Parsing Beyond CFG
Mid-term exam: preparation exercises

Laura Kallmeyer, Tatiana Bladier

Sommersemester 2018

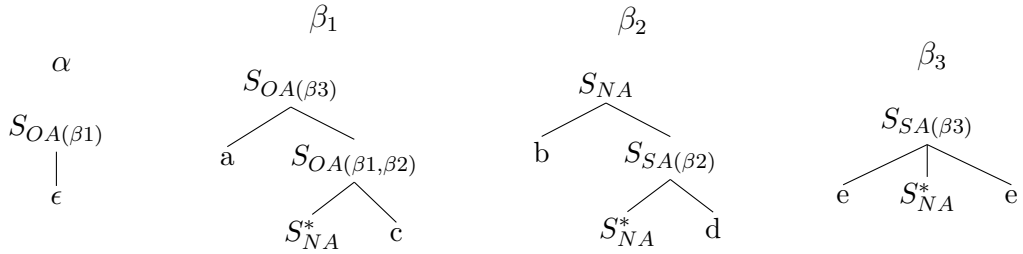
Question 1

Geben Sie für die Sprache

$$L = \{e^k a^n b^m c^n d^m e^k \mid k, n, m \geq 1\}$$

eine TAG an, die diese Sprache generiert. Verwenden Sie dabei die adjunction constraints und geben Sie in den adjunction constraints die Menge der Elementarbäume an, für die diese Beschränkungen gelten (z.B. $OA_{\beta 1}$ an einem Knoten, an dem Adjunktion von $\beta 1$ obligatorisch ist).

Lösung:



Question 2

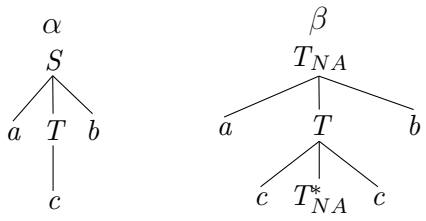
Wir haben schon gezeigt, dass $L = \{a^n b^n c^n d^n e^n \mid n \geq 0\}$ keine Tree Adjoining Language ist. Zeigen Sie, dass auch $L' = \{a^{2n} c b^n (gh)^n d^n h^n \mid n \geq 0\}$ keine Tree Adjoining Language ist.

Tipp: Nutzen Sie die Abgeschlossenheitseigenschaften der jeweiligen Sprachklassen aus.

Lösung:

Homomorphismus f mit $f(aa) \to a$, $f(c) \to \varepsilon$, $f(b) \to b$, $f(gh) \to c$, $f(d) \to d$, $f(h) \to e$ bildet L' auf L ab. D.h., wenn L' eine TAL ist, dann muss auch L eine TAL sein. Widerspruch.

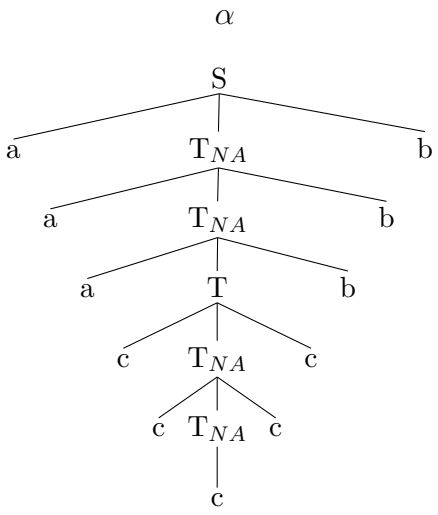
Question 3 *Consider the TAG with non-terminals $N = \{S, T\}$, terminals $T = \{a, b, c\}$ and elementary trees α and β:*



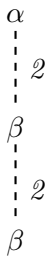
1. *Give the derived tree and derivation tree for the derivation that uses α once and β twice.*
2. *Which string language is generated by this TAG?*

Solution:

Derived tree:



Derivation tree:

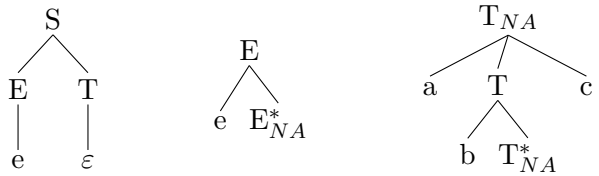


2. $\{a^{n+1}c^{2n+1}b^{n+1} \mid n \geq 0\}$

Question 4

Give a TAG generating the following language: $L = \{e^k a^n b^n c^n \mid k \geq 1, n \geq 0\}$

