Primitiv rekursive Codier- und Decodierfunktionen

Paarungsfunktionen, Codierung von Zahlenfolgen

6.26 Definition

Die **Cauchysche Paarungsfunktion** $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ wird definiert durch

$$\langle x, y \rangle = ((x + y)(x + y + 1) \text{ div } 2) + y$$

Sie ist primitiv rekursiv und bijektiv.

(x, y)	$(0,0)$	$(1,0)$	$(0,1)$	$(2,0)$	$(1,1)$	$(0,2) \cdots$
$\langle x, y \rangle$	0	1	2	3	4	$5 \cdots$

Abzählen auf geeigneten Diagonalen: $x + y$ konstant.
(x, y) kommt auf der Diagonalen $x + y + 1$ vor.

- $x + y$ komplett gefüllte Diagonalen: $1 + 2 + \cdots + (x + y) = (x + y)(x + y + 1) \text{ div } 2$
- In der Diagonalen, in der (x, y) steht, kommen noch y viele Punkte vor dem Punkt (x, y).

Also ist $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ bijektiv

Definiere folgende **Umkehrfunktionen** first, rest: $\mathbb{N} \to \mathbb{N}$ mit $\langle \text{first}(z), \text{rest}(z) \rangle = z$ und

$\text{first}(\langle x, y \rangle) = x$
$\text{rest}(\langle x, y \rangle) = y$

Paarungsfunktionen Codierung von Zahlenfolgen

6.27 Lemma Die Funktionen $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \longrightarrow \mathbb{N}$ und first, rest: $\mathbb{N} \longrightarrow \mathbb{N}$ sind primitiv rekursiv.

Beweis:

Nach Def. von $\langle x, y \rangle = z$, folgt $x \leq z$ und $y \leq z$.

Dann

$$\text{first}(z) = \mu x \leq z.\exists y \leq z.\langle x, y \rangle = z$$

$$\text{rest}(z) = \mu y \leq z.\exists x \leq z.\langle x, y \rangle = z$$

Nach Lemma 6.24 und Lemma 6.22 sind first und rest primitiv rekursiv.

Codierung endlicher Zahlenfolgen

6.28 Definition Sei $x_0, \ldots, x_n$ Zahlenfolge. Die **Codierung** $[x_0, \ldots, x_n] \in \mathbb{N}$ wird induktiv über n definiert durch

- $[\,] = 0$ (Codierung der leeren Folge)
- $[x_0, \ldots, x_n] = \langle x_0, [x_1, \ldots, x_n] \rangle$

Die **Folgenzugriffsfunktion** get : $\mathbb{N}^2 \longrightarrow \mathbb{N}$ sei definiert durch

- $\text{get}(z, 0) = \text{first}(z)$
- $\text{get}(z, i + 1) = \text{get}(\text{rest}(z), i)$

Die Folgenzugriffsfunktion liegt in $\mathcal{P}(\mathbb{N})$.

Wertverlaufsrekursion

Beachte: Die Folgenkodierung ist eindeutig bis auf Nullen am rechten Folgenende, d. h. es gilt $[x_0, \ldots, x_n] = [x_0, \ldots, x_n, 0, \ldots, 0]$. Die Elemente x_0 bis x_n mit $x_n \neq 0$ können eindeutig bestimmt werden.

6.29 Lemma

Sind $g : \mathbb{N}^{n+1} \longrightarrow \mathbb{N}$ und $h : \mathbb{N}^{n+2} \longrightarrow \mathbb{N}$ primitiv rekursiv, dann auch $f : \mathbb{N}^{n+1} \longrightarrow \mathbb{N}$ mit

- $f(\vec{x}, 0) = g(\vec{x}, 0)$
- $f(\vec{x}, y+1) = h(\vec{x}, [f(\vec{x}, y), \ldots, f(\vec{x}, 0)], y)$

Hier greift die Rekursion auf beliebig viele Vorgängerwerte zurück.

