Beispiele: Natürliche Zahlen

Erweiterung von E zu E' mit Regel:

$\hat{f}(x, y) = [is\_null(x) \rightarrow y, \hat{0}]$ ($\hat{f}$ überladen).

$\hat{f}$ implementiert die Funktion $F : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

$$F(x, y) = \begin{cases} y & x = 0 \\ 0 & x \neq 0 \end{cases} \qquad \begin{array}{l} \hat{f}(\hat{0}, \hat{y}) \xrightarrow{*} \hat{y} \\ \hat{f}(\hat{s}(x), \hat{y}) \xrightarrow{*} \hat{0} \end{array}$$

Es gilt aber

$\hat{f}(x, \hat{g}(x)) =_{E'} [is\_null(x) \rightarrow \hat{g}(x), \hat{0}]) =_{E'} \hat{f}(x)$

Es ist aber nicht

$f(n) = F(n, g(n))$ für $n > 0$

Will man alle berechenbaren Funktionen implementieren, so lassen sich die Kleeneschen Rekursionsgleichungen nicht direkt Übertragen, da hierbei die Komposition partieller Funktionen benötigt wird.

Darstellung Primitiv Rekursive Funktionen

Die Klasse $\mathfrak{P}$ enthält die Funktionen
$s = \lambda x.x + 1, \pi_i^n = \lambda x_1, ..., x_n.x_i,$ sowie $c = \lambda x.0$ auf $\mathbb{N}$ und
ist abgeschlossen gegen Einsetzung und primitive Rekursion, d.h.

$f(x_1, ..., x_n) = g(h_1(x_1, ..., x_n), ..., h_r(x_1, ..., x_n))$ bzw.

$$f(x_1, ..., x_n, 0) = g(x_1, ..., x_n)$$
$$f(x_1, ..., x_n, y + 1) = h(x_1, ..., x_n, y, f(x_1, ..., x_n, y))$$

Behauptung: $f \in \mathfrak{P}$ ist implementierbar durch $(\hat{f}, E_{\hat{f}}, \mathfrak{I})$

Idee: Zeige für geeignetes $E_{\hat{f}}$:

$\hat{f}(\hat{k_1}, ..., \hat{k_n}) \xrightarrow{*}_{E_{\hat{f}}} \widehat{f(k_1, ..., k_n)}$ mit $E_{\hat{f}}$ konfluent und terminierend.

Annahme: *FUNKT* (Signatur) enthalte für $n \in \mathbb{N}$ abzählbar viele Funktionssymbole mit Stelligkeit n.

Darstellung Primitiv Rekursive Funktionen

Satz 10.5 *Zu jeder endlichen Menge* $A \subset FUNKT \setminus \{\hat{0}, \hat{s}\}$ *die Ausnahmemenge, und jeder Funktion* $f : \mathbb{N}^n \to \mathbb{N}, f \in \mathfrak{P}$ *gibt es* $\hat{f} \in FUNKT$ *und* $E_{\hat{f}}$ *endlich konfluent und terminierend mit* $(\hat{f}, E_{\hat{f}}, \mathfrak{I})$ *implementiert* f *und keine der Gleichungen in* $E_{\hat{f}}$ *enthält Funktionssymbole aus* A.

Beweis: Induktion über Aufbau von $\mathfrak{P}$: $\hat{0}, \hat{s} \notin A$. Setze $A' = A \cup \{\hat{0}, \hat{s}\}$

