

Vorlesung Perlen der Informatik

Aufgabe 1 Intuitive Berechenbarkeit

Sind die folgenden Funktionen berechenbar? Für welche kann man sogar einen Algorithmus explizit angeben? Dabei sei $g : \mathbb{N} \rightarrow \mathbb{N}$ eine totale, berechenbare Funktion und $h : \mathbb{N} \rightarrow \mathbb{N}$ eine nicht berechenbare Funktion. Weiter ist über die Funktionen g und h nichts bekannt.

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

$$f_2(n) = \begin{cases} 1 & \text{falls es eine Primzahl } p \geq n \text{ gibt} \\ 0 & \text{sonst} \end{cases}$$

$$f_3(n) = \begin{cases} 1 & \text{falls es einen Primzahlzwilling } (p, p+2) \text{ mit } p \geq n \text{ gibt} \\ 0 & \text{sonst} \end{cases}$$

Hinweis: Ein Primzahlzwilling ist ein Paar von Primzahlen, deren Differenz 2 beträgt (z.B. (11, 13)). Es ist nicht bekannt, ob es unendlich viele Primzahlzwillinge gibt.

$$f_4(n) = \begin{cases} 1 & \text{falls } g(n) = n \\ 0 & \text{sonst} \end{cases}$$

$$f_5(n) = \begin{cases} 0 & \text{falls } n = 0 \\ h(n) & \text{sonst} \end{cases}$$

$$f_6(n) = \begin{cases} n & \text{falls es ein } m \text{ gibt mit } h(m) \leq m \\ 0 & \text{sonst} \end{cases}$$

Aufgabe 2 Konsequenzen des Satzes von Rice

Der Satz von Rice ist ein sehr starkes Unentscheidbarkeitsresultat über Eigenschaften von Programmen. Dennoch gibt es viele Eigenschaften von Programmen, die mit Hilfe von geeigneten Tools geprüft werden können, z.B.

- ob alle Variablen vor Verwendung initialisiert werden
- ob das Programm typkorrekt ist, also keine Typfehler zur Laufzeit auftreten können
- ob das Programm mögliche Sicherheitslücken wie z.B. Buffer-Overflows enthält

Warum ist die Existenz solcher Verfahren kein Widerspruch zum Satz von Rice? Was hat der Satz von Rice andererseits für Konsequenzen für die Verfahren?

Aufgabe 3 Satz von Rice

Benutzen Sie den Satz von Rice, um die Unentscheidbarkeit folgender Probleme zu zeigen. Geben Sie zunächst jeweils die Menge an, durch die das Problem beschrieben wird.

- Es soll entschieden werden, ob ein Programm eine totale Funktion berechnet. (Totalitätsproblem)
- Es soll entschieden werden, ob zwei Programme dieselbe Funktion berechnen. (Äquivalenzproblem)

Aufgabe 4 Unendliche Mengen

Zwei Definitionen der *Unendlichkeit* einer Menge A lauten wie folgt:

1. Es existiert eine Funktion $f : A \rightarrow A$, die *injektiv*, aber nicht *surjektiv* ist.
2. Es existiert eine *injektive* Funktion $g : \mathbb{N} \rightarrow A$.

Zeigen Sie die Äquivalenz dieser beiden Definitionen.