Institut für Informatik
der Universität München
Prof. Dr. M. Hofmann
Dr. M. Lange

SS 05

20.6.05

Übungen zur Vorlesung
Informatik IV

Blatt 10
Abgabe spätestens am 27.6.05, 14:00 Uhr

Aufgabe 52: **6 Punkte**

In dieser Aufgabe zeigen wir, dass der `GOTO`-Befehl in `GOTO`-Programmen essentiell ist. Mit $\texttt{GOTO}^-$ bezeichnen wir die Klasse der `GOTO`-Programme, die den `GOTO`-Befehl nicht verwenden (und damit natürlich auch kein `IF` ... `THEN GOTO`...).

a) Zeigen Sie, dass für jede $\texttt{GOTO}^-$-berechenbare Funktion $f : \mathbb{N}^n \to \mathbb{N}$ eine Konstante $c \in \mathbb{N}$ existiert, so dass für alle $x_1, \ldots, x_n \in \mathbb{N}$ gilt:

$$f(x_1, \ldots, x_n) \leq c \cdot \max(\{1\} \cup \{x_i \mid i = 1, \ldots, n\})$$

b) Geben Sie ein `GOTO`-Programm an, welches eine Funktion f berechnet, die nicht die Eigenschaft aus (a) hat. Beweisen Sie, dass dies nicht der Fall ist!

Aufgabe 53: **6 Punkte**

Wir setzen es als bekannt voraus, dass die Subtraktion $- : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, wobei $x - y$ wie üblich definiert ist, falls $x \geq y$, und $x - y = 0$ sonst, primitiv rekursiv ist.

a) Konstruieren Sie — evtl. unter Verwendung obiger Subtraktion — primitiv rekursive Funktionen $\max, \min : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, welche jeweils das Maximum bzw. Minimum zweier Zahlen berechnen.

b) Konstruieren Sie eine μ-rekursive Funktion $\texttt{ggT} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, welche den größten gemeinsamen Teiler zweier Zahlen berechnet. *Hinweis:* Modellieren Sie Euklids Algorithmus.

Aufgabe 54: **4 Punkte**

In dieser Aufgabe betrachten wir zwei modifizierte Halteprobleme. Gegeben ist eine Turing-Maschine $\mathcal{A}$ über dem Eingabealphabet Σ, eine Eingabe $w \in \Sigma^*$ und ein $k \in \mathbb{N}$. Geben Sie zu jedem der beiden folgenden Probleme an, ob sie entscheidbar sind oder nicht. Wenn Ihre Antwort "ja" ist, dann begründen Sie dies durch Angabe eines entsprechenden Algorithmus. Wenn Ihre Antwort "nein" ist, dann beweisen Sie dies, indem Sie folgendes zeigen: Wenn es entscheidbar wäre, dann müsste auch ein anderes Problem entscheidbar sein, von dem mal allerdings bereits weiß, dass es nicht entscheidbar ist.

a) $\mathcal{A}$ hält, angesetzt auf w, nach $\leq k$ Schritten.

b) $\mathcal{A}$ hält, angesetzt auf w, nach $> k$ Schritten.

Aufgabe 55: **4 Punkte**

In Aufgabe 46 wurde das Problem, zu einer gegebenen Funktion zu bestimmen, ob diese total ist, als intuitiv nicht berechenbar identifiziert. In dieser Aufgabe wird zuerst einmal gezeigt, dass dies auch wirklich so ist.

a) Das Totalitätsproblem ist das folgende: Gegeben eine berechenbare Funktion $f : \mathbb{N}^n \to \mathbb{N}$ (z.B. als Turing-Maschine, `WHILE`-Programm, etc.), entscheide, ob f total ist. Zeigen Sie, dass das Totalitätsproblem unentscheidbar ist.

b) Für zwei Turing-Maschinen $\mathcal{A}_1$ und $\mathcal{A}_2$ über dem Alphabet Σ schreiben wir $\mathcal{A}_1 \sqsubseteq \mathcal{A}_2$, falls $\mathcal{A}_1$ höchstens dann hält, wenn auch $\mathcal{A}_2$ hält, wenn also für alle $w \in \Sigma^*$ gilt: falls $\mathcal{A}_1$ angesetzt auf w hält, dann hält auch $\mathcal{A}_2$ angesetzt auf w.

   Zeigen Sie, dass das folgende Problem unentscheidbar ist: Gegeben zwei Turing-Maschinen $\mathcal{A}_1$ und $\mathcal{A}_2$, gilt $\mathcal{A}_1 \sqsubseteq \mathcal{A}_2$?

Hinweis: Zeigen Sie bei (a), dass auch das Halteproblem entscheidbar wäre, wenn das Totalitätsproblem entscheidbar ist. Benutzen Sie bei (b) das Resultat aus (a) in derselben Weise. In beiden Fällen bietet es sich an, für ein $w \in \Sigma^*$ die konstante Funktion $f_w : \Sigma^* \to \Sigma^*$, $f(v) = w$ zu benutzen.