Beweis:

Die Hilfsfunktion $F(\vec{x}, y) = [f(\vec{x}, y), \ldots, f(\vec{x}, 0)]$ ist primitiv rekursiv, da

- $F(\vec{x}, 0) = \langle g(\vec{x}, 0), 0 \rangle$
- $F(\vec{x}, y+1) = \langle h(\vec{x}, F(\vec{x}, y), y), F(\vec{x}, y) \rangle$

und somit ist

$f(\vec{x}, y) = \text{get}(F(\vec{x}, y), 0) = \text{first}(F(\vec{x}, y))$

primitiv rekursiv.

Beispiel

6.30 Beispiel

1. Sei f definiert durch
   - $f(x, 0) = 1$
   - $f(x, y + 1) = f(x, y \text{ div } 2) + 1$

   Dann ist $f \in \mathcal{P}(\mathbb{N})$: Wähle im obigen Lemma $h(x, u, y) = succ(\text{get}(u, y \text{ div } 2))$.
2. **Folgenverkettung $*$: $\mathbb{N}^2 \longrightarrow \mathbb{N}$**
   (nicht Multiplikation von Zahlen!)
   $[x_0, \ldots, x_n] * [y_0, \ldots, y_m] = [x_0, \ldots, x_n, y_0, \ldots, y_m]$
   wobei $x_n \neq 0$ (falls $n > 0$).

   Behauptung: $*$ ist primitiv rekursiv.
   Beweis: Betrachte

$$0 * v = v$$
$$(u + 1) * v = \langle \text{first}(u + 1), \text{rest}(u + 1) * v \rangle$$

   wegen $\text{rest}(u + 1) \leq u$ folgt die Behauptung durch Wertverlaufsrekursion (hier wird die Voraussetzung $x_n \neq 0$ benötigt).
   Es gilt nämlich:

$$[i] = \langle i, 0 \rangle = \underbrace{\frac{i \cdot (i + 1)}{2}}_{\geq i} > 0 \quad (i \neq 0)$$

Beispiel (Fort.)

$$\underbrace{[x_0, \ldots, x_n]}_{u+1, x_n \neq 0} = \langle x_0, [x_1, \ldots, x_n] \rangle =$$

$$\underbrace{\frac{(x_0 + [x_1 \ldots x_n])(x_0 + [x_1 \ldots x_n] + 1)}{2}}_{\geq 1} + [x_1, \ldots, x_n]$$

D. h.

- $\mathsf{first}(u+1) = x_0 < u$ (Listenlänge ≥ 2)
- $\mathsf{rest}(u+1) = [x_1, \ldots, x_n] \leq u$

Wir werden diese Funktionen noch benötigen, und zwar bei der Arithmetisierung der While-Programme. Programme sind Folgen von Anweisungen. Wir werden Anweisungen durch Zahlen codieren und dementsprechend Programme durch die Codierungen der Zahlenfolgen darstellen. Um die Interpreterfunktion zu simulieren benötigen wir die Folgenzugriffsfunktion und die Folgenverkettung.

Simultane Rekursion

6.31 Definition

Eine Funktion $f = (f_1, \ldots, f_m) : \mathbb{N}^n \longrightarrow \mathbb{N}^m$ heißt primitiv rekursiv, falls jede Komponentenfunktion $f_i : \mathbb{N}^n \longrightarrow \mathbb{N}\ i = 1, \ldots, m$ primitiv rekursiv ist.

6.32 Lemma

Sind $g : \mathbb{N}^{n+1} \longrightarrow \mathbb{N}^m$ und $h : \mathbb{N}^{n+m+1} \longrightarrow \mathbb{N}^m$ primitiv rekursiv, dann ist auch die Funktion $f : \mathbb{N}^{n+1} \longrightarrow \mathbb{N}^m$ mit

- $f(\vec{x}, 0) \rightleftharpoons g(\vec{x}, 0)$
- $f(\vec{x}, y+1) \rightleftharpoons h(\vec{x}, f(\vec{x}, y), y)$

($\rightleftharpoons$ ist als Gleichheit von Vektoren zu lesen).

Beweis:

Die Hilfsfunktion

$$F(\vec{x}, y) = [f_1(\vec{x}, y), \ldots, f_m(\vec{x}, y)]$$

ist primitiv rekursiv.

