7. Beispiele primitiv rekursiver Funktionen

Wir beweisen eine Reihe von stärkeren Abschlusseigenschaften der (primitiv) rekursiven Funktionen und benutzen diese zum Nachweis der primitiven Rekursivität gängiger Funktionen.

Weiter führen wir (primitiv) rekursive Mengen und Prädikate ein.

Schliesslich beschreiben wir eine primitiv rekursive Kodierung von endlichen Zahlenfolgen. Diese werden wir im Beweis des Äquivalenzsatzes benutzen.

7.1 Explizite Definitionen

Nach Definition sind die Klassen der primitiv rekursiven und partiell rekursiven Funktionen gegen simultane Substitution abgeschlossen: Für primitiv (partiell) rekursive $g^{(m)}, h_1^{(n)}, ..., h_m^{(n)}$ ist

$$f(x_1, ..., x_n) = g(h_1(x_1, ..., x_n), ..., h_m(x_1, ...x_n))$$

wiederum primitiv (partiell) rekursiv.

Wir wollen nun zeigen, dass F(PRIM) und F(REK) allgemeiner gegen beliebige explizite Definitionen abgeschlossen sind. D.h. sind $g_1, ..., g_m$ primitiv (partiell) rekursive Funktionen und gilt

$f(x_1, ..., x_n) =$ beliebiger korrekt geformter Term über $g_1, ..., g_m$

so ist f wiederum primitiv (partiell) rekursiv.

Der Nachweis hiervon ist etwas technisch. Zunächst müssen wir präzisieren, was wir unter einer expliziten Definition verstehen.

Funktionsterme

Seien $g_1^{(n_1)}, \ldots, g_k^{(n_k)}$ Funktionen. Die (Funktions-)Terme über $g_1, \ldots, g_k$ sind induktiv definiert durch:

(T 1) Jede Zahl und jede (Zahl-)Variable ist ein Term.

(T 2) Sind $t_1, \ldots, t_{n_i}$ Terme, so ist auch $g_i(t_1, \ldots, t_{n_i})$ ein Term.

Mit $V(t)$ bezeichnen wir die im Term t vorkommenden Variablen.

Ein Term t, der keine Variablen enthält ($V(t) = \emptyset$) heißt *geschlossener* Term.

Terme und Funktionen

- Geschlossene Terme stellen Zahlen dar.

- Ist t ein Term, in dem höchstens die Variablen $v_1, \ldots, v_n$ vorkommen, so kann man t als n-stellige Funktion $f = f_{t;v_1,\ldots,v_n}$ auffassen:

$$f(x_1, \ldots, x_n) = t_{v_1,\ldots,v_n}[x_1, \ldots, x_n],$$

  wobei $t_{v_1,\ldots,v_n}[x_1, \ldots, x_n]$ der geschlossene Term ist, den man aus t erhält, indem man alle Vorkommen der Variablen $v_1, \ldots, v_n$ durch die Zahlen $x_1, \ldots, x_n$ ersetzt.

- Eine Funktion f ist explizit über $g_1, ..., g_m$ definierbar, wenn f auf diese Art durch einen Term über $g_1, ..., g_m$ darstellbar ist.

BEISPIEL

Seien $+$ und $*$ die (2-stellige) Addition und Multiplikation. Dann ist

$$t = +(*(2, *(v_1, v_1)), +(*(3, v_2), 4))$$

ein Term über $+, *$, in dem die beiden Variablen v_1 und v_2 vorkommen.

Der geschlossene Term

$$t_{v_1,v_2,v_3}[7, 5, 8] = +(*(2, *(7, 7)), +(*(3, 5), 4))$$

stellt die Zahl 117 dar.

Die 3-stellige Funktion $f = f_{t;v_1,v_2,v_3}$ ist die Funktion

$$f(x, y, z) = t_{v_1,v_2,v_3}[x, y, z] = +(*(2, *(x, x)), +(*(3, y), 4))$$

also in üblicher Infixschreibweise

$$f(x, y, z) = 2 * x * x + 3 * y + 4 = 2x^2 + 3y + 4.$$

Ähnlich sieht man, dass jedes Polynom (in mehreren Veränderlichen) explizit über $+$ und $*$ definierbar ist.

SATZ. F(PRIM) und F(REK) sind gegen explizite Definitionen abgeschlossen.

BEWEIS. Sei $F =$ F(PRIM) oder $F =$ F(REK), seien $g_1^{(n_1)}, \ldots, g_k^{(n_k)} \in F$ und sei (für $\vec{x} = (x_1, \ldots, x_n)$ und $\vec{v} = (v_1, \ldots, v_n)$) f durch

$$f(\vec{x}) = f_{t;v_1,\ldots,v_n}(x_1, \ldots, x_n) = t_{v_1,\ldots,v_n}[x_1, \ldots, x_n] = t_{\vec{v}}[\vec{x}]$$

definiert, wobei t ein Term über $g_1^{(n_1)}, \ldots, g_k^{(n_k)}$ sei.

Wir zeigen $f \in F$ durch Induktion nach dem Aufbau des Termes t.

1. $t = m \in \mathbb{N}$. Dann ist $f(\vec{x}) = m_{\vec{v}}[\vec{x}] = m$, d.h. $f = C_m^n \in F$.

2. $t = v_i \in V$. Dann gilt $f(\vec{x}) = (v_i)_{v_1,\ldots v_n}[x_1, \ldots, x_n] = x_i$, d.h. $f = U_i^n \in F$.