Solution:

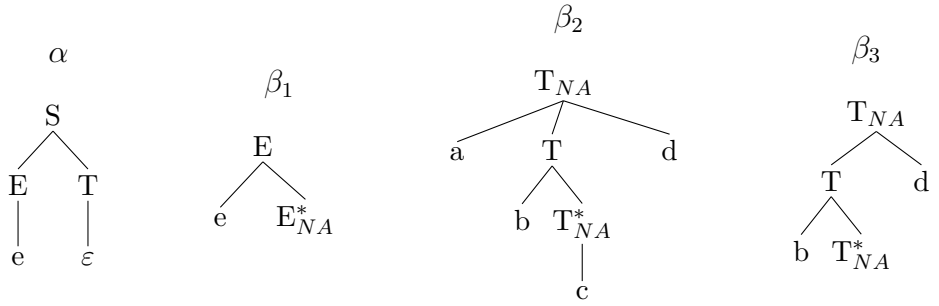


Question 5

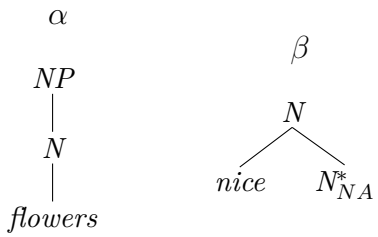
Give a TAG generating the following language:

$$L = \{e^k a^m b^n c^m d^n \mid k \geq 1, n \geq m \geq 0\}$$

Solution:



Question 6 *Consider the following TAG (non-terminals $\{N, NP\}$, terminals $\{$nice, flowers$\}$):*



Complete the following trace for a CYK parsing of the input "nice flowers" using this TAG. Indicate for each step the antecedent items and the operation that has been used.

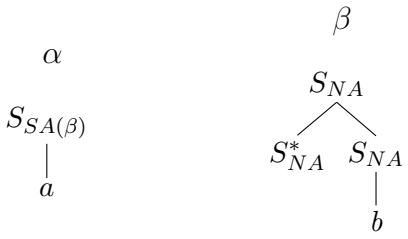
Give only the successful items.

id	*item*	*operation*	*antecedent items*
1	$[\beta, 1_\top, 0, -, -, 1]$	*Lex-scan*	–
2	$[\alpha, 11_\top, 1, -, -, 2]$	*Lex-scan*	–
3	$[\beta, 2_\top, 1, 1, 2, 2]$	*Foot-predict*	–
4	$[\alpha, 1_\bot, 1, -, -, 2]$	*Move-unary*	*2*
5	...		

Solution:

id	item	operation	antecedent items
1	$[\beta, 1_\top, 0, -, -, 1]$	Lex-scan	–
2	$[\alpha, 11_\top, 1, -, -, 2]$	Lex-scan	–
3	$[\beta, 2_\top, 1, 1, 2, 2]$	Foot-predict	–
4	$[\alpha, 1_\bot, 1, -, -, 2]$	Move-unary	2
5	$[\beta, \varepsilon_\bot, 0, 1, 2, 2]$	Move-binary	1,3
6	$[\beta, \varepsilon_\top, 0, 1, 2, 2]$	Null-adjoin	5
7	$[\alpha, 1_\top, 0, -, -, 2]$	Adjoin	4,6
8	$[\alpha, \varepsilon_\bot, 0, -, -, 2]$	Move-unary	7
9	$[\alpha, \varepsilon_\top, 0, -, -, 2]$	Null-adjoin	8

Question 7 *Betrachten Sie folgende TAG:*



Geben Sie die trace des Early Parsers für die Eingabe $w = ab$ an. Geben Sie nur die Items an, die zum korrekten parse führen. Erklären Sie für jedes Item aus welcher Regel es generiert wurde und welches Antezedens dafür verwendet wurde (die Items werden über die Nummern in der ersten Spalte identifiziert).

