RWTH Aachen | **SS 2017**
Lehrgebiet Theoretische Informatik | Blatt 6
Rossmanith–Dreier–Hark–Kuinke | 12.6.2017

Übung zur Vorlesung Formale Sprachen, Automaten und Prozesse

Aufgabe T18

Entwerfen sie für die folgenden Sprachen L_i jeweils eine kontextfreie Grammatik, die L_i erzeugt.

a) $L_1 = \{\, a^n b^n \mid n \in \mathbf{N} \,\}$ über dem Alphabet $\{a, b\}$

b) $L_2 = \{\, w \in \Sigma^* \mid w = w^R \,\}$, die Sprache der Palindrome über $\Sigma = \{a, b\}$

c) $L_3 = \{\, w \in \Sigma^* \mid w \neq w^R \,\}$, die Sprache der Nicht-Palindrome über $\Sigma = \{a, b\}$

d) L_4, die Sprache der Texte über dem Alphabet $\{a, \ldots, Z, 0, \ldots, 9\}$ mit `<i>`*italic*`</i>`- und `<b>`**boldface**`</b>`-Tags. Achten Sie darauf, daß die Tags vernünftig geschachtelt sind. Zum Beispiel ist `<b> <i> </b> </i>` nicht gültig.

Lösungsvorschlag:

a) G_1: $S \to aSb \mid \epsilon$

b) G_2: $S \to aSa \mid bSb \mid a \mid b \mid \epsilon$

c) G_3: $S \to aSa \mid bSb \mid aAb \mid bAa,\ A \to aA \mid bA \mid \epsilon$

d) G_4: $S \to SS \mid$ `<b>`S`</b>` $\mid$ `<i>`S`</i>` $\mid aS \mid bS \mid \ldots \mid zS \mid 0S \mid \ldots \mid 9S \mid \epsilon$

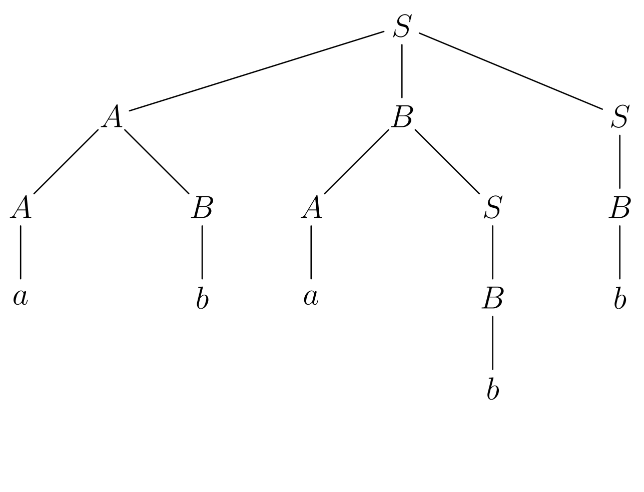
Aufgabe T19

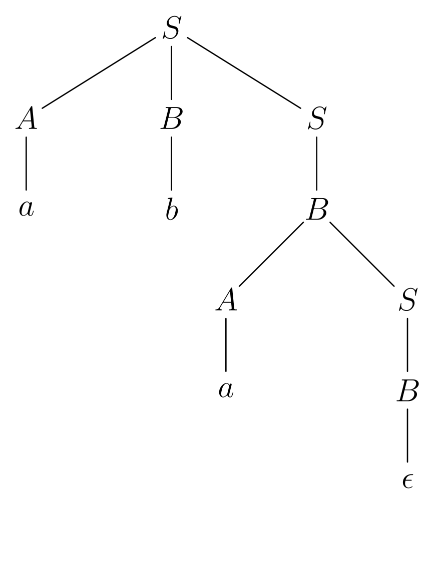
Erstellen Sie Ableitungsbäume und Linksableitungen für die Grammatik

$$S \to ABS \mid B, \qquad A \to AB \mid a, \qquad B \to AS \mid BS \mid b \mid \epsilon$$

und die Wörter aba und $ababb$. Ist diese Grammatik eindeutig?

