**Formale Systeme, Automaten, Prozesse, SS 2020**
Prof. Dr. P. Rossmanith
J. Dreier, T. Hartmann, H. Lotze, D. Mock

RWTH AACHEN UNIVERSITY

Datum: 18.05.2020

# Übungsblatt mit Lösungen 06

## 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 {$a, \ldots, Z, 0, \ldots, 9$, `<b>`, `</b>`, `<i>`, `</i>`} 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>` $\mid aS \mid bS \mid \ldots \mid zS \mid 0S \mid \ldots \mid 9S \mid \epsilon$

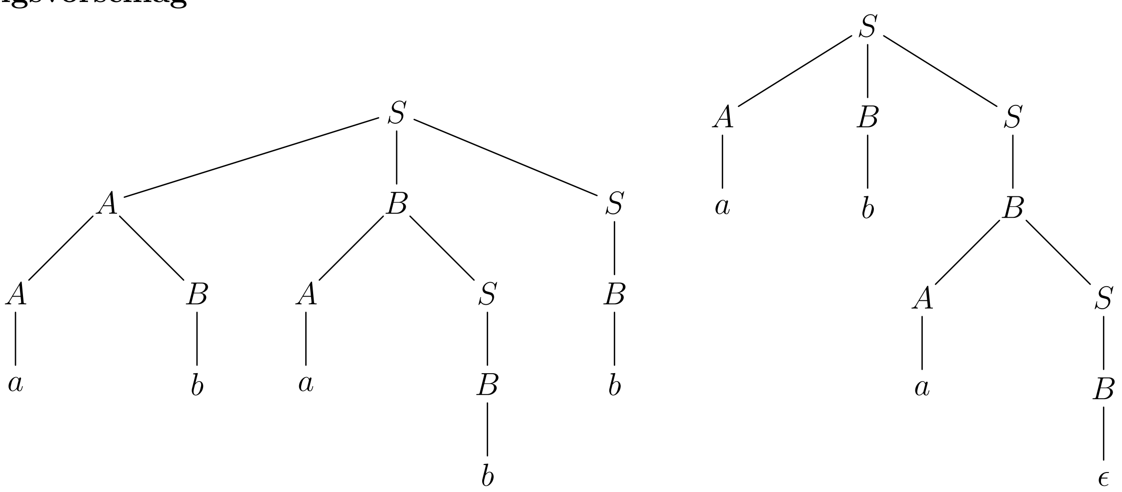
## Aufgabe T19

Erstellen Sie Ableitungsbäume und Linksableitungen für die Grammatik

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

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

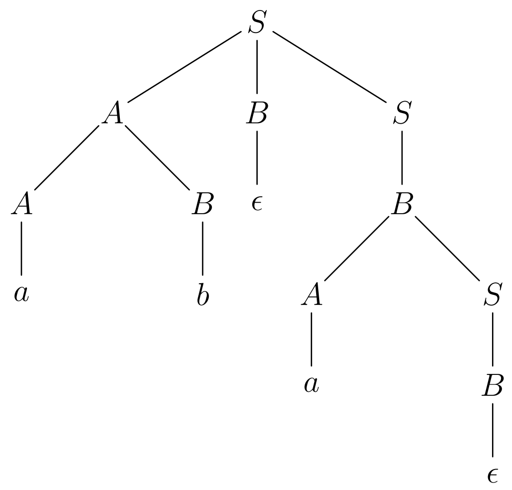
