2. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2016/2017

Lösung!

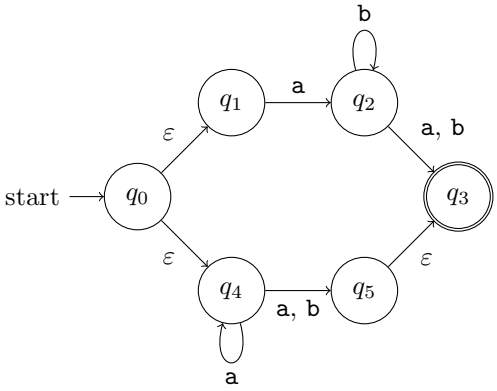
Beachten Sie:

- Bringen Sie den Aufkleber mit Ihrem Namen und Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte						Erreichte Punkte					
	a	b	c	d	e	Σ	a	b	c	d	e	Σ
1	2	3	3	–	–	8				–	–	
2	4	–	–	–	–	4		–	–	–	–	
3	7	2	3	2	–	14					–	
4	6	–	–	–	–	6		–	–	–	–	
5	2	3	4	3	3	15						
6	1	2	3	–	–	6				–	–	
7	7×1					7						
Σ						60						

Problem 1: Automatenkonstruktion $2 + 3 + 3 = 7$

(a) Sei folgender nichtdeterministischer endlicher Automat $\mathcal{A}$ gegeben:



Dabei führen alle nicht explizit angegebenen Übergänge in einen impliziten Fehlerzustand. Geben Sie einen regulären Ausdruck an, der die Sprache $L(\mathcal{A})$ erzeugt und dabei höchstens zweimal das Vereinigungssymbol $\cup$ enthält. *Lösung:*

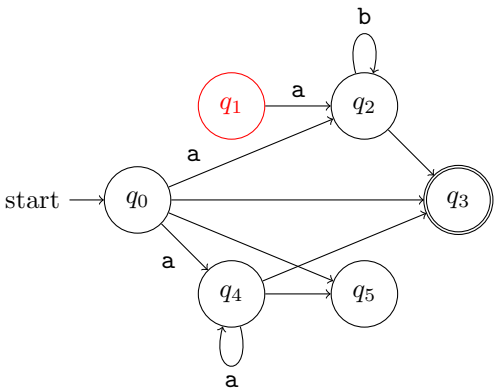
$$L(\mathcal{A}) = (\mathtt{ab}^* \cup \mathtt{a}^*)\,(\mathtt{a} \cup \mathtt{b})$$

(b) Geben Sie für jeden Zustand dessen ε-Abschluss an und geben Sie für $\mathcal{A}$ einen äquivalenten endlichen nichtdeterministischen Automaten $\mathcal{A}'$ an, der keinen ε-Übergang enthält. Der Automat $\mathcal{A}'$ soll dieselbe Zustandsmenge wie $\mathcal{A}$ haben. Markieren Sie alle überflüssigen Zustände.

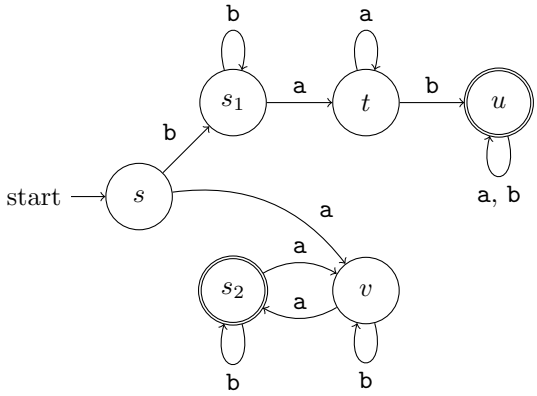
Lösung: Die ε-Abschlüsse für $\mathcal{A}$ lauten:

$$\begin{aligned}
E(q_0) &= \{q_0, q_1, q_4\} \\
E(q_1) &= \{q_1\} \\
E(q_2) &= \{q_2\} \\
E(q_3) &= \{q_3\} \\
E(q_4) &= \{q_4\} \\
E(q_5) &= \{q_5, q_3\}
\end{aligned}$$

Der zu $\mathcal{A}$ äquivalente NEA $\mathcal{A}'$ sieht wie folgt aus, wobei unbeschriftete Kanten dem Übergang $\{\mathtt{a}, \mathtt{b}\}$ entsprechen. Überflüssige Zustände sind rot markiert.



