Grundlagen der Programmierung

Grenzen von Algorithmen:

Berechenbarkeit ♦ Unentscheidbarkeit

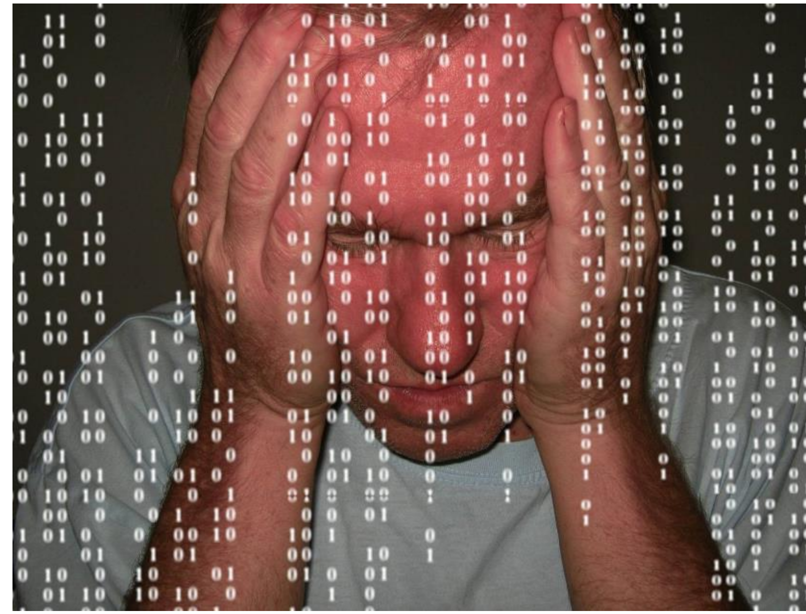
Fazit

- Die meisten Funktionen sind nicht algorithmisch berechenbar.

- *Frage:*
  Gibt es eine (in der Informatik interessante) Funktion, die nicht algorithmisch berechenbar ist?

Motivation

Wie lange soll
*ich **noch** warten?*



Kommt da noch ein Ergebnis?

Die Fragestellung

Hält ein gegebenes Programm an,
oder wird es nie terminieren?

Das Dilemma

- kann nicht getestet werden ...
- z.B. bei Schleifen: unbegrenzt viele Programmläufe
- kann von (Benutzer-) Eingaben abhängen ...

Beispiel

```
def mod(a, b):
    r = a
    while (r >= b):
        r = r - b
    return r
```

... berechnet den Rest bei **`a/b`** für positive **`a`**, **`b`**

... **terminiert** nur, falls **`b > 0`** oder **`a < b`**

Der große Traum

Algorithmus, der

- Programm und Eingabe analysiert und
- **entscheidet**, ob das Programm für die gegebene Eingabe stoppt oder nicht.

➢ **Entscheider** für das **Halteproblem**

Das Halteproblem ist ein sogenanntes Entscheidbarkeitsproblem.
Entscheidbarkeitsprobleme sind Fragestellungen, die für alle Eingaben mit ja (1) oder nein (0) beantwortet werden sollen.

Das (vereinfachte) Halteproblem

Eingabe:
Programm/Algorithmus ***A*** und natürliche Zahl ***n***

Ausgabe: **1** falls ***A*** bei Eingabewert ***n*** terminiert
0 sonst

... also Vereinfachung durch Beschränkung auf Programme mit nur einer natürlichen Zahl als Eingabe ...

Die optimistische Annahme

Sei *E* ein Algorithmus, der das Halteproblem *entscheidet*,

also mit

$$f_E(A, n) = \begin{cases} 1 & \text{falls } A \text{ terminiert bei Eingabe } n \\ 0 & \text{sonst} \end{cases}$$

f_E ist die von Algorithmus *E* berechnete Funktion.

Beispiel-Antworten von *E* (Ausschnitt)

n	1	2	3	4	5	6	7
A_1	1	1	1	1	1	1	1
A_2	1	0	0	0	0	0	0
A_3	0	1	0	1	0	1	0
A_4	1	1	0	1	1	1	0
A_5	0	0	0	0	0	0	0
A_6	1	0	1	0	0	1	0
A_7	1	1	1	0	0	0	1

Modifikation von *E*

Konstruieren Algorithmus ***E´*** mit

$$f_{E'}(n) = \begin{cases} 1 & \text{falls } f_E(A_n, n) = 0 \\ \textit{nicht def.} & \text{falls } f_E(A_n, n) = 1 \end{cases}$$

Die Antworten von *E*

n	1	2	3	4	5	6	7
A_1	1	1	1	1	1	1	1
A_2	1	0	0	0	0	0	0
A_3	0	1	0	1	0	1	0
A_4	1	1	0	1	1	1	0
A_5	0	0	0	0	0	0	0
A_6	1	0	1	0	0	1	0
A_7	1	1	1	0	0	0	1

Die Antworten von *E´*

n	1	2	3	4	5	6	7
	-						
		1					
			1				
				-			
					1		
						-	
							-

E' in der Tabelle mit Antworten von E bei $E' = A_k$

n	1	2	3	4	…	k
A_1	1	1	1	1		1
A_2	1	0	0	0		0
A_3	0	1	0	1		1
A_4	1	1	0	1		1
⋮						
A_k	0	1	1	0		?

$$f_E(A_k,k) = f_E(E',k)$$
$$= ?$$

Der Widerspruch

Wegen $f_{E'}(k) = \begin{cases} 1 & \text{falls } f_E(A_k, k) = 0 \\ \textit{nicht def.} & \text{falls } f_E(A_k, k) = 1 \end{cases}$

gilt:

$E' = A_k$ hält bei Eingabe k gdw. $f_E(A_k, k) = 0$. ↯

→ *Entscheider **E** existiert nicht !!!*

→ **Das Halteproblem ist unentscheidbar.**

Weitere unentscheidbare Probleme

- **Eingabe:** Programm und Funktion
  Ausgabe: Berechnet das Programm die Funktion?

- **Eingabe:** zwei Programme
  Ausgabe: Berechnen die Programme dieselbe Funktion?

- *viele Weitere*

- **Interpretation:** *Es gibt keinen (feststehenden) Algorithmus, der o.g. Fragen **für alle** denkbaren Eingaben beantwortet!*

Entscheidbare Probleme

- **Eingabe:** natürliche Zahl *n*
  Ausgabe: 1, falls *n* Primzahl, sonst 0

- **Frage:** Gibt es intelligentes Leben außerhalb unseres Sonnensystems?

  Eingabe: -
  Ausgabe: *Entweder* JA *oder* NEIN

  Der Algorithmus, der JA ausgibt, existiert.
  Ebenso der Algorithmus, der NEIN ausgibt.