
Einführung in die Theoretische Informatik

Abgabetermin: 21. Juni 2010 vor 12.00 Uhr in die THEO Briefkästen

Hausaufgabe 1 (5 Punkte)

Für Zwecke dieser Aufgabe nennen wir einen deterministischen Kellerautomaten $K = (Q, \Sigma, \Delta, \delta, q_0, Z_0, F)$, dessen Übergangsfunktion δ so beschaffen ist, dass der Kellerinhalt nie verändert wird, einen ϵ -DFA. Sei $L(K)$ die Sprache, die von einem ϵ -DFA K mit Endzuständen akzeptiert wird.

Geben Sie ein direktes (nicht über ϵ -NFA) Verfahren an, das zu einem beliebigen ϵ -DFA K einen deterministischen endlichen Automaten A definiert, der die Sprache $L(K)$ erkennt!

Hausaufgabe 2 (5 Punkte)

Wir bezeichnen die Anzahl der Vorkommen eines Buchstaben x in einem Wort w mit $|w|_x$. Sei $\Sigma = \{a, b\}$ und

$$L = \{w \in \Sigma^* \mid w = uav, u, v \in \Sigma^*, |u|_a = |v|_b\}.$$

1. Geben Sie eine Grammatik $G = (V, \Sigma, P, S)$ an mit $L_G = L$.
2. Leiten Sie nach Satz der Vorlesung systematisch einen zu G äquivalenten Kellerautomaten K her.

Hausaufgabe 3 (5 Punkte)

1. Geben Sie eine deterministische Turingmaschine A an, die für ein Eingabewort $w \in \{0, 1\}^+$ eine Berechnung durchführt, so dass am Ende der Berechnung das Wort ww^Rw^R auf dem Band steht und der Kopf im Endzustand auf dem Anfang von ww^Rw^R steht. Kommentieren Sie Ihre Lösung durch eine informelle Beschreibung Ihrer Lösungsidee.
2. Geben Sie eine Turingmaschine B an, die die Sprache $L = \{ww^Rw^R \mid w \in \{0, 1\}^*\}$ akzeptiert. Kommentieren Sie Ihre Lösung durch eine informelle Beschreibung Ihrer Lösungsidee.

Vorbereitung 1

Die Auswertung von Polynomen $p(x) = \sum_{i=0}^n a_i x^i$ vom Grad $n \geq 0$ für bestimmte x nach dem Horner-Algorithmus $p_i := p_{i-1}x + a_{n-i}$ für $i = 0, \dots, n$ mit $p_{-1} = 0$ liefert $p(x) = p_n$. Wir wollen den Horner-Algorithmus dazu benutzen, eine in Binärdarstellung ohne führende Nullen gegebene natürliche Zahl $y = \sum_{i=0}^n b_i 2^i$ mit $b_i \in \{0, 1\}$ in eine gleichwertige unäre Strichdarstellung $u(y) \in \{\mid\}^*$ zu transformieren.

Sei $\beta_k = b_{k-1}b_{k-2} \dots b_0$. Die Berechnung nach Horner protokollieren wir als Folge von „Konfigurationen“, d. h. Tripeln $(u(p_i), q, \beta_{n-i})$ mit

$$u(p_i) = u(2p_{i-1} + b_{n-i}) = u(p_{i-1})u(p_{i-1})u(b_{n-i}),$$

wobei wir $q = q_0 = 0$ setzen wollen.

Bestimmen Sie $u(10)$, indem Sie die Folge der genannten Konfigurationen für $y = 10$ berechnen.

Vorbereitung 2

Beweisen oder widerlegen Sie:

Für jede Turingmaschine T gibt es eine Turingmaschine T' mit nur einem Zustand, die die Sprache von T akzeptiert.

Tutoraufgabe 1

Sei L die zu dem regulären Ausdruck $0|(1(0|1)^*)$ gehörige Sprache. Für eine Zahl $n \in \mathbb{N}$ bezeichne $b(n) \in L$ bzw. $u(n) \in \{\mid\}^*$ deren binäre bzw. unäre Darstellung.

Es ist eine (deterministische) Ein-Band-Turing-Maschine zu konstruieren, die für $n \neq 0$ die Eingabe $b(n)$ in $u(n)$ umwandelt. Die Eingabe und die Ausgabe sollen sich auf dem sonst leeren Band befinden, und der Lese-/Schreibkopf der Maschine bei Start und Ende der Berechnung soll links auf dem ersten beschriebenen Feld des Bandes stehen.

1. Entwerfen Sie eine Programmstruktur für die zu konstruierende Turingmaschine, indem Sie versuchen, die Gesamtlösung auf Teilprobleme zurückzuführen, für deren Lösung dann in der nachfolgenden Teilaufgabe jeweils spezielle Turingmaschinen implementiert werden können.
2. Implementieren Sie Ihre spezifizierten Turingmaschinen und fügen Sie die Programme zu einem Gesamtprogramm zusammen zur Umwandlung einer binären in eine unäre Darstellung einer natürlichen Zahl.

Tutoraufgabe 2

Ein *Queue-Automat* funktioniert ähnlich wie ein Kellerautomat, nur dass der Keller durch eine Queue („Warteschlange“) ersetzt ist.

Formal ist ein Queue-Automat ein Tupel $A = (Q, \Sigma, \Gamma, q_0, Z_0, \delta)$. Dabei sind Q, Σ, Γ endliche Mengen, $q_0 \in Q, Z_0 \in \Gamma$, und die Übergangsfunktion δ hat den Typ $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma^*)$, wobei $|\delta(q, b, Z)| < \infty$ für alle $q \in Q, b \in \Sigma \cup \{\epsilon\}, Z \in \Gamma$.

Eine *Konfiguration* eines Queue-Automaten ist ein Tripel $(q, w, \gamma) \in Q \times \Sigma^* \times \Gamma^*$. Dabei ist q der aktuelle Zustand, w das noch zu lesende Wort und γ der aktuelle Inhalt der Queue. Daraus ergibt sich die Übergangsrelation \rightarrow_A zwischen Konfigurationen:

$$(q, bw, Z\gamma) \rightarrow_A (q', w, \gamma\gamma'), \quad \text{wenn} \quad (q', \gamma') \in \delta(q, b, Z).$$

Beachten Sie, dass der einzige Unterschied zum Kellerautomaten in der Definition von \rightarrow_A liegt. Hier wird γ' hinter das γ geschrieben und nicht davor. Die (mit leerer Queue) akzeptierte Sprache eines Queue-Automaten ist

$$L_\epsilon(A) = \{w \in \Sigma^* \mid \exists q' \in Q. (q_0, w, Z_0) \rightarrow_A^* (q', \epsilon, \epsilon)\}.$$

Die graphische Darstellung von Queue-Automaten ist die gleiche wie bei Kellerautomaten.

1. Geben Sie einen Queue-Automaten A an mit $L(A) = \{a^n b^n a^n \mid n \in \mathbb{N}\}$.
2. Zeigen Sie, dass jede (deterministische) Turingmaschine durch einen Queue-Automaten simuliert werden kann.