(c) Zeigen Sie mit einem der aus Vorlesung oder Übung bekannten Verfahren, dass der folgende deterministische endliche Automat minimal ist.



Lösung: Automatenminimierungsverfahren aus der Übung:

Zeichen	Partition
ε	$\{s_2, u\}$, $\{s, s_1, t, v\}$
a	$\{s_2\}$ $\{u\}$, $\{s, s_1, t\}$, $\{v\}$
a	$\{s_2\}$ $\{u\}$, $\{s\}$, $\{s_1, t\}$, $\{v\}$
b	$\{s_2\}$ $\{u\}$, $\{s\}$, $\{s_1\}$, $\{t\}$, $\{v\}$

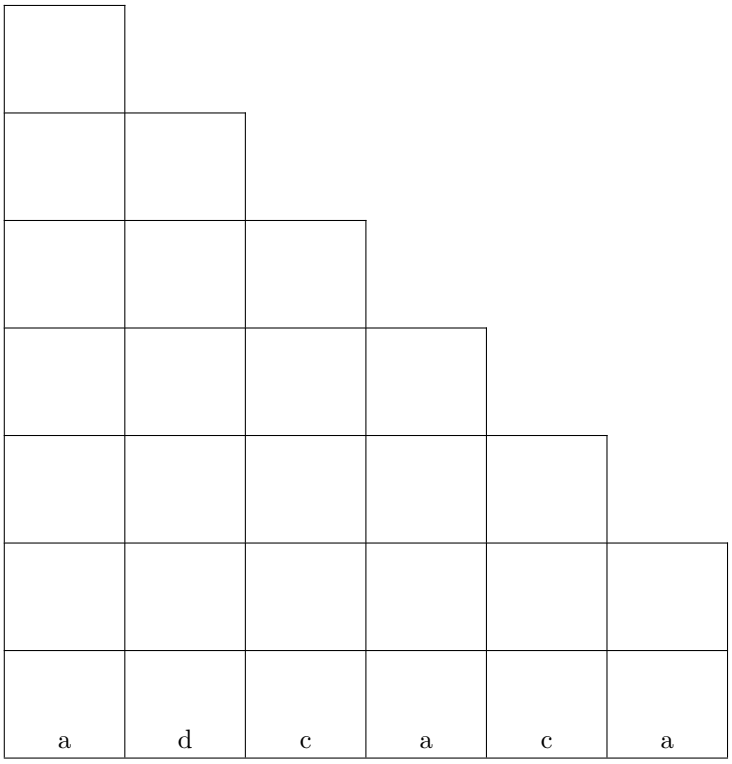
Der Automat kann also nicht weiter minimiert werden.

Problem 2: Kontextfreie Grammatiken 4

Gegeben ist folgende kontextfreie Grammatik $G = (\{a, b, c, d\}, \{A, B, C, D, E\}, A, R)$. Die Regelmenge R enthält folgende Regeln:

$$\begin{aligned} A &\to AB \mid EC \mid a \\ B &\to CA \\ C &\to CC \mid DC \mid c \\ D &\to d \\ E &\to CA \mid DA \end{aligned}$$

Überprüfen Sie mittels des aus der Vorlesung bekannten CYK-Algorithmus, ob das Wort *adcaca* in $L(G)$ enthalten ist.



Lösung:

A					
	B, E				
A	A,E	B, E			
	B,E	A	A		
	C	B, E		B, E	
A	D	C	A	C	A
a	d	c	a	c	a

Problem 3: $\mathcal{NP}$-Vollständigkeit $7 + 2 + 3 + 2 = 14$

(a) Sei M ein Spielbrett mit jeweils $m \in \mathbb{N}$ Spalten und Zeilen, die das Spielbrett in m^2 Zellen aufteilen. In jeder Zelle liegt entweder ein roter, ein blauer, oder kein Stein. Eine Belegung der Zellen durch Steine heißt Konfiguration. Eine Konfiguration ist *gewinnend*, wenn jede Zeile mindestens einen Stein enthält und jede Spalte nur Steine gleicher Farbe enthält. Betrachten Sie folgendes Problem.

Problem SOLITAIRE

Gegeben: Eine Konfiguration M.

Frage: Kann M durch Entfernen von Steinen zu einer gewinnenden Konfiguration M' transformiert werden?