3. $t = g_i^{(n_i)}(t_1, \ldots, t_{n_i})$. Nach I.V. gilt dann $f_j \in F$ für die durch t_j $(1 \leq j \leq n_i)$ definierten Funktionen $f_j := f_{t_j;v_1,\ldots,v_n}$ und es gilt

$$\begin{array}{rclcl} f(\vec{x}) & = & t_{\vec{v}}[\vec{x}] & = & g_i((t_1)_{\vec{v}}[\vec{x}], \ldots, (t_{n_i})_{\vec{v}}[\vec{x}]) \\ & = & g_i(f_1(\vec{x}), \ldots, f_{n_i}(\vec{x})) & = & g_i(f_1, \ldots, f_{n_i})(\vec{x}), \end{array}$$

weshalb $f \in F$, da F gegen simultane Substitution abgeschlossen ist.

Im Folgenden werden wir den Abschluss von F(PRIM) (und F(REK)) unter expliziten Definitionen stillschweigend verwenden. Hierbei werden wir die definierenden Terme nicht formal angeben und auch häufig – soweit üblich – die Infixschreibweise verwenden.

Definieren wir z.B. eine Funktion f durch eine primitive Rekursion

$$f(x, 0) = t_0$$

$$f(x, y+1) = t_1$$

wobei die Terme t_0 und t_1 über primitiv rekursiven Funktionen gebildet sind, wobei bei t_1 zusätzlich $f(x, y)$ als Teilterm vorkommen kann, so ist wegen des Abschlusses von F(PRIM) unter expliziten Funktionen unmittelbar klar, dass $f = \mathsf{PR}(g, h)$ für primitiv rekursives g und h ist, also f selbst auch primitiv rekursiv ist.

Wir geben hierfür einige Beispiele (wobei wir die Rekursionsgleichungen für $+$ und $*$ wiederholen).

BEISPIELE. Die folgenden Funktionen sind primitiv rekursiv:

- $f_1(x, y) = x + y$ (Summe)

$$f_1(x, 0) = x$$
$$f_1(x, y + 1) = S(f_1(x, y))$$

$g_1(x) = x$ ist explizit über $\emptyset$ definiert
$h_1(x, y, z) = S(z)$ ist explizit über $\{S\}$ definiert

- $f_2(x, y) = x * y$ (Produkt)

$$f_2(x, 0) = 0$$
$$f_2(x, y + 1) = f_2(x, y) + x$$

$g_2(x) = 0$ ist explizit über $\emptyset$ definiert
$h_2(x, y, z) = z + x = f_1(z, x)$ ist explizit über $\{+\}$ definiert

- $f_3(x) = x \dot{-} 1$ (Vorgänger)

$$f_3(0) = 0$$
$$f_3(y+1) = y$$

$g_3() = 0$ ist explizit über $\emptyset$ definiert
$h_3(y, z) = y$ ist explizit über $\emptyset$ definiert

- $f_4(x, y) = x \dot{-} y$ (Differenz auf $\mathbb{N}$)

$$f_4(x, 0) = x$$
$$f_4(x, y+1) = f_4(x, y) \dot{-} 1$$

$g_4(x) = x$ ist explizit über $\emptyset$ definiert
$h_4(x, y, z) = z \dot{-} 1 = f_3(z)$ ist explizit über $\{f_3\}$ definiert

- $f_5(x,y) = |x-y|$ (Absolute Differenz)

  $$f_5(x,y) = |x-y| = (x \dot{-} y) + (y \dot{-} x)$$

  f_5 ist explizit über $\{+, \dot{-}\} = \{f_1, f_4\}$ definiert

- $f_6(x,y) = \max(x,y)$ (Maximum)

  $$f_6(x,y) = \max(x,y) = x + (y \dot{-} x)$$

  f_6 ist explizit über $\{+, \dot{-}\} = \{f_1, f_4\}$ definiert

- $f_7(x,y) = \min(x,y)$ (Minimum)

  $$f_7(x,y) = \min(x,y) = \max(x,y) \dot{-} |x-y|$$

  f_7 ist explizit über $\{\max, \dot{-}, |x-y|\} = \{f_6, f_4, f_5\}$ definiert

- $f_8(x) = \overline{sg}(x) = \begin{cases} 1 & \text{falls } x = 0 \\ 0 & \text{falls } x > 0 \end{cases}$ (Negiertes Vorzeichen)

  $f_8(x) = 1 \dot{-} x$

  f_8 ist explizit über $\{\dot{-}\}$ definiert

- $f_9(x) = sg(x) = \begin{cases} 0 & \text{falls } x = 0 \\ 1 & \text{falls } x > 0 \end{cases}$ (Vorzeichen, Signum)

  $f_9(x) = 1 \dot{-} \overline{sg}(x)$

  f_9 ist explizit über $\{\dot{-}, \overline{sg}\}$ definiert

7.2 Beschränkte Summen, Produkte, Maxima und Minima sowie Iterationen

Bislang haben wir gezeigt, dass die binäre Summen-, Produkt-, Maxima- und Minimabildung primitiv rekursiv ist. Wir erweitern dies nun auf endliche Folgen:

Zu einer (partiellen) Funktion $g^{(n+1)}$ definieren wir hierzu:

$$\begin{array}{rcl}
\sigma(g)(\vec{x},u,o) & = & \sum_{i=u}^{o} g(\vec{x},i) \\
\sigma_<(g)(\vec{x},y) & = & \sum_{i<y} g(\vec{x},i) \\
\pi(g)(\vec{x},u,o) & = & \prod_{i=u}^{o} g(\vec{x},i) \\
\pi_<(g)(\vec{x},y) & = & \prod_{i<y} g(\vec{x},i) \\
\mathrm{max}(g)(\vec{x},y) & = & \mathrm{max}\{g(\vec{x},i) : i<y\} \\
\mathrm{min}(g)(\vec{x},y) & = & \mathrm{min}\{g(\vec{x},i) : i<y\}
\end{array}$$

Hierbei vereinbart man, dass leere Summen, Maxima und Minima den Wert 0, leere Produkte den Wert 1 haben. Bei partiellem g sind die neu definierten Funktionen an den Stellen definiert, die von dem Wert von g nur an solchen Stellen abhängen, an denen g definiert ist.

