Linguistics 201 – Introduction to Linguistic Theory November 20, 2007
Fall 2007 – Section D

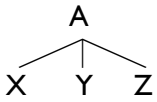
Phrase Structure Grammars

In this section of the class, we'll develop a grammar for an artificial language, *"MiniEnglish"*. We'll describe this grammar with the help of context free rules, or phrase structure rules. Our goal is to make MiniEnglish as similar to English as possible, that is to make our MiniEnglish grammar produce all and only sentences that are grammatical sentences of English.

The rule format we use looks like this:

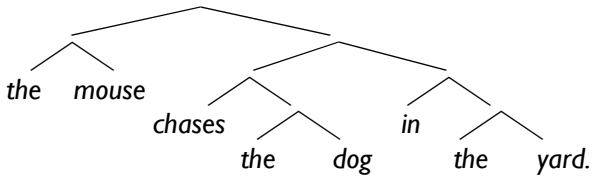
A → X Y Z

This rule tells you that a structure like the one below is valid according to the rule:



We will use the rules to create tree structures that hopefully will match the tree structures that we found with our constituency tests.

For instance, we found the constituency tree below for the sentence "The mouse chases the dog in the yard" with our test.



The following grammar can produce this tree:

Grammar		Lexicon	
i.	S → NP VP	N:	mouse, dog, yard
ii.	VP → (Aux) V'	V:	chase
iii.	V' → V' PP		
iv.	V' → V (NP)	Det:	the
v.	PP → P NP	P:	in
vi.	NP → Det N'		
vii.	N' → N' PP		
viii.	N' → N		

The easiest way to use this grammar is to start to *parse* the sentence bottom up.

Step 1:

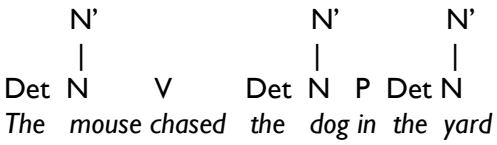
Assign lexical categories to the words

Det N V Det N P Det N
The mouse chased the dog in the yard.

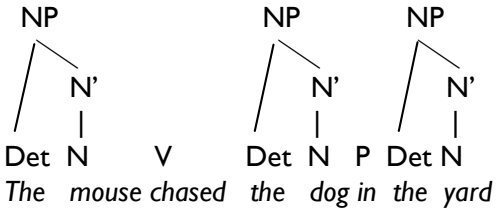
Step 2:

Combine the individual words into small trees according to our rules. Let the phrase structure tree be your guide here:

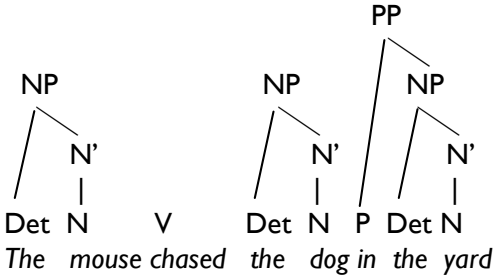
According to rule viii for instance, Ns have N's as their mother node (we'll add something to this rule later, so it's not always non-branching).



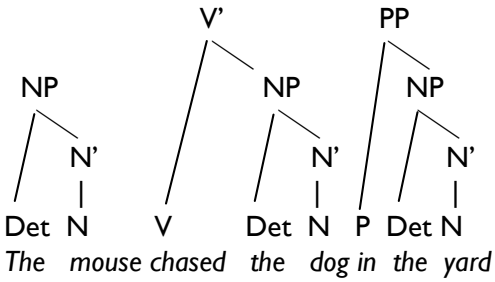
According to rule vi, we can combine Det and N' to an NP:



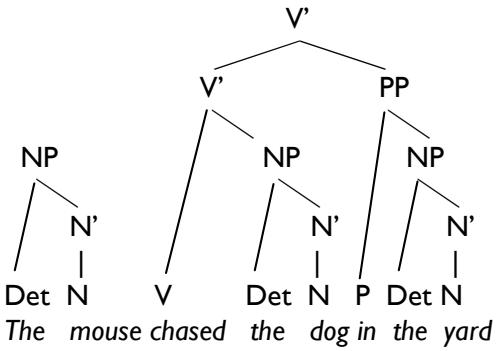
According to rule v, we can combine P and NP to a PP:



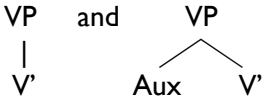
According to rule v, we can combine V and NP to a V':



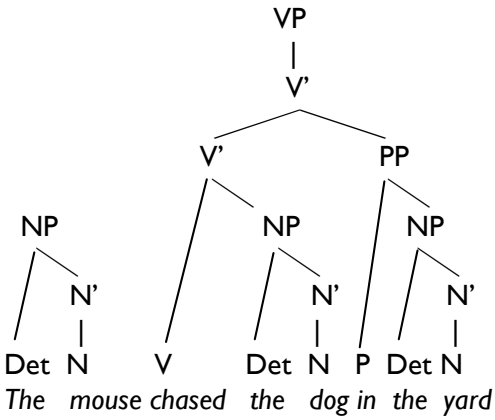
According to rule iii, we can combine V' and PP to a V':



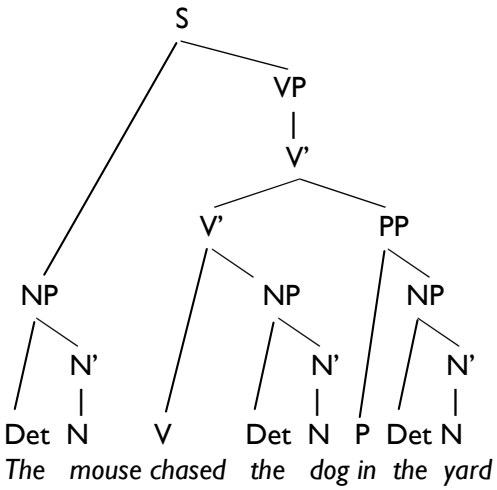
Now we can use rule ii, VP → (Aux) V'. This rule uses parentheses: ( ). The parentheses mean here that something is optional, in this case "Aux". So this rule licenses two possible structures:



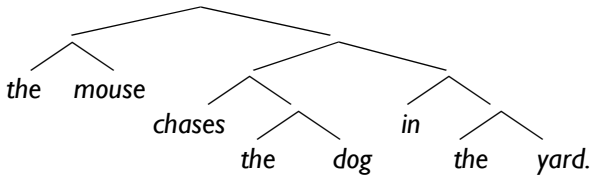
We could also simply write two rules instead: VP → V' and VP → Aux V'. The parentheses are just an abbreviation for that. Rule ii allows us to go from V' to VP then:



Now we can use rule i to combine the NP and the VP into an S:



You see that (with the exceptions of the non-branching nodes) this tree has the same structure as our constituent structure tree:



We'll keep revising the phrase structure grammar to come closer to our goal of producing all and only grammatical sentences of English, and to learn something about the shape of different phrases of English at the same time, so likely the grammar you have on the homework looks a little different from the grammar above, but things should still work exactly the same. Start with the words, and then build the tree. For every branch that you draw, you should have a rule in the grammar allowing you to do so.

Ambiguity

Sometimes our grammar will allow us to draw more than one structure for a particular sentence. This is a case of *structural ambiguity*. In the sentence above, we could have also used the PP "in the yard" to modify the noun "dog" – forming a constituent "dog in the yard" (accordingly you could have used the rule N' → N' PP instead of V' → V' PP). Typically a sentence that is structurally ambiguous has two different meanings. Here "in the yard" either tells you more about the dog, it's the "dog in the yard", or it tells you more about where the "chasing the dog" event took place, namely "in the yard". Here the two meanings are a bit hard to tell apart, but we'll see more clear-cut cases in class.