Zeigen Sie, dass SOLITAIRE $\mathcal{NP}$-vollständig ist. Verwenden Sie dazu, dass das 3-SAT-Problem $\mathcal{NP}$-vollständig ist.

Lösung:

Solitaire $\in \mathcal{NP}$. Für jede Zeile und Spalte kann in $O(m)$ Zeit die gegebenen Eigenschaften überprüft werden. Somit kann in $O(n^2)$ Zeit überprüft werden, ob eine Spielekonfiguration M' eine gewinnende Spielkonfiguration ist. Zu dem kann in $O(n^2)$ Zeit überprüft werden, ob M' nur durch entfernen von Steinen von M aus erreicht werden kann (Falls eine Zelle $M'(i,j)$ einen roten (blauen) Stein enthält, dann enthält auch $M(i,j)$ einen roten (blauen) Stein.

Reduktion von 3-SAT. Sei (V, C) eine 3-SAT Instanz mit Variablen $V = \{x_1, x_2, \ldots, x_m\}$ und Klauseln $\{c_1, c_2, \ldots, c_k\}$. Sei M ein Spielbrett mit k Zeilen und m Spalten (die Anzahl der Spalten können später aufgefüllt werden). Platziere in Zelle $M(i,j)$ genau dann einen roten Stein, wenn das Literal x_j in c_i vorkommt. Platziere dort einen blauen Stein, wenn das Literal $\overline{x_j}$ in c_i vorkommt.

Korrekheit. Sei I eine erfüllende 3-SAT Instanz. Ist x_j wahr (falsch) entferne alle blauen (roten) Steine aus Spalte j. Da jede Klausel mindestens eine erfüllendes Literall enthält, enthält jede Zeile mindestens einen Stein.

Sei I' eine erfüllende SOLITAIRE Instanz. Enthält eine Spalte j nur rote Steine, dann wird x_j wahr zu geordnet. Enthält eine Spalte nur blaue Steine, dann wird der entsprechende Variable falsch zu geordnet. Da jede Zeile mindestens einen Stein enthält, enthält jede Klausel ein wahres Literal.

Doktor Meta forscht in seinem Labor am Geheimproblem X. Er hat bereits eine polynomielle Transformation von SAT auf X gefunden, die SAT-Instanzen der Größe n auf X-Instanzen der Größe n^3 abbildet. In einem Moment der Genialität gelingt es Doktor Meta nun, eine Laufzeitschranke für SAT von $\Omega(c^n)$ für eine Konstante $c > 1$ zu beweisen.

(b) Was folgt aus der Laufzeitschranke für SAT über die Frage $\mathsf{P} \stackrel{?}{=} \mathcal{NP}$? Begründen Sie!

Lösung: Die Schranke zeigt $\text{SAT} \notin \mathsf{P}$, mit $\text{SAT} \in \mathcal{NP}$ folgt $\mathsf{P} \neq \mathcal{NP}$.

(c) Welche Laufzeitschranke ergibt sich für X? Begründen Sie!

Lösung: Es ergibt sich eine Laufzeitschranke für X von $\Omega(c^{n^{1/3}})$.

(d) Können Sie mit der Laufzeitschranke für X die Frage $\mathsf{P} \stackrel{?}{=} \mathcal{NP}$ beantworten? Begründen Sie!

Lösung: Nein. Es folgt zwar, dass X ein $\mathcal{NP}$-schweres Problem ist, aber man kann nicht ohne Beschränkung der Allgemeinheit davon ausgehen, dass X in $\mathcal{NP}$ liegt.

Problem 4: Berechenbarkeit 6

Sei T_w die Turing-Maschine mit Gödelnummer w. Zeigen Sie, dass die Sprache

$$L = \left\{(u, v) \in \{0,1\}^\star \times \{0,1\}^\star \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\right\}$$

nicht entscheidbar ist. Verwenden Sie für Ihren Beweis, dass das Halteproblem nicht eintscheidbar ist.

Hinweis: Das Wort w^R bezeichnet das Spiegelwort von w.