SATZ. F(PRIM) und F(REK) sind gegen beschränkte Summenbildung, beschränkte Produktbildung, beschränkte Maximumbildung und beschränkte Minimumbildung abgeschlossen.

BEWEIS. Wir zeigen hier die Behauptung für die Summenbildung und für $\mathsf{F} = \mathsf{F}(\mathsf{PRIM})$. Sei also $g^{(n+1)} \in \mathsf{F}(\mathsf{PRIM})$.

$\sigma_<(g) \in \mathsf{F}(\mathsf{PRIM})$:

$$\begin{array}{lcl} \sigma_<(g)(\vec{x}, 0) & = & 0 \\ \sigma_<(g)(\vec{x}, y+1) & = & \sigma_<(g)(\vec{x}, y) + g(\vec{x}, y) \end{array}$$

$\sigma(g) \in \mathsf{F}(\mathsf{PRIM})$ folgt hieraus mit der expliziten Definition

$$\sigma(g)(\vec{x}, u, o) = \sigma_<(g)(\vec{x}, o+1) \dot{-} \sigma_<(g)(\vec{x}, u)$$

Iteration

SATZ. F(PRIM) und F(REK) sind gegen Iteration abgeschlossen. D.h. für $f^{(1)} \in$ F(PRIM) (bzw. F(REK)) gilt Iter(f) $\in$ F(PRIM) (bzw. F(REK)), wobei Iter(f)$(x, n) = f^n(x)$.

BEWEIS. Es gilt

$$\begin{array}{lcl} \text{Iter}(f)(x, 0) & = & x \\ \text{Iter}(f)(x, n+1) & = & f(\text{Iter}(f)(x, n)) \end{array}$$

(Für partielles f beachte man, dass Iter(f)$(x, n)\downarrow$ g.d.w. $f^m(x)\downarrow$ für $1 \leq m \leq n$.)

7.3 Primitiv rekursive und rekursive Mengen

DEFINITION. Eine Menge $M \subseteq \mathbb{N}^n$ ist *(primitiv) rekursiv* g.d.w. die charakteristische Funktion c_M von M (primitiv) rekursiv ist.

BEMERKUNGEN. 1. Wir schreiben $M \in$ F(PRIM) bzw. $M \in$ F(REK), falls M primitiv rekursiv bzw. rekursiv ist.

2. Mehrstellige Mengen nennen wir auch *Relationen*. Ferner identifizieren wir Mengen mit *Prädikaten*. D.h. eine Eigenschaft P (von natürlichen Zahlen) identifizieren wir mit der Menge der Zahlen, die diese Eigenschaft haben. Also

$$P(\vec{x}) \text{ („}P\text{ trifft auf }\vec{x}\text{ zu“)} \;\Leftrightarrow\; \vec{x} \in P \;\Leftrightarrow\; c_P(\vec{x}) = 1.$$

Durch diese Identifizierung entsprechen sich aussagenlogische Operationen (Junktoren) und elementare mengentheoretische Operationen:

Prädikate	Mengen
$\neg P$ (Negation: P trifft *nicht* zu)	$\overline{P}$ (Komplement)
$P \vee Q$ (Disjunktion: P *oder* Q trifft zu)	$P \cup Q$ (Vereinigung)
$P \& Q$ (Konjunktion: P *und* Q treffen zu)	$P \cap Q$ (Durchschnitt)

Man kann jeden aussagenlogischen Junktor mittels der oben genannten Junktoren darstellen. Man könnte nach den DeMorganschen Regeln sogar auf $\vee$ oder & verzichten, da

$$\begin{aligned} P \vee Q &\Leftrightarrow \neg(\neg P \,\&\, \neg Q) \\ P \& Q &\Leftrightarrow \neg(\neg P \vee \neg Q) \end{aligned}$$

gelten. Hier betrachten wir als zusätzlichen Junktor nur noch die Implikation

$$P \Rightarrow Q :\Leftrightarrow \neg P \vee Q.$$

THEOREM. F(PRIM) und F(REK) sind gegen die aussagenlogischen Operationen $\neg$, $\vee$ und & abgeschlossen.

BEWEIS.

$$\begin{array}{rcl} c_{\neg P}(\vec{x}) & = & \overline{sg}(c_P(\vec{x})) \\ c_{P\vee Q}(\vec{x}) & = & \max(c_P(\vec{x}), c_Q(\vec{x})) \\ c_{P\&Q}(\vec{x}) & = & \min(c_P(\vec{x}), c_Q(\vec{x})) \end{array}$$

KOROLLAR. Die Relationen $=, <, \leq, >, \geq, \neq, \ldots$ über $\mathbb{N}$ sind primitiv rekursiv.