- $\hat{s}$ implementiert s mit $E_{\hat{s}} = \emptyset$
- $\hat{\pi}_i^n \in FUNKT^n \setminus A'$ implem. π_i^n mit $E_{\hat{\pi}_i^n} = \{\hat{\pi}_i^n(x_1, \ldots, x_n) \to x_i\}$
- $\hat{c} \in FUNKT^1 \setminus A'$ implementiert c mit $E_{\hat{c}} = \{\hat{c}(x) \to 0\}$
- Einsetzung: $[\hat{g}, E_{\hat{g}}, A_0]$, $[\hat{h}_i, E_{\hat{h}_i}, A_i]$ mit $A_i = A_{i-1} \cup \{f \in FUNKT : f \in E_{\hat{h}_{i-1}}\} \setminus \{\hat{0}, \hat{s}\}$. Sei $\hat{f} \in FUNKT \setminus A'_r$ und $E_{\hat{f}} = E_{\hat{g}} \cup \bigcup_1^r E_{\hat{h}_i} \cup \{\hat{f}(x_1, \ldots, x_n) \to \hat{g}(\hat{h}_1(\ldots), \ldots, \hat{h}_r(\ldots))\}$
- Primitive Rekursion: Analog mit den üblichen Gleichungen.

Darstellung Primitiv Rekursive Funktionen

Alle Regeln sind links-linear ohne Überlappungen $\rightsquigarrow$ Konfluenz.
Terminierungskriterium: Sei $\mathfrak{J}: \ FUNKT \to (\mathbb{N}^* \to \mathbb{N})$, d.h $\mathfrak{J}(f) : \mathbb{N}^{st(f)} \to \mathbb{N}$, streng monoton in allen Argumenten. Ist E ein Regelsystem, $l \to r \in E$, $b: VAR \to \mathbb{N}$ (Belegung), gilt $\mathfrak{J}[b](l) > \mathfrak{J}[b](r)$, so terminiert E.
Idee: Verwende die Ackermannsche Funktion als Schranke:
$A(0,y) = y+1, A(x+1,0) = A(x,1), A(x+1,y+1) = A(x, A(x+1,y))$
A ist streng monoton,
$A(1,x) = x+2, A(x,y+1) \leq A(x+1,y), A(2,x) = 2x+3$
Zu $n \in \mathbb{N}$ gibt es ein β_n mit $\quad \sum_1^n A(x_i,x) \leq A(\beta_n(x_1,\dots,x_n),x)$

Definiere $\mathfrak{J}$ durch $\mathfrak{J}(\hat{f})(k_1,\dots,k_n) = A(p_{\hat{f}}, \sum k_i)$ mit geeigneten $p_{\hat{f}} \in \mathbb{N}$.

- $p_{\hat{s}} := 1 :: \mathfrak{J}[b](\hat{s}(x)) = A(1,b(x)) = b(x)+2 > b(x)+1$
- $p_{\hat{\pi}_i^n} := 1 :: \mathfrak{J}[b](\hat{\pi}_i^n(x_1,\dots,x_n)) = A(1, \sum_1^n b(x_i)) > b(x_i)$
- $p_{\hat{c}} := 1 :: \mathfrak{J}[b](\hat{c}(x)) = A(1,b(x)) > 0 = \mathfrak{J}[b](\hat{0})$

Darstellung Primitiv Rekursive Funktionen

- Einsetzung: $f(x_1, ..., x_n) = g(h_1(...), ..., h_r(...))$.
  Setze $c^* = \beta_r(p_{\hat{h}_1}, ..., p_{\hat{h}_r})$ und $p_{\hat{f}} := p_{\hat{g}} + c^* + 2$. Überprüfe
  $\mathfrak{I}[b](\hat{f}(x_1, ..., x_n)) > \mathfrak{I}[b](\hat{g}(\hat{h}_1(x_1, ..., x_n), ..., \hat{h}_r(x_1, ..., x_n)))$
- Primitive Rekursion:
  Setze $m = \ max(p_{\hat{g}}, p_{\hat{f}})$ und $p_{\hat{f}} := m + 3$. Überprüfe
  $\mathfrak{I}[b](\hat{f}(x_1, ..., x_n, 0)) > \mathfrak{I}[b](\hat{g}(x_1, ..., x_n))$ und
  $\mathfrak{I}[b](\hat{f}(x_1, ..., x_n, \hat{s}(y))) > \mathfrak{I}[b](\hat{g}(....))$.
  Verwende $A(m + 3, k + 3) > A(p_{\hat{h}}, k + A(p_{\hat{f}}, k))$
- Durch Induktion zeige
  $\hat{f}(\hat{k}_1, ..., \hat{k}_n) \xrightarrow{*}_{E_{\hat{f}}} \widehat{f(k_1, ..., k_n)}$
- Aus Satz 10.3 folgt die Behauptung.

