Aufgabe 7.1 [Deterministisch kontextfreie Sprachen] **4 Punkte**

Entscheiden Sie für die folgenden Sprachen jeweils,

1. ob sie deterministisch kontextfrei sind; und
2. ob sie von einem DPDA, der mit leerem Keller akzeptiert, entschieden werden können.

Begründen Sie Ihre Entscheidung stichhaltig (gegebenenfalls durch Angabe geeigneter Automaten).

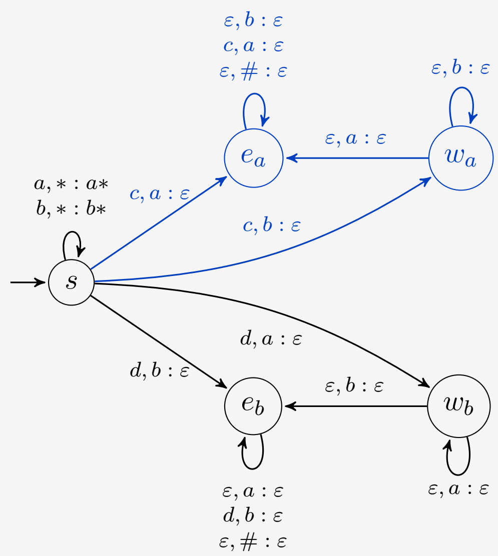
a) $L = \{wc^n \mid n \geq 1, w \in \{a,b\}^* \text{ mit } \#_a(w) = n\} \cup \{wd^n \mid n \geq 1, w \in \{a,b\}^* \text{ mit } \#_b(w) = n\}$

b) $L = \{a^k b^\ell c^m \mid k \neq m \text{ oder } \ell \neq m\}$

Lösung:

a) Die Sprache kann von einem DPDA, der mit leerem Keller akzeptiert, entschieden werden. Sie ist insbesondere deterministisch kontextfrei.

Der folgende DPDA akzeptiert mit leerem Keller und entscheidet die Sprache.



Der Automat arbeitet nach dem folgenden Prinzip: In Zustand s wird jedes gelesene a oder b auf dem Keller gespeichert, sodass die entsprechende Anzahl später für einen Vergleich mit der Anzahl von c oder d zur Verfügung steht.

Sobald das erste c oder d gelesen wurde (und mindestens ein solches Zeichen muss nach Definition der Sprache L auftreten), steht fest, ob die Anzahl von Vorkommen von a oder die Anzahl von Vorkommen von b verglichen werden muss. Wir betrachten den Fall, dass ein c gelesen wurde. Der andere Fall verläuft analog.

Nach Lesen des ersten Vorkommens von c tritt der Automat in eine zweite „Phase“ ein. In dieser Phase wird nach einem „ausgleichenden“ Vorkommen von a auf dem Keller für jedes Vorkommen von c in der Eingabe gesucht. Insbesondere wird dabei jedes b vom Keller gelöscht ohne weitere Eingabezeichen zu lesen, da die Anzahl der zuvor gelesenen Vorkommen von b für den Vergleich unerheblich ist (sowohl in Zustand e_a als auch in Zustand w_a).

Ursächlich für die **Unterscheidung der Zustände e_a und w_a** ist die Beobachtung, dass das erste Vorkommen von c *entweder* direkt durch ein a als oberstes Kellersymbol ausgeglichen werden kann *oder* erst durch ein weiter unten liegendes a.

