Technische Universität München
Fakultät für Informatik
Lehrstuhl Informatik XIV
Prof. Dr. Ernst W. Mayr
Dr. Werner Meixner

Sommersemester 2015
Lösungsblatt 11
13. Juli 2015

Theoretische Informatik

Hausaufgabe 1 (5 Punkte)

Geben Sie für die Sprache

$$L = \{ww\,;\, w \in \{0,1\}^*\}$$

einen linear beschränkten Automaten (LBA) M an, der L akzeptiert.

Lösung

Die Lösung ergibt sich durch Spezialisierung und Vereinfachung der Lösung der Tutoraufgabe 4 von Blatt 8 für $\Sigma = \{0,1\}$. Wir setzen $\Gamma = \{0,1,a,b,A,B,X,\Box\}$.

$$\begin{array}{lcllcl}
\delta(q_0,\Box) & = & \{(q_f,\Box,N)\}\,, & & & \\
\delta(q_0,0) & = & \{(q_1,a,R)\}\,, & \delta(q_0,1) & = & \{(q_1,b,R)\}\,, \\
\delta(q_1,0) & = & \{(q_1,0,R)\}\,, & \delta(q_1,1) & = & \{(q_1,1,R)\}\,, \\
\delta(q_1,A) & = & \{(q_2,A,L)\}\,, & \delta(q_1,B) & = & \{(q_2,B,L)\}\,, \\
\delta(q_1,\Box) & = & \{(q_2,\Box,L)\}\,, & & & \\
\delta(q_2,0) & = & \{(q_3,A,L)\}\,, & \delta(q_2,1) & = & \{(q_3,B,L)\}\,, \\
\delta(q_3,0) & = & \{(q_4,0,L)\}\,, & \delta(q_3,1) & = & \{(q_4,1,L)\}\,, \\
\delta(q_3,a) & = & \{(q_5,a,N)\}\,, & \delta(q_3,b) & = & \{(q_5,b,N)\}\,, \\
\delta(q_4,0) & = & \{(q_4,0,L)\}\,, & \delta(q_4,1) & = & \{(q_4,1,L)\}\,, \\
\delta(q_4,a) & = & \{(q_0,a,N)\}\,, & \delta(q_4,b) & = & \{(q_5,b,N)\}\,.
\end{array}$$

Mit Erreichen des Zustands q_5 gilt, dass das Eingabewort gerade Länge besitzt. Ausserdem besteht jetzt das erste Teilwort aus Kleinbuchstaben und das zweite Teilwort aus Großbuchstaben. Der Schreib-/Lesekopf steht auf dem letzten Buchstaben des ersten Teilworts.

$$\begin{array}{lcllcl}
\delta(q_5,a) & = & \{(q_a,X,R)\}\,, & \delta(q_5,b) & = & \{(q_b,X,R)\}\,, \\
\delta(q_a,X) & = & \{(q_a,X,R)\}\,, & \delta(q_b,X) & = & \{(q_b,X,R)\}\,, \\
\delta(q_a,A) & = & \{(q_a,A,R)\}\,, & \delta(q_b,A) & = & \{(q_b,A,R)\}\,, \\
\delta(q_a,B) & = & \{(q_a,B,R)\}\,, & \delta(q_b,B) & = & \{(q_b,B,R)\}\,, \\
\delta(q_a,\Box) & = & \{(q'_a,\Box,L)\}\,, & \delta(q_b,\Box) & = & \{(q'_b,\Box,L)\}\,, \\
\delta(q'_a,A) & = & \{(q_6,\Box,L)\}\,, & \delta(q_b,\Box) & = & \{(q_6,\Box,L)\}\,, \\
\delta(q_6,A) & = & \{(q_6,A,L)\}\,, & \delta(q_6,B) & = & \{(q_6,B,L)\}\,, \\
\delta(q_6,X) & = & \{(q_6,X,L)\}\,, & & & \\
\delta(q_6,a) & = & \{(q_5,a,N)\}\,, & \delta(q_6,b) & = & \{(q_5,b,N)\}\,, \\
\delta(q_6,\Box) & = & \{(q_f,\Box,N)\}\,. & & &
\end{array}$$

(5P)

Hausaufgabe 2 (5 Punkte)

Sei $P(k, \overline{x})$ ein primitiv-rekursives $(n+1)$-stelliges Prädikat, wobei $\overline{x} = (x_1, \ldots, x_n)$ eine Abkürzung sei für die letzten n Stellen und $n = 0$ den einstelligen Fall $P(k)$ bedeute. In den Beweisen dürfen erweiterte Komposition und erweiterte Schemata benützt werden.