Darstellung Rekursiver Funktionen

Minimierung:: μ-Operator $\mu_y[g(x_1, ..., x_n, y) = 0] = z$ gdw
i) $g(x_1, ..., x_n, i)$ definiert $\neq 0$ für $0 \leq i < z$ ii) $g(x_1, ..., x_n, z) = 0$

Reguläre Minimierung: μ wird auf totale Funktionen mit
$\forall x_1, ..., x_n \exists y : g(x_1, ..., x_n, y) = 0$

$\mathfrak{R}$ abgeschlossen gegen Ersetzung, Primitive Rekursion und Reguläre Minimierung.

Zeige: Reguläre Minimierung Implementierbar mit Ausnahmemenge A.
$\hat{g}, E_{\hat{g}}$ implementiere g wobei $\hat{g}(\hat{k}_1, ..., \hat{k}_{n+1}) \xrightarrow{*}_{E_{\hat{g}}} \widehat{g(k_1, ..., k_{n+1})}$
Seien $\hat{f}, \hat{f}^+, \hat{f}^*$ neu setze $E_{\hat{f}} := E_{\hat{g}} \cup \{\hat{f}(x_1, ..., x_n) \rightarrow \hat{f}^*(x_1, ..., x_n, \hat{0}),$
$\hat{f}^*(x_1, ..., x_n, y) \rightarrow \hat{f}^+(\hat{g}(x_1, ..., x_n, y), x_1, ..., x_n, y),$
$\hat{f}^+(\hat{0}, x_1, ..., x_n, y) \rightarrow y, \hat{f}^+(\hat{s}(x), x_1, ..., x_n, y) \rightarrow, \hat{f}^*(x_1, ..., x_n, \hat{s}(y))\}$

Behauptung: $(\hat{f}, E_{\hat{f}})$ implementiert die Minimierung von g.

Darstellung Rekursiver Funktionen

Voraussetzung: Zu $k_1, \ldots, k_n \in \mathbb{N}$ gibt es kleinstes $k \in \mathbb{N}$ mit $g(k_1, \ldots, k_n, k) = 0$

Behauptung: Für alle $i \in \mathbb{N}, i \leq k$ gilt $\hat{f}^*(\hat{k}_1, \ldots, \hat{k}_n, (k \,\hat{-}\, i)) \xrightarrow{*}_{E_{\hat{f}}} \hat{k}$

Bew: Induktion nach i:

- $i = 0 :: \hat{f}^*(\hat{k}_1, \ldots, \hat{k}_n, \hat{k}) \to \hat{f}^+(\hat{g}(\hat{k}_1, \ldots, \hat{k}_n, \hat{k}), \hat{k}_1, \ldots, \hat{k}_n, \hat{k}) \xrightarrow{*}_{E_{\hat{g}}}$
  $\hat{f}^+(g(k_1, \hat{\ldots}, k_n, k), \hat{k}_1, \ldots, \hat{k}_n, \hat{k}) \to \hat{k}$
- $i > 0 :: \hat{f}^*(\hat{k}_1, \ldots, \hat{k}_n, k - (\hat{i} + 1)) \to$
  $\hat{f}^+(\hat{g}(\hat{k}_1, \ldots, \hat{k}_n, k - (\hat{i} + 1)), \hat{k}_1, \ldots, \hat{k}_n, k - (\hat{i} + 1) \xrightarrow{*}_{E_{\hat{g}}}$
  $\hat{f}^+(\hat{s}(\hat{x}), \hat{k}_1, \ldots, \hat{k}_n, k - (\hat{i} + 1) \to \hat{f}^*(\hat{k}_1, \ldots, \hat{k}_n, \hat{s}(k - (\hat{i} + 1))) =$
  $\hat{f}^*(\hat{k}_1, \ldots, \hat{k}_n, k \,\hat{-}\, i)) \xrightarrow{*}_{E_{\hat{g}}} \hat{k}$
  Für geeignetes x und Ind. Voraussetzung.
- $E_{\hat{f}}$ ist konfluent und nach Satz 10.3 implementiert $(\hat{f}, E_{\hat{f}})$ die totale Funktion f.
- $E_{\hat{f}}$ ist nicht terminierend.$g(k, m) = \delta_{k,m} \leadsto \hat{f}^*(\hat{k}, k \,\hat{+}\, 1)$ liefert NT-Kette. Lässt sich erreichen.

