

Parsing Beyond Context-Free Grammars

Zwischenklausur

Laura Kallmeyer, Tatiana Bladier

Sommersemester 2018, Heinrich-Heine-Universität Düsseldorf

Klausurdauer: 90 Minuten.

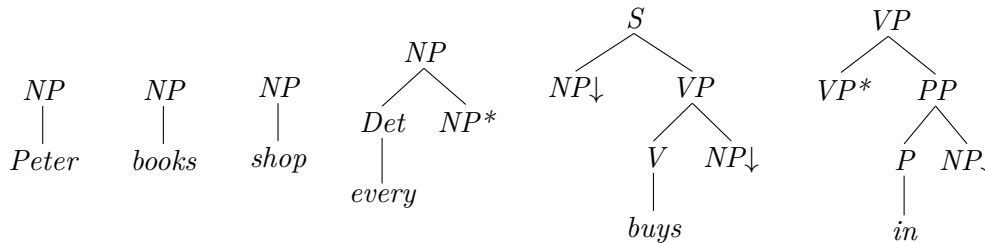
Hilfsmittel: Sämtliche Unterrichtsmaterialien und Notizen in nicht-elektronischer Form.

Aufgaben, die mit **(BA)** gekennzeichnet sind, sind nur für die BA-APs vorgesehen, und die Aufgaben markiert mit **(MA)** sind nur für MA-APs. Alle anderen Aufgaben sind sowohl für Bachelor- als auch für Master-APs.

Aufgabe 1

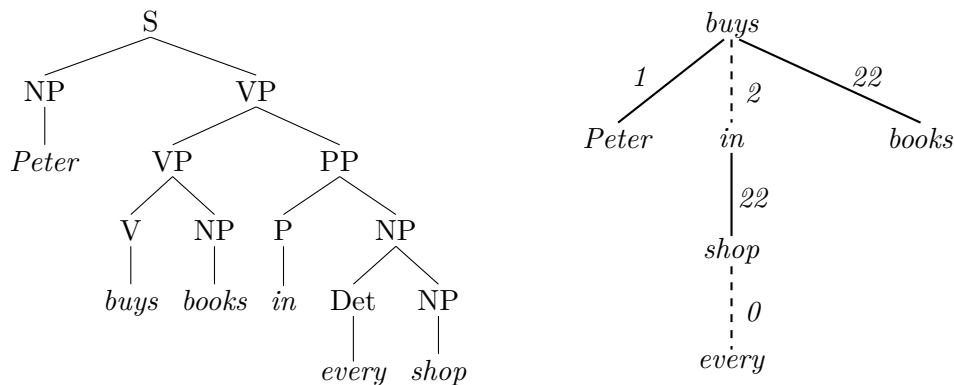
(7 Punkte)

Betrachten Sie die LTAG G_{LTAG} mit folgenden Elementarbäumen:



Wie sieht der abgeleitete Baum (derived tree) und der Ableitungsbaum (derivation tree) für den Satz "Peter buys books in every shop" aus, wenn man G_{LTAG} der Ableitung zugrunde legt?

Lösung:



Aufgabe 2 (BA)

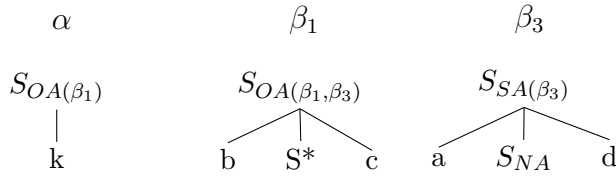
(5 Punkte)

Geben Sie für die Sprache

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

eine TAG an, die diese Sprache generiert. Verwenden Sie dabei Adjunktionsconstraints, indem Sie in den Constraints jeweils die Menge der Elementarbäume an, für die diese Beschränkungen gelten (z.B. $OA(\beta_a, \beta_b)$ an einem Knoten, an dem Adjunktion von β_a oder β_b obligatorisch ist).

