
Einführung in die Theoretische Informatik

Abgabetermin: 28. Juni 2010 vor 12.00 Uhr in die THEO Briefkästen

Hausaufgabe 1 (5 Punkte)

Seien $\Sigma = \{a, b\}$ und $L = \{w \in \Sigma^* \mid w = uav, u, v \in \Sigma^*, |u|_a = |v|_a, |u|_b = |v|_b\}$.
Geben Sie einen 2-Kellerautomaten an (Definition siehe Turaufgabe 1), der L akzeptiert.

Hausaufgabe 2 (5 Punkte)

Seien $\Sigma = \{a_1, a_2, \dots, a_n\}$ ein beliebiges n -elementiges Alphabet und $\Sigma' = \Sigma \cup \{\#\}$.
Geben Sie eine Turingmaschine $M = (Q, \Sigma', \Gamma, \delta, q_0, \square, F)$ mit höchstens 5 Zuständen an, die bei leerer Eingabe das Alphabet Σ in der Form $\#a_1\#a_2 \dots \#a_n$ auf das Band schreibt und mit dem Kopf auf dem ersten, rechtsstehenden Zeichen der Ausgabe anhält.

Hausaufgabe 3 (5 Punkte)

Eine 1-Band Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ heißt linear beschränkter Automat, wenn sie für keine Eingabe ein Leerzeichen \square überschreibt.
Sei $\Sigma = \{0, 1\}$. Geben Sie für die kontextsensitive Sprache $L = \{ww \mid w \in \Sigma^*\}$ einen linear beschränkten Automaten M an, der L akzeptiert.

Hausaufgabe 4 (5 Punkte)

Zeigen Sie, dass man **case**-Anweisungen der folgenden Form für festes n durch LOOP-Programme simulieren kann:

case x_i **do** 0: P_0 ; 1: P_1 ; ... n : P_n **od**

Falls $0 \leq x_i \leq n$ gilt, dann wird das LOOP-Programm P_{x_i} ausgeführt und die **case**-Anweisung beendet. Ist $x_i > n$, so wird die **case**-Anweisung übersprungen. Keines der LOOP-Programme P_j enthalte die Variable x_i .
Verwenden Sie nur Zuweisungen der Form $x_k := x_j \pm c$, den LOOP-Befehl und Initialisierungen der Form $x_k := c$.

Vorbereitung 1

Sei $f : \mathbb{N} \rightarrow \mathbb{N}$ diejenige Funktion, die definiert ist durch die Startwerte $f(0) = 1$, $f(1) = 1$, $f(2) = 1$ zusammen mit der Rekursion

$$f(n+1) = n \cdot f(n-1) + f(n-2) \quad \forall n \in \mathbb{N}, n \geq 2.$$

Geben Sie ein LOOP-Programm an, das den Funktionswert von f für $n \geq 3$ in der Variablen x_3 berechnet.

Vorbereitung 2

Geben Sie eine auf den in der Vorlesung definierten Regeln für primitiv rekursive Funktionen basierende Definition an für $fac(n) = n!$.

Tutoraufgabe 1

Ein 2-Kellerautomat $K = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, Z'_0, F)$ ist ein Kellerautomat, der über einen zweiten Keller verfügt. Der zweite Keller wird mit Z'_0 initialisiert. Die Übergangsfunktion $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \times \Gamma \rightarrow \mathcal{P}_e(Q \times \Gamma^* \times \Gamma^*)$ beschreibt die Vorgehensweise des 2-KA wie folgt (\mathcal{P}_e bezeichnet die Menge aller endlichen Teilmengen): Liest der 2-KA im Zustand q die Eingabe b (auch $b = \epsilon$ ist möglich), sind Z_1, Z_2 die obersten Zeichen der beiden Keller und gilt $(q', \alpha_1, \alpha_2) \in \delta(q, b, Z_1, Z_2)$, dann kann der 2-KA in den Zustand q' übergehen und hierbei Z_1 durch α_1 und Z_2 durch α_2 ersetzen.

Zeigen Sie: Jede 1-Band-Turingmaschine $T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ kann durch einen 2-Kellerautomaten $K = (Q', \Sigma, \Gamma', \delta', q'_0, Z_0, Z'_0, F')$ simuliert werden.

Tutoraufgabe 2

Zeigen Sie durch Rückführung auf die Definition, dass die folgenden Funktionen primitiv rekursiv sind.

1. $twopow(n) = 2^n$,
2. $tower(n) = 2^{2^{2^{\cdot^{\cdot^2}}}}$ (d.h. $2^{(2^{(2^{\cdot^{\cdot^2}})})}$, Turm der Höhe n),
3. $ifthen(n, a, b)$ mit

$$ifthen(n, a, b) = \begin{cases} a & n \neq 0, \\ b & n = 0. \end{cases}$$

Tutoraufgabe 3

Sei $f : \mathbb{N} \rightarrow \mathbb{N}$ diejenige Funktion, die für alle $n \in \mathbb{N}, n \neq 0$ durch die Rekursion

$$f(n+1) = f(n) \cdot f(n-1)$$

mit den Startwerten $f(0) = 1$ und $f(1) = 2$ definiert ist.

1. Zeigen Sie, dass f primitiv-rekursiv ist.
2. Zeigen Sie, dass f streng monoton wachsend ist, d. h., es gilt $f(n) < f(n+1)$ für alle $n \in \mathbb{N}$ mit $n \neq 1$.
3. Ist der Wertebereich von f entscheidbar?