Simultane Rekursion (Forts.)

Da $F(\vec{x}, 0) = [g_1(\vec{x}, 0), \ldots, g_m(\vec{x}, 0)]$

und

$F(\vec{x}, y+1) = [h_1(\vec{x}, F(\vec{x}, y), y), \ldots, h_m(\vec{x}, F(\vec{x}, y), y)]$

und für festes m die Funktion $[k_1, \ldots, k_m] : \mathbb{N}^n \longrightarrow \mathbb{N}$ primitiv rekursiv ist für $k_i : \mathbb{N}^n \longrightarrow \mathbb{N}$, $k_i \in \mathcal{P}(\mathbb{N})$.

Aus $f_i(\vec{x}, y) = \text{get}(F(\vec{x}, y), i)$ $i = 1, \ldots, m$ folgt die Behauptung.

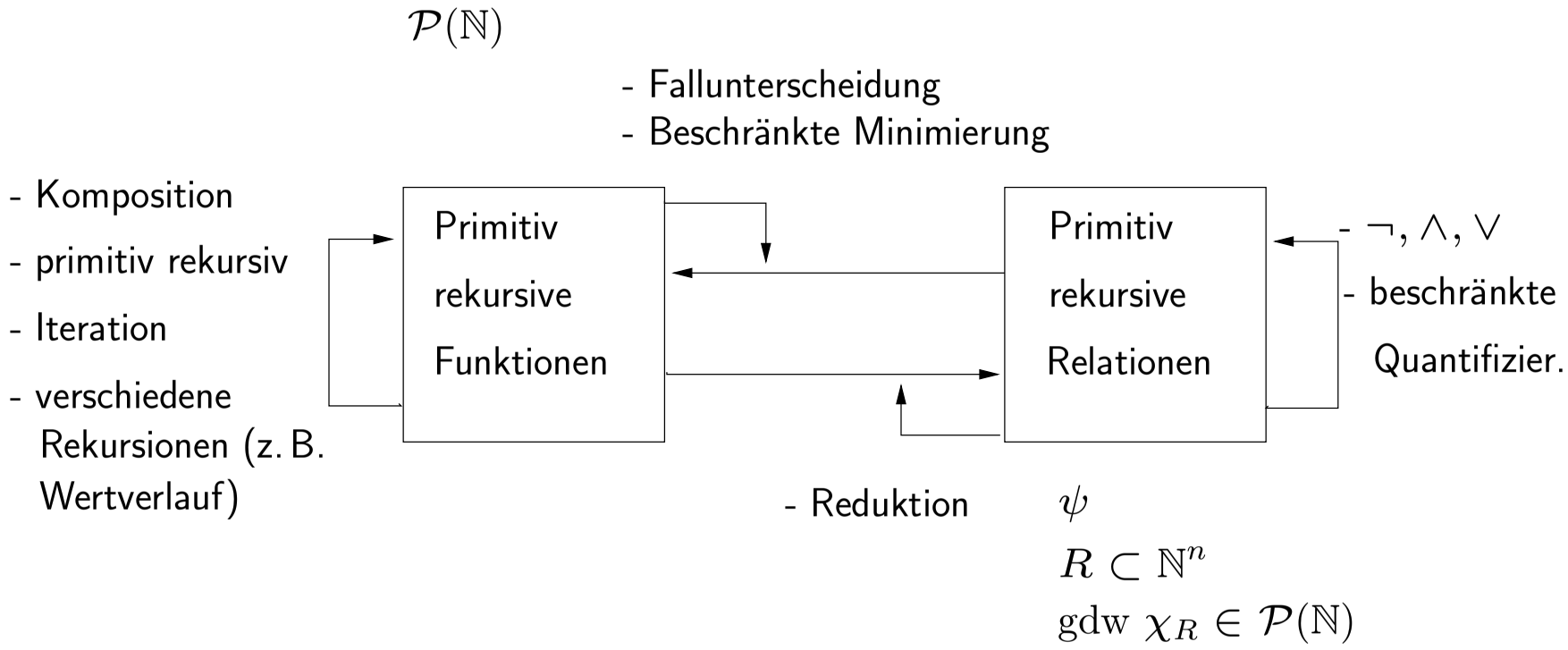
6.33 Beispiel

Sei $paar : \mathbb{N} \longrightarrow \mathbb{N}^2$ die Umkehrung der Cauchyschen Paarungsfunktion. Es gilt