Der Automat geht in den Zustand e_a über, sobald das erste c des Eingabewortes durch ein a auf dem Keller „ausgeglichen“ wurde. Bei einem Eingabewort wie $w = bac$ ist das unmittelbar der Fall, wie man an der Konfigurationenfolge

$$(s, bac, \#) \vdash \dots \vdash (s, c, ab\#) \vdash (e_a, \varepsilon, b\#) \vdash \dots$$

entnehmen kann. Bei anderen Wörtern, beispielsweise bei $w' = abbc$ jedoch müssen zunächst die Vorkommen von b vom Keller entfernt werden; der „Ausgleich“ findet später statt. Dazu geht der Automat in den „Wartezustand“ w_a über. Hier ergibt sich eine Konfigurationenfolge der Form

$$(s, abbc, \#) \vdash \dots \vdash (s, c, bba\#) \vdash (w_a, \varepsilon, ba\#) \vdash (w_a, \varepsilon, a\#) \vdash (e_a, \varepsilon, \#).$$

Diese Unterscheidung ist nur für das erste gelesene c nötig, um die von s ausgehenden Transitionen deterministisch konstruieren zu können. Das unterste Kellersymbol kann dann nur in Zustand e_a entfernt werden, das heißt, nachdem das erste c durch ein Vorkommen von a ausgeglichen wurde und außerdem jedes weitere Vorkommen von c durch ein a auf dem Keller ausgeglichen wurde (Transition $(e_a, c, a, e_a, \varepsilon)$).

b) **Die Sprache ist *nicht* determinisitsch kontextfrei. Insbesondere gibt es keinen DPDA, der sie entscheidet – unabhängig vom Akzeptanzmodell.**

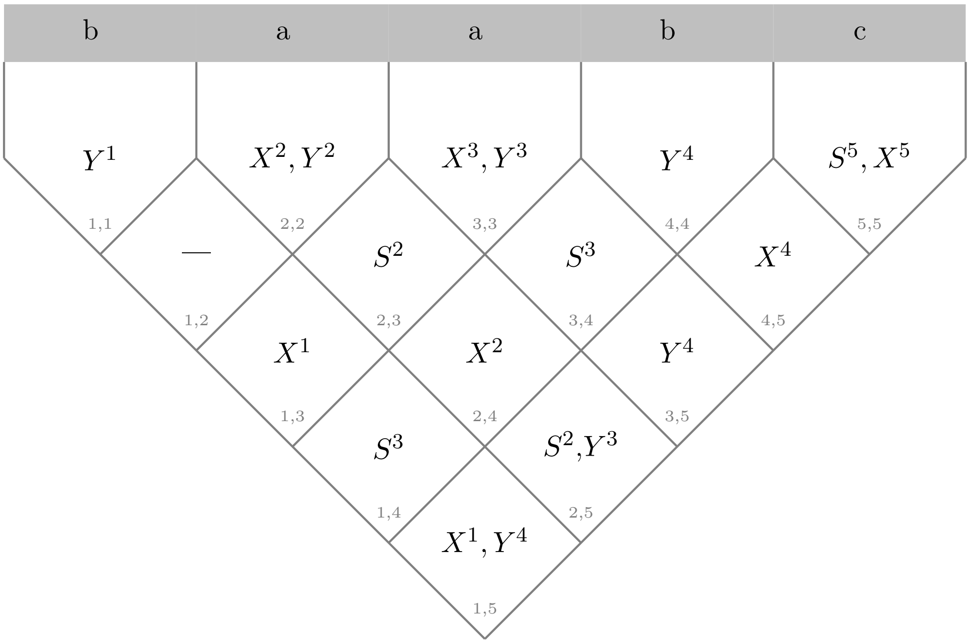
Die Behauptung ergibt sich aus folgender Argumentation unter Ausnutzung von Abschlusseigenschaften. Für einen Widerspruchsbeweis nehmen wir an, dass L deterministisch kontextfrei ist. Da die Klasse der deterministisch kontextfreien Sprachen unter Komplementbildung abgeschlossen ist, ist dann auch die Sprache $\overline{L} = \{a, b, c\}^* - L$ *deterministisch kontextfrei* und somit insbesondere *kontextfrei* (da L eine Sprache über dem Alphabet $\{a, b, c\}$ ist).[a]

Die Sprache $\overline{L}$ enthält alle Wörter, die von der Form $a^k b^k c^k$ sind oder aber *nicht* von der Form der Wörter, die durch den regulären Ausdruck $a^* b^* c^*$ beschrieben werden. Da die Klasse der kontextfreien Sprachen unter Schnittbildung mit regulären Sprachen abgeschlossen ist, ist dann auch die Sprache $\overline{L} \cap L(aa^* bb^* cc^*) = \{a^k b^k c^k \mid k \in \mathbb{N}\}$ kontextfrei – im Widerspruch zur Vorlesung. Demnach ist die Annahme, dass L deterministisch kontextfrei ist, falsch.

