

Übung zur Vorlesung
Theoretische Informatik I

Prof. Dr. Christoph Kreitz
Universität Potsdam, Theoretische Informatik, WS 2008/09

Blatt 14 (Version 1) — Abgabetermin: 13.02.2009, 12:30 Uhr

Vorbereitung auf die nächste Vorlesung: Arbeiten Sie sich in das Thema “Eigenschaften von $\mathcal{L}_0/\mathcal{L}_1$ -Sprachen” ein. Verwenden Sie hierzu z.B. die Vorlesungsfolien der Einheit 4.3, die Kapitel 8 der Bücher von Hopcroft, Motwani und Ullman bzw. von Vossen und Witt, eines der anderen empfohlenen Bücher oder das Internet.

Klausuranmeldung: Informatikstudenten, die nach der neuen Studienordnung studieren, müssen sich bis spätestens 8 Werktage vor der Klausur, also bis zum 10. Februar, für die Klausur anmelden.

Klausurzulassung: Für die Zulassung zur Klausur sind 72 Punkte aus Hausaufgaben, Probeklausur und Quiz hinreichend. Studenten, die bisher noch nicht die erforderlichen Punkte erreicht, aber Lösungen zu mindestens 3 der letzten 5 Übungsblätter (9–13) eingereicht haben, werden ebenfalls zur Klausur zugelassen, wenn sie die (Bonus-)Hausaufgaben dieses Übungsblattes bearbeiten und hierfür mindestens 4 der 16 möglichen Punkte erhalten. Beachten Sie, daß Sie die Lösungen bis Freitag, den 13. Februar einwerfen müssen, damit eine Korrektur rechtzeitig erfolgen kann. Bitte haben Sie Verständnis dafür, daß wir nur Lösungen von Teilnehmern korrigieren können, die noch nicht zur Klausur zugelassen sind.

Aufgabe 14.1 (Typ-1 Sprachen)

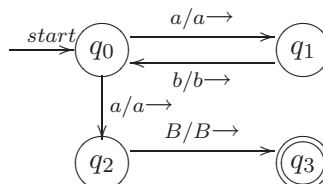
Konstruieren Sie eine expansive Grammatik mit maximal 3 Hilfssymbolen für $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

1. Beschreiben Sie zunächst informal die grundsätzliche Arbeitsweise der Grammatik.
2. Erklären Sie anschließend die Rolle der einzelnen Hilfssymbole ihrer Grammatik.
3. Geben Sie schließlich alle Komponenten der Grammatik G an.
4. Ist G kontextsensitiv? Geben Sie eine kurze Begründung.
5. Ist L kontextfrei? Geben Sie eine kurze Begründung.

Tipp: In expansiven Grammatiken ist es möglich, Hilfssymbole von links nach rechts durch ein Wort wandern zu lassen und am Ende in ein Terminalsymbol umzuwandeln.

Aufgabe 14.2 (Typ-0 Sprachen)

Betrachten Sie die NTM $M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \{a, b, B\}, \delta, q_0, B, \{q_3\})$, wobei δ durch den folgenden Graphen gegeben ist.



1. Erklären Sie wie die Maschine arbeitet und geben Sie dann die Sprache $L = L(M)$ an.
2. Beweisen Sie, daß $L = L(M)$ ist. Verwenden Sie Konfigurationen. Es ist nicht nötig, den Induktionsbeweis über das Verhalten bei q_0 explizit auszuführen.
3. Konstruieren Sie eine zu M äquivalente DTM M' . Erklären Sie Ihre Konstruktion.
4. Simulieren Sie M durch eine TM M'' mit halbseitig unendlichem Band und binärem Bandalphabet. Verwenden Sie dazu $\Sigma'' = \{I\}$ als Eingabealphabet und $\Gamma'' = \{I, B\}$ als Bandalphabet von M'' .

Aufgabe 14.3 (Konstruktion von Turingmaschinen)

1. Geben Sie eine NTM M_1 an, die für ein Wort $w \in \{0, 1\}^*$ prüft, ob w die Zeichenkette 001 enthält!
2. Konstruieren Sie eine Turingmaschine M_2 , die bei Eingabe eines Wortes $v\#w$ mit $v, w \in \{0, 1\}^*$ prüft, ob v im Wort w als Teilwort vorkommt, d.h. ob $w = uvx$ für gewisse Wörter $u, x \in \{0, 1\}^*$ ist.
3. Beschreiben Sie eine Turingmaschine M_3 , welche bei Eingabe eines Wortes $w\#w'$ mit $w, w' \in \{0, 1\}^*$ testet, ob w' der lexikographischen Vorgänger w_v von w ist.