Darstellung Partiell Rekursiver Funktionen

Problem: Rekursionsgleichungen (Kleenesche Normalform) lassen sich nicht direkt verwenden. Argumente müsen "Zahl" als Wert besitzen. (Siehe Beispiel). Manche Argumente lassen sich noch retten:

$f(x, y) = g(h_1(x, y), h_2(x, y), h_3(x, y))$. g, h_1, h_2, h_3 seien durch Gleichungsmengen als partielle Funktionen implementierbar.

Behauptung: f ist implementierbar. Seien dazu $\hat{f}, \hat{f}_1, \hat{f}_2$ neu und setze:

$\hat{f}(x, y) =$
$\hat{f}_1(\hat{h}_1(x, y), \hat{h}_2(x, y), \hat{h}_3(x, y), \hat{f}_2(\hat{h}_1(x, y)), \hat{f}_2(\hat{h}_2(x, y)), \hat{f}_2(\hat{h}_3(x, y)))$

$\hat{f}_1(x_1, x_2, x_3, \hat{0}, \hat{0}, \hat{0}) = \hat{g}(x_1, x_2, x_3), \quad \hat{f}_2(\hat{0}) = \hat{0}, \quad \hat{f}_2(\hat{s}(x)) = \hat{f}_2(x)$

$(\hat{f}, E_{\hat{g}}, E_{\hat{h}_1}, E_{\hat{h}_2}, E_{\hat{h}_3} \cup \ REST)$ implementiert f.

Satz 10.3 kann nicht angewendet werden.

$(\hat{f}, E_{\hat{g}}, E_{\hat{h}_1}, E_{\hat{h}_2}, E_{\hat{h}_3} \cup \ REST)$ implementiert f.

