties that do not conflict from the input to the output (in blue)

The nominative singular common noun lexical rule with input *student*:

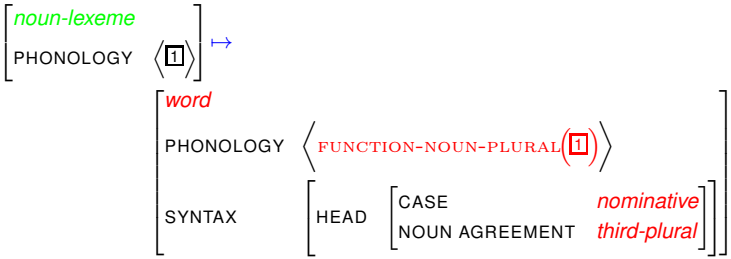


Exercises

1. Write the accusative singular common noun lexical rule.
2. Apply the lexical rule to the lexeme *student*.

The nominative plural common noun lexical rule

The nominative plural common noun lexical rule:



Why do we use a function to compute the phonology of the output?

The nominative plural common noun lexical rule

Question: Why do we use a function to compute the phonology of the output?

Answer: because the plural of nouns is not always the singular + s:

Singular	Plural
ox	oxen
child	children
fish	fish
sheep	sheep
woman	women
dog	dogs
cat	cats

Note:

- The output phonology depends on what the input is **and**
- for every input, there is exactly one output.
- This is exactly the kind of dependence that can be captured with a function.

The nominative plural common noun lexical rule

So we define the function FUNCTION-NOUN-PLURAL as follows:

FUNCTION-NOUN-PLURAL(ox) = oxen
FUNCTION-NOUN-PLURAL(child) = children
FUNCTION-NOUN-PLURAL(fish) = fish
FUNCTION-NOUN-PLURAL(sheep) = sheep
FUNCTION-NOUN-PLURAL(woman) = women
. . .

For every other X:
FUNCTION-NOUN-PLURAL(X) = Xs

Examples:

FUNCTION-NOUN-PLURAL(dog) = dogs
FUNCTION-NOUN-PLURAL(cat) = cats

Exercise

1. Apply the nominative plural common noun lexical rule to the lexeme *student*!

One problem

- The common noun words created by the four lexical rules lack one important property that the original listed words had.
- Can you think what that property is?

One problem

- The common noun words created by the four lexical rules lack one important property that the original listed words had.
- Can you think what that property is?

Answer: agreement between the common noun and its determiner!

Solution

The Noun-determiner Agreement Constraint:

$$\left[\begin{array}{l}\textit{word}\\ \text{SYNTAX}\ \left[\begin{array}{ll}\text{HEAD} & \textit{noun}\\ \text{VALENCE} & \left[\text{SPECIFIER}\ \left\langle\left[\textit{determiner}\right]\right\rangle\right]\end{array}\right]\end{array}\right] \Rightarrow \left[\begin{array}{l}\textit{word}\\ \text{SYNTAX}\ \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{l}\textit{noun}\\ \text{NOUN AGREEMENT}\ \boxed{1}\end{array}\right]\\ \text{VALENCE} & \left[\text{SPECIFIER}\ \left\langle\left[\begin{array}{l}\textit{determiner}\\ \text{DETERMINER-AGREEMENT}\ \boxed{1}\end{array}\right]\right\rangle\right]\end{array}\right]\end{array}\right]$$

- This constraint says: Every noun word with a determiner as specifier agrees with that determiner.
- Note that we have introduced a third arrow, which will only be used in constraints: $\Rightarrow$.