$paar(0) = (0, 0)$ und

$$paar(n+1) = \begin{cases} (y+1, 0) & \text{falls } \; paar(n) = (0, y) \\ (x-1, y+1) & \text{sonst, wobei } paar(n) = (x, y) \end{cases}$$

Zeige: $paar(n) = (first(n), rest(n))$.



- $f \in \mathcal{P}(\mathbb{N})$, so ist $graph(f)$ primitiv rekursiv.

Sind alle totalen „berechenbaren“ Funktionen primitiv rekursiv?

- Numeriere effektiv alle primitiv rekursiven Ausdrücke, z. B. lexikographische Anordnung der Wörter über dem Alphabet $\{NULL, SUCC, PROJ, KOMP, REK, (,), 0, \ldots, 9, , \}$
- Sei π_i der i-te primitiv rekursive Ausdruck in dieser Nummerierung (Länge + Lexikographisch).
- Definiere Diagonalfunktion $d : \mathbb{N} \longrightarrow \mathbb{N}$ durch

$$d(n) = f^{(1)}_{\pi_n}(n) + 1$$

Dann ist d total und „effektiv berechenbar“.

6.34 Satz Diagonalschluss für primitiv rekursive Funktionen

Die Funktion d ist nicht primitiv rekursiv.

Beweis:

Wäre d primitiv rekursiv, dann gäbe es einen primitiv rekursiven Ausdruck π_n, so dass $d = f^{(1)}_{\pi_n}$. Dann aber

$f^{(1)}_{\pi_n}(n) = d(n) = f^{(1)}_{\pi_n}(n) + 1$ ↯

Universelle Funktionen für $\mathcal{P}(\mathbb{N})$

6.35 Folgerung Aufzählung der einstelligen Funktionen in $\mathcal{P}(\mathbb{N})$
Es gibt keine **universelle** primitiv rekursive Funktion $f : \mathbb{N}^2 \to \mathbb{N}$ für $\mathcal{P}(\mathbb{N})$, d. h. eine Funktion $f \in \mathcal{P}(\mathbb{N})$ mit der Eigenschaft $\forall g \in \mathcal{P}(\mathbb{N}) \exists i \in \mathbb{N} : f(i, \cdot) = g(\cdot)$ (i heißt **Index** für g).

Beweis: Sei $f : \mathbb{N}^2 \to \mathbb{N}$ universelle Funktion für $\mathcal{P}(\mathbb{N})$.

Sei $d'(x) = f(x, x) + 1$.

Angenommen $f \in \mathcal{P}(\mathbb{N})$. Dann $d' \in \mathcal{P}(\mathbb{N})$ und es gibt $i \in \mathbb{N}$ mit $d'(x) = f(i, x)$ für $x \in \mathbb{N}$.
Insbesondere $f(i, i) = d'(i) = f(i, i) + 1$ ↯

Gesucht: Konkrete Funktionen, die nicht primitiv rekursiv sind.

6.36 Beispiel Ackermann Funktion

- $A(0, y) = y + 1$
- $A(x + 1, 0) = A(x, 1)$
- $A(x + 1, y + 1) = A(x, A(x + 1, y))$

A ist eine totale Funktion.

A ist „eine Art“ universelle Funktion für $\mathcal{P}(\mathbb{N})$.

