Kurs 1653
Einführung in die Theoretische Informatik A
Lösungshinweise zur Klausur vom 15. März 2008

Aufgabe 1

(a) Wir beweisen die Behauptung durch vollständige Induktion nach k.

Induktionsanfang $k = 0$:

Offensichtlich gilt

$$\mathrm{ES}^{3\cdot 0}(1,(0,n,0,\ldots)) = (1,(\sum_{i=n-0+1}^{n} i, n-0, 0, \ldots)),$$

denn $\sum_{i=n+1}^{n} i = 0$.

Induktionsschritt $k \to k+1$:

Sei $k+1 < n$, dann gilt insbesondere $k < n$ und wir dürfen die Induktionsannahme verwenden:

$$\begin{aligned}
&\mathrm{ES}^{3(k+1)}(1,(0,n,0,\ldots)) \\
&\quad = \mathrm{ES}^{3}(1,(\sum_{i=n-k+1}^{n} i, n-k, 0, \ldots)) \qquad \text{[Ind.ann.]} \\
&\quad = \mathrm{ES}^{2}(2,((\sum_{i=n-k+1}^{n} i) + (n-k), n-k, 0, \ldots)) \\
&\quad = \mathrm{ES}(3,(\sum_{i=n-k}^{n} i, n-k-1, 0, \ldots)) \\
&\quad = (1,(\sum_{i=n-(k+1)+1}^{n} i, n-(k+1), 0, \ldots)) \quad [\text{da } n-(k+1) \neq 0]
\end{aligned}$$

womit die Behauptung bewiesen wäre.

(b) Es gilt

$$\begin{aligned}
&\mathrm{ES}^{3(n-1)+3}(1,(0,n,0,0,\ldots)) \\
&\quad = \mathrm{ES}^{3}(1,(\sum_{i=n-(n-1)+1}^{n} i, n-(n-1), 0, \ldots)) \quad [\text{nach Teil (a)}] \\
&\quad = \mathrm{ES}^{2}(2,((\sum_{i=2}^{n} i) + 1, 1, 0, \ldots)) \\
&\quad = \mathrm{ES}(3,(\sum_{i=1}^{n} i, 0, 0, \ldots)) \\
&\quad = (4,(\sum_{i=1}^{n} i, 0, 0, \ldots)).
\end{aligned}$$

Da die letztgenannte Konfiguration eine Endkonfiguration ist, ergibt sich daraus

$$\mathrm{GS}(1,(0,n,0,0,...)) = (4,(\sum_{i=1}^{n} i,0,0,...)).$$

Mit dem Flussdiagramm F aus der Aufgabenstellung erhalten wir also

$$\begin{aligned} f_M(n) &= \mathrm{AC} \circ f_F \circ \mathrm{EC}^{(1)}(n) \\ &= \mathrm{AC} \circ f_F(0,n,0,0,...) \\ &= \mathrm{AC}(\sum_{i=1}^{n} i,0,0,...) \\ &= \sum_{i=1}^{n} i. \end{aligned}$$

Aufgabe 2

Es sei $m : \mathbb{N}^2 \to \mathbb{N}$ die durch

$$m(x,y) := x \cdot y$$

für alle $x, y \in \mathbb{N}$ definierte Multiplikationsfunktion. Wir geben ein WHILE-Programm P an, so dass

$$\mathrm{AC} \circ \tau(P) \circ \mathrm{EC}^{(2)} = m$$

gilt. Wir definieren hierzu:

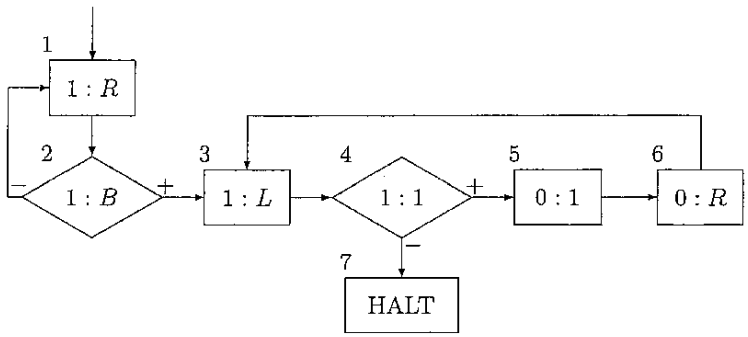
$$\begin{aligned} P &:= (\text{WHILE } R_1 \neq 0 \text{ DO } ((Q_1;Q_2);R_1-)) \qquad \text{mit} \\ Q_1 &:= (\text{WHILE } R_2 \neq 0 \text{ DO } (R_0+;(R_2-;R_3+))) \\ Q_2 &:= (\text{WHILE } R_3 \neq 0 \text{ DO } (R_2+;R_3-)) \end{aligned}$$