Wende Definition 10.1 an:
$\curvearrowright$ Für Zahlenterme sei $f(\Im(t_1), \Im(t_2)) = \Im(t)$. Es gibt Zahlenterme $T_i \ (i = 1, 2, 3)$ mit
$g(\Im(T_1), \Im(T_2), \Im(T_3)) = \Im(t)$ und $h_i(\Im(t_1), \Im(t_2)) = \Im(T_i)$.
Voraussetzung: $\hat{g}(T_1, T_2, T_3) =_{E_{\hat{f}}} t$ und $\hat{h}_i(t_1, t_2) =_{E_{\hat{f}}} T_i(i = 1, 2, 3)$ Da die T_i Zahlenterme: $\hat{f}_2(T_i) =_{E_{\hat{f}}} \hat{0}$ d.h. $\hat{f}_2(\hat{h}_i(t_1, t_2)) =_{E_{\hat{f}}} \hat{0} \ \ (i = 1, 2, 3)$.
Also
$\hat{f}(t_1, t_2) =_{E_{\hat{f}}} \hat{f}_1(T_1, T_2, T_3, \hat{0}, \hat{0}, \hat{0}) \rightsquigarrow \hat{f}(t_1, t_2) =_{E_{\hat{f}}} t (=_{E_{\hat{f}}} \hat{g}(T_1, T_2, T_3)$
q $\curvearrowleft$ Für Zahlenterme t_1, t_2, t gelte $\hat{f}(t_1, t_2) =_{E_{\hat{f}}} t$, dann
$\hat{f}_1(\hat{h}_1(t_1, t_2), \hat{h}_2(t_1, t_2), \hat{h}_3(t_1, t_2), \hat{f}_2(\hat{h}_1(t_1, t_2), ...) =_{E_{\hat{f}}} t$. Wäre für ein $i = 1, 2, 3 \quad \hat{f}_2(\hat{h}_i(t_1, t_2))$ nicht $E_{\hat{f}}$ gleich $\hat{0}$, so sind in der $E_{\hat{f}}$ Äquivalenzklasse nur $\hat{f}_1$ Terme. Es git also Zahlenterme T_1, T_2, T_3 mit $\hat{h}_i(t_1, t_2) =_{E_{\hat{f}}} = T_i \ (i = 1, 2, 3)$ (Sonst nur $\hat{f}_2$ Terme äquivalent zu $\hat{f}_2(\hat{h}_i(t_1, t_2)\ )$. Voraussetzung:
$\rightsquigarrow h_i(\Im(T_1), \Im(T_2)) = \Im(T_i), \qquad g(\Im(T_1), \Im(T_2), \Im(T_3)) = \Im(t)$

$\mathfrak{R}_p$ und Normierte Register Maschinen

Definition 10.2 *Programmterme für RM: P_n $(n \in \mathbb{N})$ Sei $0 \leq i \leq n$*
Funktionssymbole: a_i, s_i Konstanten , $\circ$ 2-stellig , W^i 1-stellig
Gewünschte Interpretation:
a_i :: *Inhalt Register i um 1 Erhöhen.*
s_i :: *Inhalt Register i um 1 Erniedrigen $(\dot{-}1)$*
$\circ(M_1, M_2)$ :: *Verkettung $M_1 M_2$ (Erst M_1, dann M_2)*
$W^i(M)$ :: *Solange Inhalt von Register i nicht 0, führe M aus Abk: $(M)_i$*

Beachte: $P_n \subseteq P_m$ für $n \leq m$

Semantik durch partielle Funktionen: $M_e : P_n \times \mathbb{N}^n \to \mathbb{N}^n$

- $M_e(a_i, \langle x_1, \ldots, x_n \rangle) = \langle \ldots x_{i-1}, x_i + 1, x_{i+1} \ldots \rangle$ $(s_i :: x_i \dot{-} 1)$
- $M_e(M_1 M_2, \langle x_1, \ldots, x_n \rangle) = M_e(M_2, M_e(M_1, \langle x_1, \ldots, x_n \rangle))$
- $M_e((M)_i, \langle x_1, \ldots, x_n \rangle) = \begin{cases} \langle x_1, \ldots, x_n \rangle & x_i = 0 \\ M_e((M)_i, M_e(M, \langle x_1, \ldots, x_n \rangle)) & \text{sonst} \end{cases}$

Implementierung normierter Register Maschinen

Lemma 10.1 *M_e kann durch Gleichungssystem implementiert werden.*

Beweis: Sei tup_n n-stelliges Funktionssymbol. Für $t_i \in \mathbb{N}$ $(0 < i \leq n)$ sei $tup_n(\hat{t}_1, \dots, \hat{t}_n)$ Interpretation von $\langle t_1, \dots, t_n \rangle$. Programmterme werden durch sich selbst interpretiert (es sind ja Terme). Für $m \geq n$ ::

$$\begin{array}{lll} P_n & tup_m(\hat{t}_1, \dots, \hat{t}_m) & \text{Syntaktische Ebene} \\ \mathfrak{I} \downarrow & \mathfrak{I} \downarrow & \\ P_n & \langle t_1, \dots, t_m \rangle & \text{Interpretation} \end{array}$$