Lösung:



Aufgabe 2 (MA)

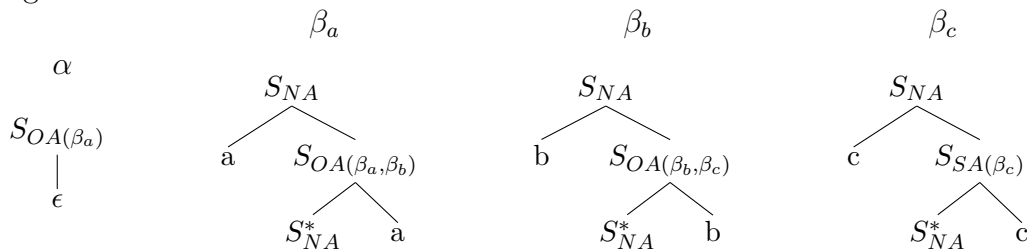
(5 Punkte)

Geben Sie für die Sprache

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

eine TAG an, die diese Sprache generiert. Verwenden Sie dabei Adjunktionsconstraints, indem Sie in den Constraints jeweils die Menge der Elementarbäume an, für die diese Beschränkungen gelten (z.B. $OA(\beta_a, \beta_b)$ an einem Knoten, an dem Adjunktion von β_a oder β_b obligatorisch ist).

Lösung:



Aufgabe 3 (BA)

(6 Punkte)

Nehmen Sie an, dass wir schon gezeigt haben, dass die Sprache

$$L_1 = \{a^n b^n c^n d^n e^n \mid n \geq 1\}$$

keine Tree Adjoining Language ist. Zeigen Sie, dass auch die Sprache

$$L_2 = \{a^n f^m b^n f^m c^n f^m d^n f^m e^n f^m \mid n, m \geq 1\}$$

keine Tree Adjoining Language ist.

Hinweis: Benutzen Sie dabei die Abgeschlossenheit von Tree Adjoining Languages unter Anwendung von Homomorphismen.

Lösung:

Homomorphismus g mit $g(f) \rightarrow \epsilon, g(a) = a, g(b) = b, g(c) = c, g(d) = d, g(e) = e$ bildet L_2 auf L_1 ab. D.h., wenn L_2 eine TAL ist, dann muss auch L_1 eine TAL sein. Widerspruch.

Aufgabe 3 (MA)

(6 Punkte)

Nehmen Sie an, dass wir schon gezeigt haben, dass die Sprache

$$L_1 = \{a^n b^n c^n d^n e^n \mid n \geq 1\}$$

keine Tree Adjoining Language ist. Zeigen Sie, dass auch die Sprache

$$L_2 = \{w_1 f^m w_2 f^m e^n \mid w_1, w_2 \in \{a, b, c, d, e\}^*, |w_1|_a = |w_1|_b = |w_2|_c = |w_2|_d = n, n \geq 1, m \geq 1\}$$

keine Tree Adjoining Language ist.

Hinweis: Nutzen Sie die Abgeschlossenheit von TALs unter Schnittbildung mit regulären Sprachen und unter Anwendung von Homomorphismen. Konkret: Schneiden Sie bei Ihrer Beweisführung L_2 mit einer geeigneten regulären Sprache und wenden Sie anschließend einen geeigneten Homomorphismus auf das Ergebnis der Schnittbildung an.

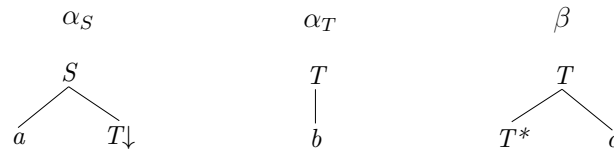
Lösung:

Angenommen, L_2 ist eine TAL. Dann muss der Schnitt von L_2 mit der von $a^*b^*f^*c^*d^*f^*e^*$ denotierten Sprache ebenfalls eine TAL sein. Diese neue Sprache ist $L'_2 = \{a^n b^n f^m c^n d^n f^m e^n \mid n, m \geq 1\}$. Wenn L'_2 eine TAL ist, muss auch das Bild von L'_2 unter einem Homomorphismus, der f auf ϵ und alle anderen Terminalen auf sich selbst abbildet, eine TAL sein. Dieses Bild ist jedoch L_1 , von der wir schon wissen, dass sie keine TAL ist. Widerspruch.

Aufgabe 4 (CYK)

(12 Punkte)

Betrachten Sie die folgende TAG:



Betrachten Sie nun die folgende trace des CYK Parsers für die Eingabe $w = abc$ an. Vervollständigen Sie in den Zeilen 5 bis 11 die Items und die Bezeichnung der Operationen unter Rücksicht der vorgegebenen Antezedenz-Items (die Items werden über die Nummern in der ersten Spalte identifiziert). Nur die Items, die zu einem erfolgreichen Parse führen, sollen in der Tabelle angegeben werden.