Unser WHILE-Programm arbeitet dabei folgendermaßen:

- Q_1 addiert den Inhalt von R_2 auf R_0 und sichert dabei den Wert von R_2 gleichzeitig nach R_3.
- Q_2 kopiert den gesicherten Inhalt von R_2 wieder von R_3 nach R_2.
- $(Q_1;Q_2)$ addiert somit den Inhalt von R_2 auf R_0, ohne ihn dabei zu "zerstören".
- P addiert also insgesamt den Inhalt von R_2 R_1-mal auf R_0, d. h. berechnet das Produkt von R_1 und R_2.

Aufgabe 3

(a) Für die Turingmaschine M über $\Sigma := \{0,1\}$, die durch das folgende Flussdiagramm gegeben ist, gilt $f(w) = f_M(w)$ für alle $w \in \Sigma^*$.



(b) In der ersten Schleife (Marken 1 und 2) wird auf Band 1 der Kopf von M an das rechte Ende der Eingabe $w \in \Sigma^*$ verschoben. Danach wird die Eingabe von rechts nach links abgearbeitet (Marke 1 bis 6). Für jede gelesene 1 auf Band 1 wird von links nach rechts eine 1 auf Band 0 geschrieben.

Sobald auf Band 1 das erste Zeichen ungleich 1 erreicht wird, hält die Maschine an. Dann stehen auf Band 0 links neben dem Kopf genau $z(w)$ Einsen.

Aufgabe 4

Es sei $r : \mathbb{N} \to \mathbb{N}$ eine primitiv-rekursive Funktion.

(a) Wir zeigen, dass die Funktion $f : \mathbb{N}^2 \to \mathbb{N}$, die definiert ist durch

$$f(x,y) := |x - r(y)|$$

für alle $x, y \in \mathbb{N}$, eine primitiv-rekursive Funktion ist.

Mit dem Zusammenhang $|x - y| = (x \dot{-} y) + (y \dot{-} x)$ für alle $x, y \in \mathbb{N}$ gilt

$$\begin{aligned}
f(x,y) &= |x - r(y)| \\
&= (x \dot{-} r(y)) + (r(y) \dot{-} x) \\
&= d(x, r(y)) + d(r(y), x) \\
&= s(d(\mathrm{pr}_1^2(x,y), r(\mathrm{pr}_2^2(x,y))), d(r(\mathrm{pr}_2^2(x,y)), \mathrm{pr}_1^2(x,y))) \\
&= \mathrm{Sub}(s, \mathrm{Sub}(d, \mathrm{pr}_1^2, \mathrm{Sub}(r, \mathrm{pr}_2^2)), \\
&\qquad \mathrm{Sub}(d, \mathrm{Sub}(r, \mathrm{pr}_2^2), \mathrm{pr}_1^2))(x,y)
\end{aligned}$$

für alle $x, y \in \mathbb{N}$ mit den Funktionen s für die Summe und d für die arithmetische Differenz. Es folgt

$$f = \mathrm{Sub}(s, \mathrm{Sub}(d, \mathrm{pr}_1^2, \mathrm{Sub}(r, \mathrm{pr}_2^2)), \mathrm{Sub}(d, \mathrm{Sub}(r, \mathrm{pr}_2^2), \mathrm{pr}_1^2)) .$$

Die Projektionen pr_1^2 und pr_2^2 sind als primitiv-rekursive Grundfunktionen nach Definition 6.2.1 primitiv-rekursiv. Die Summe s und die arithmetische Differenz d sind nach Satz 6.2.2 primitiv-rekursiv und r ist nach Voraussetzung primitiv-rekursiv. f entsteht also durch die Substitution Sub aus primitiv-rekursiven Funktionen. Daher ist f nach Definition 6.2.1 primitiv-rekursiv.

