



# 5. Algorithmen

# Zu lösende Probleme

1. Berechne  $2^n$  .
2. Berechne die Fakultät einer nat. Zahl:  
$$n! = 1 * 2 * \dots n$$
3. Entscheide, ob die nat. Zahl  $n$  eine Primzahl ist!
4. Ist ein gegebenes Computerprogramm korrekt?
5. Gegeben ist ein beliebiges Computerprogramm. Entscheide, ob es immer nach endlich vielen Schritten abbricht (kein unendlicher Zyklus).

z. B.

```
class T { ...  
    while (x >= 0)  
        ... ;  
}
```

# Programme lösen Probleme

... durch Algorithmen, die über Daten operieren

Programme = Daten + Algorithmen

Was ist ein Algorithmus?

Berechnungsvorschrift:  
Eingaben → Resultat

Eingaben → Berechnungsvorschrift → Resultat

# Was ist ein Algorithmus?

## Algorithmus:

Verfahren zur Berechnung gesuchter Werte aus gegebenen Werten, das in einer präzisen endlichen Beschreibung angegeben ist, die auf der schrittweisen Ausführung von elementaren (und berechenbaren) Verarbeitungsoperationen beruht.

# Was wird benötigt?

Exakte Vorschrift zum Berechnen:

- Gegeben:  
Elementare Operationen
- Festgelegt:  
Abfolge der Operationen
- Wodurch: präzise endliche Beschreibung
  - jedoch nicht gesichert: Berechnung endlich  
(d. h. bricht überhaupt nach endlich vielen Schritten ab)

## Algorithmus:

Verfahren zur Berechnung gesuchter Werte aus gegebenen Werten, das in einer präzisen endlichen Beschreibung angegeben ist, die auf der schrittweisen Ausführung von elementaren (und berechenbaren) Verarbeitungsoperationen beruht.

```
x = 3;  
while (x > 0)  
{ y = y * 2 }
```

➔ Definition muss präzisiert werden:  
Welche genaueren Informationen müssten geliefert werden, um konkrete Algorithmen zu formulieren?

# Präzisierung des Begriffs 'Algorithmus'

- Was sind elementare, berechenbare Operationen?
- Wie können sie kombiniert werden?
- Wie beschreibt man?

# Beispiel für einen Algorithmus


Schritt 0: Eingabe von  $n$

Schritt 1: falls  $n < 0$  :  
Ausgabe "Fehler: negative Zahl" , STOP

Schritt 2:  $\text{resultat} := 1$

Schritt 3: falls  $n = 0$  :  
Ausgabe "Resultat:" resultat, STOP

Schritt 4:  $\text{resultat} := \text{resultat} * n$   
 $n := n - 1$   
weiter bei Schritt 3



Was wird  
berechnet?

Probleme im Beispiel:

- Was sind die Elementaroperationen?
- Wie werden sie kombiniert?

# Antworten zum Beispiel:

- Berechnet wird:  $n! = 1 * 2 * \dots * n$   
(aber: Reihenfolge umgekehrt)
  
- Elementaroperationen:
  - Eingabe, Ausgabe
  - Tests:  $< =$
  - Arithmetik:  $*$ ,  $-$
  - Wertzuweisung an Variable:  $n := n+1$
  - STOP
  
- Kombination der Elementaroperationen:
  - Sequentielle Ausführung der Schritte  
(Schritt 0 → Schritt 1 ... bzw. innerhalb eines Schritts)
  - Bedingte Ausführung (falls ...)
  - Zyklus: Wiederholung von Schritten (Schritt 4 → Schritt 3)

Hinweis: alle Algorithmen bzw. Programme kommen mit diesen drei Kombinationsmöglichkeiten aus



# Kunst des Programmierens: Algorithmen entwickeln . . .

Angebot einer Programmiersprache

elementare  
Verarbeitungsoperationen:

+

-

<

/

\*

<=

Eingabe

u. a.

Ausgabe

Kombinationsmöglichkeiten

bedingte Ausführung:

if B ...

Wiederholung:

while B ...

Sequenz:

S1; S2; ...

Algorithmus:

```
Schritt 1:  
Schritt 2:  
...
```

Programm:

```
class T {  
    if ...  
    while ...  
}
```

**Kunst des Programmierens:**

- Daten darstellen
- Algorithmen entwickeln . . .

# Algorithmen zur ...

## Berechnung:

(Funktions-)Werte ausrechnen:  $2^n$ ,  $n!$

## Entscheidung:

- Zugehörigkeit zu einer Menge:  
Ist eine Zahl  $n$  eine Primzahl?  
 $n \in \{m \mid m \text{ Primzahl}\}$
- Korrektheit eines Programms: Leistet das Programm das Verlangte?  
→ allgemein: nicht entscheidbares Problem; Algorithmus existiert nicht
- Terminiert ein Programm:  
Bricht es bei einer beliebigen Eingabe nach endlich vielen Schritten ab  
oder entsteht ein unendlicher Zyklus?  
→ allgemein: nicht entscheidbares Problem; Algorithmus existiert nicht  
("Halteproblem")

**Es gibt Probleme, für die es keine Algorithmen gibt**

Unterschied:

Entscheidung - Berechnung?

# Darstellungsformen für Algorithmen:

- Folge von textuellen Anweisungen
- Flussdiagramm
- Pseudocode
- Computerprogramm
- ...

für abstrakte Untersuchungen (Berechnungstheorie):

- rekursive Funktionen
- TURING-Maschinen
- Zähler-Automaten
- ...

→ Grundlagen der  
Theoretischen Informatik

# „Alltags-“ Beispiel für Algorithmus: Partnerwahl

Kennengelernt haben wir uns über einen Algorithmus.  
Die Frau, die am häufigsten anruft, die ist es.  
Silke hat zweimal angerufen.



Quelle: OI, Berliner Zeitung