1. Sei $\max \emptyset = 0$. Zeigen Sie, dass die folgende Funktion $q(m, \overline{x})$ primitiv-rekursiv ist.

$$q(m, \overline{x}) = \max\{k\,;\, k \leq m \,\wedge\, P(k, \overline{x})\}\,.$$

   Die entsprechende Aussage aus der Vorlesung ist zum Beweis nicht verwendbar. Begründen Sie diesen Sachverhalt!

2. Zeigen Sie, dass der folgende beschränkte Existenzquantor primitiv-rekursiv ist.

$$Q(m, \overline{x}) := \exists k \leq m :\; P(k, \overline{x})\,.$$

   Hinweis: Es ist von Vorteil, zunächst im Fall $\hat{P}(0, \overline{x}) = 0$ für alle $m \geq 0$ die folgende Gleichung zu beweisen:

$$\hat{Q}(m, \overline{x}) = 1 \dot{-} (1 \dot{-} q(m, \overline{x}))\,.$$

Lösung

1. Wir benutzen die Erweiterungen der Komposition bzw. des rekursiven Schemas wie folgt:

$$\begin{aligned} q(0, \overline{x}) &= 0\,, \\ q(m+1, \overline{x}) &= q(m, \overline{x}) + \hat{P}(m+1, \overline{x}) \cdot ((m+1) \dot{-} q(m, \overline{x}))\,. \end{aligned}$$

   Die Funktion $q(n)$ aus der Vorlesung betraf nur den Fall $n = 0$, d.h. $\overline{x}$ war leer.

   (2P)

2. Wir betrachten den Fall $\hat{P}(0, \overline{x}) = 0$.

   $m = 0$:

   $\hat{Q}(0, \overline{x}) = 0 = 1 \dot{-} (1 \dot{-} q(0, \overline{x}))\,.$

   $m \to m+1$:

   Es gilt $Q(m+1, \overline{x}) = Q(m, \overline{x}) \vee P(m+1, \overline{x})\,.$

   Fall $\hat{Q}(m, \overline{x}) = 1$:

   Wegen $1 = \hat{Q}(m, \overline{x}) = 1 \dot{-} (1 \dot{-} q(m, \overline{x}))$ gilt $q(m, \overline{x}) \geq 1$ und mithin $q(m+1, \overline{x}) \geq 1\,.$ Daraus folgt die Rechnung:

$$\begin{aligned} \hat{Q}(m+1, \overline{x}) &= \hat{Q}(m, \overline{x}) \\ &= 1 \dot{-} (1 \dot{-} q(m, \overline{x})) \\ &= 1 \dot{-} (1 \dot{-} q(m+1, \overline{x}))\,. \end{aligned}$$

Fall $\hat{Q}(m,\overline{x}) = 0$:
Zunächst folgt $\hat{Q}(m,\overline{x}) = \hat{P}(m,\overline{x}) = 0\,.$
Aus $0 = \hat{Q}(m,\overline{x}) = 1 \dot{-} (1 \dot{-} q(m,\overline{x}))$ folgt des Weiteren $q(m,\overline{x}) = 0$:

$$\begin{aligned}
& 1 \dot{-} (1 \dot{-} q(m{+}1,\overline{x})) \\
&\quad = \; 1 \dot{-} (1 \dot{-} (q(m,\overline{x}) + \hat{P}(m{+}1,\overline{x}) \cdot ((m+1) \dot{-} q(m,\overline{x}))) \\
&\quad = \; 1 \dot{-} (1 \dot{-} \hat{P}(m{+}1,\overline{x}) \cdot (m+1)) \\
&\quad = \; \hat{P}(m{+}1,\overline{x}) \\
&\quad = \; \hat{Q}(m{+}1,\overline{x})\,.
\end{aligned}$$

Wir betrachten den Fall $\hat{P}(0,\overline{x}) = 1$.

Dann ist klar, dass für alle m die folgende triviale Gleichung gilt:

$$\hat{Q}(m,\overline{x}) = 1\,.$$

Zusammenfassend kann nun die Funktion $\hat{Q}(m,\overline{x})$ durch Komposition primitiv rekursiver Funktionen dargestellt werden. Dabei wird die Fallunterscheidung durch Komposition dargestellt wie folgt:

$$\hat{Q}(m,\overline{x}) = \hat{P}(0,\overline{x}) + (1 \dot{-} \hat{P}(0,\overline{x}))(1 \dot{-} (1 \dot{-} q(m,\overline{x}))\,.$$

(3P)

Bemerkung: Mit dem genannten Vorteil ist hier gemeint, dass die Funktion $\hat{Q}(m,\overline{x})$ der Teilaufgabe 2 gestützt auf das Ergebnis von Teilaufgabe 1 durch Komposition und ohne neuerliche Verwendung des Rekursionschemas konstruiert wurde. Natürlich ist auch eine alternative Konstruktion mit Verwendung des Rekursionschemas möglich.

