Goethe-Universität Frankfurt am Main
Institut für Informatik
Theorie komplexer Systeme
Prof. Dr. Nicole Schweikardt

24. Mai 2012

Theoretische Informatik 2

Sommersemester 2012

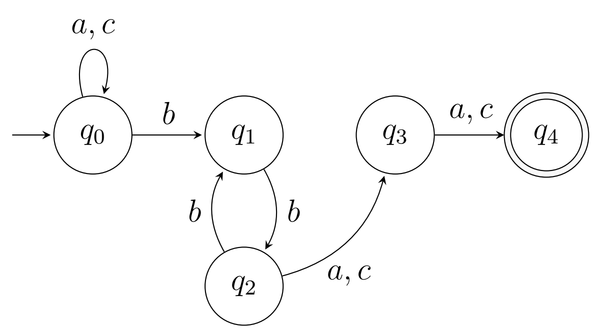
Übungsblatt 6

Abgabe: *bis 31. Mai 2012 8:14*

Aufgabe 1: **($9 + 9 + 9 = 27$ Punkte)**

Reguläre Grammatiken werden auf Grund der Form ihrer Produktionen auch *rechts-lineare* Grammatiken genannt. Eine Grammatik $G = (\Sigma, V, S, P)$ heißt *linear*, wenn ihre Produktionen alle von der Form $A \to \epsilon$ oder $A \to uBv$, für $u, v \in \Sigma^*$ und $A, B \in V$, sind. Eine Sprache heißt *linear*, wenn es eine lineare Grammatik gibt, die sie erzeugt.

(a) **(i)** Konstruieren Sie eine reguläre Grammatik, die die Sprache des folgenden Automaten erzeugt:



Formen Sie die Grammatik anschließend in eine lineare Grammatik mit möglichst wenigen Variablen um.

(ii) Sei $G := (\{a, b\}, \{E, O\}, E, P)$ die lineare Grammatik mit der folgenden Produktionsmenge P:

$$E \to aO \mid bO \mid bbE \mid bb$$
$$O \to aE \mid bE \mid abaE \mid aba$$

Erzeugt G eine reguläre Sprache? Falls ja, geben Sie eine reguläre Grammatik an, die die gleiche Sprache erzeugt, und wandeln sie diese anschließend in einen endlichen Automaten um, der die Sprache erkennt. Falls nicht, beweisen Sie, dass die Sprache nicht regulär ist.

(b) Beweisen oder widerlegen Sie folgende Behauptung:

Eine Sprache ist genau dann linear, wenn sie regulär ist.

Aufgabe 2: **($12 + 12 = 24$ Punkte)**

Geben Sie für die folgenden Sprachen kontextfreie Grammatiken an und beweisen Sie, dass diese Grammatiken die Sprachen erzeugen:

(a) $L_1 := \{a^n b^m a^{n+m} : n, m \in \mathbb{N}\}$,

(b) $L_2 := \{w \in \{0,1\}^* : |w|_1 \geq |w|_0\}$.

(Zur Erinnerung: Für alle Alphabete Σ, $w \in \Sigma^*$ und $a \in \Sigma$ ist $|w|_a$ die Anzahl der Vorkommen des Buchstabens a in w).

Aufgabe 3: **($10 + 13 = 23$ Punkte)**

Wir wollen zeigen, dass es einen Algorithmus gibt, der bei Eingabe zweier regulärer Ausdrücke R, S einen regulären Ausdruck T mit $L(T) = L(R) \cap L(S)$ und $|T| = 2^{\mathcal{O}(|R|\cdot|S|)}$ konstruiert. Geben Sie hierfür Algorithmen $\mathcal{A}_1$ und $\mathcal{A}_2$ an, die Folgendes tun:

(a) Bei Eingabe von R, S gibt $\mathcal{A}_1$ einen NFA A mit $L(A) = L(R) \cap L(S)$ aus, dessen Zustandsmenge die Größe $\mathcal{O}(|R| \cdot |S|)$ hat.

(b) Bei Eingabe eines NFAs A mit der Zustandmenge Q erzeugt $\mathcal{A}_2$ einen regulären Ausdruck T der Länge $2^{\mathcal{O}(|Q|)}$ mit $L(T) = L(A)$.

Aufgabe 4: **($14 + 12 = 26$ Punkte)**

Eine *unäre* Sprache ist eine Sprache über einem Alphabet Σ mit $|\Sigma| = 1$. Im Folgenden sei $\Sigma := \{a\}$, und für jede unäre Sprache L sei $M(L) := \{i \in \mathbb{N} : a^i \in L\}$.

Eine Menge von natürliche Zahlen heißt *linear*, wenn es Zahlen $j_0, \ldots, j_n \in \mathbb{N}$ gibt, so dass die Menge von der Form $\{j_0 + k_1 j_1 + \cdots + k_n j_n : k_1, \ldots, k_n \in \mathbb{N}\}$ ist. Eine Menge heißt *semilinear*, wenn sie eine endliche Vereinigung linearer Mengen ist. Insbesondere ist also auch die leere Menge semilinear.

(a) Zeigen Sie: Eine unäre Sprache L ist genau dann regulär, wenn $M(L)$ semilinear ist.

(Überlegen Sie sich hierzu, von welcher besonders einfachen Gestalt ein DFA sein muss, wenn er eine unäre Sprache erkennt.)

(b) Zeigen Sie: Für jede (nicht zwingend reguläre) unäre Sprache L ist L^* regulär.

(Zeigen Sie hierfür, dass $M(L^*)$ semilinear ist. Bilden Sie die linearen Mengen mit Hilfe der Länge des kürzesten Wortes in $L \setminus \{\epsilon\}$.)