Id	*Item*	*Operation*	*Antezedens-Items*
1	$[\alpha, \varepsilon, la, 0, -, -, 0, 0]$	*Initialize*	–
2	$[\beta, \varepsilon, la, 0, -, -, 0, 0]$	*...*	*1*
3	$[\beta, \varepsilon, lb, 0, -, -, 0, 0]$	*...*	*2*
4	$[\beta, 1, la, 0, -, -, 0, 0]$	*...*	*3*

Lösung:

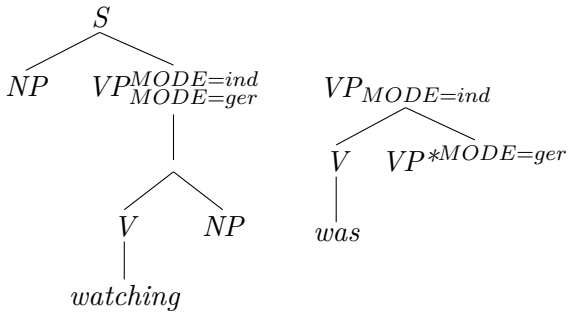
Id	Item	Operation	Antezedens-Items
1	$[\alpha, \varepsilon, la, 0, -, -, 0, 0]$	Initialize	–
2	$[\beta, \varepsilon, la, 0, -, -, 0, 0]$	PredictAdjoinable	1
3	$[\beta, \varepsilon, lb, 0, -, -, 0, 0]$	PredictNoAdjunction	2
4	$[\beta, 1, la, 0, -, -, 0, 0]$	MoveDown	3
5	$[\beta, 1, lb, 0, -, -, 0, 0]$	PredictNoAdjunction	4
6	$[\alpha, \varepsilon, lb, 0, -, -, 0, 0]$	PredictAdjoned	5
7	$[\alpha, 1, la, 0, -, -, 0, 0]$	MoveDown	6
8	$[\alpha, 1, ra, 0, -, -, 1, 0]$	ScanTerm	7
9	$[\alpha, \varepsilon, rb, 0, -, -, 1, 0]$	MoveUp	8
10	$[\beta, 1, rb, 0, 0, 1, 1, 0]$	CompleteFoot	9,5
11	$[\beta, 2, la, 0, -, -, 1, 0]$	MoveRight	10
12	$[\beta, 2, lb, 1, -, -, 1, 0]$	PredictNoAdjunction	11
13	$[\beta, 21, la, 1, -, -, 1, 0]$	MoveDown	12
14	$[\beta, 21, ra, 1, -, -, 2, 0]$	ScanTerm	13
15	$[\beta, 2, rb, 1, -, -, 2, 0]$	MoveUp	13
16	$[\beta, 2, ra, 1, -, -, 2, 0]$	CompleteNode	15,11
17	$[\beta, \varepsilon, rb, 0, 0, 1, 2, 0]$	MoveUp	16
18	$[\beta, \varepsilon, ra, 0, 0, 1, 2, 0]$	CompleteNode	16,2
19	$[\alpha, \varepsilon, rb, 0, -, -, 2, 1]$	Adjoin	19,9
20	$[\alpha, \varepsilon, ra, 0, -, -, 2, 0]$	CompleteNode	20,1

Question 8 (Features)

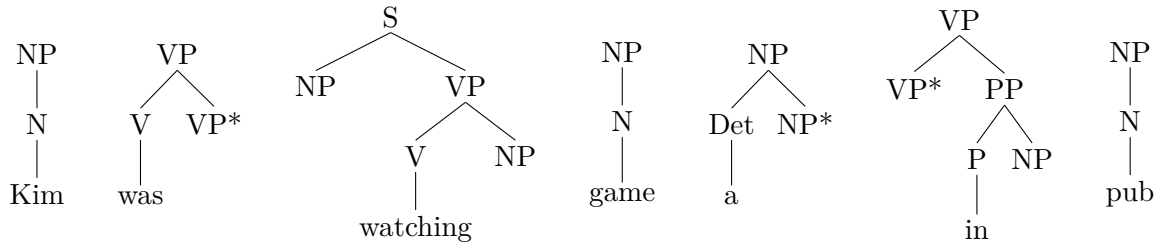
(1) *a. Kim was watching a game in a pub.*
*b. *Kim watching a game.*