Hausaufgabe 3 (5 Punkte)

Ganzzahlige Division $Div(m,n)$ von natürlichen Zahlen m und n ist definiert durch

$$Div(m,0) = 0 \quad \text{und} \quad Div(m,n) = \max\{k\,;\, k{\cdot}n \le m\} \quad \text{für} \quad n \neq 0\,.$$

Zeigen Sie, dass $Div(m,n)$ primitiv-rekursiv ist.

Hinweis: Verwerten Sie die Erkenntnisse aus Hausaufgabe 2 und definieren Sie ein Prädikat $P(k,m,n) := (k{\cdot}n \le m) \wedge (n \neq 0)$. Beweisen Sie zunächst, dass $P(k,m,n)$ primitiv rekursiv ist.

Lösung

P ist primitiv-rekursiv wegen $\hat{P}(k,m,n) = (1 \dot{-} (k{\cdot}n \dot{-} m)) \cdot (1 \dot{-} (1 \dot{-} n))\,.$ (1P)

Nun folgt für $n \neq 0$:

$$\begin{aligned}
Div(m,n) &= \max\{k\,;\, k{\cdot}n \le m\} \\
&= \max\{k\,;\, k \le m \,\wedge\, k{\cdot}n \le m \,\wedge\, n \neq 0\} \\
&= \max\{k\,;\, k \le m \,\wedge\, P(k,m,n)\} \\
&= q(m,m,n)\,.
\end{aligned}$$

Die Gleichung gilt sogar für $n = 0$, denn für $n = 0$ gilt $\hat{P}(k,m,0) = 0$.
Mithin folgt $q(m,m,0) = 0$.
Daraus folgt $Div(m,0) = q(m,m,0) = 0$.

Damit ist $Div(m,n)$ primitiv rekursiv. (4P)

Hausaufgabe 4 (5 Punkte)

Wir betrachten die Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$, die für alle $n \geq 3$ der linearen Rekursion $f(n) = f(n-1) + f(n-3)$ genügt. Außerdem gelte $f(0) = 1, f(1) = 2, f(2) = 3$.

1. Zeigen Sie, dass f primitiv-rekursiv ist.
2. Sei W_f der Wertebereich von f, d. h. $W_f = \{f(n)\,;\, n \in \mathbb{N}_0\}$. Zeigen Sie, dass W_f entscheidbar ist.
3. Sei $g : \mathbb{N}_0 \to \mathbb{N}_0$ die Umkehrfunktion von f auf dem Wertebereich W_f von f, d. h., dass für alle $n \in \mathbb{N}_0$ gilt $n = g(f(n))$ und für $y \notin W_f$ gilt, dass $g(y)$ nicht definiert ist. Zeigen Sie, dass g WHILE-berechenbar ist.

Lösung

1. Dass f primitiv-rekursiv ist, folgt aus dem folgenden *LOOP*-Programm, das den Funktionswert von f für $n \geq 3$ in der Variablen x_2 berechnet.

   $x_0 := 1;\ x_1 := 2;\ x_2 := 3;\ x_3 := n - 2;$
   $LOOP\ x_3\ DO$
   $x_4 := x_0 + x_2;\ x_0 := x_1;\ x_1 := x_2;\ x_2 := x_4;$
   $END;$

   (1P)

2. Offenbar ist f streng monoton wachsend, d. h., es gilt $f(n-1) < f(n)$ für alle $n \in \mathbb{N}$.

   Außerdem gilt $n < f(n)$ für alle $n \geq 0$.

   Die charakteristische Funktion $\chi_{W_f}(m)$ ist damit wie folgt berechenbar.

   $$\chi_{W_f}(m) = \begin{cases} 1 & : \quad (\exists n \leq m)[f(n) = m]\,, \\ 0 & : \quad \text{sonst}\,. \end{cases}$$

   (2P)

3. Wir betrachten folgendes Programm, das offenbar durch ein *WHILE*-Programm darstellbar ist.

   $n := 0;\ z := f(n);$
   while $z \neq y$ **do**
   $n := n + 1;\ z := f(n)\ ;$
   end

   $g(y)$ wird in der Variablen n berechnet. (2P)

Zusatzaufgabe 9 (wird nicht korrigiert)

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 $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \times \Gamma \to \mathcal{P}_e(Q \times \Gamma^* \times \Gamma^*)$ 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 (deterministische) Turingmaschine $T = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$ kann durch einen 2-Kellerautomaten $K = (Q', \Sigma, \Gamma', \delta', q_0', Z_0, Z_0', F')$ simuliert werden.
Hinweis: Bei einer Simulation müssen die Berechnungen bzw. Konfigurationsänderungen zweier Machinen einander zugeordnet werden können und die akzeptierten Sprachen müssen gleich sein.