BEWEIS. Es gilt

$$c_{\leq}(x, y) = sg((y+1) \dot{-} x),$$

weshalb $\leq \in$ F(PRIM). Die übrigen Relationen erhält man aus $\leq$ durch Anwendung der aussagenlogischen Operationen:

$$\begin{array}{rcl} x = y & \Leftrightarrow & x \leq y \ \& \ y \leq x \\ x \neq y & \Leftrightarrow & \neg(x = y) \\ x < y & \Leftrightarrow & x \leq y \ \& \ x \neq y \end{array}$$

u.s.w.

BEMERKUNG. Die Gleichheit auf $\mathbb{N}^n$ ($n \geq 1$) ist ebenfalls primitiv rekursiv. Dies folgt aus der folgenden Charakterisierung der charakteristischen Funktion $c_= : \mathbb{N}^{2n} \to \mathbb{N}$:

$$c_=(x_1, \ldots, x_n, y_1, \ldots, y_n) = \overline{sg}(|x_1 - y_1| + \cdots + |x_n - y_n|)$$

D.h. $c_=$ ist explizit über $+$, $|x - y|$ und $\overline{sg}$ definierbar.

Alternativ können wir die Gleichheit auf $\mathbb{N}^n$ auch durch die Gleichheit auf $\mathbb{N}$ mit Hilfe von Konjunktionen beschreiben:

$$(x_1, \ldots, x_n) = (y_1, \ldots, y_n) \Leftrightarrow x_1 = y_1 \ \& \ \ldots \ \& \ x_n = y_n$$

Auch hieraus ergibt sich die primitive Rekursivität der $2n$-stelligen Gleichheitsrelation unmittelbar.

Wie bei den Funktionen sind die (primitiv) rekursiven Mengen gegen explizite Definitionen abgeschlossen, da die (primitive) Rekursivität einer Menge ja über die (primitive) Rekursivität ihrer charakteristischen Funktion definiert ist. Insbesondere gilt:

LEMMA. Die Klasse der (primitiv) rekursiven Prädikate ist gegen Einsetzung totaler (primitiv) rekursiver Funktionen abgeschlossen. D.h. für (primitiv) rekursives $P \subseteq \mathbb{N}^m$ und totale (primitiv) rekursive Funktionen $f_1^{(n)}, \ldots, f_m^{(n)}$ ist das Prädikat $Q \subseteq \mathbb{N}^n$ mit

$$Q(\vec{x}) \Leftrightarrow P(f_1(\vec{x}), \ldots, f_m(\vec{x}))$$

wiederum (primitiv) rekursiv.

BEWEIS. Es gilt $c_Q = c_P(f_1, \ldots, f_m)$.

Aus den vorhergehenden Ergebnissen folgt, dass der Graph einer totalen (primitiv) rekursiven Funktion ebenfalls (primitiv) rekursiv ist. Im Fall der Rekursivität (nicht aber der primitiven Rekursivität) gilt auch die Umkehrung. (Man vergleiche dies mit unserer früheren Beobachtung, dass eine Funktion genau dann berechenbar ist, wenn ihr Graph entscheidbar ist!)

SATZ. Eine totale Funktion $f : \mathbb{N}^n \to \mathbb{N}$ ist genau dann rekursiv, wenn ihr Graph rekursiv ist. Ist f primitiv rekursiv, so ist der Graph von f sogar primitiv rekursiv.

BEWEIS. Sei f (primitiv) rekursiv. Dann erhält man den Graphen G_f von f durch Einsetzen von f in das Gleichheitsprädikat:

$$(\vec{x}, y) \in G_f \iff f(\vec{x}) = y.$$

Ist umgekehrt der Graph G_f von f rekursiv, so besitzt f die Darstellung

$$f(\vec{x}) = \mu y((\vec{x}, y) \in G_f) = \mu y(\overline{sg}(c_{G_f}))(\vec{x}).$$

Eine weitere wichtige Abschlusseigenschaft der (primitiv) rekursiven Mengen ist der Abschluss gegen beschränkte Quantoren: Für $P \subseteq \mathbb{N}^{n+1}$ entstehen $P_\exists$, $P_\forall \subseteq \mathbb{N}^{n+1}$ aus P durch Anwendung des *beschränkten Existenz- bzw. Allquantors*:

$$\begin{array}{lll} P_\exists(\vec{x}, y) & :\Leftrightarrow & \exists z < y(P(\vec{x}, z)) & :\Leftrightarrow \exists z(z < y \,\&\, P(\vec{x}, z)) \\ P_\forall(\vec{x}, y) & :\Leftrightarrow & \forall z < y(P(\vec{x}, z)) & :\Leftrightarrow \forall z(z < y \Rightarrow P(\vec{x}, z)) \end{array}$$

SATZ: Die Klassen $\mathsf{F} = \mathsf{F(PRIM)}, \mathsf{F(REK)}$ sind gegen den beschränkten Existenz- und Allquantor abgeschlossen.

BEWEIS. $P \in \mathsf{F} \Rightarrow P_\exists \in \mathsf{F}$:

$$\begin{array}{lll} P_\exists(\vec{x}, y) & \Leftrightarrow & \exists z < y(P(\vec{x}, z)) \\ & \Leftrightarrow & \exists z < y(c_P(\vec{x}, z) = 1) \\ & \Leftrightarrow & \sum_{z<y} c_P(\vec{x}, z) \geq 1 \\ & \Leftrightarrow & \sigma_<(c_P)(\vec{x}, y) \geq 1 \end{array}$$

Für $P_\forall$ beobachtet man

$$P_\forall(\vec{x}, y) \Leftrightarrow \forall z < y(P(\vec{x}, z)) \Leftrightarrow \neg\exists z < y(\neg P(\vec{x}, z)) \Leftrightarrow \neg(\neg P)_\exists(\vec{x}, z)$$

BEMERKUNGEN.