Lösungsvorschlag:

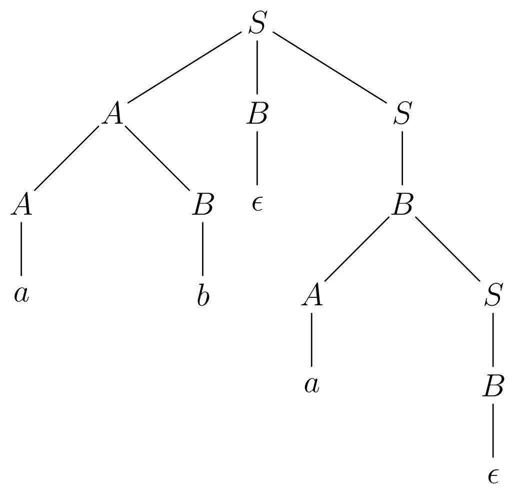




$S \Rightarrow ABS \Rightarrow aBS \Rightarrow abS \Rightarrow abB \Rightarrow abAS \Rightarrow abaS \Rightarrow abaB \Rightarrow aba$

$S \Rightarrow ABS \Rightarrow ABBS \Rightarrow aBBS \Rightarrow abBS \Rightarrow abASS \Rightarrow abaSS \Rightarrow abaBS \Rightarrow\Rightarrow ababS \Rightarrow ababB \Rightarrow ababb$.

Die Grammatik ist nicht eindeutig, da man *aba* auch so ableiten könnte:

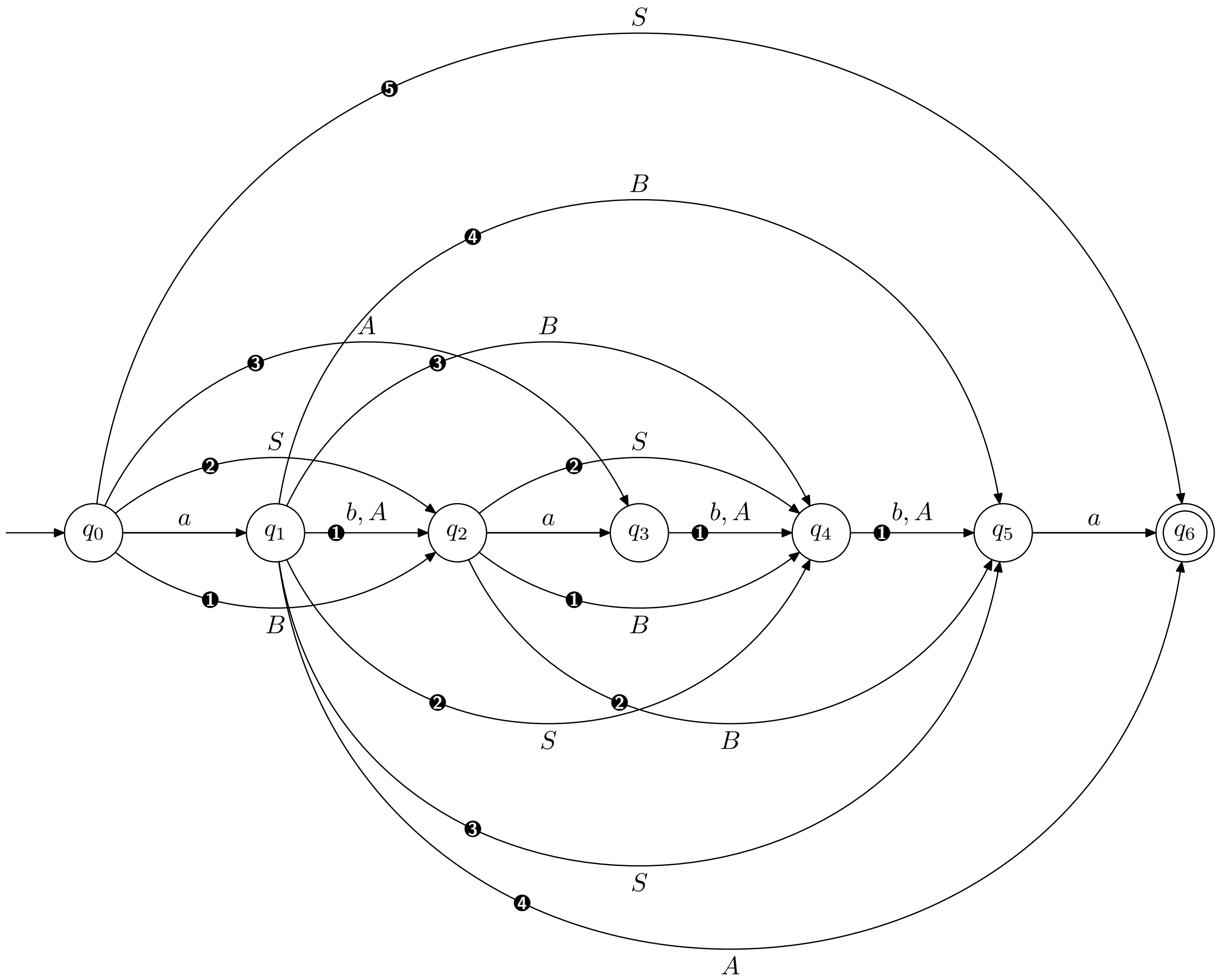


Aufgabe T20
Gegeben sei die Grammatik mit den Produktionen

$$S \to aA \mid bB \mid SAB, \quad A \to Sa \mid b, \quad B \to BA \mid AS \mid ab$$

Benutzen Sie einen saturierten NFA für eine geeignete Sprache L, um anhand von $pre^*_G(L)$ zu überprüfen, ob G das Wort *ababba* erzeugen kann.
Bilden Sie zwei Gruppen, um zu untersuchen, ob die folgenden α auf *ababba* ableitbar sind. Die erste Gruppe soll dies anhand der Grammatik G selbst, die zweite anhand von $pre^*_G(L)$ erledigen: (a) $\alpha = aSA$, (b) $\alpha = BBa$, (c) $\alpha = bBAS$, (d) $\alpha = aA$, (e) $\alpha = B$, (f) $\alpha = Ba$, (g) $\alpha = BA$, (h) $\alpha = AS$

Lösungsvorschlag:
Es gilt G kann *ababba* erzeugen genau dann, wenn $S \in pre^*_G(\{ababba\})$. Wir nehmen $L = \{ababba\}$ und den sich einfach ergebenden NFA mit sieben Zuständen, der L erkennt. Dann konstruieren wir den Automaten für $pre^*_G(L)$ mit Hilfe des Algorithmus aus der Vorlesung. Die Sättigung tritt nach fünf Durchgängen ein.
In der letzten Sättigungsrunde ergibt sich ein mit S beschriebener Übergang vom Start- in den Endzustand. Das bedeutet, daß das Startsymbol S in $pre^*_G(\{ababba\})$ enthalten ist. Folglich kann G auch das Wort *ababba* erzeugen. Die acht Nebenaufgaben sind dann schnell erledigt. α ist auf *ababba* ableitbar genau dann wenn es von dem Automaten erkannt wird. (a) nein, (b) ja, (c) nein, (d) ja, (e) nein, (f) nein, (g) nein, (h) nein.


S
5
B
4
A
B
3
3
S
S
2
2
a
b, A
a
b, A
b, A
a
q_0
q_1
q_2
q_3
q_4
q_5
q_6
1
1
1
1
1
B
B
2
2
S
B
3
S
4
A