[a] Um einen Widerspruch herzuleiten, genügt es nun zu zeigen, dass die Sprache $\overline{L}$ *nicht* kontextfrei ist. Dies könnte beispielsweise durch Anwendung des Pumping-Lemmas gezeigt werden, die Argumentation hier greift jedoch abermals eine Abschlusseigenschaft auf.

Aufgabe 7.2 [Das Wortproblem für kontextfreie Grammatiken] 5 Punkte

a) Gegeben sei das folgende vollständige Tableau, welches der erweiterte CYK-Algorithmus für das Wort $w = baabc$ und die Grammatik G berechnet hat.



Beurteilen Sie für jedes der drei Wörter $w_1 = baabc$, $w_2 = baab$ und $w_3 = aabc$ anhand des Tableaus, ob sie in der Sprache $L(G)$ enthalten sind. Bestimmen Sie für jedes Wort w_i, für $i \in \{1, 2, 3\}$, das in der Sprache enthalten ist, anhand des Tableaus einen Ableitungsbaum. Setzen Sie dazu voraus, dass die Grammatik G genau die Regeln

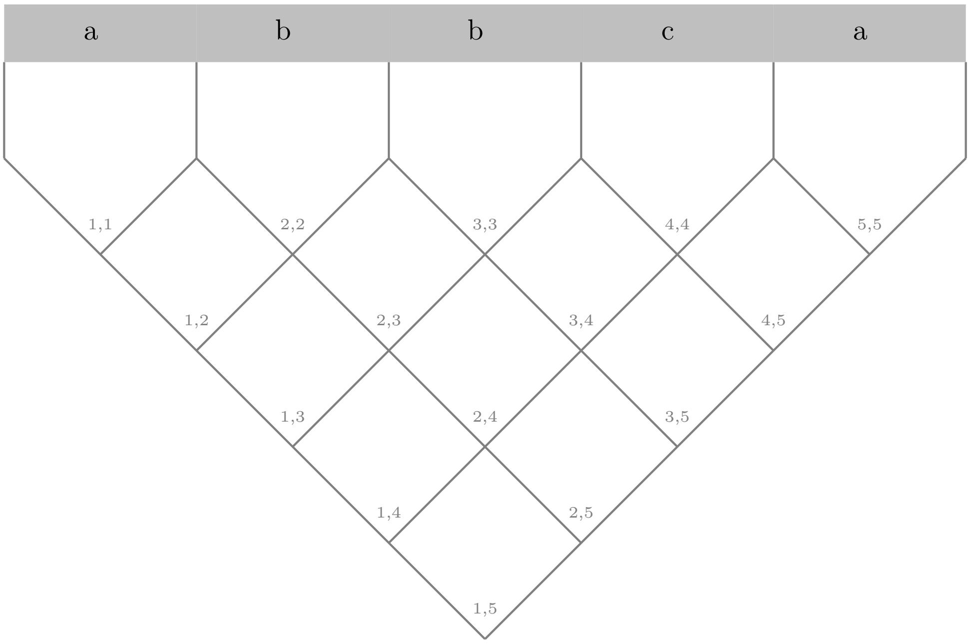
$$\begin{array}{lcl} S & \to & c \mid XY \\ X & \to & a \mid c \mid YS \\ Y & \to & a \mid b \mid SX \end{array}$$

enthält. **(2 Punkte)**

b) Geben Sie das vollständige Tableau (inklusive Mehrfachvorkommen von Variablen) für den erweiterten CYK-Algorithmus für das Wort $w = abbca$ und die folgende Grammatik an.

