Technische Universität München
Fakultät für Informatik
Prof. Tobias Nipkow, Ph.D.
Dr. Werner Meixner, Alexander Krauss

Sommersemester 2008
Übungsblatt 9
16. Juni 2008

Einführung in die Theoretische Informatik

Abgabetermin: 23. Juni 2008 **vor** *der Vorlesung in die THEO Briefkästen*

Hausaufgabe 1 (5 Punkte)

Wir wollen eine Kodierung von Listen beliebiger Länge in natürliche Zahlen mit primitiver Rekursion programmieren. Im Gegensatz zur $\langle\cdot,\ldots,\cdot\rangle$-Funktion zur Kodierung von k-Tupeln, soll hier auch die *Länge* der Liste eindeutig aus der Zahl abzulesen sein.
Dafür soll eine Konstante $\mathit{empty} \in \mathbb{N}$ und primitiv rekursive Funktionen

$$\begin{aligned} \mathit{cons} &: \mathbb{N} \times \mathbb{N} \to \mathbb{N} & \qquad \mathit{head} &: \mathbb{N} \to \mathbb{N} \\ \mathit{isempty} &: \mathbb{N} \to \{0,1\} & \mathit{tail} &: \mathbb{N} \to \mathbb{N} \\ \mathit{drop} &: \mathbb{N} \times \mathbb{N} \to \mathbb{N} \end{aligned}$$

definiert werden. Dabei repräsentiert *empty* die leere Liste, und $\mathit{cons}(n, l)$, die Liste l, an die man vorne das Element n anhängt. Bei einer nichtleeren Liste l soll $\mathit{head}(l)$ das erste Element liefern, und $\mathit{tail}(l)$ die Restliste. $\mathit{drop}(n, l)$ soll die ersten n Elemente aus der Liste l entfernen.

1. Geben Sie mit Hilfe der Paarfunktion c aus der Vorlesung und den passenden Projektionen p_1 und p_2 primitiv rekursive Definitionen für *empty*, *cons*, *isempty*, *head*, *tail* und *drop*, und zeigen Sie, dass folgende Gleichungen erfüllt sind:

$$\begin{aligned} \mathit{isempty}(\mathit{empty}) &= 1 & \qquad \mathit{drop}(0, l) &= l \\ \mathit{isempty}(\mathit{cons}(n, l)) &= 0 & \mathit{drop}(n+1, l) &= \mathit{tail}(\mathit{drop}(n, l)) \\ \mathit{head}(\mathit{cons}(n, l)) &= n \\ \mathit{tail}(\mathit{cons}(n, l)) &= l \end{aligned}$$

   Gehen Sie in dieser Teilaufgabe strikt nach Definition 4.30 vor, und geben Sie jeweils die Instanz des Schemas der primitiven Rekursion bzw. Komposition an, die Sie verwenden.

2. Geben Sie eine primitiv rekursive Definition von $\mathit{nth} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, so dass $\mathit{nth}(n, l)$ das n-te Element der Liste l liefert. *Hinweis: Verwenden Sie drop*

3. Warum lässt sich die Funktion $\mathit{len} : \mathbb{N} \to \mathbb{N}$, die die Länge einer Liste liefert, nicht ohne weiteres mit dem Schema der primitiven Rekursion definieren?

   Zeigen Sie, dass *len* trotzdem primitiv rekursiv ist. *Hinweis: Verwenden Sie drop und beschränkte Maximierung.*

Hausaufgabe 2 (5 Punkte)

Sei $a : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ die Ackermann-Funktion.

1. Zeigen Sie, dass $f(m, n) := (a(m, n))^2$ nicht primitiv rekursiv ist.

2. Zeigen Sie, dass $g(m, n) := \min(3, a(m, n))$ primitiv rekursiv ist.

Hausaufgabe 3 (5 Punkte)

Die Funktion $a' : \mathbb{N} \to \mathbb{N}$ sei gegeben durch $a'(n) = a(n, n)$ mit der Ackermann-Funktion a. Sei $W_{a'} = \{a'(n) \mid n \in \mathbb{N}\}$. Zeigen Sie, dass die Umkehrfunktion $b' : W_{a'} \to \mathbb{N}$ von a' μ-rekursiv ist.

