Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr

SS 2002
Semestralklausur
20. Juli 2002

Einführung in die Informatik IV

Mit $BIN = L(1(0+1)^* + 0)$ wird die Menge der Binärdarstellungen der natürlichen Zahlen (Darstellung zur Basis 2) bezeichnet. Die Funktion $bin : \mathbb{N}_0 \to BIN$ ordnet jeder natürlichen Zahl ihre Binärdarstellung zu. Es liegt eine Gödelisierung von Turingmaschinen mit Werten in BIN vor.

Aufgabe 1 (15 Punkte)

Zeigen oder widerlegen Sie: Die Sprache

$$L = \{w \in BIN \mid M_w \text{ akzeptiert mindestens 3 verschiedene Wörter aus } \{0,1\}^*\}$$

ist rekursiv.

Aufgabe 2 (10 Punkte)

Zeigen oder widerlegen Sie: Die Sprache

$$L = \{w \in BIN \mid \exists x \in BIN : M_w \text{ hält auf Eingabe } x\}$$

ist rekursiv aufzählbar.

Aufgabe 3 (15 Punkte)

Wir betrachten kontextfreie Grammatiken über einem Terminalalphabet Σ. Zeigen Sie, dass folgendes Problem unentscheidbar ist:
Gegeben sei eine beliebige, kontextfreie Grammatik G. Ist dann $L(G) = \Sigma^*$?

Aufgabe 4 (15 Punkte)

Mit $bin(n)$ bzw. $un(n)$ bezeichen wir die binäre bzw. unäre Darstellung von $n \in \mathbb{N}_0$ (über dem Alphabet {0,1} bzw. {|}). Welche der folgenden Sprachen sind kontextfrei? Begründen Sie Ihre Antwort.

(a) $L_1 = \{bin(n)\#un(n) \mid n \in \mathbb{N}\}$.

(b) $L_2 = \{bin(n)\#un(\lceil \log_2(n+1) \rceil) \mid n \in \mathbb{N}\}$.

Beachten Sie: $\mathbb{N} = \{1, 2, 3, \ldots\}$.

Aufgabe 5 (10 Punkte)

Verwenden Sie für diese Aufgabe keine Begriffsäquivalenzen, d.h. Ihre Begründung darf sich nur auf LOOP/WHILE-Programme und keine anderen Konstrukte (primitive Rekursion, Turingmaschinen, etc.) beziehen.

(a) Begründen Sie, warum LOOP-berechenbare Funktionen total sind.

(b) Warum können WHILE-berechenbare Funktionen auch partiell sein?

Aufgabe 6 (15 Punkte)

Geben Sie LOOP-Programme an, die folgende Programmabläufe simulieren. Verwenden Sie nur Zuweisungen der Form $x_i := x_j \pm c$, den **loop** Befehl und Initialisierungen $x_i := c$.

(a) **case** x_i **do**

$$\begin{array}{l} 0 : P_0; \\ 1 : P_1; \\ \vdots \\ n : P_n \end{array}$$

od

Falls $0 \leq x_i \leq n$ gilt, 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 $P_0, P_1, \ldots, P_n$ enthält die Variable x_i.

(b) Die Anweisung

$$\textbf{for } x_i := x_m \textbf{ to } x_l \textbf{ do } P \textbf{ od}$$

wobei P ein LOOP-Programm ist, in dem die Variablen x_i, x_l, x_m nicht vorkommen.

Aufgabe 7 (20 Punkte)

Zeigen Sie, dass folgende Funktionen primitiv rekursiv sind, **ohne** LOOP-Programme zu benutzen. Die folgenden Funktionen können Sie als primitiv rekursiv voraussetzen: $+, \cdot$, $fac(n) = n!$, $f(n) = \sum_{k=1}^{n} g(k, n)$, wobei g primitiv rekursiv ist, $n \dot{-} m = max(0, n - m)$, $kleiner(n, m) = 1$, falls $n < m$ und 0 sonst. Beachten Sie: $0! = 1$ und $m, n \in \mathbf{N}_0$.

(a) $\min(n, m) := \begin{cases} n, & \text{falls } n \leq m, \\ m, & \text{sonst} \end{cases}$

(b) $\text{tfac}(n) := \begin{cases} 1, & \text{falls, } \exists\, k \in \mathbf{N}\ n = k! \\ 0, & \text{sonst} \end{cases}.$

(c) Seien f_1, f_2 primitiv rekursiv mit Werten in $\{0, 1\}$.

$$f(n) := \begin{cases} 1, & \text{falls } f_i(n) = 1 \text{ für mindestens ein } i \in \{1, 2\}, \\ 0, & \text{sonst} \end{cases}$$

(d) Sei f primitiv rekursiv mit Werten in $\{0, 1\}$. Dann sei

$$g(n) := \begin{cases} 1, & \text{falls } \forall k \in \{0, 1, 2, \ldots, n\}\ f(k) = 1, \\ 0, & \text{sonst} \end{cases}$$