Die (primitiv) rekursiven Mengen sind - wie wir noch sehen werden - jedoch nicht gegen die unbeschränkten Quantoren abgeschlossen.

Wegen des Abschlusses gegen Einsetzung primitiv rekursiver Funktionen ist mit $P(\vec{x}, y)$ nicht nur $\exists y < z(P(\vec{x}, y))$ wiederum primitiv rekursiv, sondern auch $\exists y < f(z)\ (P(\vec{x}, y))$ für jedes $f \in$ F(PRIM). Insbesondere ist also auch (mit $f = S$) $\exists y \leq z(P(\vec{x}, y))$ primitiv rekursiv. (Analog für den beschränkten Allquantor.)

Allgemein stellen die bislang gezeigten Abschlusseigenschaften der primitiv rekursiven Mengen sicher, dass jedes Prädikat, das explizit definiert ist über primitiv rekursive Prädikate, die durch die aussagenlogischen Junktoren verknüpft, beschränkt quantifiziert, und in die primitiv rekursive Funktionen eingesetzt sein dürfen, wiederum primitiv rekursiv ist.

BEISPIEL

Die Menge $PZ \subseteq \mathbb{N}$ der Primzahlen ist wegen der folgenden Darstellung primitiv rekursiv.

$$PZ(n) \Leftrightarrow n \geq 2 \;\&\; \forall x < n(\forall y < n(x \cdot y \neq n))$$

Der μ-Operator, der ja die kleinste Nullstelle einer Funktion berechnet, kann auf beliebige Prädikate sinngemäß wie folgt übertragen werden: Für $P \subseteq \mathbb{N}^{n+1}$ ist $f^{(n)} = \mu(P)$ die partielle Funktion, die bei Eingabe $\vec{x}$ das kleinste y mit $P(\vec{x}, y)$ ausgibt und undefiniert ist, falls solch ein y nicht existiert. D.h. aber gerade, dass $\mu(P)(\vec{x}) = \mu(c_{\overline{P}})(\vec{x})$, weshalb – für rekursives P – $\mu(P) \in \mathsf{F}(\mathsf{REK})$ gilt.

Die Klasse F(PRIM) ist gegen den folgenden *beschränkten μ-Operator* abgeschlossen:

$$\mu_b(P)(\vec{x}, y) = \begin{cases} \mu z < y(P(\vec{x}, z)) & \text{falls} \quad \exists z < y(P(\vec{x}, z)) \\ 0 & \text{sonst} \end{cases}$$

(Im folgenden werden wir entsprechend $\mu z < y(P(\vec{x}, y))$ als 0 lesen, falls es kein $z < y$ mit $P(\vec{x}, z)$ gibt.)

SATZ. F(PRIM) ist gegen den beschränkten μ-Operator abgeschlossen.

BEWEIS. Zu $P \subseteq \mathbb{N}^n$ aus F(PRIM) definieren wir zunächst

$$Q(\vec{x}, y) \;\Leftrightarrow\; P(\vec{x}, y) \,\&\, \forall z < y(\neg P(\vec{x}, z))$$

Dann ist $Q \in$ F(PRIM) und $Q(\vec{x}, y)$ gilt genau für das kleinste y mit $P(\vec{x}, y)$ (falls P überhaupt auf ein $(\vec{x}, y)$ zutrifft). Es folgt hieraus, dass

$$\begin{array}{rcl} \mu_b(P)(\vec{x}, y) & = & \mu z < y(P(\vec{x}, z)) \\ & = & \mu z < y(Q(\vec{x}, z)) \\ & = & \sum_{z<y}(z \cdot c_Q(\vec{x}, z)) \end{array}$$

weshalb $\mu_b(P) \in$ F(PRIM).

BEISPIEL

Die Funktion

$$div(x,y) = \begin{cases} x : y & \text{falls } y \text{ Teiler von } x \\ 0 & \text{sonst} \end{cases}$$

ist primitiv rekursiv, da

$$div(x,y) = \mu z \leq x \; (z \cdot y = x).$$

7.4 Weitere Abschlusseigenschaften

Fallunterscheidungen

Wir schließen unsere Untersuchung der Abschlusseigenschaften von F(PRIM) und F(REK) mit dem Nachweis ab, dass diese Klassen F gegen endliche Fallunterscheidungen abgeschlossen sind:

Sind $P_1, \ldots, P_k$ n-stellige Prädikate aus F, die $\mathbb{N}^n$ zerlegen (d.h. $P_1 \cup \cdots \cup P_k = \mathbb{N}^n$ und $P_1, \ldots, P_k$ sind paarweise disjunkt), und sind $f_1^{(n)}, \ldots, f_k^{(n)}$ (partielle) Funktionen aus F, so ist auch

$$f(\vec{x}) = \begin{cases} f_1(\vec{x}) & \text{falls} \quad P_1(\vec{x}) \\ \cdot & \\ \cdot & \\ \cdot & \\ f_k(\vec{x}) & \text{falls} \quad P_k(\vec{x}) \end{cases}$$

wiederum in F.

SATZ. F(REK) und F(PRIM) sind gegen endliche Fallunterscheidungen abgeschlossen.

