Übung 11: Berechenbarkeitstheorie

Theoretische Informatik Sommersemester 2014

Markus Kaiser

4. Juli 2014

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.

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

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

Wir schreiben dann $A \leq B$.

Intuitiv gilt:

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

Definition (Spezielles Halteproblem)

Gegeben ein Wort $w \in \{0,1\}^*$ und die durch Gödelisierung kodierte Turingmaschine M_w.
Die als spezielles Halteproblem bezeichnete Sprache K enthält alle Turingmaschinen, die bei sich selbst als Eingabe halten.

$$K := \{w \mid M_w[w] \downarrow\}$$

Satz

Das spezielle Halteproblem ist semi-entscheidbar, aber es ist nicht entscheidbar.
$\overline{K}$ ist also nicht semi-entscheidbar.

Definition (Allgemeines Halteproblem)

Gegeben Wörter $w, x \in \{0, 1\}^*$ und die durch Gödelisierung kodierte Turingmaschine M_w.
Die als allgemeines Halteproblem bezeichnete Sprache H enthält alle Paare M_w und x, sodass M_w bei Eingabe x hält.

$$H := \{w\#x \mid M_w[x] \downarrow\}$$

Satz

Das allgemeine Halteproblem ist semi-entscheidbar, aber es ist nicht entscheidbar.
$\overline{H}$ ist also nicht semi-entscheidbar.

- Es ist $K \leq H$. Warum?

Definition (Rekursiv aufzählbar)

Eine Menge A heißt rekursiv aufzählbar wenn $A = \emptyset$ oder es eine berechenbare totale Funktion $f : \mathbb{N} \to A$ gibt, so dass

$$A = \{f(0), f(1), \dots\} = \bigcup_{n \in \mathbb{N}} \{f(n)\}$$

Satz

Sei A formale Sprache, dann ist äquivalent:

- *A ist Typ 0 Sprache*
- *A rekursiv aufzählbar*
- *A semi-entscheidbar, also χ'_A berechenbar*
- *$A = L(M)$ für eine TM M*
- *A ist Bild oder Urbild einer berechenbaren Funktion*

Satz (Rice)

Sei F eine Menge berechenbarer Funktionen.
Sei weder $F = \emptyset$ noch $F = \{f \mid f \text{ berechenbar}\}$ (F nicht trivial).
Der Satz von Rice besagt, dass es dann unentscheidbar ist, ob die von einer gegebenen TM M_w berechnete Funktion in F ist.
Die formale Sprache

$$\{w \in \Sigma^* \mid \varphi_w \in F\}$$

ist unentscheidbar.

Intuitiv gilt:

- Nicht-triviale semantische Eigenschaften von Programmen sind unentscheidbar.
- Termination ist unentscheidbar.