Eigenschaften der Ackermann Funktion

1. $A(x, y) > y$.
2. $A(x, y_1) > A(x, y_2)$, falls $y_1 > y_2$ (Monotonie).
3. $A(x + 1, y) \geq A(x, y + 1)$.
4. $A(x + 2, y) > A(x, 2y)$.
5. A ist total. (wie zeigt man dies!)
6. A ist effektiv berechenbar.

proc AKM(**in** $x, y : nat$, **out** $z : nat$)
  {true}**call** $AKM(x, y, z)\{z = A(x, y)\}$
  begin
    if $x = 0$
      then $z := y + 1$;
      else
        if $y = 0$
          then call $AKM(pred(x), 1, z)$;
          else
            call $AKM(x, pred(y), z_1)$;
            call $AKM(pred(x), z_1, z)$;
        end;
    end;
  end.

Die Prozedur „sollte“ korrekt kommentiert sein! (Nachweis!) über $(Nat, \ldots, A)$.

Eigenschaften der Ackermann Funktion (Forts.)

7. Für jede primitiv rekursive Funktion $f : \mathbb{N}^n \to \mathbb{N}$ gibt es ein $r \in \mathbb{N}$, so dass

$$f(\vec{x}) \leq A(r, \max\{x_1, \ldots, x_n\})$$

Beweis: Induktion über Aufbau von $\mathcal{P}(\mathbb{N})$.

- Grundfunktionen: wähle $r = 0$
  0
  $x_1 + 1 \quad \leq A(0, \max\{x_1, \ldots, x_n\}) = \max\{x_1, \ldots, x_n\} + 1$
  x_i
- Sei $f = g \circ (h_1, \ldots, h_m)$ und $y = \max\{x_1, \ldots, x_n\}$.
  wobei für g durch r, h_i durch s_i beschränkt sei, d. h.
  $g(\vec{x}) \leq A(r, y)$, $h_i(\vec{x}) \leq A(s_i, y)$.

$$\begin{aligned}
f(\vec{x}) &= g \circ (h_1, \ldots, h_m)(\vec{x}) = g(h_1(\vec{x}), \ldots, h_m(\vec{x})) = \\
&\leq A(r, \max\{h_1(\vec{x}), \ldots, h_m(\vec{x})\}) \\
&\overset{(2)}{\leq} A(r, \max\{A(s_1, y), \ldots, A(s_m, y)\}) \\
&\overset{(3)}{=} A(r, A(\max\{s_1, \ldots, s_m\}, y)) \\
&\overset{(2)(3)}{\leq} A(\max\{s_1, \ldots, s_m, r\}, A(\max\{s_1, \ldots, s_m, r\} \\
&\qquad\qquad +1, y)) \\
&\overset{def}{=} A(\max\{s_1, \ldots, s_m, r\} + 1, y + 1) \\
&\overset{(3)}{\leq} A(\max\{s_1, \ldots, s_m, r\} + 2, y)
\end{aligned}$$

Wähle $\max\{s_1, \ldots, s_m, r\} + 2$ als Konstante für f.

Eigenschaften der Ackermann Funktion (Forts.)

- Sei $f = R(g, h)$.
  g sei durch r, h durch s beschränkt.
  Hilfsbehauptung:
  $f(\vec{x}, z) \leq A(\max\{r, s\} + 1, y + z)$ mit
  $y = \max\{x_1, \ldots, x_n\}$.
  Beweis: Induktion über z. (einfach).
  Setze $p := \max\{r, s\} + 1$. Dann

$$A(p, y+z) \overset{(2)}{\leq} A(p, 2\max\{y, z\}) \overset{(4)}{\leq} A(p+2, \max\{y, z\})$$

  $p + 2$ kann als Konstante für f gewählt werden.

8. Die Ackermannfunktion ist nicht primitiv rekursiv, d. h.
   $A \notin \mathcal{P}(\mathbb{N})$.

Beweis:

Angenommen $A \in \mathcal{P}(\mathbb{N})$. Dann ist f mit $f(x, y) = A(x, y) + 1$ primitiv rekursiv. Es gibt (wegen 7.) ein $r \in \mathbb{N}$ mit
$f(x, y) = A(x, y) + 1 \leq A(r, \max\{x, y\})$.
Insbesondere für $x = y = r$

$$f(r, r) = A(r, r) + 1 \leq A(r, r) \qquad \lightning$$