Lösung

Die Idee für die Simulation von $T = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$ ist, eine Turing-Konfiguration (α, q, β) darzustellen durch die Konfiguration $(q, \epsilon, \alpha^R Z_0, \beta Z_0')$, wobei α^R gleich dem gespiegelten Wort α ist. Konfigurationsübergänge werden im Kellerautomaten stets mit leeren Eingaben ϵ modelliert.
Beim Start von T liegt das Wort x auf dem Band, der Kopf von T befindet sich über dem ersten Element von $x\Box$. Dies bedeutet, dass wir diese Anfangskonfiguration nach dem Start von K erst herstellen müssen. Dazu dient die folgende Konstruktion.

Sei $K = (Q \cup \{q_0', q_1'\}, \Sigma, \Gamma \cup \{Z_0, Z_0'\}, \delta', q_0', Z_0, Z_0', F)$.
Der 2-Kellerautomat legt zuerst die Eingabe auf den linken Keller. Danach wird der Inhalt des linken Kellers auf den rechten Keller gelegt, wobei der linke Keller fast, d. h. bis auf Z_0, geleert wird. Die beiden Keller entsprechen dann dem Teil des Bandes der Turing-Maschine, der links bzw. rechts vom Lesekopf ist.

$$\begin{array}{lll}
\delta'(q_0', a, *, *) & = \{(q_0', a*, *)\}\,, & \text{Eingabe auf linken Keller legen.} \\
\delta'(q_0', \epsilon, *, *) & = \{(q_1', *, *)\}\,, & \text{Eingabe fertig.} \\
\delta'(q_1', \epsilon, A, *) & = \{(q_1', \epsilon, A*)\}\,, & A \neq Z_0,\ \text{linken Keller} \to \text{rechten Keller.} \\
\delta'(q_1', \epsilon, Z_0, *) & = \{(q_0, Z_0, *)\}\,. &
\end{array}$$

Nun beginnt die eigentliche Simulation der Turingmaschine T, wobei δ' definiert ist für $a \neq Z_0$, $b \neq Z_0'$ durch

$$\begin{array}{rcl}
\delta'(q, \epsilon, a, b) & = & \{(q', ca, \epsilon)\,;\, (q', c, R) \in \delta(q, b)\} \\
& & \cup\, \{(q', \epsilon, ac)\,;\, (q', c, L) \in \delta(q, b)\} \\
& & \cup\, \{(q', a, c)\,;\, (q', c, N) \in \delta(q, b)\}\,,
\end{array}$$

für $a = Z_0, b \neq Z_0'$ durch

$$\begin{array}{rcl}
\delta'(q, \epsilon, Z_0, b) & = & \{(q', cZ_0, \epsilon)\,;\, (q', c, R) \in \delta(q, b)\} \\
& & \cup\, \{(q', Z_0, \Box c)\,;\, (q', c, L) \in \delta(q, b)\} \\
& & \cup\, \{(q', Z_0, c)\,;\, (q', c, N) \in \delta(q, b)\}\,,
\end{array}$$

für $a \neq Z_0, b = Z_0'$ durch

$$\begin{aligned} \delta'(q, \epsilon, a, Z_0') \quad &= \quad \{(q', ca, Z_0')\,;\,(q', c, R) \in \delta(q, \Box)\} \\ &\quad \cup \{(q', \epsilon, acZ_0')\,;\,(q', c, L) \in \delta(q, \Box)\} \\ &\quad \cup \{(q', a, cZ_0')\,;\,(q', c, N) \in \delta(q, \Box)\}\,, \end{aligned}$$

für $a = Z_0, b = Z_0'$ durch

$$\begin{aligned} \delta'(q, \epsilon, Z_0, Z_0') \quad &= \quad \{(q', cZ_0, Z_0')\,;\,(q', c, R) \in \delta(q, \Box)\} \\ &\quad \cup \{(q', Z_0, \Box cZ_0')\,;\,(q', c, L) \in \delta(q, \Box)\} \\ &\quad \cup \{(q', Z_0, cZ_0')\,;\,(q', c, N) \in \delta(q, \Box)\}\,. \end{aligned}$$

Hinweis: *Die Vorbereitungsaufgaben bereiten die Tutoraufgaben vor und werden in der Zentralübung unterstützt. Tutoraufgaben werden in den Übungsgruppen bearbeitet. Hausaufgaben sollen selbstständig bearbeitet und zur Korrektur und Bewertung abgegeben werden.*

Vorbereitung 1

1. Im Folgenden bezeichne $a(n,m)$ die Ackermann-Funktion.

   Berechnen Sie $a(1,6)$ und $a(2,1)$!

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

   (i) $W_a = \{a(n,m)\,;\, n,m \in \mathbb{N}_0\}$.   (ii) $W'_a = \{a(n,n)\,;\, n \in \mathbb{N}_0\}$.

