4.3 Das Halteproblem

In der Berechenbarkeitstheorie stellt sich letztendlich die Frage, ob es auch Probleme gibt, die nicht nur „umständlich“ (also z.B. mit $O(x^n)$) zu berechnen sind, sondern sogar unmöglich zu berechnen sind. Wir schauen uns zwei Beispiele an, die diese Grenzen der Berechenbarkeit aufzeigen.

Archilles vs. die Schildkröte

Eines der bekanntesten Paradoxa der Mathematik ist das Wettrennen zwischen Archilles und einer Schildkröte. Dabei geht man davon aus, dass Archilles gegen eine Schildkröte im Wettlauf antritt.
Um den Wettlauf etwas fairer zu gestalten, bekommt Archilles ein Handicap: Die Schildkröte bekommt einen Vorsprung von 10m auf der 100m Strecke. Nun ist leicht erkennbar, dass Archilles, egal wie schnell er ist, eine gewisse Zeit braucht, um die Schildkröte einzuholen und diese Zeit nutzt die Schildkröte natürlich, um den Vorsprung weiter auszubauen.
In dem Moment also, wo Archilles die Startposition der Schildkröte erreicht hat, hat diese wieder einen Vorsprung erzielt. Dieser Vorsprung ist zwar kleiner, doch das Spiel wiederholt sich: Bis der Vorsprung aufgeholt ist, ist die Schildkröte wieder dem Ziel näher gekommen.
Das Paradoxon ergibt sich daraus, dass auf diese Weise Archilles das Rennen nie gewinnen kann, egal wie schnell er rennt.
Ähnlich ist es, wenn ein Rechner versucht, die Zeit in der Archilles die Schildkröte einholt z.B. rekursiv zu berechnen.

```
double Zeit(double Abstand, double VArchilles, double VSchild)
{
      double tAufholen = Abstand/VArchilles;
      return tAufholen + Zeit(VSchild*tAufholen, VArchilles, VSchild);
}
```

Die einzige Möglichkeit hier eine Endlose Rekursion zu verhindern ist, sich darauf zu verlassen, dass tAufholen irgendwann auf „0“ gerundet wird und eine bedingte Anweisung außen herum zu bauen, dass in diesem Fall „0“ zurückgegeben wird. Der genaue Wert der Zeit ist jedoch nicht berechenbar.

Das Halteproblem

Das Beispiel mit Archilles wirkt oft etwas konstruiert, da es eine einfache Möglichkeit gibt, die Zeit zu berechnen: Man berechne den Schnitt zweier Bewegungsdiagramme.
Ein Beispiel aus der Informatik jedoch zeigt, dass es unmöglich ist, alles mit dem Computer auszurechnen.
Das Halteproblem ist die Frage danach, ob man bei jedem beliebigen Algorithmus entscheiden kann, ob er in einer endlichen Zeit ein Ergebnis liefert (terminiert) oder nicht (z.B. Endlosschleife). Man kann beweisen, dass das Halteproblem nicht entscheidbar ist[1].

Der Beweis geht ungefähr wie auf der folgenden Seite:

[1] D.h. die Frage, ob man zu allen denkbaren Algorithmen entscheiden kann, ob diese terminieren oder nicht kann nicht beantwortet werden!

1. Um zu zeigen, dass nicht alle Algorithmen bei allen denkbaren Eingabewerten auf ihr Terminierungsverhalten untersucht werden können, reicht es, einen Algorithmus zu finden, bei dem es nicht funktioniert.

2. Zuerst brauchen wir einen Halteproblem-Algorithmus, der auf irgendeine Weise überprüft, ob ein Algorithmus für eine Eingabe terminiert oder nicht. Dieser Algorithmus soll durch die Methode
`boolean EndlichkeitTesten(String Quellcode, String Eingabe)`
dargestellt werden. Dabei gibt `EndlichkeitTesten(…)` folgende Rückgaben:
    a. TRUE, wenn der Algorithmus in „`Quelltext`" für die „`Eingabe`" terminiert
    b. FALSE, sonst

3. Man stelle sich vor, dass es eine Tabelle gibt, in der alle Algorithmen der Welt in Zeilen untereinander angeordnet sind.
    a. In der ersten Spalte der Tabelle ist jeweils der Quellcode jedes Algorithmus zu finden.
    b. In der zweiten Spalte der Tabelle jeweils der Rückgabewert der Methode `EndlichkeitTesten(…)` mit der ersten Spalte als Eingabewert für den Quellcode, getestet auf alle denkbaren Eingaben.

Quellcode	EndlichkeitTesten(Quellcode)
A	TRUE
B	
C	
…	
G	?
H	?
…	…

Abbildung 1: Tabelle aller Algorithmen

4. In dieser Tabelle muss dann auch folgender Algorithmus „G" stecken:

```
void Gegenbeispiel(String methode)
{
      Solange EndlichkeitTesten(methode, methode)
            //tue nichts
      EndeSolange
}
```

5. Wenn man nun das Verhalten von `Gegenbeispiel( G )` betrachtet kommt man zu folgendem Problem:
    a. Wenn das `Gegenbeispiel` terminiert gibt `EndlichkeitTesten(G, G)` TRUE aus. Damit ist allerdings die Wiederholung im `Gegenbeispiel` eine Endlosschleife und somit müsste die Endlichkeit eigentlich FALSE sein.
    b. Wenn das `Gegenbeispiel` nicht terminiert, gibt `EndlichkeitTesten(G, G)` FALSE aus. Allerdings wird dann die Wiederholung im `Gegenbeispiel` nie ausgeführt und somit müsste das `Gegenbeispiel` sofort terminieren und damit die Endlichkeit eigentlich TRUE sein.

6. Dieser Widerspruch lässt sich nicht auflösen und daher ist die Frage, ob dieses `Gegenbeispiel` für alle Eingabewerte terminiert nicht entscheidbar und somit ist automatisch das Halteproblem für alle denkbaren Algorithmen nicht entscheidbar.[2]

[2] Ein anschaulicheres Beispiel: Stellen Sie sich vor, es gibt einen Barber (ein Friseur für Bärte), der sich vorgenommen hat, nur Leute zu rasieren, die sich nicht selber rasieren. Was glauben Sie: Rasiert sich dieser Mann selber?

Aufgaben:

a) Setzen Sie den Beispielalgorithmus zu Archilles und der Schildkröte mit verschiedenen Datentypen um. Wie genau kann der exakte Wert berechnet werden? Überprüfen Sie anhand einer händischen Rechnung.
b) …