Hausaufgabe 4 (5 Punkte)

Wir wollen untersuchen, ob sich (ähnlich wie bei den WHILE-Programmen) auch jedes LOOP-Programm in eine "Normalform"

$$\texttt{LOOP } X \texttt{ DO } P \texttt{ END}$$

bringen lässt, so dass P keine Schleifen mehr enthält.

1. Zeigen Sie: Für jedes LOOP-Programm P *ohne Schleifen* gibt es eine Konstante k, so dass für jede Variable x_i gilt:

   $$[x_i]' \leq [x_i] + k$$

   Dabei bezeichnet $[x_i]$ den Wert der Variablen x_i beim Start des Programms, und $[x_i]'$ den Wert der Variablen nach Programmende.

2. Zeigen Sie, dass es kein LOOP-Programm in Normalform geben kann, welches die Quadratfunktion $n \mapsto n^2$ berechnet.

__Hinweis:__ Die als Vorbereitung *bezeichneten Aufgaben werden nicht bewertet und dienen der häuslichen Vorbereitung der Tutoraufgaben, die ebenfalls nicht bewertet werden. Die Abgabe einer Bearbeitung der Vorbereitungsaufgaben zusammen mit der Bearbeitung der Hausaufgaben wird empfohlen. Tutoraufgaben werden in den Übungsgruppen bearbeitet.*

Vorbereitung 1

Welche Aussagen sind wahr? Geben Sie jeweils eine knappe Begründung an.

1. Jede unentscheidbare Sprache enthält eine entscheidbare Teilmenge.
2. Jede Teilmenge einer entscheidbaren Sprache ist entscheidbar.
3. Für jede unentscheidbare Sprache A gibt es eine echte Obermenge, die ebenfalls unentscheidbar ist.
4. Aus 'A entscheidbar' und '$A \cap B$ entscheidbar' folgt 'B entscheidbar'.

Tutoraufgabe 1

1. Sei a die Ackermann-Funktion. Zeigen Sie, dass die folgenden Wertemengen entscheidbar sind.

   (i) $W_a = \{a(n,m) \,|\, n,m \in \mathbb{N}\}$. (i) $W'_a = \{a(n,n) \,|\, n \in \mathbb{N}\}$.

2. Gegeben sei eine berechenbare Auflistung (Codierung) aller Turingmaschinen, die jedem Wort $w \in \{0,1\}^*$ eine Turingmaschine M_w zuordnet. Zeigen Sie, dass die Sprache $L = \{w \,|\, L(M_w) \text{ ist semi-entscheidbar}\}$ entscheidbar ist.

Vorbereitung 2

Wir betrachten wieder die in der Vorlesung beschriebene Kodierung von Turingmaschinen durch Wörter über $\Sigma^* = \{0,1\}^*$. Für ein $w \in \Sigma^*$ beschreibt $\varphi_w : \Sigma^* \to \Sigma^*$ dann die Funktion, die von der durch die Turingmaschine M_w berechnet wird. Finden Sie informelle Beschreibungen für die folgenden Mengen:

1. $A = \{w \in \Sigma^* \mid \varphi_w = \Omega\}$.
2. $B = \{w \in \Sigma^* \mid \varphi_w(101) \neq \bot\}$.
3. $C = \{w \in \Sigma^* \mid \varphi_w(\epsilon) = w\}$.
4. $D = \{(u,v,w) \in \Sigma^* \times \Sigma^* \times \Sigma^* \mid \varphi_u(w) = \varphi_v(w)\}$.

Tutoraufgabe 2

Zeigen Sie die Unentscheidbarkeit der folgenden Mengen und wenden Sie zum Beweis Techniken der Reduzierbarkeit eines Problems A auf ein Problem B an.

1. $H_I = \{w \,|\, \forall y.\ \varphi_w(y) = y\}$.
2. $H_{\Sigma^*} = \{w \,|\, M_w \text{ hält für alle Eingaben}\}$.
3. Sei $L = \{a^n b^n \,|\, n \in \mathbb{N}\}$. Dann ist $A = \{w \,|\, L(M_w) = L\}$ unentscheidbar.

Tutoraufgabe 3

Zeigen Sie mit Hilfe des Satzes von Rice:

1. $L_1 = \{w \mid L(M_w) \text{ ist kontextfrei}\}$ ist unentscheidbar.
2. $L_2 = \{w \mid M_w \text{ berechnet } 3n + 5\}$ ist unentscheidbar.