(b) Wir zeigen, dass die Funktion $g : \mathbb{N}^2 \to \mathbb{N}$, die definiert ist durch

$$g(x, y) := \sum_{i=1}^{y} r(i+1)$$

für alle $x, y \in \mathbb{N}$, eine primitiv-rekursive Funktion ist.

Hierzu definieren wir zunächst eine Funktion $h : \mathbb{N}^3 \to \mathbb{N}$ durch

$$h(x, y, z) := z + r(y+2)$$

für alle $x, y, z \in \mathbb{N}$. Es gilt dann

$$\begin{aligned} h(x, y, z) &= z + r(y+2) = z + r(S(S(y))) \\ &= \mathrm{Sub}(s, \mathrm{pr}_3^3, \mathrm{Sub}(r, \mathrm{Sub}(S, \mathrm{Sub}(S, \mathrm{pr}_2^3))))(x, y, z) \end{aligned}$$

für alle $x, y, z \in \mathbb{N}$ mit der Nachfolgerfunktion S. Es folgt

$$h = \mathrm{Sub}(s, \mathrm{pr}_3^3, \mathrm{Sub}(r, \mathrm{Sub}(S, \mathrm{Sub}(S, \mathrm{pr}_2^3)))) .$$

Die Projektionen pr_2^3 und pr_3^3, sowie die Nachfolgerfunktion S sind als primitiv-rekursive Grundfunktionen nach Definition 6.2.1 primitiv-rekursiv. Die Summe s ist nach Satz 6.2.2 primitiv-rekursiv und r ist nach Voraussetzung primitiv-rekursiv. h entsteht also durch die Substitution Sub aus primitiv-rekursiven Funktionen. Daher ist h nach Definition 6.2.1 primitiv-rekursiv.

Es gilt nun

$$g(x, 0) = \sum_{i=1}^{0} r(i+1) = 0 = Z(x)$$

für alle $x \in \mathbb{N}$ mit der einstelligen Nullfunktion Z, sowie

$$\begin{aligned} g(x, y+1) &= \sum_{i=1}^{y+1} r(i+1) \\ &= \sum_{i=1}^{y} r(i+1) + r(y+1+1) \\ &= g(x, y) + r(y+2) \\ &= h(x, y, g(x, y)) \end{aligned}$$

für alle $x, y \in \mathbb{N}$. Es folgt $g = \mathrm{Prk}(Z, h)$, d. h. g entsteht durch primitive Rekursion aus der einstelligen Nullfunktion Z und der Funktion h. Die einstellige Nullfunktion Z ist als primitiv-rekursive Grundfunktion nach Definition 6.2.1 primitiv rekursiv. Da auch h — wie wir bereits gezeigt haben — primitiv-rekursiv ist, entsteht g durch primitive Rekursion aus primitiv-rekursiven Funktionen. g ist also nach Definition 6.2.1 primitiv-rekursiv.

(c) Wir zeigen, dass die Funktion $h :\subseteq \mathbb{N} \to \mathbb{N}$, die definiert ist durch

$$h(x) := \begin{cases} \min\{n \in \mathbb{N} \mid r(n) = x\} & \text{falls } (\exists n \in \mathbb{N}) r(n) = x, \\ \text{div} & \text{sonst,} \end{cases}$$

für alle $x \in \mathbb{N}$, eine μ-rekursive Funktion ist.

Mit dem Zusammenhang "$x = y \iff |x - y| = 0$" für alle $x, y \in \mathbb{N}$ erhalten wir unter Verwendung der Funktion f aus Aufgabenteil (a)

$$\begin{aligned} h(x) &= \begin{cases} \min\{n \in \mathbb{N} \mid r(n) = x\} & \text{falls } (\exists n \in \mathbb{N}) r(n) = x, \\ \text{div} & \text{sonst,} \end{cases} \\ &= \begin{cases} \min\{n \in \mathbb{N} \mid |x - r(n)| = 0\} & \text{falls } (\exists n \in \mathbb{N})\, |x - r(n)| = 0, \\ \text{div} & \text{sonst,} \end{cases} \\ &= \begin{cases} \min\{n \in \mathbb{N} \mid f(x, n) = 0\} & \text{falls } \{n \in \mathbb{N} \mid f(x, n) = 0\} \neq \emptyset, \\ \text{div} & \text{sonst,} \end{cases} \\ &= \mu n[f(x, n) = 0] \\ &= \tilde{\mu}(f)(x) \end{aligned}$$

für alle $x \in \mathbb{N}$. Es folgt $h = \tilde{\mu}(f)$, d. h. h entsteht durch μ-Rekursion aus der Funktion f. Da f nach Aufgabenteil (a) primitiv-rekursiv ist, entsteht h durch μ-Rekursion aus einer primitiv-rekursiven Funktion. h ist also nach Definition 6.3.1 μ-rekursiv.