BEWEIS (für F(PRIM)): Seien $f_1, \ldots, f_k, P_1, \ldots, P_k$ primitiv rekursiv und sei f hieraus wie oben definiert. Dann gilt

$$f(\vec{x}) = \sum_{i=1}^{k} f_i(\vec{x}) \cdot c_{P_i}(\vec{x}).$$

Als Anwendung des Satzes zeigen wir, dass die Klassen F(REK) und F(PRIM) gegen endliche Varianten abgeschlossen sind. Wir nennen hierbei $f^{(n)}$ eine *endliche Variante* von $g^{(n)}$ und schreiben $f \stackrel{*}{=} g$, wenn sich f und g nur in endlich vielen Argumenten unterscheiden. Entsprechend ist M eine *endliche Variante* von M', $M \stackrel{*}{=} M'$, falls $c_M \stackrel{*}{=} c_{M'}$.

SATZ. Jede endliche Menge ist primitiv rekursiv. F(PRIM) und F(REK) sind gegen endliche Varianten abgeschlossen.

BEWEIS. Die (n-stellige) leere Menge $\emptyset^{(n)}$ ist primitiv rekursiv, da $c_{\emptyset^{(n)}} = C_0^n$. Eine endliche nichtleere Menge $M = \{\vec{x}_1, \ldots, \vec{x}_m\} \subseteq \mathbb{N}^n$ ist primitiv rekursiv, da

$$M(\vec{x}) \Leftrightarrow \vec{x} = \vec{x}_1 \vee \cdots \vee \vec{x} = \vec{x}_m.$$

Seien nun $f^{(n)}, g^{(n)}$ totale Funktionen mit $f \overset{*}{=} g$ und $f \in$ F(PRIM). Seien $\vec{x}_1, ..., \vec{x}_k$ die Eingaben, für die sich f und g unterscheiden, und seien $y_1, ..., y_k$ die Werte von g an diesen Stellen. Dann gilt

$$g(\vec{x}) = \begin{cases} y_1 & \text{falls } \vec{x} = \vec{x}_1 \\ \cdots & \cdots \\ y_k & \text{falls } \vec{x} = \vec{x}_k \\ f(\vec{x}) & \text{sonst,} \end{cases}$$

wobei die zur Fallunterscheidung verwendeten Prädikate (co-)endlich und die Teilfunktionen konstant bzw. f sind.

7.5 Primitiv rekursive Kodierung von Folgen

Im Rest dieses Kapitels geben wir primitiv rekursive Kodierungen von endlichen Zahlenfolgen an. Hierzu betrachten wir zunächst Paare von Zahlen, dann Zahlenfolgen beliebiger aber fester Länge (Vektoren), und schließlich beliebige endliche Zahlenfolgen.

Kodierung von Zahlenpaaren

Man erhält eine bijektive Abbildung $\tau : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, d.h. eine Nummerierung der Zahlenpaare, indem man – anschaulich gesprochen – den rechten oberen Quadranten der (diskreten) Zahlenebene vollständig durchläuft, wobei man alle Nebendiagonalen der Reihe nach jeweils von (rechts) unten nach (links) oben durchläuft (Diagramm: s. Tafel).

Formale Definition von τ

Da die n-te Nebendiagonale $ND_n = \{(x, y) : x+y = n\}$ aus $n+1$ Zahlenpaaren besteht und die Nebendiagonale ND_n in der Reihenfolge

$$(n, 0), (n-1, 1), (n-2, 2), \ldots, (0, n)$$

durchlaufen wird, gilt

$$\begin{aligned}
\tau(x, y) &= [(\textstyle\sum_{n=0}^{x+y-1} |ND_n|) + y + 1] - 1 \\
&= (\textstyle\sum_{n=0}^{x+y-1} (n+1)) + y \\
&= (\textstyle\sum_{n=1}^{x+y} n) + y \\
&= \tfrac{(x+y)(x+y+1)}{2} + y \\
&= \tfrac{1}{2}(x^2 + 2xy + y^2 + x + 3y) \\
&= div(x^2 + 2xy + y^2 + x + 3y, 2)
\end{aligned}$$

SATZ. Die Funktion $\tau : \mathbb{N}^2 \to \mathbb{N}$ hat folgende Eigenschaften:

(i) τ ist bijektiv.

(ii) τ ist monoton, d.h.

$$x_1 \leq x_1' \ \ \& \ \ x_2 \leq x_2' \ \ \Rightarrow \ \ \tau(x_1, x_2) \leq \tau(x_1', x_2').$$

Ferner gilt $x_1, x_2 \leq \tau(x_1, x_2)$.

(iii) τ ist primitiv rekursiv.

(iv) Die *Projektionsfunktionen* π_1 und π_2 mit $\pi_i(\tau(x_1, x_2)) = x_i$ $(i = 1, 2)$ sind ebenfalls primitiv rekursiv.

BEWEIS. Wir verzichten auf den Beweis von (i), der anschaulich klar ist. Teil (ii) folgt unmittelbar aus der Definition von τ und der Monotonie der Addition und Multiplikation. (iii) folgt ebenfalls direkt aus der Definition von τ, da diese eine explizite Definition über den primitiv rekursiven Funktionen $+$, $*$, div ist. Zum Beweis von (iv) genügt es, wegen der oben gezeigten Abschlusseigenschaften von F(PRIM) die Projektion π_1 (und analog π_2) durch

$$\pi_1(x) = \mu x_1 \leq x(\exists x_2 \leq x(\tau(x_1, x_2) = x))$$

zu charakterisieren.