Geben Sie die Übergangstabelle für δ an und begründen Sie die Korrektheit der Turingmaschine.

Hinweise: Die lexikographische Ordnung der Worte über $\{0, 1\}$ beginnt mit der folgenden Liste:
 $\varepsilon - 0 - 1 - 00 - 01 - 10 - 11 - 000 - 001 - 010 \dots$ Der lexikographische Vorgänger von ε ist ε .
Es macht Sinn, die Fälle $w = 0^k$ und $w = u10^k$ ($k \in \mathbb{N}, u \in \{0, 1\}^*$) zu unterscheiden.

Hausaufgabe 14.4 (Reguläre Sprachen – 6 Bonuspunkte)

Gegeben sind die DEAs $A_1 = (\{1, 2, 3, 4, 5, 6, 7, 8\}, \{a, b\}, \delta_1, 1, \{3\})$, und $A_2 = (\{A, B, C, D\}, \{a, b\}, \delta_2, A, \{C\})$, wobei δ_1 und δ_2 wie folgt gegeben sind.

	Zustand	a	b		Zustand	a	b
$\rightarrow$	1	5	2	$\rightarrow$	A	D	B
	2	3	4		B	C	B
*	3	5	3	*	C	D	C
	4	3	4		D	D	D
	5	6	5				
	6	5	7				
	7	5	5				
	8	5	3				

1. Erzeugen Sie aus A_2 eine rechtslineare Grammatik G_2 mit $L(G_2) = L(A_2)$. (1 Punkte)
2. Prüfen Sie, ob A_1 und A_2 äquivalent sind und konstruieren Sie einen minimalen Automaten A'_1 mit $L(A'_1) = L(A_1)$. (2 Punkte)
3. Wandeln Sie A_1 mit Hilfe des Zustandseleminationsverfahrens in einen äquivalenten regulären Ausdruck R_1 um, der maximal 6 Symbole enthält. (1 Punkte)
4. Beweisen Sie mittels Induktion, dass A_1 tatsächlich die Sprache $L_1 = L(R_1)$ akzeptiert, ohne dabei die Aussage zu verwenden, daß A_1 äquivalent zu R_1 ist. (2 Punkte)

Hausaufgabe 14.5 (Kontextfreie Sprachen – 7 Bonuspunkte)

Betrachten Sie die kontextfreien Grammatik $G = (\{S, A, B, C, D, E, F, \{0, 1, 2\}, P, S)$ mit

$$\begin{aligned}
 P = \{ & S \rightarrow SA \mid CC \mid 2S2S, \\
 & A \rightarrow F, \\
 & B \rightarrow 0 \mid AD, \\
 & C \rightarrow 2 \mid 21 \mid 12 \mid SS, \\
 & D \rightarrow 0E \mid 1, \\
 & E \rightarrow D1 \mid 0B \mid 1E \\
 & F \rightarrow 2F \mid 0F \mid FE \}
 \end{aligned}$$

1. Transformieren Sie G in eine äquivalente Grammatik G' ohne unnütze Symbole. (1 Punkte)
2. Wandeln Sie G in einen PDA A um, der $L(G)$ mit mit Endzuständen akzeptiert. (2 Punkte)
3. Wandeln Sie G in eine äquivalente Grammatik G'' in Chomsky Normalform um (1 Punkte)
4. Prüfen Sie mit Hilfe des Cocke-Younger-Kasami Algorithmus, ob die Wörter 121 und 212 zu $L(G)$ gehören. (1 Punkte)
5. Beweisen Sie durch (vollständige) Induktion über die Länge der Ableitung, daß für jedes Wort $w \in L(G)$ gilt $\#_2(w) \geq \#_1(w)$ (2 Punkte)

Hausaufgabe 14.6 (Nichtreguläre Sprachen – 3 Bonuspunkte)

1. Beweisen Sie ausschließlich durch Anwendung von Abschlußseigenschaften, daß $L_1 = \{w \mid w \in \{a, b\}^* \text{ und } w \text{ enthält doppelt so viele } a \text{ wie } b\}$ nicht regulär ist. (1 Punkte)
2. Beweisen Sie mit dem Pumping Lemma, daß $L_2 = \{a^n b^m \mid n, m \in \mathbb{N} \wedge \exists t \geq 1. n = tm\}$ nicht regulär ist. (2 Punkte)