$$\begin{array}{lcl} S & \to & AB \mid BB \\ A & \to & a \mid SB \mid SC \\ B & \to & b \mid CA \\ C & \to & a \mid c \mid CB \end{array}$$

(3 Punkte)



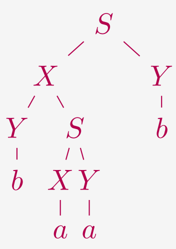
Lösung:

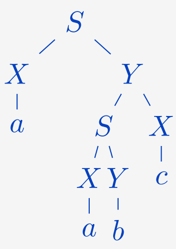
a) Nach Definition gilt für eine kontextfreie Grammatik G mit Startsymbol S, dass ein Wort w genau dann in $L(G)$ liegt, wenn es von S abgeleitet werden kann, $S \Rightarrow^* w$.

Entsprechend gilt für die vorgegeben Wörter

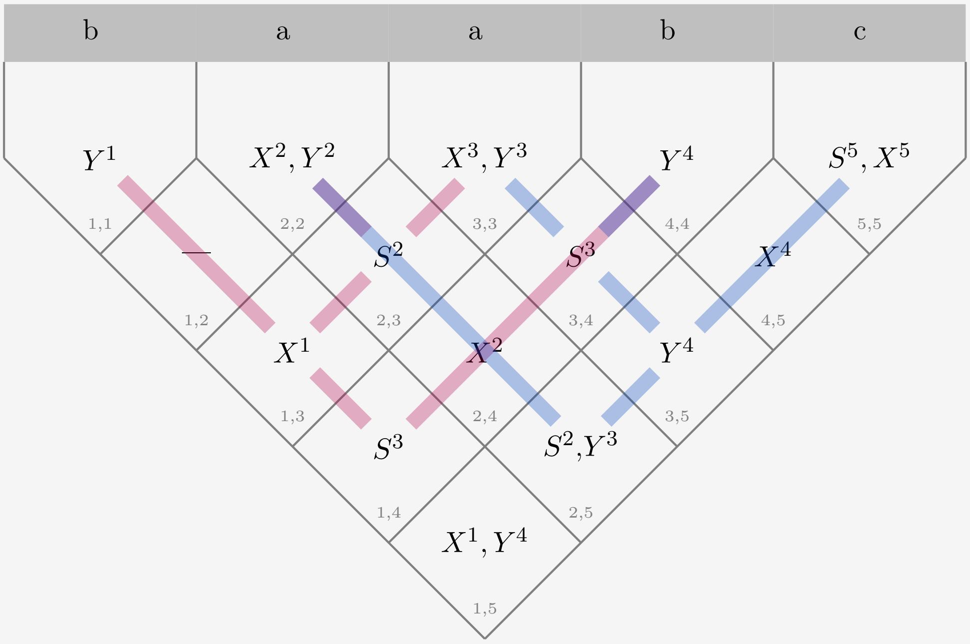
- $baabc = w[1,5] \notin L(G)$, da kein k mit $S_k \in V_{1,5}$ existiert;
- $baab = w[1,4] \in L(G)$, da ein k mit $S_k \in V_{1,4}$ existiert ($k = 3$);
- $aabc = w[2,5] \in L(G)$, da ein k mit $S_k \in V_{2,5}$ existiert ($k = 2$).

