

# Zusammenfassung Turingmaschinen

- Turingmaschinen erlauben “komplexe” Berechnungen
- Es lassen sich sogar Turingmaschinen konstruieren, die andere Turingmaschinen simulieren  $\Rightarrow$  Universelle Turingmaschinen
- Können Turingmaschinen “alles” berechnen?

*DIE KLASSE DER TURING-BERECHENBAREN FUNKTIONEN STIMMT MIT DER KLASSE DER INTUITIV BERECHENBAREN FUNKTIONEN ÜBEREIN.*

- Bedeutet also: Funktion berechenbar  $\Leftrightarrow$  Es gibt eine Turingmaschine, die die Funktion berechnet
- Auch andere theoretische Berechnungsmodelle (Registermaschinen, while-Programme) haben sich als äquivalent zu Turingmaschinen erwiesen
- Gibt es Funktionen, die **nicht** (turing-)berechenbar sind?

Fleißige Biber

---

- Eine Biber-Turingmaschine ist eine Turingmaschine mit
  - $n$  “nutzbaren” Zuständen
  - einem Haltezustand
  - Bandalphabet  $\{0, 1\}$
- Sie arbeitet nach folgenden Regeln:
  - Es erfolgt immer eine Bewegung des Schreib-/Leseskopfes nach links oder rechts
  - Das Band ist anfangs komplett mit 0 gefüllt, es gibt keine leeren Felder
  - Die Maschine muss nach endlich vielen Schritten halten
- Wie viele Einsen (“Baustämme”) auf dem Band lassen sich von einer solchen Maschine erzeugen?

# Aufgabe 1

- Bestimme, wie viele Biber-Turingmaschinen mit einem Zustand (plus Haltezustand) es gibt
  - Welche davon terminieren sicher nicht?
- Entwirf mindestens zwei verschiedene Biber-Turingmaschinen mit zwei Zuständen (plus Haltezustand) mit JFlap und führe sie aus
  - Versuche, möglichst viele Einsen zu erzeugen

## Anzahl möglicher Maschinen (1)

- Wie viele Biber-Turingmaschinen mit  $n$  Zuständen (plus Haltezustand) gibt es?

- Mögliche Argumente für die Zustandsübergangsfunktion:

$$\cdot \delta(q_0, 0), \delta(q_0, 1), \delta(q_1, 0), \delta(q_1, 1), \dots, \delta(q_{n-1}, 0), \delta(q_{n-1}, 1)$$

$\Rightarrow 2n$  mögliche Zustandsübergänge

- Wahlmöglichkeiten eines jeden Zustandsübergangs:

- Einer von  $n + 1$  Folgezuständen
- Eine Bandbewegung (links oder rechts)
- Ein zu schreibendes Bandsymbol (0 oder 1)

$\Rightarrow (n + 1) \cdot 2 \cdot 2$  Möglichkeiten für **einen** Zustandsübergang

- **Jeder** der  $2n$  Übergänge kann unabhängig von den anderen aus  $(n + 1) \cdot 2 \cdot 2$  Kombinationen wählen

## Anzahl möglicher Maschinen (2)

Insgesamt

$$((n + 1) \cdot 2 \cdot 2)^{2n}$$

Möglichkeiten (und damit Biber-Turingmaschinen mit  $n$  Zuständen (plus Haltezustand))

- $\Sigma(n)$  liefert die maximale Zahl an Einsen, die eine Biber-Turingmaschine mit  $n$  Zuständen (plus Haltezustand) schreiben kann
- $\Sigma(n)$  wächst extrem schnell
- $\Sigma(n)$  ist nicht (turing)berechenbar



# Das Halteproblem

---

# Halteproblem - Problemstellung

- Gibt es eine Turingmaschine, die prüfen kann, ob eine andere Turingmaschine mit einer bestimmten Eingabe hält?
- Konkret: Wir suchen eine Turingmaschine  $T_H$ , die
  - als Eingabe eine Turingmaschine  $T_M$  sowie eine Eingabe  $I$  erhält
  - prüft, ob  $T_M$  mit der Eingabe  $I$  terminiert oder nicht
- Man kann zeigen, dass eine solche Turingmaschine  $T_H$  nicht existieren kann
- Das Halteproblem ist daher nicht entscheidbar