Primitiv rekursive Kodierung von Vektoren

Durch Iteration von τ können wir n-Tupel von Zahlen, d.h. Folgen fester Länge über $\mathbb{N}$ kodieren:

DEFINITION. Die Funktionen $\tau_n : \mathbb{N}^n \to \mathbb{N}$ ($n \geq 1$) sind induktiv definiert durch

$$\tau_1(x_1) = x_1$$

$$\tau_{n+1}(x_1, x_2, \ldots, x_{n+1}) = \tau(x_1, \tau_n(x_2, \ldots, x_{n+1}))$$

SATZ. Für $n \geq 1$ gilt:

(i) $\tau_n : \mathbb{N}^n \to \mathbb{N}$ ist bijektiv.

(ii) τ_n ist monoton und $x_1, \ldots, x_n \leq \tau_n(x_1 \ldots x_n)$.

(iii) τ_n ist primitiv rekursiv.

(iv) Die zu τ_n gehörenden Projektionsfunktionen $\pi_i^n : \mathbb{N} \to \mathbb{N}$ mit

$$\pi_i^n(\tau_n(x_1, \ldots, x_n)) = x_i \ \ (1 \leq i \leq n)$$

sind primitiv rekursiv.

BEWEISIDEE. Beweis durch Induktion nach n. Der Fall $n = 1$ ist trivial. Der Induktionsschritt benutzt bei (i) - (iii) die entsprechenden Eigenschaften von τ. Zum Beweis von (iv) beobachtete man:

$$\begin{aligned}\pi_i^{n+1}(x) = \mu x_i \leq x(\exists x_1 \leq x \ldots \exists x_{i-1} \leq x \ \exists x_{i+1} \leq x \ldots \\ \ldots \exists x_{n+1} \leq x(\tau_{n+1}(x_1, \ldots, x_{n+1}) = x))\end{aligned}$$

Kodierung von endlichen Folgen variabler Länge

Abschließend können wir mit Hilfe der Funktionen τ_n nun endliche Zahlenfolgen variabler Länge kodieren.

DEFINITION. Sei $\mathbb{N}^*$ die Menge aller endlichen Folgen über $\mathbb{N}$. Die Funktion $\tau^* : \mathbb{N}^* \to \mathbb{N}$ ist definiert durch:

$$\begin{aligned} &\tau^*(\lambda) = 0 \\ &\tau^*(x_1, \dots, x_n) = \tau(n-1, \tau_n(x_1, \dots, x_n)) + 1 \end{aligned}$$

Grob gesprochen ist $\tau^*(x_1, \dots, x_n)$ das kodierte Paar bestehend aus der Länge n der Folge und der Kodierung $\tau_n(x_1, \dots, x_n)$ der Folge als Vektor (fester Länge). Um τ^* surjektiv zu machen, ersetzt man jedoch n durch $n-1$ und um $\tau^*(x_1, \dots, x_n) \neq \tau^*(\lambda)$ zu sichern, erhöht man den Wert um 1.

LEMMA. $\tau^* : \mathbb{N}^* \to \mathbb{N}$ ist eine Bijektion.

BEWEIS. Dies ergibt sich leicht aus der Bijektivität der Funktionen τ und τ_n für $n \geq 1$.

BEMERKUNGEN.

Der Funktionstyp von τ^* schließt aus, dass τ^* primitiv rekursiv ist: τ^* ist vom Typ $\mathbb{N}^* \to \mathbb{N}$ während primitiv rekursive Funktionen vom Typ $\mathbb{N}^n \to \mathbb{N}$ sind.

Im Folgenden werden wir aber zeigen, dass alle wichtigen Informationen über die Folge $x_1, \ldots, x_n$ aus der Zahl $\tau^*(x_1, \ldots, x_n)$ auf primitiv rekursive Weise entnommen werden können.

Auch sollte man beachten, dass wegen der expliziten Definition

$$\tau^*(x_1, \ldots, x_n) = \tau(n-1, \tau_n(x_1, \ldots, x_n)) + 1$$

von τ^* über den primitiv rekursiven Funktionen $\dot{-}$, $+$, τ und τ_n die Einschränkung von τ^* auf $\mathbb{N}^n$ für jedes feste $n \geq 1$ eine primitiv rekursive Funktion ist.

Die wie folgt definierten 1-stelligen Funktionen l, head, tail und 2-stelligen Funktionen π^*, $\circ$ sind primitiv rekursiv:

$$l(\tau^*(\vec{x})) = |\vec{x}|$$

$$\text{head}(\tau^*(\vec{x})) = \begin{cases} x_1 & \text{falls } \vec{x} = (x_1, \ldots, x_n) \\ 0 & \text{sonst} \end{cases}$$

$$\text{tail}(\tau^*(\vec{x})) = \begin{cases} \tau^*(x_2, \ldots, x_n) & \text{falls } \vec{x} = (x_1, \ldots, x_n) \text{ mit } n \geq 2 \\ \tau^*(\lambda)(= 0) & \text{sonst} \end{cases}$$

$$\pi^*(\tau^*(\vec{x}), i) = \begin{cases} x_i & \text{falls } \vec{x} = (x_1, \ldots, x_n) \text{ mit } n \geq i \geq 1 \\ 0 & \text{sonst} \end{cases}$$

$$\tau^*(\vec{x}) \circ \tau^*(\vec{y}) = \tau^*(\vec{x}, \vec{y})$$

Für die durch x kodierte Folge $\vec{x}$ gibt also $l(x)$ die Länge von $\vec{x}$, head(x) die erste Komponente x_1 von $\vec{x}$ und $\pi^*(x, i)$ allgemein die i-te Komponente x_i von $\vec{x}$ an, während tail(x) die Restfolge von $\vec{x}$ nach Streichen von x_1 ist. Entsprechend kodiert $\circ$ die Konkatenation.