Aufgabe 5

(a) Wir beweisen, dass es eine totale berechenbare Funktion $f \in \mathrm{R}^{(2)}$ gibt, so dass

$$\varphi_{f(x,y)}(z) = \max\{\varphi_x(z), \varphi_y(z)\}$$

für alle $x, y, z \in \mathbb{N}$ gilt.

Hierfür definieren wir $g :\subseteq \mathbb{N}^2 \to \mathbb{N}$ durch

$$g(\langle x, y\rangle, z) := \max\{\varphi_x(z), \varphi_y(z)\}$$

für alle $x, y, z \in \mathbb{N}$. Es gilt dann

$$g(\langle x, y\rangle, z) = \max\{\varphi_x(z), \varphi_y(z)\} = \max(u_\varphi(x, z), u_\varphi(y, z))$$

für alle $x, y, z \in \mathbb{N}$. Die Maximumfunktion max ist nach Satz 3.2.5 berechenbar und u_φ ist nach dem utm-Theorem berechenbar. Daher ist g als Zusammensetzung berechenbarer Funktionen auch berechenbar, d. h. wir erhalten $g \in \mathrm{P}^{(2)}$. Nach dem smn-Theorem existiert somit eine totale berechenbare Funktion $r \in \mathrm{R}^{(1)}$ mit

$$\varphi_{r\langle x,y\rangle}(z) = g(\langle x, y\rangle, z) = \max\{\varphi_x(z), \varphi_y(z)\}$$

für alle $x, y, z \in \mathbb{N}$.

Wir definieren nun $f : \mathbb{N}^2 \to \mathbb{N}$ durch

$$f(x, y) := r\langle x, y\rangle = r \circ \pi(x, y)$$

für alle $x, y \in \mathbb{N}$, d.h. $f = r \circ \pi$. Da π ebenso wie r eine totale berechenbare Funktion ist, ist auch die Verkettung f eine totale berechenbare Funktion. Es gilt also $f \in \mathrm{R}^{(2)}$, sowie

$$\varphi_{f\langle x,y\rangle}(z) = \varphi_{r\langle x,y\rangle}(z) = \max\{\varphi_x(z), \varphi_y(z)\}$$

für alle $x, y, z \in \mathbb{N}$.

(b) Wir zeigen, dass es eine nicht berechenbare Funktion $f :\subseteq \mathbb{N} \to \mathbb{N}$ gibt, so dass $f(x) = x$ für alle $x \in \mathrm{Def}(f)$ gilt.

Hierzu definieren wir $f :\subseteq \mathbb{N} \to \mathbb{N}$ durch

$$f(x) := \begin{cases} x & \text{falls } x \notin K_\varphi, \\ \mathrm{div} & \text{sonst,} \end{cases}$$

für alle $x \in \mathbb{N}$. Damit gilt $f(x) = x$ für alle $x \in \mathrm{Def}(f)$. Weiterhin gilt $\mathrm{Def}(f) = \mathbb{N} \setminus K_\varphi$, d. h. der Definitionsbereich von f stimmt mit dem Komplement des Selbstanwendbarkeitsproblems K_φ überein.

Wir nehmen nun an, dass unsere Funktion f berechenbar ist. Dann ist nach Definition 8.2.1 $\mathbb{N} \setminus K_\varphi = \mathrm{Def}(f)$ rekursiv-aufzählbar. $\mathbb{N} \setminus K_\varphi$ ist aber nach Satz 8.3.1 nicht rekursiv-aufzählbar. Widerspruch!

Unsere Annahme war also falsch, d. h. die Funktion f ist nicht berechenbar.

(c) Es sei $f :\subseteq \mathbb{N} \to \mathbb{N}$ eine berechenbare Funktion. Wir zeigen, dass es unendlich viele Zahlen $i \in \mathbb{N}$ mit $\varphi_i = f$ gibt.

