Prof. J. Esparza
S. Sickert, J. Krämer

Einführung in die theoretische Informatik

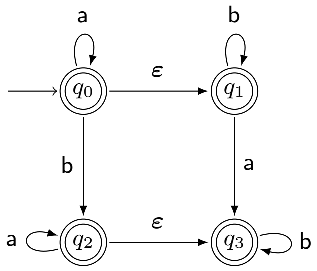
Sommersemester 2017 – Abgabeblatt Lösungsskizze 2

Abgabeblatt

Wir unterscheiden zwischen Übungs- und Abgabeblättern. Mit diesem *Abgabeblatt* fragen wir eine *Auswahl* an Kernaspekten ab, die auf den Trainingsblättern 4 bis 9 zu finden sind. Auch die Kernaspekte, die sich nicht auf Abgabeblättern finden, sind für die Klausur relevant. Wenn Sie die Abgabeblätter regelmäßig abgeben, können Sie zusätzlich einen *Bonus* von 0,3 (nach Bestehen) für beide Klausuren erreichen. Die genauen Regularien, wann Sie einen Notenbonus erhalten und wann nicht, finden Sie auf der Vorlesungshomepage.

AUFGABE 2.1. (*Myhill-Nerode-Äquivalenzklassen*)

Gegeben sei folgender ε-NFA $N = (Q, \Sigma, \delta, q_0, F)$ über dem Alphabet $\Sigma = \{a, b\}$: 4 Punkte



Bestimmen Sie die Anzahl der Myhill-Nerode-Äquivalenzklassen der Sprache $L(N)$, indem Sie den kanonischen Minimalautomaten (Definition 3.58) für $L(N)$ konstruieren. Geben Sie zu jeder Klasse einen regulären Ausdruck an, der alle Wörter der Klasse beschreibt.

Hinweise:

- Sie erhalten 0 Punkte auf diese Aufgabe, falls Ihr konstruierter DFA nicht isomorph zu dem kanonischen Minimalautomaten für L ist. Prüfen Sie deshalb Ihre Lösung für verschiedene Wörter $w \in \Sigma^*$.
- Sie erhalten 0 Punkte auf diese Aufgabe, wenn Sie nicht die Tabelle, die Sie zur Minimierung Ihres DFAs verwendet haben, angegeben.
- Beschriften Sie die Zustände des Automaten mit dem lexikographisch kleinsten Element der Äquivalenzklasse.

Lösungsskizze

Potenzmengenkonstruktion:

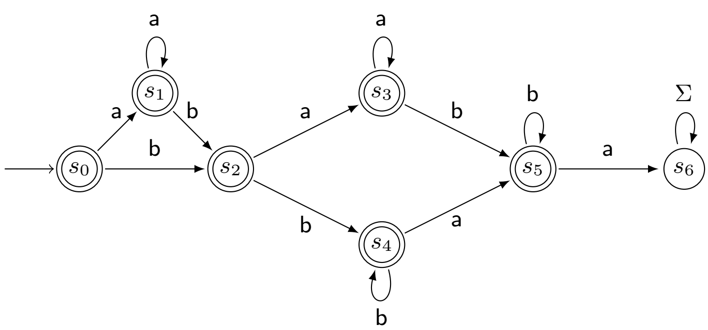
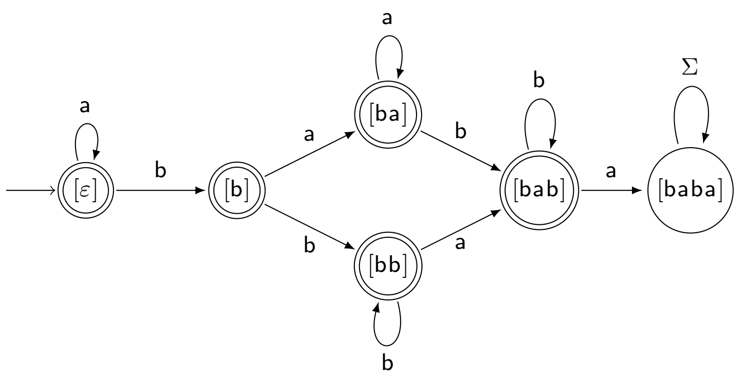


