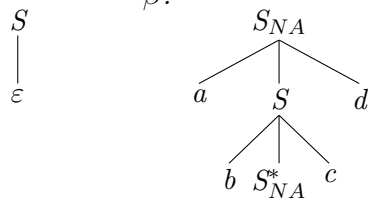
Parsing Beyond CFG
Homework 2: TAG

Laura Kallmeyer

Question 1

Consider a TAG G with start symbol S and only the following two trees:

α: β:



1. *What is the string language generated by this TAG?*
2. *Now consider a TAG G' that contains the same trees but without adjunction constraints. Does it generate the same string language as G? If not, give an example of a string that is only in one of the two languages.*
3. *What is the string language of G'?*

Solution:

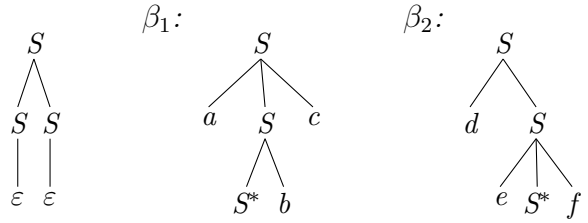
1. $L(G) = \{a^n b^n c^n d^n \mid n \geq 0\}$
2. By using β twice, adjoining each time to the root (i.e., β adjoins to the root of β, and this derived auxiliary tree adjoins at the root of α, we obtain a tree for the string $ababcdcd \notin L(G)$.
3. Clearly, if the leftmost terminal is a a (resp. a b), the rightmost is a d (resp. a c). The same holds for the next pair of terminals when moving towards the middle of a word, and so on. Furthermore, whenever adding a pair a and b to the first half of a word, the a precedes the b. Otherwise, the new pair can be added in any position. This means that, in the first half of the word, w_1, we have $|w_1|_a = |w_1|_b$ and for any two v_1, v_2 with $w_1 = v_1 v_2$, $|v_1|_a \geq |v_2|_a$ holds.

   $L(G') = \{w_1 w_2 \mid w_1 \in \{a,b\}^*, |w_1|_a = |w_1|_b$, for any v_1, v_2 with $w_1 = v_1 v_2$, $|v_1|_a \geq |v_2|_a$ holds, and, furthermore, w_2 is the image of w_1^R under a homomorphism f with $f(a) = d, f(b) = c\}$

Question 2

Now consider the following elementary trees:

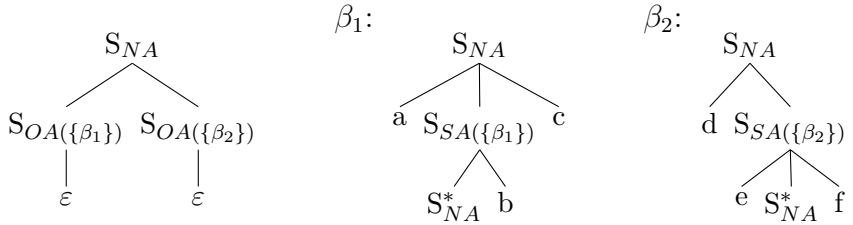
α: β_1: β_2:



Add adjunction constraints of the form NA (stands for $f_{SA}(v) = \emptyset$), OA(X) (stands for $f_{OA}(v) = 1$ and $f_{SA}(v) = X$) or SA(X) (stands for $f_{OA}(v) = 0$ and $f_{SA}(v) = X$) to the trees so that the TAG generates $\{a^n b^n c^n d^m e^m f^m \mid n, m \geq 1\}$.

Solution

α: β_1: β_2:



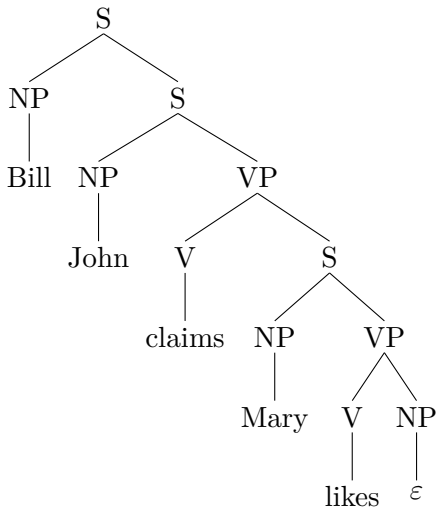
Question 3

Consider the derivation for "Bill John claims Mary likes" on slide 25, ignoring the subscript i.

1. *Give the resulting derived tree.*
2. *Give the resulting derivation tree. (As names for the trees, you can use their lexical anchors.)*

Solution

1.



2.