1. *Give the elementary trees that generate the sentence in (1-a) with separate elementary trees for all lexical elements in the sentence (no features).*

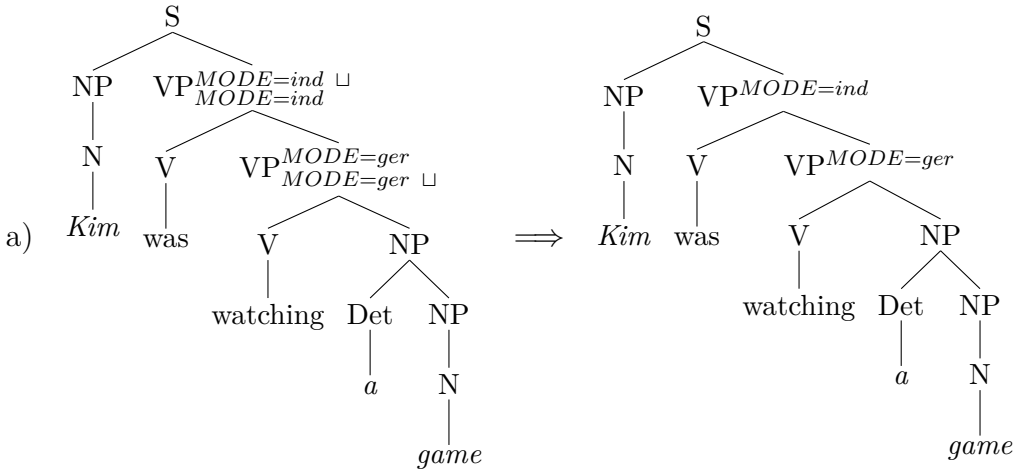
2. *Consider the following two modifications for the verbal elementary trees. Using the* Mode *feature and feature unifications constraints:*
a) give a FTAG analysis of the sentence "Kim was watching a game" and
b)show how they exclude the sentence (1-b).

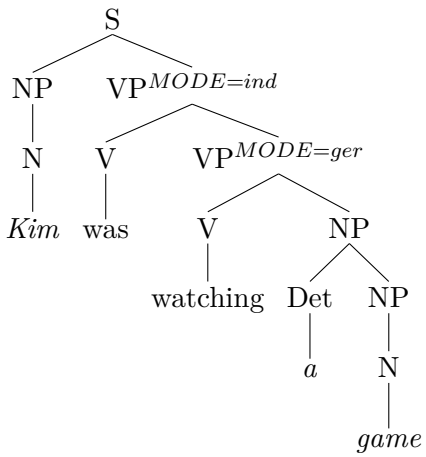


Solution:

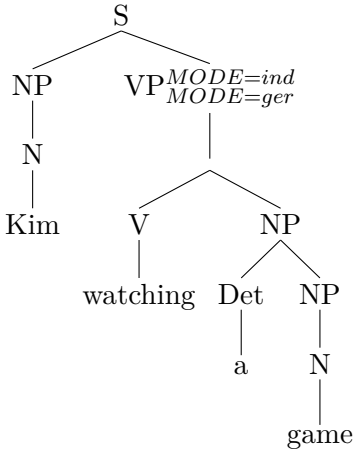
1.


2.

a)