### Lösungsvorschlag



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

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

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

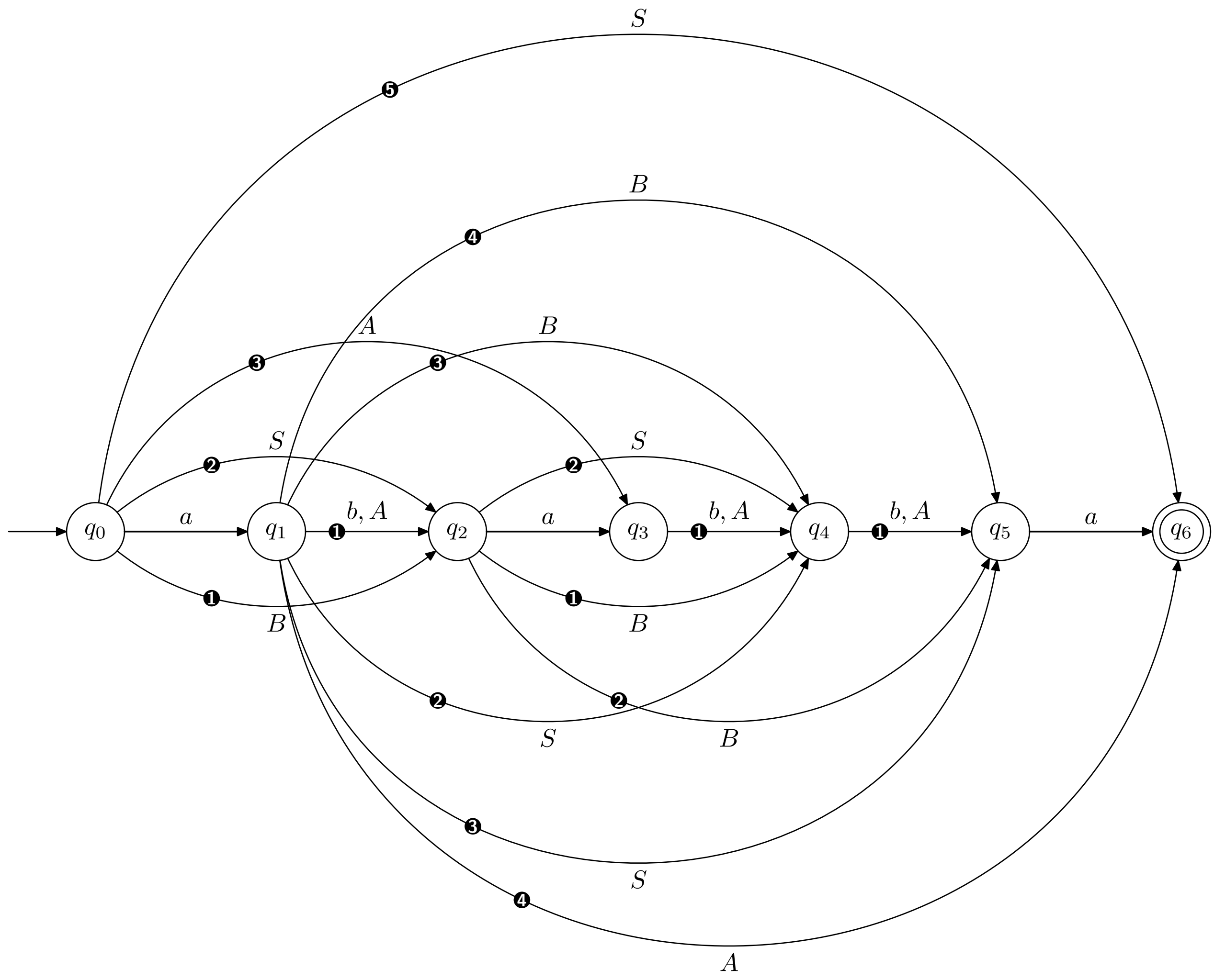



**Aufgabe T20**
Gegeben sei die Grammatik mit den Produktionen

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

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 $\alpha$ 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. $\alpha$ 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.

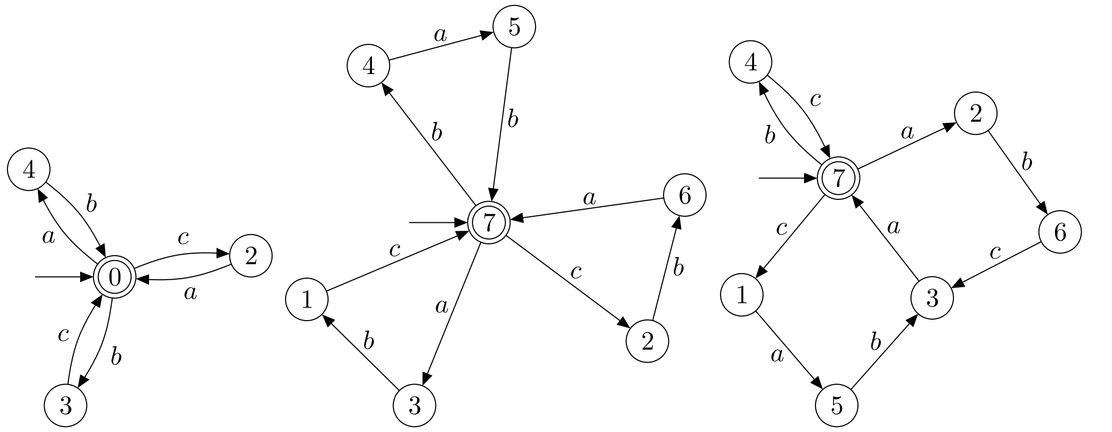



**Aufgabe H16** (10 Punkte)
Konstruieren Sie einen minimalen NFA für die Sprache

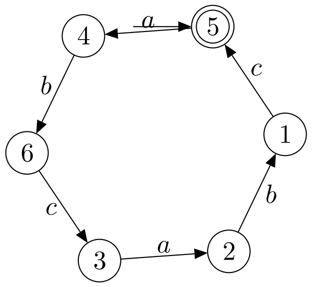
$$L((ab + bc + ca)^*) \cap L((abc + bab + cba)^*) \cap L((abca + bc + caba)^*)$$