Sei $F := \{f\}$. Dann ist F nicht leer, d. h. $F \neq \emptyset$. F ist auch nicht gleich $\mathrm{P}^{(1)}$, d. h. es gilt $F \neq \mathrm{P}^{(1)}$, da es mehr als eine berechenbare einstellige Funktion gibt. Nach dem Satz von Rice ist daher die Menge

$$\varphi^{-1}[F] = \{i \in \mathbb{N} \mid \varphi(i) \in F\} = \{i \in \mathbb{N} \mid \varphi_i = f\}$$

nicht rekursiv. Nach Satz 8.1.3 sind alle endlichen Teilmengen von $\mathbb{N}$ rekursiv. Da $\varphi^{-1}[F]$ nicht rekursiv ist, muss $\varphi^{-1}[F]$ also unendlich sein. Damit gibt es unendlich viele Zahlen $i \in \mathbb{N}$ mit $\varphi_i = f$.

Aufgabe 6

(a) Wir definieren eine Funktion $f :\subseteq \mathbb{N} \to \mathbb{N}$ durch

$$f\langle i, n\rangle := u_\varphi(i, n)$$

für alle $i, n \in \mathbb{N}$. Nach dem utm-Theorem ist f berechenbar und es gilt $\mathrm{Def}(f) = H$. Damit ist H rekursiv-aufzählbar.

(b) $\Longleftarrow$: Nach (a) ist H rekursiv-aufzählbar. Gilt nun $A \leq H$, so folgt mit Satz 8.3.5, daß auch A rekursiv-aufzählbar ist.

$\Longrightarrow$: Sei nun A rekursiv-aufzählbar. Dann gibt es ein $i \in \mathbb{N}$ mit $\mathrm{Def}(\varphi_i) = A$. Die Funktion $f : \mathbb{N} \to \mathbb{N}$ mit $f(n) = \langle i, n\rangle$ für alle $n \in \mathbb{N}$ ist offenbar berechenbar und es gilt

$$\begin{aligned} f(n) \in H &\iff \langle i, n\rangle \in H \\ &\iff n \in \mathrm{Def}(\varphi_i) = A \end{aligned}$$

für alle $n \in \mathbb{N}$, d.h. $A \leq H$.

Aufgabe 7

Es sei $\mathrm{sq} : \mathrm{P}^{(1)} \to \mathrm{P}^{(1)}$ die durch

$$\mathrm{sq}(f)(n) := (f(n))^2 \quad \text{für alle } f \in \mathrm{P}^{(1)} \text{ und } n \in \mathbb{N}$$

auf der Menge $\mathrm{P}^{(1)}$ der einstelligen berechenbaren Funktionen definierte Funktion. Wir zeigen, dass sq eine (φ, φ)-berechenbare Funktion ist.

Wir definieren hierzu eine Funktion $h :\subseteq \mathbb{N}^2 \to \mathbb{N}$ durch

$$h(i, n) := (\varphi_i(n))^2 = u_\varphi(i, n) \cdot u_\varphi(i, n)$$

für alle $i, n \in \mathbb{N}$. u_φ ist nach dem utm-Theorem berechenbar und die Multiplikation ist nach Satz 3.2.5 berechenbar. Daher ist h als Zusammensetzung berechenbarer Funktionen eine berechenbare zweistellige Funktion, d. h. $h \in \mathrm{P}^{(2)}$. Nach dem smn-Theorem existiert also eine total-rekursive Funktion $g \in \mathrm{R}^{(1)}$ mit

$$\varphi_{g(i)}(n) = h(i, n) = (\varphi_i(n))^2$$

für alle $i, n \in \mathbb{N}$. Es folgt

$$\mathrm{sq}(\varphi_i(n)) = (\varphi_i(n))^2 = \varphi_{g(i)}(n)$$

für alle $i, n \in \mathbb{N}$ und somit

$$\mathrm{sq} \circ \varphi(i) = \varphi \circ g(i)$$

für alle $i \in \mathbb{N}$. Nach Definition 9.1.3 ist sq daher eine (φ, φ)-berechenbare Funktion.

Aufgabe 8

Es sei $\rho :\subseteq \Sigma^\omega \to \mathbb{R}$ die Cauchy-Darstellung der reellen Zahlen. Wir zeigen, dass die Funktion $t :\subseteq \mathbb{R} \to \mathbb{R}$, die definiert ist durch

$$t(x) := \begin{cases} -1 & \text{falls } x < 0, \\ 1 & \text{falls } x > 0, \\ \mathrm{div} & \text{falls } x = 0, \end{cases}$$

für alle $x \in \mathbb{R}$, eine (ρ, ρ)-berechenbare Funktion ist.