Ableitungsbäume für *baab* und *aabc* sind nachfolgend dargestellt:

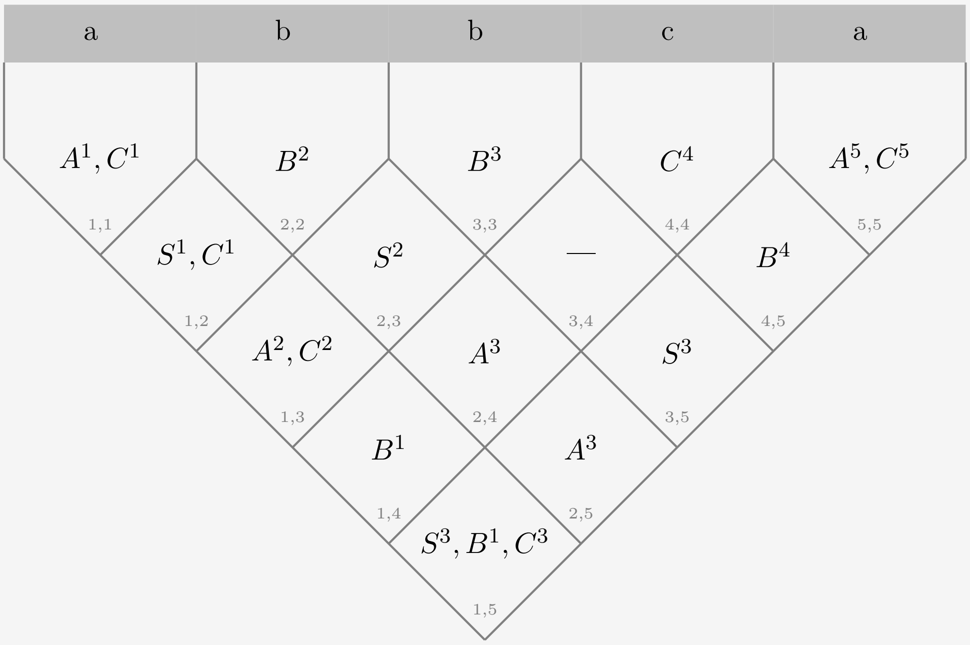




Diese Ableitungsbäume finden sich im Tableau wieder.



b) Für die angegebene Grammatik und das Wort *abbca* ergibt sich folgendes Tableau.



Aufgabe 7.3 [LL(1)- und LR(1)-Grammatiken] 6 Punkte

a) Beurteilen Sie für jede der folgenden drei Grammatiken, ob es sich um eine LL(1)-Grammatik handelt, und begründen Sie Ihr Urteil. **(5 Punkte)**

Grammatik G_1:

$$\begin{aligned} S &\to ab \mid eAS \mid fAC \\ A &\to ea \mid CB \\ B &\to \varepsilon \mid bcB \\ C &\to \varepsilon \mid d \end{aligned}$$

Grammatik G_2:

$$\begin{aligned} S &\to acB \mid ccB \mid bC \\ B &\to \varepsilon \mid bC \\ C &\to aB \end{aligned}$$

Grammatik G_3:

$$\begin{aligned} S &\to bA \mid BCA \\ A &\to a \mid b \\ B &\to \varepsilon \mid cS \\ C &\to \varepsilon \mid dC \end{aligned}$$

b) Ist die folgende Grammatik G_4 eine LR(1)-Grammatik? Begründen Sie Ihr Urteil.

$$\begin{aligned} S &\to aA \mid abBE \\ A &\to bCa \\ B &\to C \\ C &\to cc \\ E &\to ab \end{aligned}$$

(1 Punkt)

Lösung:

a) **Grammatik G_1 ist keine LL(1)-Grammatik.** Die Grammatik verletzt Bedingung ii): $\mathsf{FOLLOW(A)} \cap \mathsf{FIRST(ea)} = \{e\} \neq \emptyset$.