Tabelle:

	s_0	s_1	s_2	s_3	s_4	s_5	s_6
s_0	—	—	—	—	—	—	—
s_1	$\equiv$	—	—	—	—	—	—
s_2	aba	aba	—	—	—	—	—
s_3	ba	ba	ba	—	—	—	—
s_4	aa	aa	aa	aa	—	—	—
s_5	a	a	a	a	a	—	—
s_6	ε	ε	ε	ε	ε	ε	—

(Kanonischer) Minimaler DFA für $L(N)$:



Es gibt somit sechs Äquivalenzklassen: $[\varepsilon] = \mathsf{L}(\mathsf{a}^*)$, $[\mathsf{b}] = \mathsf{L}(\mathsf{a}^*\mathsf{b})$, $[\mathsf{ba}] = \mathsf{L}(\mathsf{a}^*\mathsf{ba}^+)$,$[\mathsf{bb}] = \mathsf{L}(\mathsf{a}^*\mathsf{bb}^+)$, $[\mathsf{bab}] = \mathsf{L}(\mathsf{a}^*\mathsf{ba}^+\mathsf{b}^+ \mid \mathsf{a}^*\mathsf{bb}^+\mathsf{ab}^*)$ und $[\mathsf{baba}] = \mathsf{L}((\mathsf{a}^*\mathsf{ba}^+\mathsf{b}^+ \mid \mathsf{a}^*\mathsf{bb}^+\mathsf{ab}^*)\mathsf{a}(\mathsf{a} \mid \mathsf{b})^*)$.

AUFGABE 2.2. (*Kontextfreie Grammatiken*) 1 Punkte

Die Definition aus den Folien für Typ-2 Grammatiken verbietet ε-Produktionen (mit der Ausnahme $\mathsf{S} \to \varepsilon$). In der Literatur werden aber auch Grammatiken $\mathsf{G} = (\mathsf{V}, \Sigma, \mathsf{P}, \mathsf{S})$ mit $\mathsf{P} \subseteq \mathsf{V} \times (\mathsf{V} \cup \Sigma)^*$ als kontextfrei bezeichnet. Zeigen Sie, dass diese Definitionen die gleiche Klasse an Sprachen beschreibt, d.h. $\mathsf{L}(\mathsf{G})$ ist kontextfrei für jedes G mit $\mathsf{P} \subseteq \mathsf{V} \times (\mathsf{V} \cup \Sigma)^*$.

Lösungsskizze

Anwendung des ε-Entfernungsschritt (Lemma 4.26) liefert das gewünschte Ergebnis.

AUFGABE 2.3. (*Chomsky-Normalform, Cocke-Younger-Kasami-Algorithmus*) 8 Punkte

Wir betrachten folgende Grammatik $\mathsf{G} = (V, \Sigma, P, S)$ für (stark vereinfachte) arithmetische Ausdrücke mit $\Sigma = \{a, b, 0, 1, 2, +, -, \cdot, (,)\}$:

$$\begin{aligned}
S &\to M + M \mid T + T \mid M \mid T \\
D &\to M - M \mid T - T \mid M \mid T \\
M &\to T \cdot M \mid M \\
T &\to F\,V \mid K \mid (S) \\
F &\to \varepsilon \mid K \\
K &\to 0 \mid 1 \mid 2 \\
V &\to a \mid b
\end{aligned}$$

(a) Geben Sie eine Grammatik G' mit $\mathsf{L}(\mathsf{G}) = \mathsf{L}(\mathsf{G}')$, die nur noch nützliche Symbole enthält, an.
(b) Geben Sie eine Grammatik G'' mit $\mathsf{L}(\mathsf{G}') = \mathsf{L}(\mathsf{G}'')$, die in Chomsky-Normalform ist. Geben Sie hierbei alle vier Schritte an.
(c) Bestimmen Sie unter Verwendung des Cocke-Younger-Kasami-Algorithmus, ob $a + b + 2$ und $(2b)$ in $\mathsf{L}(\mathsf{G})$ enthalten sind. Geben Sie hierbei auch Ihre verwendeten Tabellen an.

Hinweise:

- Beachten Sie die aktualisierten Folien! Das Verfahren eliminiert nun erst am Ende Kettenproduktionen.