Lösung

1. (i) Die Rekursionsgleichungen liefern

$$\begin{aligned} a(1,6) &= a(0,a(1,5)) = a(1,5)+1 \\ &= a(1,4)+2 \\ &= a(1,3)+3 \\ &= a(1,2)+4 \\ &= a(1,1)+5 \\ &= a(1,0)+6 \\ &= a(0,1)+6 = 2+6 = 8\,. \end{aligned}$$

   (ii) Wir rechnen mit Rekursionsgleichungen in ausführlicher Notation

$$\begin{aligned} a(2,1) &= a(1,a(2,0)) \\ &= a(1,a(1,1)) \\ &= a(1,a(0,a(1,0))) \\ &= a(1,a(0,a(0,1))) \\ &= a(1,a(0,2)) \\ &= a(1,3)) \\ &= a(0,a(1,2)) \\ &= a(0,a(0,a(1,1))) \\ &= a(0,a(0,a(0,a(1,0)))) \\ &= a(0,a(0,a(0,a(0,1)))) \\ &= a(0,a(0,a(0,2))) \\ &= a(0,a(0,3)) \\ &= a(0,4) \\ &= 5\,. \end{aligned}$$

2. (i) Es gilt $W_a \subseteq \mathbb{N}_0$. Grundsätzlich kann man Entscheidbarkeitsfragen auf die Definition zurückführen. Danach ist W_a genau dann entscheidbar, wenn die charakteristische Funktion $\chi_{W_a} : \mathbb{N}_0 \to \{0, 1\}$ berechenbar ist.

   Bekanntlich ist a eine berechenbare Funktion. Leider (oder nicht leider) ist es so, dass nicht jede berechenbare Funktion einen entscheidbaren Wertebereich besitzt. Wir müssen also den konkreten Wertebereich von a analysieren bzw. zeigen, dass für χ_{W_a} ein Algorithmus existiert.

   Nun liefert uns bereits die erste definierende Gleichung der Ackermann-Funktion $a(0, n) = n + 1$ für alle $n \in \mathbb{N}_0$ den Beweis, dass $\mathbb{N} \subseteq W_a$ gilt. Tatsächlich gilt sogar $\mathbb{N} = W_a$, was aber nicht bewiesen werden muss, denn sicher gilt nun entweder $W_a = \mathbb{N}$ oder $W_a = \mathbb{N}_0$. In beiden Fällen ist χ_{W_a} total und berechenbar.

   (ii) Wir berechnen $\chi_{W'_a}(x) = c$ mit dem folgenden WHILE-Programm, wobei wir uns auf ein WHILE-Programm zur Berechnung der Ackermann-Funktion $a(n, n)$ stützen.

   $x_0 := x$; $c_0 := 0$;
   $n_0 := x_0$;
   LOOP n_0 DO
   $n_0 := n_0 - 1$;
   $a_0 := a(n_0, n_0)$;
   IF $a_0 = x_0$ THEN $c_0 := 1$ END END ;
   $c := c_0$

   Die Korrektheit des Programms ergibt sich aus der strengen Monotonie von $a(n, n)$ zusammen mit $n < a(n, n)$.

   Mit den Eigenschaften der Ackermann-Funktion nach Vorlesung folgt die strenge Monotonie von a', insbesondere

$$a'(n) = a(n, n) < a(n, n + 1) < a(n + 1, n) < a(n + 1, n + 1) = a'(n + 1) .$$

   Nach Vorlesung gilt auch

$$n < a(n, n) .$$

Vorbereitung 2

Sei $a : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ die Ackermann-Funktion.

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

2. Die Funktion $a' : \mathbb{N}_0 \to \mathbb{N}_0$ sei gegeben durch $a'(n) = a(n, n)$.
   Sei $W_{a'} = \{a'(n) \,;\, n \in \mathbb{N}_0\}$.

   Zeigen Sie, dass die Umkehrfunktion $b' : W_{a'} \to \mathbb{N}_0$ von a' μ-rekursiv ist.

Lösung

1. Wir nehmen an, f sei primitiv-rekursiv, und geben eine primitiv-rekursive Definition von a an. Dazu benötigen wir die (ganzzahlige) Quadratwurzelfunktion, die man leicht mit Hilfe der beschränkten Maximierung als primitiv rekursiv nachweisen kann:
$$sqrt(n) = max\,\{i \le n\,;\, i^2 \dot{-} n = 0\}$$
Dann ist $a(m,n) = sqrt(f(m,n))$, und somit wäre a primitiv-rekursiv... ein Widerspruch.

2. Die Ackermann-Funktion ist nach Vorlesung total und berechenbar, und folglich auch μ-rekursiv. Wegen $a'(n) = a(\pi_1^1(n), \pi_1^1(n))$ ist a' ebenfalls μ-rekursiv.

   Nach Aufgabenstellung können wir von der Existenz einer eindeutigen Umkehrfunktion $b' : W_{a'} \to \mathbb{N}_0$ ausgehen. Die Existenz vo b' folgt z. B. aus der strengen Monotonie der Ackermann-Funktion.