Diese Bedingung müsste jedoch erfüllt sein, da für die Variable $X = A$ zwei Ersetzungen $\alpha = CB$ und $\beta = ba$ existieren, sodass $\alpha \Rightarrow CB \Rightarrow B \Rightarrow \varepsilon$ gilt. Ferner belegen die Ableitungen $S \Rightarrow eAS$ und $S \Rightarrow fAC$, dass die Variablen S und C unmittelbar nach der Variablen A stehen können. Dementsprechend gilt

$$\mathsf{FOLLOW(A)} = (\mathsf{FIRST(S)} \cup \mathsf{FIRST(C)}) - \{\varepsilon\} = \{a, e, f, d\}.$$

Grammatik G_2 ist eine LL(1)-Grammatik. Wir zeigen für jede Variable, dass sie die Bedingungen i) und ii) erfüllt:

- **Variable S:**
  - Es gelten
    1. $\mathsf{FIRST(acB)} \cap \mathsf{FIRST(ccB)} = \{a\} \cap \{c\} = \emptyset$,
    2. $\mathsf{FIRST(acB)} \cap \mathsf{FIRST(bC)} = \{a\} \cap \{b\} = \emptyset$ und
    3. $\mathsf{FIRST(ccB)} \cap \mathsf{FIRST(bC)} = \{c\} \cap \{b\} = \emptyset$.

    Mithin gilt für alle Kombination von Ersetzungen α und β für die Variable S, dass $\mathsf{FIRST}(\alpha) \cap \mathsf{FIRST}(\beta) = \emptyset$ ist.
  - Da $S \not\Rightarrow^* \varepsilon$ gilt, ist Bedingung ii) trivialerweise erfüllt.
- **Variable B:**
  - Bedingung i) ist erfüllt, da $\mathsf{FIRST}(\varepsilon) \cap \mathsf{FIRST(bC)} = \{\varepsilon\} \cap \{b\} = \emptyset$ ist.
  - Die Grammatik ist rechtslinear. Deshalb enthält jede Satzform einer Ableitung höchstens eine Variable, welche dann am Ende der Satzform steht. Folglich ist $\mathsf{FOLLOW(B)} = \emptyset$ und mithin $\mathsf{FOLLOW(B)} \cap \mathsf{FIRST}(\beta) = \emptyset \cap \mathsf{FIRST}(\beta) = \emptyset$ für

alle rechten Regelseiten β.

- **Variable C:** Beide Bedingungen sind trivialerweise erfüllt, da es keine zwei unterschiedlichen Ersetzungen gibt.

Grammatik G_3 ist keine LL(1)-Grammatik. Die Regeln $S \to bA$ und $S \to BCA$ verstoßen gegen Bedingung i): Wegen $B \Rightarrow^* \varepsilon$ und $C \Rightarrow^* \varepsilon$ ist dann $\mathsf{FIRST}(\mathsf{BCA})$ die Menge der ersten Zeichen von

$$\mathsf{FIRST}(\mathsf{B}) \circ \mathsf{FIRST}(\mathsf{C}) \circ \mathsf{FIRST}(\mathsf{A}) = \{\varepsilon, c\} \circ \{\varepsilon, d\} \circ \{a, b\} = \{a, b, ca, cb, cda, cdb, da, db\}$$

und somit $\mathsf{FIRST}(\mathsf{BCA}) \cap \mathsf{FIRST}(\mathsf{bA}) = \{a, b, c, d\} \cap \{b\} \neq \emptyset$.

b) Grammatik G_4 ist keine LR(1)-Grammatik. Die folgenden Rechtsableitungen beweisen, dass die Grammatik Eigenschaft 1) verletzt. Die Ableitung

$$S \Rightarrow_r abBE \Rightarrow_r abBab \Rightarrow_r abCab$$

ist von der Form $S \Rightarrow_r^* abBab \Rightarrow_r abCab = \alpha\beta\sigma x$ für $\alpha = ab$, $\beta = C$, $\sigma = a$ und $x = b$.

Die Ableitung

$$S \Rightarrow_r aA \Rightarrow_r abCa$$

belegt die Existenz einer Satzform $\gamma = aA$, aus der sich mit einem Rechtsableitungsschritt die Satzform $\alpha\beta\sigma y = abCa$, für $y = \varepsilon$, gewinnen lässt. Da γ aus S abgeleitet werden kann, ist Eigenschaft 1) nicht erfüllt.