Lösung: Angenommen L ist entscheidbar, dann existiert eine Turing-Maschine $\mathcal{M}$, die L entscheidet. Das Halteproblem $\mathcal{H} = \{wv \mid T_w \text{ hält auf der Eingabe } v\}$ ist aus der Vorlesung als unentscheidbar bekannt. Konstruiere zu einer Instanz wv des Halteproblems eine Turing-Maschine $\mathcal{M}_{wv}$, die bei Eingabe von v die Turing-Maschine T_w simuliert und genau dann akzeptiert, wenn T_w stoppt. Bei allen anderen Eingaben akzeptiert $\mathcal{M}_{wv}$ nicht. Sei ferner $\mathcal{M}_\star$ eine Turing-Maschine, die alle Eingaben akzeptiert. Dann entscheidet $\mathcal{M}$ bei Eingabe von $(\langle\mathcal{M}_{wv}\rangle, \langle\mathcal{M}_\star\rangle)$, ob die Turing-Maschine T_w auf der Eingabe v hält, was ein Widerspruch zur Unentscheidbarkeit des Halteproblems ist.

Problem 5: Approximationsalgorithmen $2 + 3 + 4 + 3 + 3 = 15$

Aus Vorlesung und Übung kennen Sie das $\mathcal{NP}$-vollständige Problem VERTEX COVER:

Problem VERTEX COVER

Gegeben: ungerichteter Graph $G = (V, E)$ und $k \in \mathbb{N}$

Gesucht: Menge $V' \subseteq V$ mit $|V'| \leq k$, sodass für alle Kanten $(u, v) \in E$ gilt: $u \in V' \vee v \in V'$.

(a) Ist das Problem in der oben gegebenen Formulierung ein Optimierungsproblem, ein Optimalwertproblem, oder ein Entscheidungsproblem? Geben Sie auch die beiden anderen Formulierungen an. *Anmerkung für Lernende: Die Frage lässt sich nicht wie gefordert beantworten. Lösung:* Bei der obigen Formulierung handelt es sich weder um ein Optimierungsproblem, noch um ein Optimalwertproblem, noch um ein Entscheidungsproblem. Deren Formulierungen wären wie folgt.

Entscheidungsproblem: **Problem**

Gegeben: ungerichteter Graph $G = (V, E)$ und $k \in \mathbb{N}$

Gesucht: existiert eine Menge $V' \subseteq V$ mit $|V'| = k$, sodass für alle Kanten $(u, v) \in E$ gilt: $u \in V' \vee v \in V'$?

Optimalwertproblem: **Problem**

Gegeben: ungerichteter Graph $G = (V, E)$

Gesucht: minimales $k \in \mathbb{N}$, sodass eine Menge $V' \subseteq V$ mit $|V'| = k$ exisitiert, sodass für alle Kanten $(u, v) \in E$ gilt: $u \in V' \vee v \in V'$.

Optimierungsproblem: **Problem**

Gegeben: ungerichteter Graph $G = (V, E)$

Gesucht: Kardinalitätsminimale Menge $V' \subseteq V$, sodass für alle Kanten $(u, v) \in E$ gilt: $u \in V' \vee v \in V'$.

Es soll eine approximative Lösung für das VERTEX COVER Problem berechnet werden. Dazu wird zunächst $V' = \emptyset$ gesetzt. Solange G eine Kante enthält wird dazu eine beliebige Kante $(u, v) \in E$ gewählt, dann beide Knoten u, v zu V' hinzugefügt und aus G entfernt. Um einen Knoten w aus G zu entfernen werden alle zu w inzidenten Kanten und w selbst entfernt.

(b) Zeigen Sie, dass dieser Algorithmus tatsächlich ein VERTEX COVER berechnet. *Lösung:* Betrachte eine beliebige Kante (u, v). Da der Algorithmus erst terminiert, wenn G keine Kante mehr enthält, wird (u, v) irgendwann aus G entfernt. Dies geschieht nur dann, wenn entweder u oder v aus G entfernt wird. Dann gilt $u \in V' \vee v \in V'$.

(c) Zeigen Sie, dass dieser Algorithmus ein Approximationsalgorithmus für das VERTEX COVER Problem mit einer relativen Gütegarantie von 2 ist.

Hinweis: Die Menge der gewählten Kanten bildet ein Matching. *Lösung:* Betrachte die Menge von Kanten, die der Algorithmus auswählt. Da für jede solche Kante (u, v) danach sowohl u als auch v aus G entfernt werden, kann danach keine zu (u, v) adjazente Kante mehr gewählt werden. Die Menge der gewählten Kanten bilden also ein Matching. Jedes VERTEX COVER muss (u, v) abdecken und deshalb u oder v enthalten. Die Menge V' enthält also höchstens doppelt so viele Elemente, wie eine optimale Lösung.

