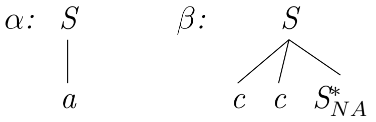
Parsing Beyond CFG
Homework 4: TAG parsing

Laura Kallmeyer

Question 1 *Consider a TAG with only the following two trees:*



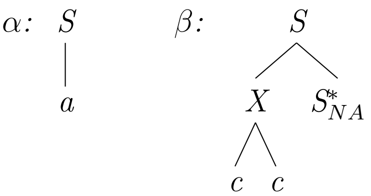
1. *Give the CYK trace (only successful items) for the input $w = cca$, using the algorithm with dotted productions.*

   Write nodes as pairs $\langle\gamma, p\rangle$ where $\gamma \in \{\alpha, \beta\}$ and p is the address of a node in γ.

   The trace should list the items together with information about the rule that has been applied and the antecedent items:

id	*item*	*operation, antecedents*
1	$[\langle\alpha,\varepsilon\rangle \to \bullet\langle\alpha,1\rangle, 2, -, -, 2]$	*axiom*
2	$[\langle\beta,\varepsilon\rangle \to \bullet\langle\beta,1\rangle\langle\beta,2\rangle\langle\beta,3\rangle, 0, -, -, 0]$	*axiom*
3	$[\langle\alpha,\varepsilon\rangle \to \langle\alpha,1\rangle\bullet, 2, -, -, 3]$	*lex-scan 1*
	...	

2. *For the binary CYK, we first have to binarize the elementary trees. Assuming that we do a left to right binarization, this leads to*



   Where X is a new non-terminal. De-binarization then only consists of removing all nodes with label X.

   Give the CYK trace (only successful items) for the input $w = cca$, using the binary CYK algorithm.

   The trace should list the items together with information about the rule that has been applied and the antecedent items:

id	*item*	*operation, antecedents*
1	$[\langle\beta, 11_\top, 0, -, -, 1]$	*lex-scan*
2	$[\langle\beta, 12_\top, 1, -, -, 2]$	*lex-scan*
	...	

Solution

1.

id	item	operation, antecedents
1	$[\langle\alpha,\varepsilon\rangle \to \bullet\langle\alpha,1\rangle, 2, -, -, 2]$	axiom
2	$[\langle\beta,\varepsilon\rangle \to \bullet\langle\beta,1\rangle\langle\beta,2\rangle\langle\beta,3\rangle, 0, -, -, 0]$	axiom
3	$[\langle\alpha,\varepsilon\rangle \to \langle\alpha,1\rangle\bullet, 2, -, -, 3]$	lex-scan 1
4	$[\langle\beta,\varepsilon\rangle \to \langle\beta,1\rangle \bullet \langle\beta,2\rangle\langle\beta,3\rangle, 0, -, -, 1]$	lex-scan 2
5	$[\langle\alpha,\varepsilon\rangle_\perp, 2, -, -, 3]$	convert 3
6	$[\langle\beta,\varepsilon\rangle \to \langle\beta,1\rangle\langle\beta,2\rangle \bullet \langle\beta,3\rangle, 0, -, -, 2]$	lex-scan 4
7	$[\langle\beta,\varepsilon\rangle \to \langle\beta,1\rangle\langle\beta,2\rangle\langle\beta,3\rangle\bullet, 0, 2, 3, 3]$	foot adjunction 6, 5
8	$[\langle\beta,\varepsilon\rangle_\perp, 0, 2, 3, 3]$	convert 7
9	$[\langle\beta,\varepsilon\rangle_\top, 0, 2, 3, 3]$	null adjoin 8
10	$[\langle\alpha,\varepsilon\rangle_\top, 0, -, -, 3]$	root adjunction 9, 5

2.

id	item	operation, antecedents
1	$[\langle\beta, 11_\top, 0, -, -, 1]$	lex-scan
2	$[\langle\beta, 12_\top, 1, -, -, 2]$	lex-scan
3	$[\langle\alpha, 1_\top, 2, -, -, 3]$	lex-scan
4	$[\langle\beta, 2_\top, 2, 2, 3, 3]$	foot-predict
5	$[\langle\beta, 1_\perp, 0, -, -, 2]$	move-binary 1, 2
6	$[\langle\alpha, \varepsilon_\perp, 2, -, -, 3]$	move-unary
7	$[\langle\beta, 1_\top, 0, -, -, 2]$	null-adjoin 5
8	$[\langle\beta, \varepsilon_\perp, 0, 2, 3, 3]$	move-binary 7,4
9	$[\langle\beta, \varepsilon_\top, 0, 2, 3, 3]$	null-adjoin 9
10	$[\langle\alpha, \varepsilon_\top, 0, -, -, 3]$	adjoin 6,9

Question 2 *We could also modify the dotted production parser towards a left-corner parser. Idea: Instead of blindly predicting every item $[v \to \bullet\gamma, i, -, -, i]$, we predict a production only when the left corner has been found.*

*The items would have the same format as in the dotted production CYK, the goal items would also be the same, and all rules except **lex-scan**, **eps-scan** and **axioms** would stay the same.*

New rules:

Lex-scan: $\dfrac{}{[v_\top, i, -, -, i+1]}$ $l(v) = w_{i+1}$ **Eps-scan**: $\dfrac{}{[v_\top, i, -, -, i]}$ $l(v) = \varepsilon$

Old rules that do not change:

Convert: $\dfrac{[v \to \gamma\bullet, i, f_1, f_2, j]}{[v_\perp, i, f_1, f_2, j]}$

Null-adjoin: $\dfrac{[v_\perp, i, f_1, f_2, j]}{[v_\top, i, f_1, f_2, j]}$ $f_{OA}(v) = 0$

Move right: $\dfrac{[v \to \gamma_1 \bullet w\gamma_2, i, f_1, f_2, j], [w_\top, j, f_3, f_4, k]}{[v \to \gamma_1 w \bullet \gamma_2, i, f_1 \oplus f_3, f_2 \oplus f_4, k]}$

Substitute: $\dfrac{[v \to \gamma_1 \bullet w\gamma_2, i, f_1, f_2, j], [u, \top, j, -, -, k]}{[v \to \gamma_1 w \bullet \gamma_2, i, f_1, f_2, k]}$ *$l(w) = l(u)$, $root(u)$, w substitution node*

Foot adjunction: $\dfrac{[v \to \gamma_1 \bullet w\gamma_2, i, -, -, j], [u_\perp, j, f_1, f_2, k]}{[v \to \gamma_1 w \bullet \gamma_2, i, j, k, k]}$ *$l(w) = l(u), foot(w)$, adj. of auxiliary tree with w allowed at u*

Root adjunction: $\dfrac{[v_\top, i, j, k, l], [u, \perp, j, f_1, f_2, k]}{[u_\top, i, f_1, f_2, l]}$ *$l(v) = l(u), root(v)$, adj. of auxiliary tree with v allowed at u*

*In addition, we need a new rule **Left-corner-predict** that starts a rule once its left corner (i.e., its first lefthand side element) has been found. It replaces the former blind predictions done in **axioms**. How should this rule look like?*

Solution:

Left-corner-predict: $\dfrac{[u_\top, i, f_1, f_2, j]}{[v \to u \bullet \gamma, i, f_1, f_2, j]}$ $v \to u\gamma$ is a rule