und geben Sie das Ergebnis auch als einen regulären Ausdruck an.

**Lösungsvorschlag**
Zunächst konstruieren wir drei NFAs für die drei regulären Ausdrücke.



Dann berechnen wir den Produktautomaten und minimieren. Wenn wir den Fangzustand weglassen, bekommen wir folgenden NFA:




Der reguläre Ausdruck ist $(abcabc)^*$, wie man jetzt leicht sieht.

**Aufgabe H17** (10 Punkte)

Sei $G$ die folgende kontextfreie Grammatik:

$$\begin{array}{lcl} S & \to & BDA \mid DAC \mid DC \\ A & \to & SA \mid aA \mid a \mid d \mid b \\ B & \to & \epsilon \mid Sc \mid cD \mid AD \\ C & \to & aD \mid dA \mid bC \mid a \mid d \mid b \\ D & \to & BS \mid cC \mid aA \end{array}$$

a) Berechnen Sie die unerreichbaren Symbole von $G$. Die unerreichbaren Symbole sind definiert als $\{ A \in N \mid \text{es gibt kein } \alpha A \beta \in (N \cup T)^* \text{ mit } S \overset{*}{\Rightarrow} \alpha A \beta \}$.

b) Die unproduktiven Symbole einer Grammatik sind definiert als:

$$\{ A \in N \mid \text{es gibt kein } w \in T^* \text{ mit } A \overset{*}{\Rightarrow} w \}$$

Wie lauten die unproduktiven Symbole von $G$?

c) Gilt $L(G) = \emptyset$?

d) Gilt $\epsilon \in L(G)$?

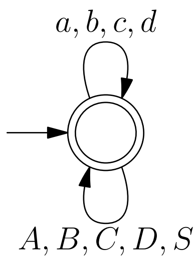
e) Die nullierbaren Symbole einer Sprache sind alle Symbole, die sich nach $\epsilon$ ableiten lassen. Wie lauten die nullierbaren Symbole von $G$?

f) Geben Sie eine kontextfreie Grammatik $G'$ an, so dass $L(G') = L(G)$. Hierbei darf $G'$ keine unproduktiven, unerreichbaren oder nullierbaren Symbole enthalten.

**Lösungsvorschlag**

a) Durch die beiden Regeln $S \Rightarrow BDA$ und $S \Rightarrow DAC$ sind schon alle Nichtterminale erreichbar.

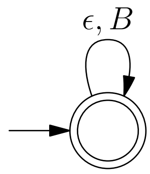
b) Die unproduktiven Symbole sind $N \setminus pre^*(T^*)$. Der Automat für $pre^*(T^*)$ lautet:




Für jedes Nichtterminal gibt es eine Satzform, die jene enthält und vom Automaten erkannt wird. Damit ist jedes Nichtterminal produktiv.

c) Da $S$ produktiv ist, kann von $S$ ein $w \in T^*$ abgeleitet werden. Somit ist die von $G$ erzeugte Sprache nicht leer.

d) Es gilt $\epsilon \in L(G)$ genau dann wenn $S \in pre^*(\{\epsilon\})$. Der Automat für $pre^*(\{\epsilon\})$ lautet:




Es gilt $pre^*(\{\epsilon\}) = B^*$. Somit enthält $L(G)$ nicht das leere Wort.

e) Die nullierbaren Symbole der Grammatik sind genau die Nichtterminale in $pre^*(\{\epsilon\})$. Diese Menge haben wir bereits in Teilaufgabe d) bestimmt, d.h. nur $B$ läßt sich nach $\epsilon$ ableiten.

f) Regeln mit unerreichbaren oder unproduktiven Symbolen dürfen direkt gelöscht werden, da sie, in einer Ableitung vom Startsymbol aus, entweder nicht verwendet werden können (unerreichbar) oder bei Verwendung eine Ableitung eines Wortes aus Terminalsymbolen unmöglich machen (unproduktiv). Nullierbare Symbole können wie folgt behandelt werden:

1) Sei $B \to \epsilon \in G$ und $B \neq S$.
2) Lösche $B \to \epsilon$ aus $G$.
3) Für jede Regel $D \to \alpha B \beta$ füge die Regel $D \to \alpha\beta$ in $G$ ein.
4) Falls keine Regel der Form $A \to \epsilon$ mehr existiert fertig, ansonsten gehe zu 1.