   Wir verwenden den μ-Operator, um die Funktion b' zu definieren. Sei $h : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$, mit
$$h(n,m) = (m \dot{-} a'(n)) + (a'(n) \dot{-} m)\ .$$
   Die Definition ist gerade so gewählt, dass $h(n,m) = 0 \iff a'(n) = m$ gilt.

   Es gilt
$$\begin{aligned}
(\mu h)(m) &= \begin{cases} \min\{n \in \mathbb{N}_0\,;\, h(n,m) = 0\} & \text{falls ein solches } n \text{ existiert und } \underbrace{h(k,m) \ne \bot \text{ für alle } k \le n \text{ gilt}}_{\text{immer erfüllt, da } h \text{ total ist}} \\ \bot & \text{sonst} \end{cases} \\
&= \begin{cases} \min\{n \in \mathbb{N}_0\,;\, a'(n) = m\} & \text{falls ein solches } n \text{ existiert} \\ \bot & \text{sonst} \end{cases} \\
&= \begin{cases} b'(m) & \text{falls } m \in W_{a'} \\ \bot & \text{sonst}\,. \end{cases}
\end{aligned}$$

Vorbereitung 3

Wir betrachten 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 durch die Turingmaschine M_w berechnet wird. Finden Sie informelle Beschreibungen für die folgenden Mengen:

1. $A = \{w \in \Sigma^*\,;\, \varphi_w = \Omega\}\,.$

2. $B = \{w \in \Sigma^*\,;\, \varphi_w(101) \ne \bot\}\,.$

Bemerkung: Die Standard-Turingmaschine, die für ein ungeeignetes Kodewort w gewählt wird, terminiert nie und berechnet deshalb die nirgends definierte Funktion Ω.

Lösung

1. Die Menge aller (Codes der) Turingmaschinen, die die überall undefinierte Funktion Ω berechnen, die also auf keiner Eingabe halten.

2. Die Menge aller (Codes der) Turingmaschinen, die auf der Eingabe 101 anhalten.

Vorbereitung 4

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

1. $H_{\Sigma^*} = \{w\,;\, M_w$ hält für mindestens eine Eingabe$\}$.

2. $C = \{w\,;\, M_w$ berechnet die Funktion g mit $g(n) = 0$ für alle n$\}$.

Lösung

Wir gehen davon aus, dass es eine universelle Turingmaschine U gibt, die die Berechnungen jeder Turingmaschine M_w auf deren Eingabe x simulieren kann, und deshalb insbesondere genau dann hält, wenn M_w hält. Außerdem wurde in der Vorlesung bewiesen, dass das Halteproblem auf leerem Band H_0 nicht entscheidbar ist.

1. Wir reduzieren das bekannte Halteproblem H_0 auf das Problem H_{Σ^*} durch Konstruktion einer totalen und berechenbaren Funktion f wie folgt.

   Es sei $w' = f(w)$ der Code einer Turingmaschine $M_{w'}$, die bei Eingabe eines Wortes y folgendes ausführt: Zunächst wird die Turingmaschine M_w bei leerer Eingabe simuliert (beispielsweise auf einem zweiten Band). Falls M_w hält, dann hält auch $M_{w'}$.

   Offenbar gilt nun $w \in H_0 \Leftrightarrow f(w) \in H_{\Sigma^*}$, d.h. f reduziert H_0 auf H_{Σ^*}.

2. Wir verfahren analog zur Lösung der vorhergehenden Aufgabe und reduzieren mit Hilfe einer Funktion f das Problem H_0 auf C:

   Es sei $w' = f(w)$ der Code einer Turingmaschine $M_{w'}$, die bei Eingabe eines Wortes y folgendes ausführt: Zunächst wird die Turingmaschine M_w bei leerer Eingabe simuliert. Falls M_w hält, dann schreibt $M_{w'}$ eine 0 auf das Band und terminiert.

   Offenbar gilt nun $w \in H_0 \Leftrightarrow f(w) \in C$, d.h. f reduziert H_0 auf C.

Tutoraufgabe 1

1. Falls A auf B mit Funktion f reduzierbar ist, dann gilt $f^{-1}(B) = A$, aber nicht notwendigerweise $f(A) = B$. Beweis!

2. Falls A reduzierbar auf B und B semi-entscheidbar ist, dann ist auch A semi-entscheidbar. Beweis!

3. Sei $B \subseteq \Sigma^*$ mit $B \neq \Sigma^*$ und $B \neq \emptyset$ entscheidbar.
   Zeigen Sie: B ist reduzierbar auf $\Sigma^* \setminus B$.

Lösung

1. Nach Definition gilt $x \in A \Leftrightarrow f(x) \in B$, und wir erinnern an die Definition $f^{-1}(B) = \{x\,;\, f(x) \in B\}$.