Lösungsskizze

(a) G':

$$
\begin{aligned}
S &\to T + T \mid T \\
T &\to F\,V \mid K \mid (S) \\
F &\to \varepsilon \mid K \\
K &\to 0 \mid 1 \mid 2 \\
V &\to a \mid b
\end{aligned}
$$

(b) (i) Einfügen von Variablen:

$$
\begin{aligned}
S &\to T X_+ T \mid T \\
T &\to F\,V \mid K \mid X_(S X_) \\
F &\to \varepsilon \mid K \\
K &\to 0 \mid 1 \mid 2 \\
V &\to a \mid b \\
X_+ &\to + \\
X_(&\to (\\
X_) &\to)
\end{aligned}
$$

(ii) Verkürzen von Produktionen:

$$
\begin{aligned}
S &\to T X \mid T \\
X &\to X_+ T \\
T &\to F\,V \mid K \mid X_(Y \\
Y &\to S X_) \\
F &\to \varepsilon \mid K \\
K &\to 0 \mid 1 \mid 2 \\
V &\to a \mid b \\
X_+ &\to + \\
X_(&\to (\\
X_) &\to)
\end{aligned}
$$

(iii) Elimination aller ε-Produktionen:

$$
\begin{aligned}
S &\to T X \mid T \\
X &\to X_+ T \\
T &\to V \mid F\,V \mid K \mid X_(Y \\
Y &\to S X_) \\
F &\to K \\
K &\to 0 \mid 1 \mid 2 \\
V &\to a \mid b \\
X_+ &\to + \\
X_(&\to (\\
X_) &\to)
\end{aligned}
$$

(iv) Elimination aller Kettenproduktionen, Ergebnis ist G'':

$$
\begin{aligned}
S &\to TX \mid FV \mid X_(Y \mid a \mid b \mid 0 \mid 1 \mid 2 \\
X &\to X_+T \\
T &\to FV \mid X_(Y \mid a \mid b \mid 0 \mid 1 \mid 2 \\
Y &\to SX_) \\
F &\to 0 \mid 1 \mid 2 \\
V &\to a \mid b \\
X_+ &\to + \\
X_(&\to (\\
X_) &\to)
\end{aligned}
$$

(c) Nach dem CYK-Algorithmus ergeben sich folgende Berechnungstabelle:

15				
14	25			
13 S	24	35 S		
12	23 X	34	45 X	
11 S,T,V	22 X_+	33 S,T,V	44 X_+	55 S,T,F
a	$+$	b	$+$	2

14 S,T			
13	24 Y		
12	23 S,T	34 Y	
11 $X_($	22 S,T,F	33 S,T,V	44 $X_)$
(	2	b	)

Also ist $a + b + 2 \notin \mathsf{L}(\mathsf{G})$ und $(2b) \in \mathsf{L}(\mathsf{G})$.

Definition (Hamming-Distanz)

Seien $u = a_1a_2 \cdots a_n$ und $v = b_1b_2 \cdots b_n$ zwei Wörter derselben Länge über dem Alphabet Σ. Die Hamming-Distanz $h(u, v)$ ist die Anzahl der Indizes $1 \le i \le n$ mit $a_i \neq b_i$. Die Hamming-Nachbarschaft ist dann definiert als $H(\mathsf{L}, i) = \{w \in \Sigma^* \mid \exists w' \in \mathsf{L}.\ |w| = |w'| \wedge h(w, w') \le i\}$.
Beispiel: Sei $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Dann $h(\mathsf{aaa}, \mathsf{aaa}) = 0$, $h(\mathsf{aaa}, \mathsf{aba}) = 1$ und $h(\mathsf{aab}, \mathsf{baa}) = 2$.

AUFGABE 2.4.

5 Punkte

Zeigen Sie, dass die Hamming-Nachbarschaft mit $i = 1$ einer kontextfreien Sprachen wieder kontextfrei ist.

(a) Geben Sie hierzu ein allgemeines Verfahren an, dass eine Grammatik G in Chomsky-Normalform in eine Grammatik G' mit $\mathsf{L}(\mathsf{G}') = H(\mathsf{L}(\mathsf{G}), 1)$ übersetzt.
(b) Wenden Sie Ihr Verfahren auf die Grammatik $\mathsf{G} = (\{\mathsf{S}\}, \{(,)\}, \mathsf{P}, \mathsf{S})$ mit $\mathsf{P} : \mathsf{S} \to \mathsf{SS} \mid (\mathsf{S}) \mid ()$ an. Hierbei müssen Sie vorher G in Chomsky-Normalform umwandeln.
(c) Begründen Sie die Korrektheit Ihres allgemeinenen Verfahrens.