(d) Zeigen Sie, dass unter der Annahme dass $\mathcal{P} \neq \mathcal{NP}$ kein Polynomialzeitapproximationsalgorithmus mit absoluter Gütegarantie für das VERTEX COVER Problem existiert. *Lösung:* Angenommen, es gäbe einen solchen Algorithmus $\mathcal{A}$ mit absoluter Gütegarantie k. Bei Eingabe einer Instanz $G = (V, E)$ wird dann einfach eine Instanz generiert, die aus $k+1$ Kopien von G besteht. Da $\mathcal{A}$ eine absolute Gütegarantie k hat, muss mindestens eine Kopie optimal gelöst worden sein. Gibt man also die beste Teilkopie zurück, hat man in Polynomialzeit eine optimale Lösung von G berechnet. Dies ist unter Annahme von $\mathcal{P} \neq \mathcal{NP}$ ein Widerspruch zur $\mathcal{NP}$-Vollständigkeit von VERTEX COVER.

(e) Der Algorithmus soll angepasst werden, um eine approximative Lösung für das INDEPENDENT SET Problem zu finden.

Problem INDEPENDENT SET

Gegeben: ungerichteter Graph $G = (V, E)$ und $k \in \mathbb{N}$

Gesucht: Menge $V' \subseteq V$ mit $|V'| \geq k$, wobei aus $v \in V'$ für alle Kanten $(u, v) \in E$ folgt, dass $u \notin V'$.

Dazu wird anstatt der Menge V' die Menge $V \setminus V'$ ausgegeben. Ist der so abgewandelte Algorithmus ein Approximationsalgorithmus für das INDEPENDENT SET Problem mit relativer Gütegarantie 2? Begründen Sie!

Lösung: Nein. Betrachte dazu eine Menge von unabhängigen Kanten. Der Algorithmus gibt dann die leere Menge aus, was keine 2-Approximation eines INDEPENDENT SET ist.

Problem 6: Kellerautomaten $1 + 2 + 3 = 6$

(a) Die kontextfreie Grammatik G_1 über dem Eingabealphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ sei definiert durch die Menge der Nichtterminalsymbole $\{S, L, R\}$ mit dem Startsymbol S und folgenden Ableitungsregeln:

$$S \to LR \mid L \mid R$$
$$L \to \mathtt{a}L\mathtt{b} \mid \mathtt{ab}$$
$$R \to \mathtt{b}R\mathtt{a} \mid \mathtt{ba}$$

Geben Sie die Sprache $L(G_1)$ in Mengenschreibweise an, die von G_1 erzeugt wird. *Lösung:*

$$L(G_1) = \left\{\mathtt{a}^i\mathtt{b}^k\mathtt{a}^j \mid i, j, k \in \mathbb{N},\, k > 0,\, i + j = k\right\}$$

(b) Geben Sie eine Grammatik G_2 in Greibach-Normalform an, sodass $L(G_2) = L(G_1)$ gilt, wobei G_1 die Grammatik aus Teilaufgabe a ist.