   Nun ist $x \in A \Rightarrow f(x) \in B$ gleichbedeutend mit $f(A) \subseteq B$, woraus insbesondere auch $A \subseteq f^{-1}(B)$ folgt. Aus $x \in A \Leftarrow f(x) \in B$ folgt $A \supseteq f^{-1}(B)$. Damit hat man $A = f^{-1}(B)$ bewiesen.

   Durch Angabe eines Beispiels zeigen wir nun, dass i. A. $f(A) \neq B$ gilt.

   Sei $\Sigma^* = \{0, 1\}^*$, $B = \{0\}^* \subseteq \Sigma^*$ und $A = \{0x\,;\, x \in \Sigma^*\}$.

   Die Abbildung $f : \Sigma^* \to \Sigma^*$ mit

   $$f(w) = \begin{cases} 0 & \text{falls } w \in A \\ 1 & \text{falls } w \notin A \end{cases}$$

   ist offenbar total, berechenbar und reduziert A auf B. Es gilt $00 \in B$, aber $00 \notin f(A)$.

   Man bemerkt, dass B beliebig vergrößert werden kann, solange $B \cap f(\overline{A}) = \emptyset$ erfüllt bleibt. Im Beispiel gilt $f(\overline{A}) = \{1\}$.

2. Sei f eine totale, berechenbare Funktion, die A auf B reduziert. Dann gilt für die charakteristische Funktion χ'_B von B

   $$w \in A \Rightarrow f(w) \in B \Rightarrow \chi'_B(f(w)) = 1$$

   und

   $$w \notin A \Rightarrow f(w) \notin B \Rightarrow \chi'_B(f(w)) = \perp$$

   Daraus folgt $\chi'_A(w) = \chi'_B(f(w))$.

   Offenbar ist χ_A berechenbar, mithin ist A semi-entscheidbar.

3. Seien $a \in \Sigma^* \setminus B$ und $b \in B$. Dann definieren wir die Abbildung $f : \Sigma^* \to \Sigma^*$ mit

   $$f(w) = \begin{cases} a & \text{für } \chi_B(w) = 1 \\ b & \text{für } \chi_B(w) = 0 \end{cases}$$

   f ist offenbar total, berechenbar und reduziert B auf $\Sigma^* \setminus B$.

Tutoraufgabe 2

1. Seien L_1 und L_2 rekursiv auflistbare Mengen. Sind die folgenden Mengen L_a und L_b rekursiv auflistbar? Beweisen Sie Ihre Antwort!

   (i) $L_a = L_1 \cup L_2$ (i) $L_b = \{x\,;\, x \in L_1 \Leftrightarrow x \in L_2\}$

2. Seien $L_n \subseteq A$ für alle $n \in \mathbb{N}_0$ rekursiv auflistbar. Zeigen Sie, dass dann

$$L = \bigcup_{i \in \mathbb{N}} L_i$$

   abzählbar, aber nicht notwendigerweise rekursiv auflistbar ist.

Lösung

1. Seien $f_1 : \mathbb{N}_0 \to L_1$ und $f_2 : \mathbb{N}_0 \to L_2$ totale, berechenbare Funktionen, die beide surjektiv sind.

   (i) Wir zeigen, dass L_a rekursiv auflistbar ist. Wir können ohne Einschränkung annehmen, dass L_1 und L_2 nicht leer sind, denn ansonsten ist L_a gleich einer der beiden Mengen, und somit rekursiv auflistbar. Es existieren also Aufzählungsfunktionen f_1 und f_2. Wir definieren

$$f_a(n) = \begin{cases} f_1(m) & \text{falls } n = 2m \\ f_2(m) & \text{falls } n = 2m+1 \end{cases}$$

   f_a ist surjektiv auf $\mathbb{N}_0 \to L_1 \cup L_2$, total und berechenbar.

   (ii) Seien $L_1, L_2 \subseteq \Sigma^*$. Dann gilt $L_b = (L_1 \cap L_2) \cup (\overline{L_1} \cap \overline{L_2})$. Wir zeigen, dass L_b im Allgemeinen nicht rekursiv auflistbar ist. Wir benützen dabei die Äquivalenz, dass eine Menge genau dann rekursiv auflistbar ist, wenn sie semi-entscheidbar ist.

   Es genügt $L_1 = \emptyset$ zu setzen und L_2 so zu wählen, dass L_2 semi-entscheidbar und das Komplement $\overline{L_2}$ nicht semi-entscheidbar ist.

   Sei $L_2 = K = \{w\ ;\ M_w[w]\downarrow\}$. Dann ist $\overline{K}$ nicht semi-entscheidbar und mithin nicht rekursiv auflistbar. Die leere Menge ist nach Definition rekursiv auflistbar.

   Wegen $L_b = \overline{L_2}$ ist damit L_b nicht rekursiv auflistbar.