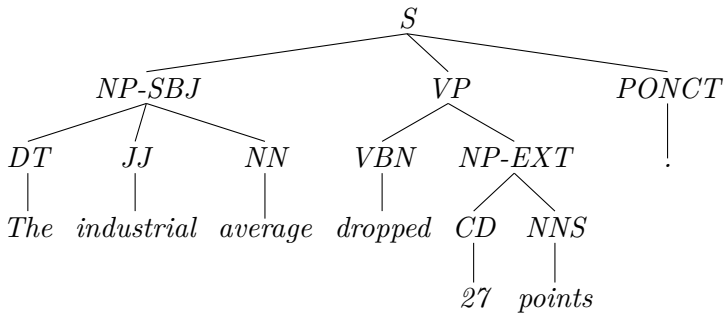


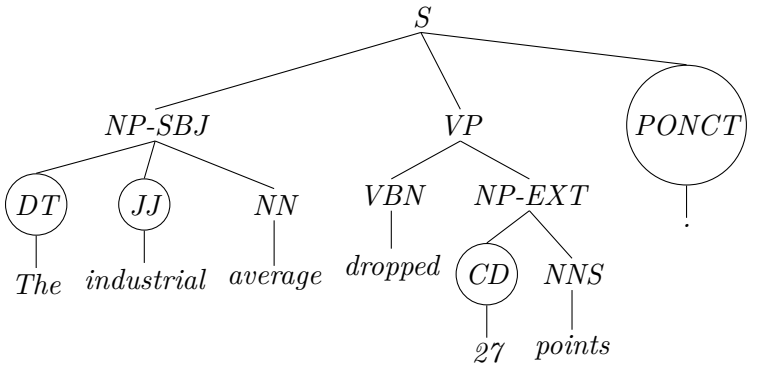
b) The analysis for the ungrammatical sentence "Kim watching a game" would lead to the following mid-result:

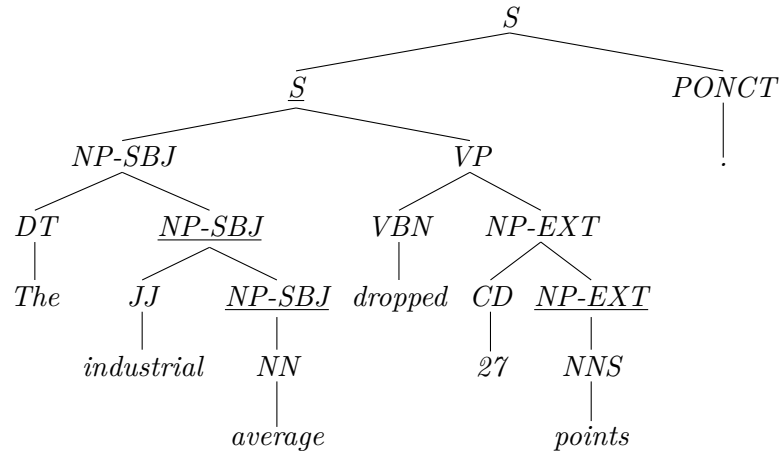


However, for the complete analysis we have to unify top and bottom feature structures at the VP node, and the unification would fail in this case, which means that the sentence cannot be generated with this LTAG.

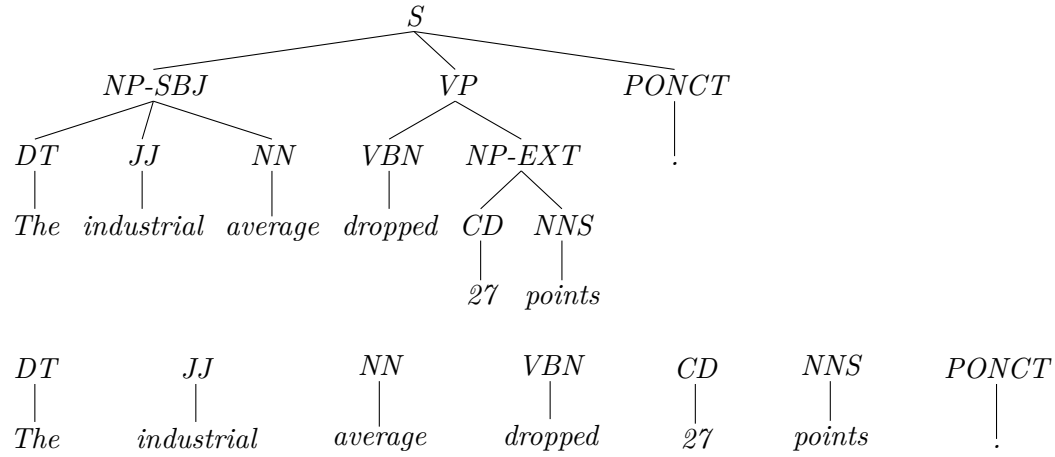
Question 9 (TAG induction (Top down approach))
In the following tree, mark every modifier node and provide a fully-bracketed tree



