

2 LOOP-Programme

Befehle:

(a) **INPUT** X_1, \dots, X_ℓ

(b) **OUTPUT** X_j

(c) $X = 0$

(d) $X = Y$

(e) $X = X + 1$

(f) **LOOP** X

⋮

END

Satz 2.1: *LOOP-berechenbare Funktionen sind primitiv-rekursiv.*

Satz 2.2: *Jede primitiv-rekursive Funktion ist **LOOP**-berechenbar.*

3 Hierarchien primitiv-rekursiver Funktionen

3.1 LOOP-Schachtelungstiefe d_L

$$d_L(\boxed{X = 0}) = d_L(\boxed{X = Y}) = d_L(\boxed{X = X + 1}) := 0$$

$$d_L\left(\begin{array}{c} \boxed{P} \\ \boxed{Q} \end{array}\right) := \max \left\{ d_L(\boxed{P}), d_L(\boxed{Q}) \right\}$$

$$d_L\left(\begin{array}{c} \text{LOOP } X \\ \boxed{P} \\ \text{END} \end{array}\right) := d_L(\boxed{P}) + 1$$

3.1.1 Programme und Funktionen der Schachtelungstiefe $\leq i$

$$\mathbb{I}_i := \left\{ \begin{array}{l} \mathbf{INPUT} X_1, \dots, X_m \\ P \\ \mathbf{OUTPUT} Y \end{array} : d_L(\boxed{P}) \leq i \right\}$$

$$\mathcal{L}_i := \{f : f \in \mathcal{PR} \wedge \exists \mathbb{IP} (\mathbb{IP} \in \mathbb{I}_i \wedge \mathbb{IP} \text{ berechnet } f)\}$$

Eigenschaft 3.1: \mathcal{L}_i ist abgeschlossen bezüglich Substitution.

3.1.2 Wachstumsfunktionen

$$\begin{aligned}
 t_0(x) &:= x + 2 \\
 t_{n+1}(0) &:= 1 \quad \text{und} \quad t_{n+1}(x+1) := t_n(t_{n+1}(x)) \quad \text{d.h.} \\
 t_{n+1}(x) &:= t_n^x(1) = \underbrace{t_n \circ \dots \circ t_n}_{x \text{ mal}}(1)
 \end{aligned}$$

Eigenschaft 3.2: *Es gelten $t_n \in \mathcal{L}_n$, sowie die Ungleichungen*

$$t_n(x) \geq x + 1 \tag{3.2.1}$$

$$t_n(x+1) > t_n(x) \tag{3.2.2}$$

$$t_{n+1}(x) \geq t_n(x) \quad , \text{ falls } x + n > 0 \tag{3.2.3}$$

$$t_n^{k+1}(x) \geq 2 \cdot t_n^k(x) \quad , \text{ falls } n > 0 \tag{3.2.4}$$

$$t_n^{k+1}(x) \geq t_n^k(x) + x \quad , \text{ falls } n > 0 \tag{3.2.5}$$

Satz 3.3: *Ist $f \in \mathcal{L}_n$, so existiert ein $k \in \mathbb{N}$ derart, dass*

$$\forall x_1 \dots \forall x_m (f(x_1, \dots, x_m) \leq t_n^k(\max\{x_1, \dots, x_m\})) .$$

3.2 Zeitkomplexität von LOOP-Programmen

$\text{Time}_{\text{IP}}(x_1, \dots, x_m) :=$ Anzahl der von IP bei Eingabe
von x_1, \dots, x_m ausgeführten Instruktionen

Lemma 3.4: *Ist $\text{IP} \in \mathbb{IL}_n$, so ist $\text{Time}_{\text{IP}} \in \mathcal{L}_n$.*

Folgerung 3.5: *Ist $f \in \mathcal{L}_n$, so gilt*

$$f(x_1, \dots, x_m) \leq \text{Time}_{\text{IP}}(x_1, \dots, x_m) + \max\{x_1, \dots, x_m\} .$$

für jedes $\text{IP} \in \mathbb{IL}_n$, welches f berechnet.

Satz 3.6: *Es seien $m \geq 1$ und $n \geq 2$. Eine m -stellige Funktion f ist genau dann in \mathcal{L}_n , wenn es ein **LOOP**-Programm IP gibt, welches f berechnet und für das*

$$\exists k(k \in \mathbb{IN} \wedge \text{Time}_{\text{IP}}(x_1, \dots, x_m) \leq t_n^k(\max\{x_1, \dots, x_m\})) \quad \textit{gilt}.$$

Ein Zusammenhang zwischen den Klassen

\mathcal{L}_n und \mathcal{L}_{n+1} .