2. Natürlich ist jede nicht rekursiv auflistbare, aber abzählbare Menge eine abzählbare Vereinigung von einelementigen Mengen, von denen jede einzelne natürlich rekursiv auflistbar ist.

Tutoraufgabe 3

Zeigen Sie mit Hilfe des Satzes von Rice:

1. $L_1 = \{w \in \Sigma^*\,;\, L(M_w) \text{ ist kontextfrei}\}$ ist unentscheidbar.
2. $L_2 = \{w \in \Sigma^*\,;\, \forall n \in \mathbb{N}_0.\, \varphi_w(n) = 3n + 5\}$ ist unentscheidbar.

Lösung

Der Satz von Rice vereinfacht die Reduktionsbeweise. Wir müssen lediglich prüfen, ob die Eigenschaften L_1 und L_2 nicht triviale Eigenschaften der berechneten Funktionen sind.

1. Es gibt mindestens eine Sprache $L \subseteq \Sigma^*$, die nicht kontextfrei ist, denn kontextfreie Sprachen sind entscheidbar, also kann das Halteproblem nicht kontextfrei sein.

   Daraus folgt $L_1 \neq \Sigma^*$.

   Andererseits gibt es mindestens eine kontextfreie Sprache über Σ, d. h. $L_1 \neq \emptyset$.

2. $L_2 \neq \emptyset$, weil es für die Berechnung von $3n + 5$ eine Turingmaschine gibt.

   Andererseits ist klar, dass nicht jede Turingmaschine die Funktion $3n+5$ berechnet, d. h. $L_2 \neq \Sigma^*$.

Tutoraufgabe 4

1. Sei $g : \mathbb{N}_0 \to \mathbb{N}_0$ total und μ-rekursiv, und sei $f : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch die Startwerte $f(0) = 1$ und $f(1) = 2$ zusammen mit der Rekursion

   $$f(n) = g(n) + f(n-1) \cdot f(n-2) \quad \text{für alle} \quad n \in \mathbb{N} \setminus \{1\}\,.$$

   Zeigen Sie die μ-Rekursivität der Funktion f, indem Sie die Erzeugungsregeln für μ-rekursive Funktionen zusammen mit einer Paarfunktion $p : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ und deren Umkehrfunktionen c_1 und c_2 anwenden.

   Hinweis: Sie dürfen zusätzlich zu den Basisfunktionen der primitiven Rekursion die folgenden Funktionen als primitiv-rekursiv annehmen: $plus(m, n)$ $(+)$, $times(m, n)$ $(\cdot)$, $pred(n)$, $p(m, n)$, $c_1(n)$, $c_2(n)$ und die konstante k-stellige Funktion c^k_n. Sie dürfen die erweiterte Komposition und das erweiterte rekursive Definitionsschema benützen. LOOP- und WHILE-Programme sind nicht erlaubt.

2. Sei $f : \mathbb{N}_0 \to \mathbb{N}_0$ definiert durch die Startwerte $f(0) = 1$ und $f(1) = 2$ zusammen mit der Rekursion

   $$f(n) = 1 + f(n-1) \cdot f(n-2) \quad \text{für alle} \quad n \in \mathbb{N} \setminus \{1\}\,.$$

   Zeigen Sie, dass f primitiv-rekursiv ist, indem Sie f durch ein LOOP-Programm darstellen. *IF THEN ELSE* Konstrukte sowie arithmetische Operationen dürfen verwendet werden.

Lösung

1. Sei $k(n) = p(f(n), f(n+1))$.
   Dann gilt

   $$\begin{aligned} k(0) &= p(1, 2)\,, \\ k(n+1) &= p(\, c_2(k(n)),\, g(n+2) + c_2(k(n)) \cdot c_1(k(n))\,)\,. \end{aligned}$$

   Mithin ist k μ-rekursiv.
   Wegen $f(n) = c_1(k(n))$ ist damit auch f μ-rekursiv.

2. Das folgende LOOP-Programm basiert auf den Variablen $x_0, \ldots, x_5$.
   In x_0 wird das Ergebnis $f(n)$ ausgegeben, x_1 enthält beim Start das Argument n.

   $x_2 := x_1 - 1;\ x_3 := 1;\ x_4 := 2;$
   $LOOP\ x_2\ DO$
   $\quad x_5 := x_4 * x_3;$
   $\quad x_5 := x_5 + 1;$
   $\quad x_3 := x_4;\ x_4 := x_5;$
   $END;$
   $x_0 := x_5$
   $IF\ x_1 = 0\ THEN\ x_0 := 1\ END;$
   $IF\ x_1 = 1\ THEN\ x_0 := 2\ END$

   Wenn wir den Satz der Vorlesung benutzen, dann folgt aus der Existenz eines LOOP-Programms für die Funktion f die primitive Rekursivität von f.