Lösungsskizze

(a) Sei $\mathsf{G} = (\mathsf{V}, \Sigma, \mathsf{P}, \mathsf{S})$ eine Grammatik in CNF. $\mathsf{G}' = (\mathsf{V} \cup \{X' \mid X \in \mathsf{V}\}, \Sigma, \mathsf{P}', \mathsf{S}')$ mit P':

$$\mathsf{P}' = \mathsf{P} \cup \{X' \to x \mid x, y \in \Sigma \wedge X \to y \in \mathsf{P}\} \cup \{X' \to Y'Z, X' \to YZ' \mid X \to YZ \in \mathsf{P}\}$$

Hinweis: Da G in CNF ist, haben wir $\varepsilon \notin \mathsf{L}(\mathsf{G})$. Falls wir zusätzlich noch ε in die Sprache aufnehmen möchten, können wir in diesem Fall das neuen Startsymbol S'' mit $\mathsf{S}'' \to \mathsf{S}' \mid \varepsilon$ einfügen.

(b)

$$\begin{aligned}
S' &\to S'S \mid SS' \mid A'X \mid AX' \mid A'B \mid AB' \\
S &\to SS \mid AX \mid AB \\
X' &\to S'B \mid SB' \\
X &\to SB \\
A' &\to (\ \mid\ ) \\
A &\to (\\
B' &\to (\ \mid\ ) \\
B &\to)
\end{aligned}$$

(c) Die Grammatik rät die Stelle an der ein Buchstabe getauscht werden könnte und propagiert dies mit $'$ bis zu der Regel die das Terminal erzeugt. Hier kann dann ein beliebiges Zeichen verwendet werden.

AUFGABE 2.5.

4 Punkte

Wir betrachten folgende (Pfeil-)Sprache über dem Alphabet $\Sigma = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$:

$$\mathsf{L} = \{w \in \Sigma^* \mid |w|_{\leftarrow} \neq |w|_{\rightarrow} \vee |w|_{\uparrow} \neq |w|_{\downarrow}\}$$

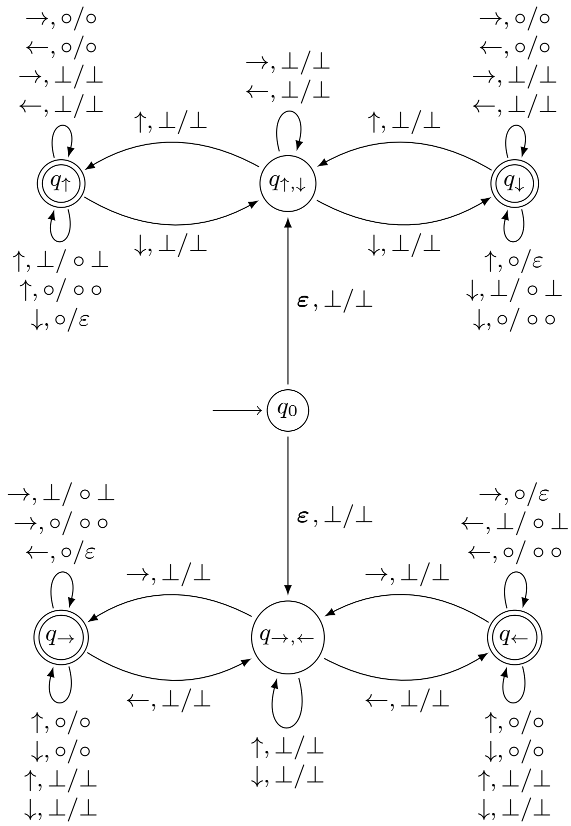
L beschreibt alle Pfade, die *nicht* im Ursprung enden. Zeigen Sie, dass L in der Klasse CFL, aber nicht in der Klasse DCFL liegt.