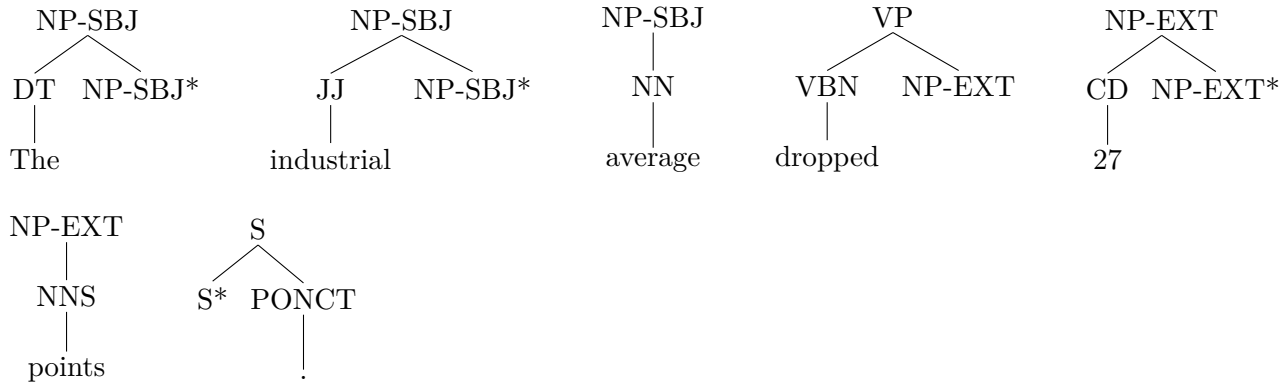




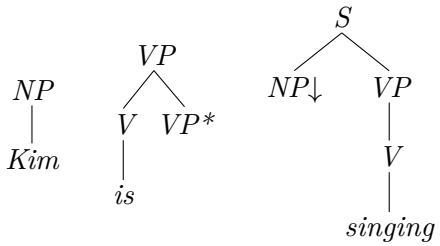
Question 10 (TAG induction (Bottom up approach))
Consider the following bracketed tree and the elementary trees extracted in the first round of the algorithm. Give the elementary trees after the second round of the algorithm.



Solution:



Question 11 (LTIG to PCFG) *Convert the following LTIG to CFG:*



Solution:

A CFG $G = \langle\ N,\ T,\ P,\ S\ \rangle$

$N = \{NP, NP_1, NP_2, VP, VP_1, VP_2, S, S_1, S_2, V, V_1, V_2\}$

$T = \{\epsilon, Kim, is, singing\}$

$S = S$

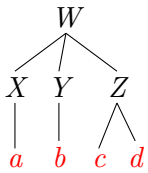
$P = \{NP_1 \to \epsilon, NP_1 \to \epsilon, VP_1 \to \epsilon, VP_2 \to \epsilon, S_1 \to \epsilon, S_2 \to \epsilon, V_1 \to \epsilon, V_2 \to \epsilon,$

$NP \to NP_1\ \ Kim\ \ NP_2, \qquad S \to S_1\ \ NP\ \ VP\ \ S_2, \qquad VP \to VP_1\ \ V\ \ VP_2,$

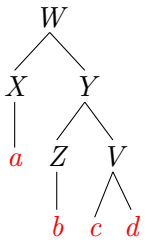
$V \to singing, \qquad VP_1 \to VP\ \ VP_1, \qquad VP \to V, \qquad V \to is\ \ V_2\}$

Question 12 (Evaluation) *Calculate the labeled F-Score for the following prediction of the parser:*

Candidate parse:

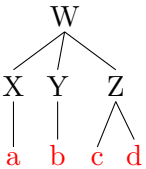


Golden standard:

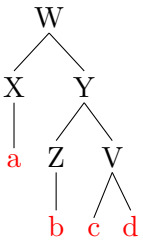


Solution:

Candidate parse:



Golden standard:



Candidate	Gold	
		precision = 4/4 = 1
		recall = 4/5 = 0.8
X $\to^*$ a	X $\to^*$ a	f-score = 0.89
Y $\to^*$ b	Z $\to^*$ b	
Z $\to^*$ cd	V $\to^*$ cd	labelled precision = 2/4 = 0.5
–	Y $\to^*$ bcd	labelled recall = 2/5 = 0.4
W $\to^*$ abcd	W $\to^*$ abcd	labelled f-score = 0.44