	Item	Operation	Antezedenz-Items
1.	$[\alpha_S, 1_\top, 0, -, -, 1]$	lex-scan (a)	–
2.	$[\alpha_T, 1_\top, 1, -, -, 2]$	lex-scan (b)	–
3.	$[\beta, 2_\top, 2, -, -, 3]$	lex-scan (c)	–
4.	$[\beta, 1_\top, 1, 1, 2, 2]$	foot-predict	–
5.			2
6.			4 und 3
7.			6
8.		adjoin	7 und 5
9.		substitute	8
10.			9 und 1
11.			10

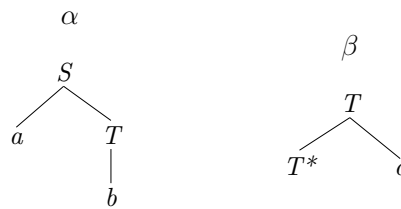
Lösung:

	Item	Operation	Antezedenz-Items
1.	$[\alpha_S, 1_\top, 0, -, -, 1]$	lex-scan (a)	–
2.	$[\alpha_T, 1_\top, 1, -, -, 2]$	lex-scan (b)	–
3.	$[\beta, 2_\top, 2, -, -, 3]$	lex-scan (c)	–
4.	$[\beta, 1_\top, 1, 1, 2, 2]$	foot-predict	–
5.	$[\alpha_T, \epsilon_\perp, 1, -, -, 2]$	move-unary	2
6.	$[\beta, \epsilon_\perp, 1, 1, 2, 3]$	move-binary	4 und 3
7.	$[\beta, \epsilon_\top, 1, 1, 2, 3]$	null-adjoin	6
8.	$[\alpha_T, \epsilon_\top, 1, -, -, 3]$	adjoin	7 und 5
9.	$[\alpha_S, 2_\top, 0, -, -, 3]$	substitute	8
10.	$[\alpha_S, \epsilon_\perp, 0, -, -, 3]$	move-binary	9 und 1
11.	$[\alpha_S, \epsilon_\top, 0, -, -, 3]$	null-adjoin	10

Aufgabe 5

(14 Punkte)

Betrachten Sie die folgende TAG:



Betrachten Sie nun die folgende trace des Early Parsers für die Eingabe $w = abc$ an. Geben Sie für die Zeilen 14 bis 20 die Items und die Bezeichnung der Operationen unter Rücksicht der vorgegebenen Antezedens-Items an (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	$[\alpha, \varepsilon, lb, 0, -, -, 0, 0]$	PredictNoAdjunction	1
3	$[\alpha, 1, la, 0, -, -, 0, 0]$	MoveDown	2
4	$[\alpha, 1, ra, 0, -, -, 1, 0]$	ScanTerm	3
5	$[\alpha, 2, la, 0, -, -, 1, 0]$	MoveRight	4
6	$[\beta, \varepsilon, la, 1, -, -, 1, 0]$	PredictAdjoinable	5
7	$[\beta, \varepsilon, lb, 1, -, -, 1, 0]$	PredictNoAdjunction	6
8	$[\beta, 1, la, 1, -, -, 1, 0]$	MoveDown	7
9	$[\beta, 1, lb, 1, -, -, 1, 0]$	PredictNoAdjunction	8
10	$[\alpha, 2, lb, 1, -, -, 1, 0]$	PredictAdjoined	9
11	$[\alpha, 3, la, 1, -, -, 1, 0]$	MoveDown	10
12	$[\alpha, 3, ra, 1, -, -, 2, 0]$	ScanTerm	11
13	$[\alpha, 2, rb, 1, -, -, 2, 0]$	MoveUp	12
14			13 und 9
15			14 und 8
16			15
17			16
18			17
19			18 und 6
20			19 und 13

Bemerkung: Es sind nur Items gelistet, die Teil eines erfolgreichen Parse sind, und die Trace ist nicht vollständig, die letzten Items fehlen.