(a) Geben Sie hierzu einen PDA M mit $\mathsf{L} = \mathsf{L}_F(M)$ an.
(b) Zeigen Sie unter Verwendung der Abschlusseigenschaften von DCFLs, dass L nicht in der Klasse DCFL liegt.

Hinweise:

- Stellen Sie sicher, dass ihr PDA folgende Wörter korrekt akzeptiert bzw. ablehnt: $\varepsilon, \uparrow, \downarrow\uparrow\downarrow\uparrow, \downarrow\leftarrow\uparrow\rightarrow\uparrow$. Falls dies nicht der Fall ist, wird die Aufgabe mit 0 Punkten bewertet.
- Verwenden Sie maximal 8 Zustände und maximal 5 Stacksymbole für Ihren PDA. Falls Sie mehr verwenden, wird die Aufgabe mit 0 Punkten gewertet.
- Sie dürfen in dieser Aufgabe Resultate, die in den Übungen bewiesen worden sind, verwenden.

Lösungsskizze
L ist in der Klasse CFL:



L ist nicht in der Klasse DCFL:
Angenommen L ist in der Klasse DCFL, dann ist auch $\overline{\mathsf{L}}$ auch in DCFL und somit auch in CFL. Dies steht aber im Widerspruch zu dem Fakt, dass $\overline{\mathsf{L}} = \{w \in \Sigma^* \mid |w|_\leftarrow = |w|_\rightarrow \wedge |w|_\uparrow = |w|_\downarrow\}$ nicht kontextfrei ist. (Siehe Übungen).

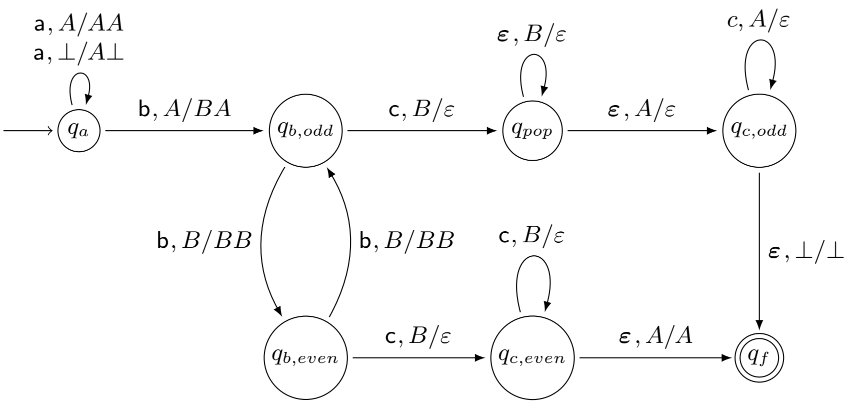
AUFGABE 2.6. 3 Punkte

Wir betrachten folgende Sprache über dem Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$:

$$\mathsf{L} = \{\mathsf{a}^n\mathsf{b}^m\mathsf{c}^m \mid n, m \in \mathbb{N} \wedge m \text{ gerade}\} \cup \{\mathsf{a}^n\mathsf{b}^m\mathsf{c}^n \mid n, m \in \mathbb{N} \wedge m \text{ ungerade}\}$$

Zeigen Sie, dass L in der Klasse DCFL liegt, indem Sie einen DPDA für L angeben.
Hinweise:
- Falls Ihr PDA nicht der Definition eines DPDAs entspricht, wird die Aufgabe mit 0 Punkten gewertet.
- Stellen Sie sicher, dass ihr DPDA folgende Wörter korrekt akzeptiert bzw. ablehnt: ε, abc, cba, abbccc, aabcc. Falls dies nicht der Fall ist, wird die Aufgabe mit 0 Punkten bewertet.
- Verwenden Sie maximal 8 Zustände und maximal 4 Stacksymbole für Ihren DPDA. Falls Sie mehr verwenden, wird die Aufgabe mit 0 Punkten gewertet.

Lösungsskizze



Hinweis: DPDAs für $\mathsf{L} = \{\mathsf{a}^n\mathsf{b}^m\mathsf{c}^m \mid n, m \in \mathbb{N}_0 \wedge m \text{ gerade}\} \cup \{\mathsf{a}^n\mathsf{b}^m\mathsf{c}^n \mid n, m \in \mathbb{N}_0 \wedge m \text{ ungerade}\}$ wurden auch als Lösungen akzeptiert.