Somit hat die resultierende Grammatik keine Regeln der Form $B \to \epsilon$ für $B \neq S$. Die Regel $S \to \epsilon$ gehört genau dann zur Grammatik, wenn $\epsilon \in L(G)$ ist.

Für die Grammatik $G$ ergibt sich also folgende Grammatik $G'$:

$$
\begin{array}{lcl}
S & \to & BDA \mid DA \mid DAC \mid DC \\
A & \to & SA \mid aA \mid a \mid d \mid b \\
B & \to & Sc \mid cD \mid AD \\
C & \to & aD \mid dA \mid bC \mid a \mid d \mid b \\
D & \to & BS \mid S \mid cC \mid aA
\end{array}
$$

**Aufgabe H18** (10 Punkte)
Die Menge aller korrekten aussagenlogischen Formeln werden über dem Terminalalphabet $\Sigma = \{(,), \wedge, \vee, \neg, 0, 1\}$ gebildet. Ihre Aufgaben sind die folgenden.

1. Geben Sie eine kontextfreie Grammatik an, deren Sprache genau alle korrekt gebildeten aussagenlogischen Formeln enthält.

2. Modifizieren Sie die Grammatik aus der obigen Teilaufgabe so, dass Sie nur noch jene Formeln enthält, deren Wahrheitswert 1 ist.

3. Geben Sie auf der Basis ihrer in Teilaufgabe 2 gebildeten Grammatik Ableitungsbäume für folgende aussagenlogischen Formeln an: $\neg 0 \wedge 1$, $\neg(\neg 1 \wedge 0)$

Erklären Sie ausführlich, wie Ihre Grammatik aus Teilaufgabe 2 funktioniert. Testen Sie ihre Grammatik aus Aufgabenteil 2 eigenständig auf weiteren aussagenlogischen Formeln, insbesondere solchen, die zu 0 evaluiert werden müssten.
Achten Sie insbesondere auf die Präzedenzen in aussagenlogischen Formeln: $\neg$ bindet am stärksten, gefolgt von $\wedge$ und $\vee$.

**Lösungsvorschlag**

Folgende Grammatik erzeugt alle syntaktisch korrekten aussagenlogischen Formeln:

$$S \to (S) \mid 0 \mid 1 \mid \neg S \mid S \wedge S \mid S \vee S$$

Für alle aussagenlogischen Formeln, die zu 1 evaluiert werden, betrachten wir die Grammatik $G = (\{D, D^+, D^-, K, K^+, K^-, P, P^+, P^-\}, \{(,), \wedge, \vee, \neg, 0, 1\}, P, D^+)$, mit folgenden Produktionsregeln $P$:

$$\begin{aligned}
&D^+ \to K^+ \mid K^+ \vee D \mid K \vee D^+\\
&K^+ \to P^+ \mid P^+ \wedge K^+\\
&P^+ \to 1 \mid (D^+) \mid \neg P^-\\
&P^- \to 0 \mid (D^-) \mid \neg P^+\\
&D^- \to K^- \mid K^- \vee D^-\\
&K^- \to P^- \mid P^- \wedge K \mid P \wedge K^-\\
&D \to D^+ \mid D^-\\
&K \to K^+ \mid K^-\\
&P \to P^+ \mid P^-
\end{aligned}$$

Zuerst die Benennung der Nichtterminale: **D**isjunktion, **K**onjunktion, **P**rimitiv, wobei das $+$ und $-$ angeben, dass es sich um einen Teilausdruck handelt, der zu 1 oder 0 evaluiert. D.h. die Grammatik wäre insbesondere in der Lage auch solche Formeln abzuleiten, die zu 0 evaluiert werden, wenn wir als Startsymbol $D^-$ wählen.

Die einzelnen Ableitungsregeln ergeben sich aus einfachen Überlegungen: Jede Aussagenlogische Formel ist auf der höchsten Abstraktionsebene eine Disjunktion, da diese am schwächsten bindet. Wir wählen solche, die zu einer 1 evaluieren, als Startsymbol.