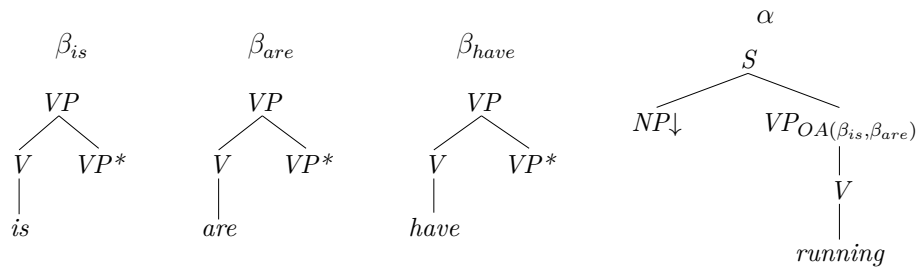
Lösung:

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

Aufgabe 6

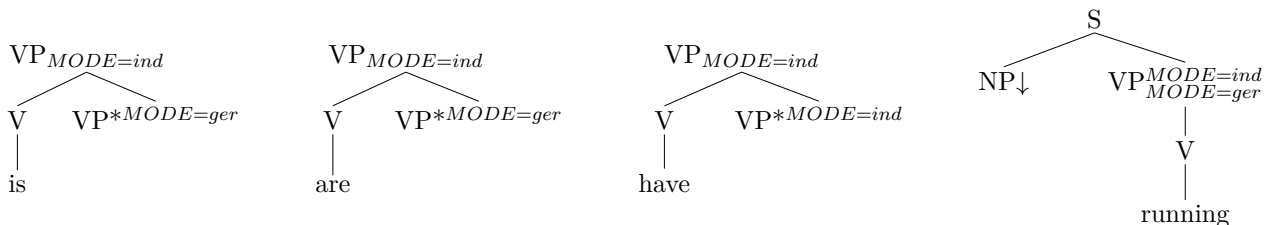
(5 Punkte)

Betrachten Sie die folgenden LTAG-Elementarbäume mit Adjunktionsconstraints:



Zeigen Sie am Beispiel des feature MODE, wie man diese adjunction constraints mit Merkmalsstrukturen (feature structures) modellieren kann.

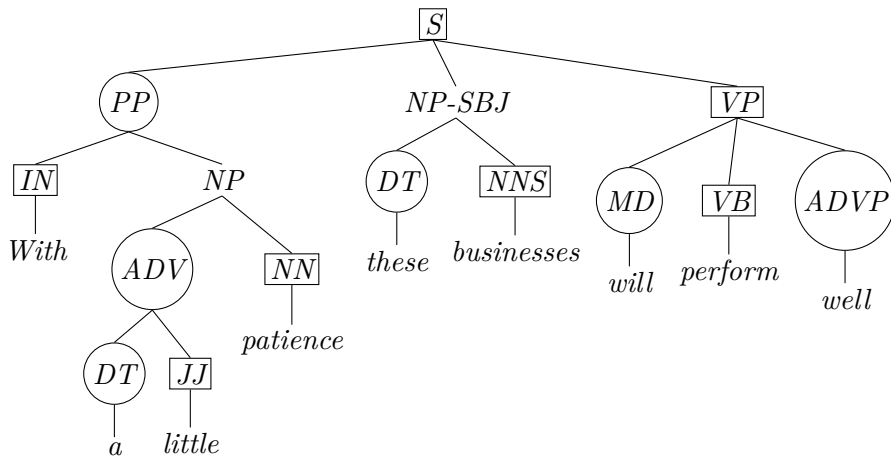
Lösung:



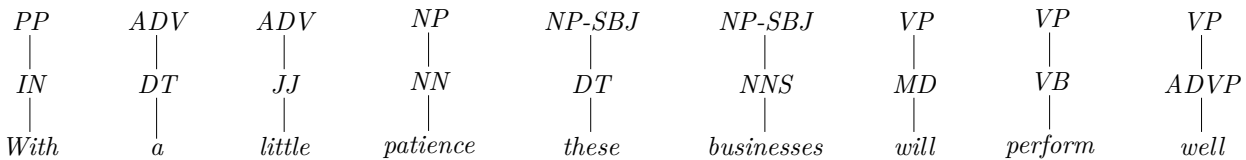
Aufgabe 7

(5 Punkte)

Betrachten Sie den folgenden Satz aus Penn Treebank, in dem die Kopf-Knoten (head nodes) in einem Viereck angegeben sind, die Modifizierer-Knoten (modifier nodes) eingekreist sind und die Argumente (arguments) keine Markierung haben:

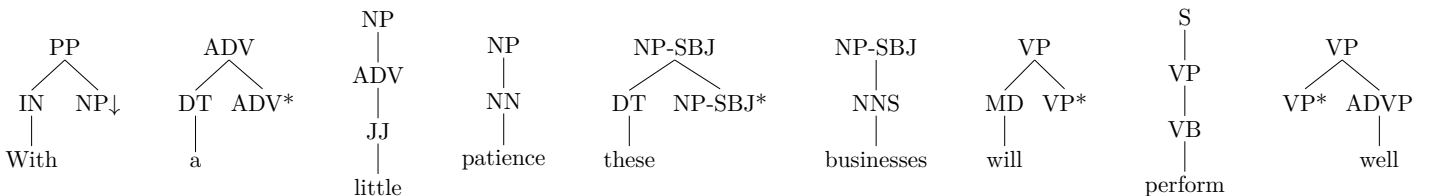


Nach den ersten Schritten des bottom-up Algorithmus für die TAG-Induktion sehen die Partiellbäume folgendermaßen aus:



Zeigen Sie die Partiellbäume nach dem nächsten Schritt des Algorithmus.

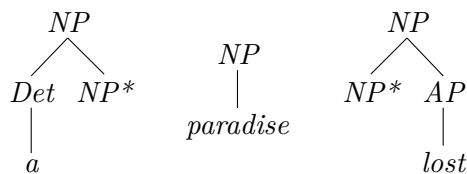
Lösung:



Aufgabe 8 (BA)

(10 Punkte)

Betrachten Sie die folgenden Elementarbäume in LTIG und konvertieren Sie diese zu CFG:



Lösung:

$$G = \langle N, T, P, S \rangle:$$

$$N = \{NP, NP_1, NP_2, AP, AP_1, AP_2, Det, Det_1, Det_2\}$$

$$T = \{a, paradise, lost\}$$

$S = NP$

$P = \{NP_1 \rightarrow \epsilon, NP_2 \rightarrow \epsilon, AP_1 \rightarrow \epsilon, AP_2 \rightarrow \epsilon, Det_1 \rightarrow \epsilon, Det_2 \rightarrow \epsilon,$

$NP \rightarrow NP_1 \text{ paradise } NP_2,$

$NP_1 \rightarrow NP \ NP_1,$

$NP \rightarrow Det,$

$Det \rightarrow a \ Det_2,$

$NP_2 \rightarrow NP \ NP_2,$

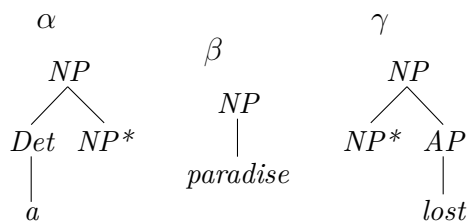
$NP \rightarrow AP,$

$AP \rightarrow AP_1 \text{ lost } \}$

Aufgabe 8 (MA)

(10 Punkte)

Betrachten Sie die folgenden Elementarbäume in LTIG und die Eingabe "a paradise lost". Geben Sie für die Zeilen 4 bis 8 die Items und die Bezeichnung der Operationen unter Rücksicht der vorgegebenen Antezedenz-Items an (die Items werden über die Nummern in der ersten Spalte identifiziert).



	Item	Operation	Antezedenz-Items
1.	$[\beta_{NP}^1 \rightarrow \bullet \beta_{paradise}^2, 0, 0]$	Initialize	–
2.	$[\alpha_{NP}^1 \rightarrow \bullet \alpha_{Det}^2 \alpha_{NP}^4, 0, 0]$	PredictLeftAdjunction	1
3.	$[\alpha_{Det}^2 \rightarrow \bullet \alpha_a^3, 0, 0]$	MoveDown	2
4.			3
5.			4,2
6.			5
7.			6,1
8.			7

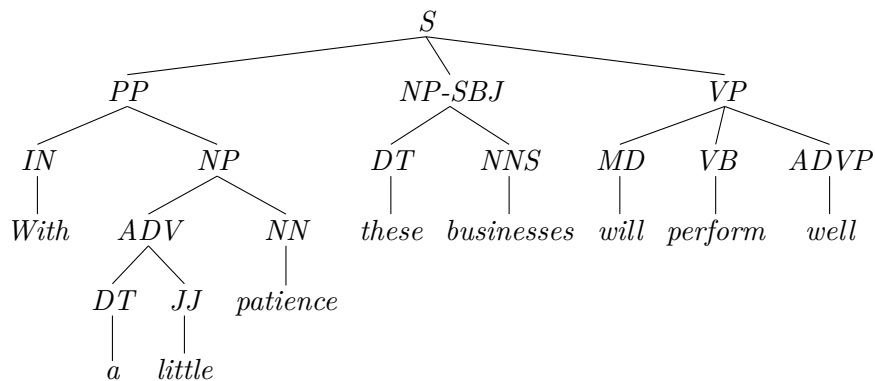
Lösung:

	Item	Operation	Antezedenz-Items
1.	$[\beta_{NP}^1 \rightarrow \bullet \beta_{paradise}^2, 0, 0]$	Initialize	–
2.	$[\alpha_{NP}^1 \rightarrow \bullet \alpha_{Det}^2 \alpha_{NP}^4, 0, 0]$	PredictLeftAdjunction	1
3.	$[\alpha_{Det}^2 \rightarrow \bullet \alpha_a^3, 0, 0]$	MoveDown	2
4.	$[\alpha_{Det}^2 \rightarrow \alpha_a^3 \bullet, 0, 1]$	Scan	3
5.	$[\alpha_{NP}^1 \rightarrow \alpha_{Det}^2 \bullet \alpha_{NP}^4, 0, 1]$	CompleteNode	4,2
6.	$[\alpha_{NP}^1 \rightarrow \alpha_{Det}^2 \alpha_{NP}^4 \bullet, 0, 1]$	ScanFoot	5
7.	$[\beta_{NP}^1 \rightarrow \bullet \beta_{paradise}^2, 0, 1]$	LeftAdjunction	6,1
8.	$[\beta_{NP}^1 \rightarrow \beta_{paradise}^2 \bullet, 0, 2]$	Scan	7

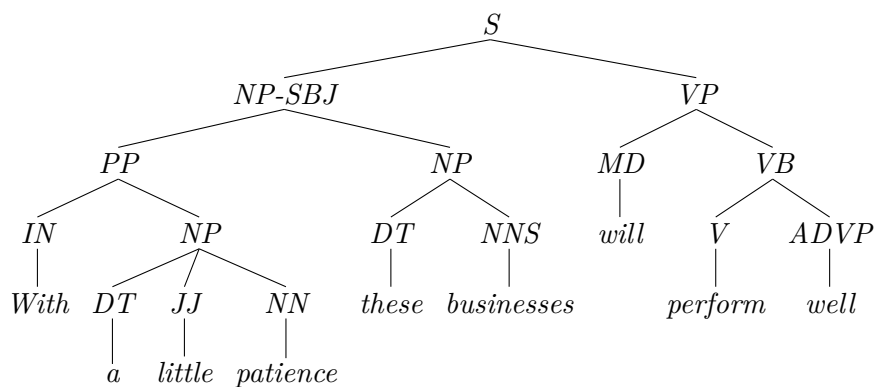
Aufgabe 9 (Evaluation (f-score, recall, precision))

(5 Punkte)

Angenommen, wir haben den folgenden Satz in der Penn Treebank:



Ein implementierter Parser hat für diesen Satz die folgende syntaktische Analyse als Output produziert:



Berechnen Sie Recall, Precision und F-Score (die Brüche muss man nicht berechnen. Es reicht aus, wenn Sie den Rechenweg angeben.)

Lösung:

Candidate

IN → With
 DT → a
 JJ → little
 –
 NN → patience
 NP →* a little patience
 PP →* With a little patience
 DT → these
 NNS → businesses
 NP →* these businesses
 NP-SBJ →* With a little patience
 these businesses
 MD → will
 VB → perform
 ADVP → well
 VB →* perform well
 VP →* will perform well
 S →* With a little patience these
 businesses will perform well

Gold

IN → With
 DT → a
 JJ → little
 ADV →* a little
 NN → patience
 NP →* a little patience
 PP →* With a little patience
 DT → these
 NNS → businesses
 NP-SBJ →* these businesses
 –
 MD → will
 VB → perform
 ADVP → well
 –
 VP →* will perform well
 S →* With a little patience these
 businesses will perform well

$$\begin{aligned}
 \text{precision} &= 14/16 = 0.875 \\
 \text{recall} &= 14/15 = 0.933 \\
 \text{f-score} &= \frac{2 \times 14/16 \times 14/15}{14/16 + 14/15} \\
 &= 0.903
 \end{aligned}$$