Satz 3.7: Sind $g_1, \dots, g_m, h_1, \dots, h_m \in \mathcal{L}_n$ und entstehen f_1, \dots, f_m durch simultane Rekursion aus $g_1, \dots, g_m, h_1, \dots, h_m$, so sind $f_1, \dots, f_m \in \mathcal{L}_{n+1}$.

Folgerung 3.8: \mathcal{L}_{n+1} ist der Abschluß der Menge $\mathcal{L}_n \cup \{\mathbf{SR}(g_1, \dots, g_m, h_1, \dots, h_m) : m \in \mathbf{IN} \wedge g_i, h_i \in \mathcal{L}_n\}$ bezüglich Substitution.

Lemma 3.9: Für $n \geq 2$ ist \mathcal{L}_n abgeschlossen bezüglich beschränkter simultaner Rekursion.

3.3 Die GRZEGORCZYK-Hierarchie

$$E_2(x) := x^2 + 2 \text{ bzw. } E_n(0) := 2 \text{ und } E_n(x+1) := E_{n-1}(E_n(x))$$

$$\begin{aligned} \mathcal{E}^n &:= \bigcup_{i \in \mathbb{N}} \mathcal{A}_i^n, \text{ wobei } \mathcal{A}_0^n := \mathcal{A}^0 \cup \{E_1, E_n\} \\ \mathcal{A}_{i+1}^n &:= \mathcal{A}_i^n \cup \{ \mathbf{S}(f; g_1, \dots, g_k) : k \in \mathbb{N} \wedge f, g_1, \dots, g_k \in \mathcal{A}_i^n \} \\ &\quad \cup \{ \mathbf{BR}(g, h; b) : g, h, b \in \mathcal{A}_i^n \} \text{ f\"ur } n \geq 3 \end{aligned}$$

Lemma 3.10: *F\"ur jedes $f \in \mathcal{E}^n$, ($n \geq 2$) gibt es ein $k \in \mathbb{N}$ derart, dass $f(x_1, \dots, x_m) \leq E_n^k(\max\{x_1, \dots, x_m\})$.*

Lemma 3.11: *Sind $g, h \in \mathcal{E}^i$ ($i \neq 1$), so gilt $\mathbf{R}(g, h) \in \mathcal{E}^{i+1}$.*

Lemma 3.12: *Ist $n \geq 3$, so wird \mathcal{E}^n mit Hilfe von Substitution und beschränkter primitiver Rekursion schon aus $\mathcal{A}_0 \cup \{E_1, t_{n-1}\}$ erzeugt.*

Folgerung 3.13: Für $n \geq 2$ gilt $\mathcal{E}^{n+1} \subseteq \mathcal{L}_n$.

Folgerung 3.14: Für $n \geq 2$ gilt: Entstehen die Funktionen f_1, \dots, f_m durch simultane Rekursion aus $g_1, \dots, g_m, h_1, \dots, h_m \in \mathcal{E}^n$, so sind $f_1, \dots, f_m \in \mathcal{E}^{n+1}$.

Folgerung 3.15: $\mathcal{E}^{n+1} \supseteq \mathcal{L}_n$ für $n \geq 0$.

Satz 3.16: Für $n \geq 2$ gilt $\mathcal{E}^{n+1} = \mathcal{L}_n$.

4 Eine effektive Numerierung von \mathcal{PR}^1

P(m) := **INPUT** X_0
 $p(m)$, wobei
 OUTPUT X_1

$p(c(n, c(i, j)))$:= $X_i = X_j$ falls $n = 0 \pmod{5}$

$p(c(n, i))$:= $X_i = 0$ falls $n = 1 \pmod{5}$

$p(c(n, i))$:= $X_i = X_i + 1$ falls $n = 2 \pmod{5}$

$p(c(n, c(i, j)))$:= $\begin{matrix} p(i) \\ p(j) \end{matrix}$ falls $n = 3 \pmod{5}$

$p(c(n, c(i, j)))$:= **LOOP** X_i
 $p(j)$ falls $n = 4 \pmod{5}$

END

$$f_m(x) := f_{\mathbf{P}(m)}(x)$$

$$\Phi(m, x) := f_m(x)$$

Lemma 4.1:

$$\mathcal{PR}^1 = \{f_m : m \in \mathbf{IN}\}$$

$$\mathcal{PR}^k = \{\mathbf{S}(f_m; c^{(k)}) : m \in \mathbf{IN}\}$$

Lemma 4.2: *Die Funktion Φ ist berechenbar, aber nicht primitiv-rekursiv.*