$D^+$: Eine Disjunktion die zu 1 evaluiert ist entweder eine Konjunktion die zu 1 evaluiert oder eine Disjunktion zweier Disjunktionen, von denen mindestens eine oder beide zu 1 evaluieren.

$K^+$: Eine Konjunktion, die zu 1 evaluiert ist entweder ein Primitiv, das zu 1 evaluiert oder eine Konjunktion zweier Konjunktionen, die beide zu 1 evaluieren.

$P^+$: Ein Primitiv, das zu 1 evaluiert wird ist entweder bereits eine 1, eine Klammerung einer zu 1 evaluierenden Disjunktion oder die Negation eines Primitivs, das zu 0 evaluiert.

$P^-$: Ein Primitiv, das zu 0 evaluiert wird ist entweder bereits eine 0, eine Klammerung einer zu 0 evaluierenden Disjunktion oder die Negation eines Primitivs, das zu 1 evaluiert.

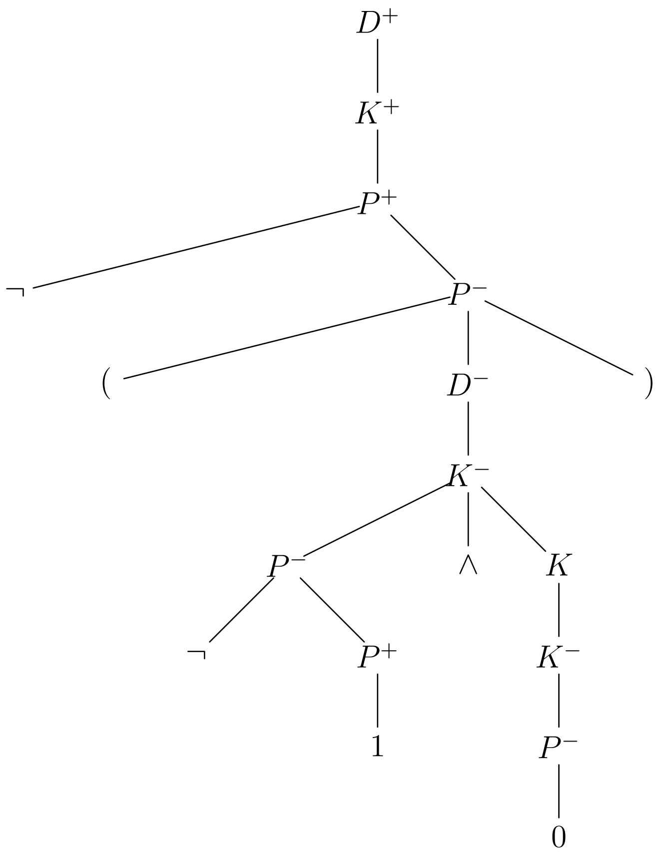
$D^-$: Eine Disjunktion, die zu 0 evaluiert wird ist entweder eine Konjunktion, die zu 0 evaluiert oder eine Disjunktion zweier Disjunktionen, die beide zu 0 evaluieren.

$K^-$: Eine Konjunktion, die zu 0 evaluiert ist entweder ein Primitiv, das zu 0 evaluiert, oder eine Konjunktion zweier Konjunktionen, bei der eine der Konjunktionen zu 0 evaluiert.

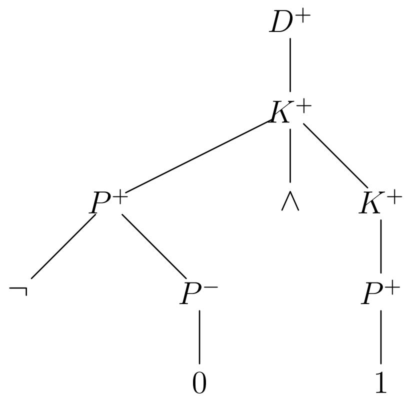
$D, K, P$: Jede Disjunktion ist entweder eine solche die zu 1 oder zu 0 evaluiert, gleiches gilt für Konjunktionen und Primitive.

Anhand dieser Beschreibungen sollte deutlich sein, dass wir von $D^+$ nur Formeln ableiten können, die zu 1 evaluieren.

Folgend noch die Ableitungsbäume für die beiden Formeln:



$D^+$
$K^+$
$P^+$
$\neg$
$($
$P^-$
$D^-$
$)$
$K^-$
$P^-$
$\wedge$
$K$
$\neg$
$P^+$
$K^-$
$1$
$P^-$
$0$

$D^+$
$K^+$
$P^+$
$\wedge$
$K^+$
$\neg$
$P^-$
$P^+$
$0$
$1$