BEWEIS.

$l \in$ F(PRIM):

$$l(x) = \begin{cases} 0 & \text{falls} \quad x = 0 \\ \pi_1(x \dot{-} 1) + 1 & \text{sonst} \end{cases}$$

head $\in$ F(PRIM):

$$\text{head}(x) = \begin{cases} 0 & \text{falls} \quad l(x) = 0 \\ \pi_2(x \dot{-} 1) & \text{falls} \quad l(x) = 1 \\ \pi_1(\pi_2(x \dot{-} 1)) & \text{sonst} \end{cases}$$

NB: $x = \tau^*(x_1) \Rightarrow \pi_2(x \dot{-} 1) = \tau_1(x_1) = x_1$

$x = \tau^*(x_1, \ldots, x_n)$ & $n \geq 2 \Rightarrow \pi_2(x \dot{-} 1) = \tau_n(x_1, \ldots, x_n) = \tau(x_1, \tau_{n-1}(x_2, \ldots, x_n))$

tail $\in$ F(PRIM):

$$\text{tail}(x) = \begin{cases} 0 & \text{falls } l(x) \leq 1 \\ \tau(l(x) \dot{-} 2, \pi_2(\pi_2(x \dot{-} 1))) + 1 & \text{sonst} \end{cases}$$

Man beachte, dass für $x = \tau^*(\vec{x})$ und $\vec{x} = (x_1, \ldots, x_n)$ mit $n \geq 2$,

$$\pi_2(\pi_2(x \dot{-} 1)) = \tau_{n-1}(x_2, \ldots, x_n),$$

woraus sich die Definition von tail(x) im 2. Fall ergibt, wenn man τ_{n-1} noch nach τ^* übersetzt.

$\pi^* \in$ F(PRIM):

Um $\pi^* \in$ F(PRIM) zu zeigen, beobachten wir zunächst, dass die Funktion end(x,y) mit

$$\text{end}(\tau^*(\vec{x}), y) = \begin{cases} \tau^*(x_{y+1}, \ldots, x_n) & \text{falls } \vec{x} = (x_1, \ldots, x_n) \text{ mit } n > y \\ \tau^*(\lambda) = 0 & \text{sonst} \end{cases}$$

primitiv rekursiv ist:

$$\begin{array}{lcl} \text{end}(x, 0) & = & x \\ \text{end}(x, y+1) & = & \text{tail}(\text{end}(x, y)) \end{array}$$

Hiermit lässt sich π^* primitiv rekursiv durch

$$\begin{array}{ll} \pi^*(x, 0) = & 0 \\ \pi^*(x, y+1) = & \begin{cases} \text{head}(\text{end}(x, y)) & \text{falls } l(\text{end}(x, y)) \geq 1 \\ 0 & \text{sonst} \end{cases} \end{array}$$

definieren.

$\circ \in$ F(PRIM): Wir zeigen zunächst, dass es eine primitiv rekursive Funktion $\sigma(n)$ gibt, sodass für jede Folge $\vec{x} = (x_1, \ldots, x_m)$ mit $x_1, \ldots, x_m, m \leq n$ gilt, dass $\tau^*(\vec{x}) < \sigma(n)$ ist. Da nach Definition von τ^* auch $x_1, \ldots, x_m, m \leq \tau^*(\vec{x})$ gilt, folgt hieraus für beliebige Folgen $\vec{x}, \vec{y}$

$$\tau^*(\vec{x}) \circ \tau^*(\vec{y}) = \tau^*(\vec{x}, \vec{y}) < \sigma(\tau^*(x) + \tau^*(y)).$$

Dies erlaubt $\circ$ mit Hilfe des beschränkten μ-Operators zu definieren:

$$\begin{aligned} x \circ y = \mu z < \sigma(x+y)(\forall i < l(x)(\pi^*(z, i+1) = \pi^*(x, i+1)) \\ \&\ \forall i < l(y)(\pi^*(z, l(x)+i+1) = \pi^*(y, i+1))) \end{aligned}$$

Definition von $\sigma \in$ F(PRIM): Für die primitiv rekursive Funktion σ_0 mit

$$\begin{array}{lcl} \sigma_0(n, 0) & = & n \\ \sigma_0(n, m+1) & = & \tau(n, \sigma_0(n, m)) \end{array}$$

gilt (nach Definition von τ_n und wegen der Monotonie von τ und τ_n):

$$\forall \vec{x} = (x_1, \ldots, x_p)([x_1, \ldots, x_p \leq n \,\&\, p \leq m+1] \Rightarrow \tau_p(\vec{x}) \leq \sigma_0(n, m)).$$

Nach Definition von τ^* können wir also definieren:

$$\sigma(n) = \tau(n, \sigma_0(n, n)) + 1.$$

Für kodierte Folgen benutzen wir auch folgende Schreibweise:

$$\begin{aligned}
\langle x_1, \ldots, x_n \rangle &:= \tau^*(x_1, \ldots, x_n) \\
\langle \rangle &:= \tau^*(\lambda) \\
(x)_i &:= \pi^*(x, i) \\
(x)_{i,j} &:= \pi^*(\pi^*(x, i), j)
\end{aligned}$$