Aufgaben zur Vorlesung **WiSe 22/23**
Berechenbarkeit und Komplexität **TU Berlin**
Weller/Froese/Kellerhals/Zschoche **25.10.2022**

2. Aufgabenblatt

(Besprechung in den Tutorien 31.10–04.11.2022)

Aufgabe 1. Berechenbar oder nicht?

Die unbewiesene Goldbachsche Vermutung lautet: „Jede gerade Zahl größer 2 lässt sich als Summe zweier Primzahlen darstellen."

1. Sei $f : \mathbb{N} \to \mathbb{N}$ eine Funktion, die bei allen Eingaben genau dann 1 ausgibt, wenn die Goldbachsche Vermutung gilt, und sonst 0. Existiert ein Algorithmus, der bei Eingabe $n \in \mathbb{N}$ nach endlicher Zeit $f(n)$ ausgibt?

   ——————Lösungsskizze——————

   Ja, da man dazu nur konstant 1 oder konstant 0 ausgeben muss. (Aber: Wir wissen derzeit nicht, welche der beiden Möglichkeiten die richtige ist. . .)

   ————————————————————

2. Existiert ein Algorithmus, der bei Eingabe einer binär kodierten natürlichen Zahl n genau dann 1 ausgibt, wenn n eine gerade Zahl größer 2 ist und sich als Summe zweier Primzahlen darstellen lässt, und sonst 0?

   ——————Lösungsskizze——————

   Ja: Alle Primzahlenpaare, die kleiner sind als n, ausprobieren.

   ————————————————————

3. Beschreiben Sie den sich ergebenden Unterschied, wenn folgende, gegenüber 2. modifizierte Aufgabe betrachtet wird: Bei Eingabe n ist genau dann 1 auszugeben, wenn n eine gerade Zahl größer 2 ist und sich als Differenz zweier Primzahlen darstellen lässt, und 0 sonst.

   ——————Lösungsskizze——————

   Der Unterschied ist, dass man nun potenziell die Differenz *aller*, d.h. unendlich vieler Primzahlpaare ausprobieren muss. Auf diese Art und Weise wird also womöglich nie eine 0 ausgegeben. Mit diesem Ansatz lässt sich die Funktion also nicht berechnen. (Genauer: Der Definitionsbereich der gegebenen Funktion sind die natürlichen Zahlen, und bei einer berechenbaren Funktion muss auf allen Eingaben des Definitionsbereichs die Berechnung nach endlicher Zeit stoppen.)

   Nach unserem bisherigen Kenntnisstand, ist unbekannt, ob diese Funktion berechenbar ist oder nicht.

   ————————————————————

Aufgabe 2. Berechenbar oder nicht?

Geben Sie an, ob folgende Funktionen berechenbar sind. Begründen Sie Ihre Antworten.

1. $f(n) = \begin{cases} 1, & \text{falls } n \text{ Studierende unter den dann geltenden Abstandsregeln} \\ & \text{die Klausur im Audimax schreiben können.} \\ 0, & \text{sonst.} \end{cases}$

   ——————Lösungsskizze——————

Ja: Es gibt eine maximale Studierendenanzahl x, die am Tag der Klausur in den Audimax passen. Also wird f von einem Algorithmus berechnet, der 1 ausgibt genau dann, wenn $n \leq x$ ist und sonst 0 ausgibt.

---

2. $$g(n) = \begin{cases} 1, & \text{falls } n \text{ Tage nach dem 24.12.2022 die Sonne nicht scheint} \\ & \text{oder schönes Wetter ist.} \\ 0, & \text{sonst.} \end{cases}$$

(Anmerkung: Sonnenschein impliziert schönes Wetter.)

——————Lösungsskizze——————

Eine Fallunterscheidung wie bei 1. funktioniert nicht, aber wir können folgendes beobachten: Für alle $n \in \mathbb{N}$ gilt $f(n) = 1$. Denn falls kein schönes Wetter ist, kann die Sonne auch nicht scheinen. Damit ist f berechenbar.

Aufgabe 3. Berechenbar oder nicht?

Im Folgenden sei $B \in \mathbb{N}$ „Die kleinste Zahl, die sich nicht mit weniger als zwanzig deutschsprachigen Wörtern definieren lässt." Desweiteren sei die Funktion $f : \mathbb{N} \to \mathbb{N}$ definiert als

$$f(n) := \begin{cases} 1, & n \leq B, \\ 0, & \text{sonst.} \end{cases}$$

Ist die Funktion f berechenbar oder nicht?

——————Lösungsskizze——————

Da die Definition von B widersprüchlich ist (denn sie beinhaltet weniger als 20 Wörter), existiert die Zahl B gar nicht. Daher ist die Funktion f also gar nicht wohldefiniert. Somit ist f die nirgends definierte Funktion und daher berechenbar.

Eine detaillierte Erläuterung der Problematik kann unter `http://de.wikipedia.org/wiki/Berry-Paradoxon` eingesehen werden.

Aufgabe 4. Berechenbarkeit

Sei $f : \mathbb{N} \to \mathbb{N}$ die Funktion aus der Vorlesung (Beispiel 3, Folie 21) mit

$$f(n) := \begin{cases} 1, & \text{falls die Dezimaldarstellung von } n \text{ in der Dezimalbruchentwicklung} \\ & \text{von } \pi \text{ vorkommt} \\ 0, & \text{sonst} \end{cases}.$$

Desweiteren definieren wir für $x \in \mathbb{N}$ die Funktion $f_x : \mathbb{N} \to \mathbb{N}$ wie folgt

$$f_x(n) := \begin{cases} 1, & \text{falls die Dezimalbruchentwicklung von } \pi \ n \text{ aufeinanderfolgende} \\ & \text{Konkatenationen der Dezimaldarstellung von } x \text{ enthält} \\ 0, & \text{sonst} \end{cases}.$$

Zum Beispiel gilt $f_{141}(1) = 1$. Die Funktion f_1 entspricht also der Funktion aus Beispiel 4 aus der Vorlesung (Folie 21).

Worin besteht das Problem in folgendem vermeintlichen „Beweis" der Berechenbarkeit von f?

„Für jedes $x \in \mathbb{N}$ ist f_x berechenbar (analog zum Beweis der Berechenbarkeit von f_1 aus der Vorlesung). Um nun die Funktion $f(n)$ zu berechnen, kann ein Algorithmus also einfach den Wert von $f_n(1)$ berechnen und ausgeben. Also ist auch f berechenbar."

Lösungsskizze

Das Problem liegt darin, dass dieser Ansatz keine endliche Beschreibung eines Algorithmus liefert. Es stimmt zwar, dass f_x für jedes $x \in \mathbb{N}$ berechenbar ist, jedoch besitzt jede Funktion f_x womöglich einen anderen Algorithmus, der sie berechnet. Um nun f zu berechnen, müsste ein Algorithmus nach obigem Ansatz also potenziell unendlich viele Algorithmen als „Subprozeduren“ aufrufen können. Dies lässt sich aber nicht durch eine endliche Beschreibung (z.B. als Programmcode) erreichen. (Zur Erinnerung: Ein Algorithmus besteht immer aus endlich vielen Anweisungen.)