Seien $w_0, w_1 \in \mathrm{Def}(\nu_{\mathrm{rat}})$ Wörter mit $\nu_{\mathrm{rat}}(w_0) = -1$ und $\nu_{\mathrm{rat}}(w_1) = 1$ (dies gilt beispielsweise für $w_0 := -1¢1$ und $w_1 := 1¢1$). Wir konstruieren eine Typ-2-Maschine M, die eine (ρ, ρ)-Realisierung von $t :\subseteq \mathbb{R} \to \mathbb{R}$ berechnet. Die Maschine M soll sich bei Eingabe einer Folge $p = u_0\#u_1\#u_2\#... \in \mathrm{Def}(\rho)$ mit $x := \rho(p)$ wie folgt verhalten: die Wörter $u_0, u_1, ...$ auf dem Eingabeband werden eingelesen, solange bis ein $i \in \mathbb{N}$ gefunden ist, für das gilt

$$\nu_{\mathrm{rat}}(u_i) < -2^{-i} \text{ oder } \nu_{\mathrm{rat}}(u_i) > 2^{-i}.$$

Trifft der erste Fall zuerst zu, dann schreibt die Maschine $q = w_0\#w_0\#w_0\#...$ auf das Ausgabeband (für dieses q gilt offenbar $\rho(q) = -1$); trifft der zweite Fall zuerst zu, dann schreibt die Maschine $q = w_1\#w_1\#w_1\#...$ auf das Ausgabeband (und hier gilt analog $\rho(q) = 1$). Wird kein passendes $i \in \mathbb{N}$ gefunden, dann sucht die Maschine unendlich lange und schreibt gar nichts auf das Ausgabeband.

Wir müssen noch begründen, dass die Maschine M korrekt arbeitet, dass also

$$t\rho(p) = \rho(q)$$

für jede Eingabe $p \in \mathrm{Def}(t\rho)$ und die zugehörige Ausgabe $q = f_M(p)$ gilt. Zunächst erwähnen wir, dass gemäß der Definition der Cauchy-Darstellung ρ der reellen Zahlen für jedes $p = u_0\#u_1\#u_2\#... \in \mathrm{Def}(\rho)$ und für alle $i \in \mathbb{N}$ gilt $|x - \nu_{\mathrm{rat}}(u_i)| \leq 2^{-i}$. Weiterhin gilt

$$|x - \nu_{\mathrm{rat}}(u_i)| \leq 2^{-i} \text{ und } \nu_{\mathrm{rat}}(u_i) < -2^{-i} \Longrightarrow x < 0$$

und analog

$$|x - \nu_{\text{rat}}(u_i)| \leq 2^{-i} \text{ und } \nu_{\text{rat}}(u_i) > 2^{-i} \Longrightarrow x > 0.$$

Daraus folgt bereits, dass wenn die Maschine M zu $p \in \text{Def}(t\rho)$ ein passendes $i \in \mathbb{N}$ findet und folglich q gemäß dem obigen Algorithmus ausgibt, dann gilt auch

$$t\rho(p) = \left\{ \begin{array}{ll} -1 & \text{falls } \rho(p) = x < 0 \\ 1 & \text{falls } \rho(p) = x > 0 \end{array} \right\} = \rho(q).$$

Zu zeigen bleibt, dass die Maschine M für alle $p = u_0\#u_1\#u_2\#... \in \text{Def}(t\rho)$ wirklich ein passendes $i \in \mathbb{N}$ findet. Ist aber $\rho(p) = x < 0$, dann gibt es auch ein $i \in \mathbb{N}$, so dass $x < -2^{-i+1}$ gilt. Daraus folgt dann

$$\nu_{\text{rat}}(u_i) \leq x + 2^{-i} < -2^{-i}.$$

Dieses i wäre also passend und würde folglich von der Maschine M gefunden. Im Falle $\rho(p) = x > 0$ zeigt man analog, dass dann ein $i \in \mathbb{N}$ mit $\nu_{\text{rat}}(u_i) > 2^{-i}$ existiert.

Insgesamt folgt daraus, dass M die gewünschte Funktion t berechnet.