

Übung 10: μ Rekursion, Entscheidbarkeit

Theoretische Informatik Sommersemester 2013

Markus Kaiser

July 2, 2013

Definition (LOOP-Programm)

Syntax von **LOOP-Programmen**.

Es ist $X \in \{x_0, x_1, \dots\}$ und $C \in \mathbb{N}$.

```
P → X := X + C
   | X := X - C
   | P; P
   | LOOP X DO P END
   | IF X = 0 DO P ELSE Q END
```

- Ausgabe steht in x_0 , Eingaben in x_1, \dots, 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 (Erweitertes PR-Schema)

Das **erweiterte Schema der primitiven Rekursion** erlaubt

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

wobei

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

Satz

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

Definition

Ein Prädikat P ist **PR**, wenn es eine PR Funktion \hat{P} gibt mit

$$\hat{P}(x) = 1 \iff P(x)$$

Definition (Beschränkte Operationen)

Ist P PR, dann auch

- der **beschränkte max-Operator**

$$\max \{x \leq n \mid P(x)\}, \quad \max \{\emptyset\} = 0$$

- der **beschränkte Existenzquantor**

$$\exists x \leq n. P(x)$$

Definition (μ -Operator)

Sei $f : \mathbb{N}^{k+1} \mapsto \mathbb{N}$ eine Funktion.

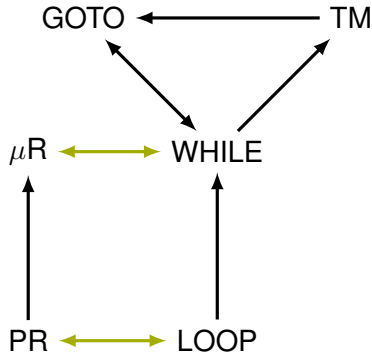
Der μ -Operator definiert eine neue Funktion $\mu f : \mathbb{N}^k \mapsto \mathbb{N}$:

$$(\mu f)(\bar{x}) := \begin{cases} \min \{n \in \mathbb{N} \mid f(n, \bar{x}) = 0\} & \text{falls } n \text{ existent}^* \\ \perp & \text{sonst} \end{cases}$$

- *Für alle $m \leq n$ muss f definiert sein: $f(m, \bar{x}) \neq \perp$
- PR + μ = μ -Rekursion
- In Pseudocode:

$$\mu f(\bar{x}) = \text{find}(0, \bar{x})$$

$$\text{find}(n, \bar{x}) = \text{if } f(n, \bar{x}) = 0 \text{ then } n \text{ else } \text{find}(n + 1, \bar{x})$$



LOOP kann in WHILE **übersetzt** werden, WHILE ist also **mindestens so mächtig** wie LOOP (sogar mächtiger).

Definition (Intuitive Berechenbarkeit)

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

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

Churchsche These (nicht beweisbar)

Turing-Maschinen können genau **alle** intuitiv berechenbaren Funktionen berechnen.

Definition (Entscheidbarkeit)

Eine Menge A heißt **entscheidbar** gdw ihre **charakteristische Funktion**

$$\chi_A(x) := \begin{cases} 1 & \text{falls } x \in A \\ 0 & \text{falls } x \notin A \end{cases}$$

berechenbar ist.

Definition (Semi-Entscheidbarkeit)

Eine Menge A heißt **semi-entscheidbar** gdw

$$\chi'_A(x) := \begin{cases} 1 & \text{falls } x \in A \\ \perp & \text{falls } x \notin A \end{cases}$$

berechenbar ist.

Definition (Reduzierbarkeit)

Eine Menge $A \subseteq \Sigma^*$ ist **reduzierbar** auf eine Menge $B \subseteq \Gamma^*$ gdw es eine totale und berechenbare Funktion $f : \Sigma^* \mapsto \Gamma^*$ gibt mit

$$\forall w \in \Sigma^*. w \in A \iff f(w) \in B$$

Wir schreiben dann $A \leq B$.

Intuition:

- B ist **mindestens so schwer** zu lösen wie A
- Ist A unlösbar, dann auch B .
- Ist B unlösbar, dann erst recht A .