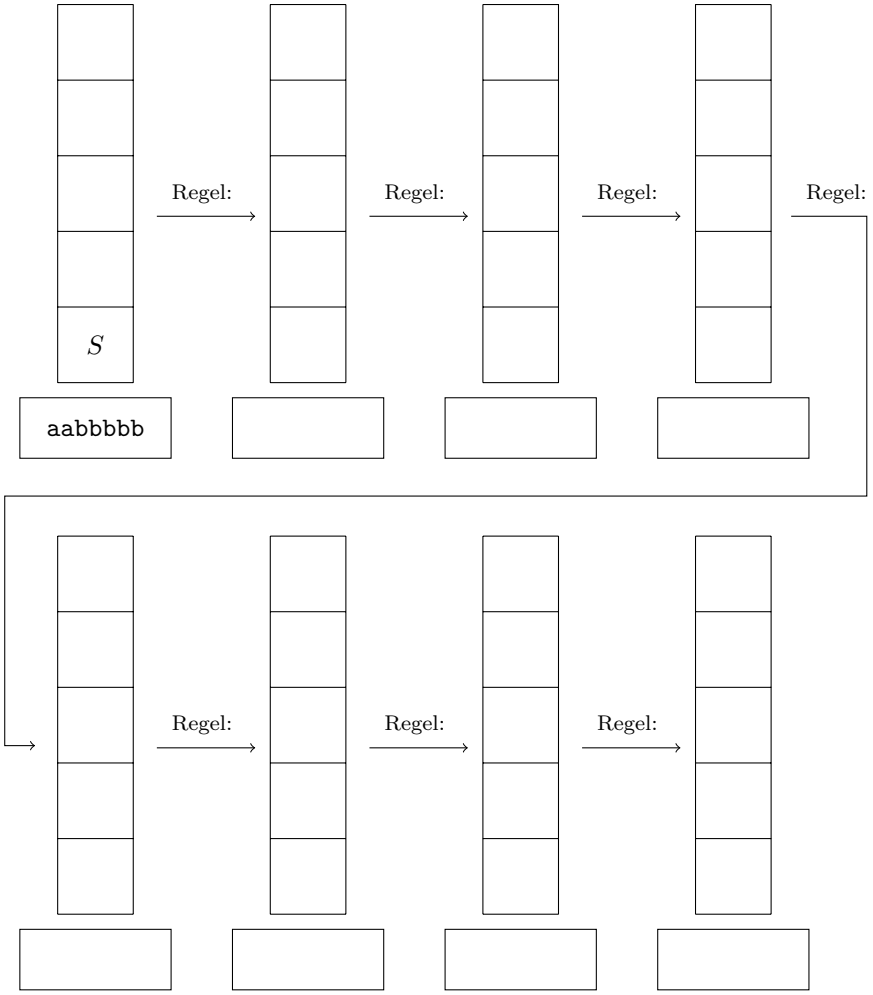
Hinweis: Sie dürfen das Verfahren aus der Vorlesung nutzen, müssen aber nicht. Lösung:

$$S \to \mathtt{a}LBR \mid \mathtt{a}BR \mid \mathtt{a}LB \mid \mathtt{a}B \mid \mathtt{b}RA \mid \mathtt{b}A$$
$$L \to \mathtt{a}LB \mid \mathtt{a}B$$
$$R \to \mathtt{b}RA \mid \mathtt{b}A$$
$$A \to \mathtt{a}$$
$$B \to \mathtt{b}$$

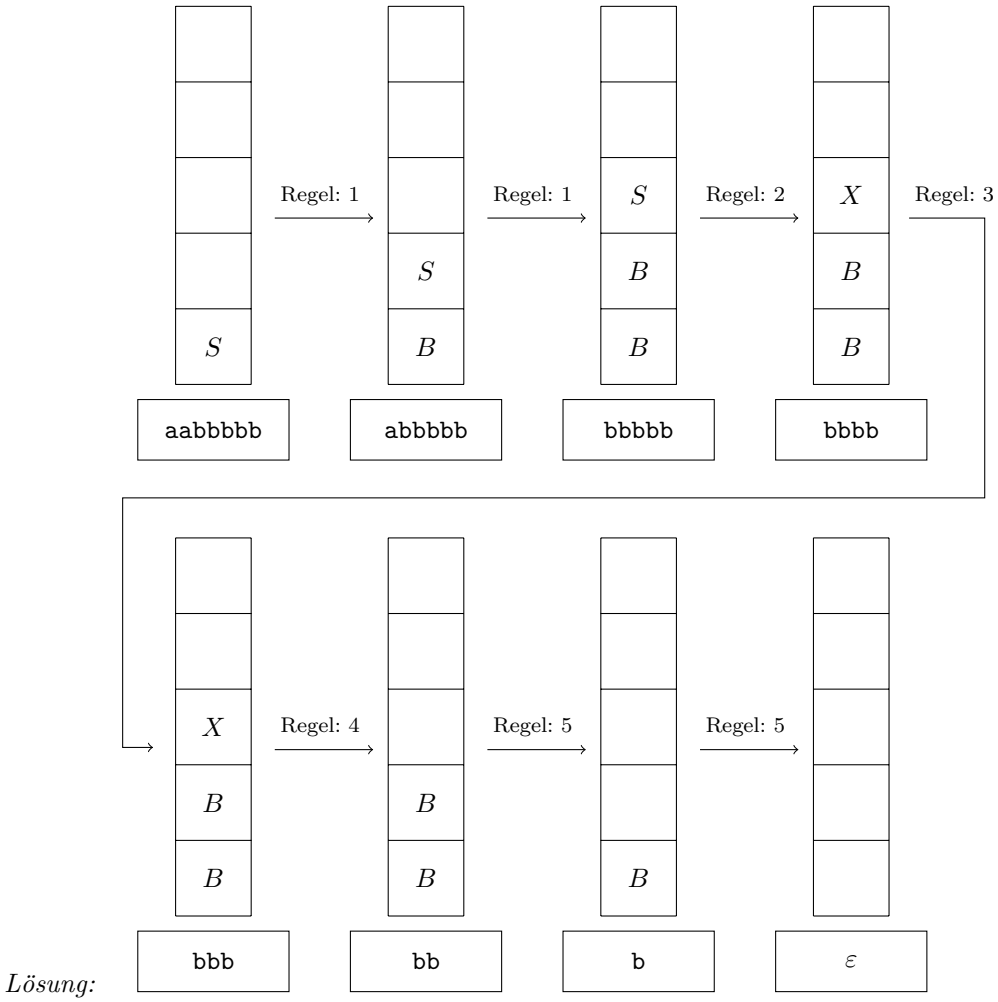
(c) Die kontextfreie Grammatik G_3 in Greibach-Normalform über dem Eingabealphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ sei definiert durch die Menge der Nichtterminalsymbole $\{S, B, X\}$, Startsymbol S und folgenden Ableitungsregeln:

$$S \to \mathtt{a}SB \tag{1}$$
$$S \to \mathtt{b}X \tag{2}$$
$$X \to \mathtt{b}X \tag{3}$$
$$X \to \mathtt{b} \tag{4}$$
$$B \to \mathtt{b} \tag{5}$$

In Vorlesung und Übung wurde ein nichtdeterministischer Kellerautomat vorgestellt, der Sprachen erkennen kann, die von Grammatiken in Greibach-Normalform erzeugt werden. Geben Sie die Konfigurationen dieses Automaten an, die bei Abarbeitung der Eingabe `aabbbbb` entstehen. Vervollständigen Sie dazu folgendes Schema mit den zu lesenden Worten, den Stackinhalten und den Regeln, die beim Übergang verwendet werden. Ist das Eingabewort in der Sprache $L(G_3)$ enthalten?



Lösung:


S
aabbbbb
Regel: 1
S B
abbbbb
Regel: 1
S B B
bbbbb
Regel: 2
X B B
bbbb
Regel: 3
X B B
bbb
Regel: 4
B B
bb
Regel: 5
B
b
Regel: 5
ε

Problem 7: Gemischtes 7

Sind die folgende Aussagen korrekt? Begründen Sie jeweils kurz Ihre Antwort.

(a) Sei ein nichtdeterministischer endlicher Automat $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ gegeben. Dann akzeptiert der Automat $(Q, \Sigma, \delta, s, Q \setminus F)$ die Sprache $L(\mathcal{A})^c$. *Lösung:* Falsch, da sich die Existenz eines akzeptierenden Berechnungspfades nicht einfach negieren lässt.

(b) Sei ein nichtdeterministischer endlicher Kellerautomat $\mathcal{A}$ gegeben. Dann existiert ein nichtdeterministischer endlichernichtdeterministischer endlicher Kellerautomat, der die Sprache $L(\mathcal{A})^c$ akzeptiert. *Lösung:* Falsch. Kontextfreie Sprachen sind unter Komplementbildung nicht abgeschlossen.

(c) Die Sprache aller binär kodierten Primzahlen kleiner als 2017^{2017} ist regulär. *Lösung:* Wahr, jede endliche Sprache ist regulär.

(d) Mit dem CYK-Algorithmus kann die Zugehörigkeit eines Wortes zu einer Typ-1 Grammatik überprüft werden. *Lösung:* Falsch, der CYK-Algorithmus prüft die Zugehörigkeit eines Wortes zu Grammatiken von Typ 2.

(e) Zu jeder entscheidbaren Sprache L existiert eine Typ-0 Grammatik, die L erzeugt. *Lösung:* Wahr, denn die Typ-0 Sprachen sind genau die semi-entscheidbaren Sprachen.

(f) Es existiert eine Sprache in $\mathcal{NP}$, die in Polynomialzeit entschieden werden kann. *Lösung:* Wahr, denn $\mathcal{P} \subset \mathcal{NP}$.

(g) Mit dem Pumpinglemma kann gezeigt werden, dass eine Sprache regulär ist. *Lösung:* Nein. Das Pumpinglemma gibt notwendige, aber nicht hinreichende Eigenschaften von regulären Sprachen an.