Übung 10: PR und LOOP

Theoretische Informatik Sommersemester 2014

Markus Kaiser

29. Juni 2014

Definition (Basisfunktionen der PR)

Die Menge der primitiv rekursiven (PR) Funktionen ist induktiv definiert. Die Basisfunktionen sind die

konstante Funktion	$f(x) = 0$
Nachfolgerfunktion	$s(n) = n + 1$
Projektionsfunktion	$\pi_i^k : \mathbb{N}^k \to \mathbb{N}, i \in [k]$
	$\pi_i^k(x_1, \ldots, x_k) = x_i$

Definition (Komposition)

Seien g und h_i PR und $\bar{x} = (x_1, \ldots, x_n)$.
Dann ist auch die Komposition f PR.

$$f(\bar{x}) = g(h_1(\bar{x}), \ldots, h_k(\bar{x}))$$

Basisfunktionen und Komposition

Die folgenden Funktionen sind schon als primitiv rekursiv bekannt.

konstante Funktion	$f(x) = 0$
Nachfolgerfunktion	$s(n) = n + 1$
Projektionsfunktion	$\pi_i^k : \mathbb{N}^k \to \mathbb{N}, i \in [k]$
Komposition	$f(\bar{x}) = g(h_1(\bar{x}), \ldots, h_k(\bar{x}))$

Definition (Primitive Rekursion)

Das Schema der primitiven Rekursion erzeugt aus g und h die neue Funktion f.

$$f(0, \bar{x}) = g(\bar{x})$$
$$f(m+1, \bar{x}) = h(f(m, \bar{x}), m, \bar{x})$$

f ist ebenfalls primitiv rekursiv.

Definition (Erweitertes PR-Schema)

Das erweiterte Schema der primitiven Rekursion erlaubt

$$f(0, \bar{x}) = t_0$$
$$f(m+1, \bar{x}) = t$$

wobei t_0 und t Terme sind für die gilt

- t_0 enthält nur PR-Funktionen und die x_i
- t enthält nur $f(m, \bar{x})$, PR Funktionen, m und die x_i.

Satz

Das erweiterte Schema der primitiven Rekursion führt nicht aus PR heraus.

Unter anderem diese Programme sind laut Vorlesung oder Übung PR:

- $succ(x) = x + 1$
- $pred(x) = \max\{0, x - 1\}$
- $add(x, y) = x + y$
- $x \dot{-} y = \max\{0, x - y\}$
- $mult(x, y) = x \cdot y$
- $div(x, y) = x \div y$ (Ganzzahldivision)
- $pow(x, y) = x^y$
- Die restliche einfache Arithmetik...

- $tower(n) = 2^{2^{2^{\cdot^{\cdot^{\cdot}}}}}$ mit $tower(4) = 2^{16}$
- $sqr(x) = x^2$
- $sqrt(x) = \sqrt{x}$
- $c(x), p_1(x), p_2(x)$ (Cantorsche Paarungsfunktion)
- $ifthen(n, a, b) = \begin{cases} a & n \neq 0 \\ b & n = 0 \end{cases}$

Definition (LOOP-Programm)

Syntax von LOOP-Programmen.
Es ist $X \in \{x_0, x_1, \ldots\}$ und $C \in \mathbb{N}$.

$$
\begin{aligned}
P \rightarrow\ & X := X + C \\
|\ & X := X - C \\
|\ & P; P \\
|\ & \textbf{LOOP } X \textbf{ DO } P \textbf{ END} \\
|\ & \textbf{IF } X = 0 \textbf{ DO } P \textbf{ ELSE } Q \textbf{ END}
\end{aligned}
$$

- Ausgabe steht in x_0, Eingaben in $x_1, \ldots, x_n$, Rest ist 0.
- **LOOP** x_i **DO** P **END** führt P genau n mal aus, wobei n der Anfangswert von x_i ist. Zuweisungen an x_i in P ändern die Anzahl der Durchläufe nicht.

Definition (Intuitive Berechenbarkeit)

Eine Funktion $f : \mathbb{N}^k \to \mathbb{N}$ heißt intuitiv berechenbar, wenn es einen Algorithmus gibt, der bei Eingabe $(n_1, \ldots, n_k) \in \mathbb{N}^k$

- nach endlich vielen Schritten mit Ergebnis $f(n_1, \ldots, n_k)$ hält, falls $f(\ldots)$ definiert ist,
- und nicht terminiert, falls $f(\ldots)$ nicht definiert ist.

Churchsche These (nicht beweisbar)

Turing-Maschinen können genau alle intuitiv berechenbaren Funktionen berechnen.

Beispiel (Berechenbarkeit)

Sind die folgenden Funktionen intuitiv berechenbar?

$$f_1(n) = \begin{cases} 1 & \text{falls } n \text{ prim} \\ 0 & \text{sonst} \end{cases}$$

$$f_2(n) = \begin{cases} 1 & \text{falls } n \text{ die ersten } n \text{ Ziffern von } \pi \text{ darstellt} \\ 0 & \text{sonst} \end{cases}$$

$$f_3(n) = \begin{cases} 1 & \text{falls in } \pi \ n \text{ Nullen am Stück vorkommen} \\ 0 & \text{sonst} \end{cases}$$

Beispiel (Berechenbarkeit)

Sind die folgenden Funktionen intuitiv berechenbar?

$$f_1(n) = \begin{cases} 1 & \text{falls } n \text{ prim} \\ 0 & \text{sonst} \end{cases}$$

$$f_2(n) = \begin{cases} 1 & \text{falls } n \text{ die ersten } n \text{ Ziffern von } \pi \text{ darstellt} \\ 0 & \text{sonst} \end{cases}$$

$$f_3(n) = \begin{cases} 1 & \text{falls in } \pi\ n \text{ Nullen am Stück vorkommen} \\ 0 & \text{sonst} \end{cases}$$

Alle drei Funktionen sind intuitiv berechenbar.