

# Einführung in die Theoretische Informatik

Woche 11

Harald Zankl

Institut für Informatik @ UIBK  
Wintersemester 2014/2015



Zusammenfassung

## Zusammenfassung der letzten LV

### Definition

eine **deterministische, einbändige Turingmaschine (TM)**  $M$  ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1  $Q$  eine endliche Menge von **Zuständen**,
- 2  $\Sigma$  eine endliche Menge von **Eingabesymbolen**,
- 3  $\Gamma$  eine endliche Menge von **Bandsymbolen**, mit  $\Sigma \subseteq \Gamma$ ,
- 4  $\vdash \in \Gamma \setminus \Sigma$ , der **linke Endmarker**,
- 5  $\sqcup \in \Gamma \setminus \Sigma$ , das **Blanksymbol**,
- 6  $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$  die **Übergangsfunktion**,
- 7  $s \in Q$ , der **Startzustand**,
- 8  $t \in Q$ , der **akzeptierende Zustand** und
- 9  $r \in Q$ , der **verwerfende Zustand** mit  $t \neq r$ .

## Zusammenfassung der letzten LV

### Satz

Es kann kein Testprogramm für „hello, world“-Programme geben.

### Definition (informell)

Ist ein Problem nicht algorithmisch lösbar, nennen wir es **unentscheidbar**.

### Satz

Die folgenden Probleme sind **unentscheidbar**:

- 1 das Halteproblem
- 2 das Postsche Korrespondenzproblem
- 3 Eindeutigkeit einer beliebigen kontextfreien Grammatik
- 4 das Wortproblem in der Gleichungslogik ( $E \models s \approx t$ )
- 5 ...

Überblick

## Inhalte der Lehrveranstaltung

### Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Formales Beweisen, Konjunktive und Disjunktive Normalformen

### Einführung in die Algebra

Boolesche Algebra, Universelle Algebra, Logische Schaltkreise

### Einführung in die Theorie der Formalen Sprachen

Grammatiken und Formale Sprachen, Reguläre Sprachen, Kontextfreie Sprachen

### Einführung in die Berechenbarkeitstheorie

Algorithmisch unlösbare Probleme, **Turing Maschinen, Registermaschinen**

### Einführung in die Programmverifikation

Prinzipien der Analyse von Programmen, Verifikation nach Hoare, Verschlüsselung und Sicherheit

# Konfigurationen

## Definition

eine **Konfiguration** einer TM  $M$  ist ein Tripel  $(p, z, n)$ , sodass

- 1  $p \in Q$  Zustand,
- 2  $z = y \sqcup^\infty$  Bandinhalt
- 3  $n \in \mathbb{N}$  Position des Lese/Schreibkopfes

$$y \in \Gamma^*$$

## Definition

**Startkonfiguration** bei Eingabe  $x \in \Sigma^*$ :

$$(s, \vdash x \sqcup^\infty, 0)$$

## Beispiel

Die Startkonfiguration bei Eingabe 0011 ist  $(s, \vdash 0011 \sqcup^\infty, 0)$ .

Sei  $M = (\{s, t, r, q_1, q_2, q_3\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1, X, Y\}, \vdash, \sqcup, \delta, s, t, r)$  mit  $\delta$ :

	$\vdash$	$\sqcup$	0	1	X	Y
s	$(s, \vdash, R)$	$(r, \sqcup, R)$	$(q_1, X, R)$	$(r, 1, R)$	$(r, X, R)$	$(q_3, Y, R)$
q <sub>1</sub>	$(r, \vdash, R)$	$(r, \sqcup, R)$	$(q_1, 0, R)$	$(q_2, Y, L)$	$(r, X, R)$	$(q_1, Y, R)$
q <sub>2</sub>	$(r, \vdash, R)$	$(r, \sqcup, R)$	$(q_2, 0, L)$	$(r, 1, R)$	$(s, X, R)$	$(q_2, Y, L)$
q <sub>3</sub>	$(r, \vdash, R)$	$(t, \sqcup, R)$	$(r, 0, R)$	$(r, 1, R)$	$(r, X, R)$	$(q_3, Y, R)$

Wir betrachten die Schrittfunktion für die Eingabe 0011:

$$\begin{aligned}
 &(s, \vdash 0011 \sqcup^\infty, 0) \xrightarrow{1/M} (s, \vdash 0011 \sqcup^\infty, 1) \xrightarrow{1/M} (q_1, \vdash X 011 \sqcup^\infty, 2) \\
 &\xrightarrow{1/M} (q_1, \vdash X 011 \sqcup^\infty, 3) \xrightarrow{1/M} (q_2, \vdash X 0 Y 1 \sqcup^\infty, 2) \xrightarrow{1/M} (q_2, \vdash X 0 Y 1 \sqcup^\infty, 1) \\
 &\xrightarrow{1/M} (s, \vdash X 0 Y 1 \sqcup^\infty, 2) \xrightarrow{1/M} (q_1, \vdash X X Y 1 \sqcup^\infty, 3) \xrightarrow{1/M} (q_1, \vdash X X Y 1 \sqcup^\infty, 4) \\
 &\xrightarrow{1/M} (q_2, \vdash X X Y Y \sqcup^\infty, 3) \xrightarrow{1/M} (q_2, \vdash X X Y Y \sqcup^\infty, 2) \xrightarrow{1/M} (s, \vdash X X Y Y \sqcup^\infty, 3) \\
 &\xrightarrow{1/M} (q_3, \vdash X X Y Y \sqcup^\infty, 4) \xrightarrow{1/M} (q_3, \vdash X X Y Y \sqcup^\infty, 5) \xrightarrow{1/M} (t, \vdash X X Y Y \sqcup^\infty, 6)
 \end{aligned}$$

# Schrittfunktion

## Definition

Schrittfunktion  $\xrightarrow{1/M}$  ist wie folgt definiert:

$$(p, z, n) \xrightarrow{1/M} \begin{cases} (q, z', n-1) & \text{wenn } \delta(p, z_n) = (q, b, L) \\ (q, z', n+1) & \text{wenn } \delta(p, z_n) = (q, b, R) \end{cases}$$

Hier ist  $z'$  das Wort, das wir aus  $z$  erhalten, wenn  $z_n$  durch  $b$  ersetzt wird.

## Definition

$\xrightarrow{*M}$  definieren wir induktiv:

- 1  $\alpha \xrightarrow{0/M} \alpha$
- 2  $\alpha \xrightarrow{k+1/M} \beta$ , wenn  $\alpha \xrightarrow{k/M} \gamma$  und  $\gamma \xrightarrow{1/M} \beta$  für eine Konfiguration  $\gamma$
- 3  $\alpha \xrightarrow{*M} \beta$ , wenn es ein  $k \in \mathbb{N}$  gibt mit  $\alpha \xrightarrow{k/M} \beta$

## Definition

eine TM  $M$

- **akzeptiert** Eingabe  $x \in \Sigma^*$ , wenn  $\exists y \in \Gamma^*, n \in \mathbb{N}$ :  
 $(s, \vdash x \sqcup^\infty, 0) \xrightarrow{*M} (t, y \sqcup^\infty, n)$
- **verwirft** Eingabe  $x \in \Sigma^*$ , wenn  $\exists y \in \Gamma^*, n \in \mathbb{N}$ :  
 $(s, \vdash x \sqcup^\infty, 0) \xrightarrow{*M} (r, y \sqcup^\infty, n)$
- **hält** bei Eingabe  $x$ , wenn  $M$  Eingabe  $x$  akzeptiert oder verwirft
- ist **total**, wenn  $M$  auf **allen** Eingaben hält

## Definition

die **Sprache** einer TM  $M$  ist wie folgt definiert:

$$L(M) := \{x \in \Sigma^* \mid M \text{ akzeptiert } x\}$$

## Satz

Sei  $M$  eine Turingmaschine. Dann ist  $L(M)$  **rekursiv aufzählbar**.

Umgekehrt gibt es zu jeder rekursiv aufzählbaren Sprache  $L$  eine Turingmaschine  $M$  mit  $L = L(M)$ .

## Definition (Berechenbarkeit mit einer TM)

- Gegeben TM  $M = (Q, \{\sqcup, \square\}, \{\vdash, \sqcup, \sqcap, \square\}, \vdash, \sqcup, \delta, s, t, r)$ .  
Eine partielle Funktion  $f: \mathbb{N}^k \rightarrow \mathbb{N}$  heißt  **$M$ -berechenbar**, wenn:

$$f(n_1, \dots, n_k) = m \quad \text{gdw.} \quad (s, \vdash \sqcap^{n_1} \square \dots \square \sqcap^{n_k} \sqcup^\infty, 0) \xrightarrow[M]{*} (t, \vdash \sqcap^m \sqcup^\infty, n)$$

Das Zeichen  $\sqcap$  kodiert Zahlen und  $\square$  dient als Trennsymbol.

- Eine partielle Funktion  $f: \mathbb{N}^k \rightarrow \mathbb{N}$  heißt **berechenbar mit einer TM**, wenn eine TM  $M$  über dem Alphabet  $\{\sqcup, \square\}$  existiert, sodass  $f$   $M$ -berechenbar.

## Church-Turing These

Jedes algorithmisch lösbare Problem ist mit einer Turingmaschine lösbar.

## Registermaschinen

## Definition

Eine **Registermaschine (RM)**  $R$  ist ein Paar  $R = ((x_i)_{1 \leq i \leq n}, P)$  sodass

- 1  $(x_i)_{1 \leq i \leq n}$  eine Sequenz von  $n$  **Registern**  $x_i$ , die **natürliche Zahlen** beinhalten
- 2  $P$  ein Programm

**Programme** sind endliche Folgen von Befehlen und induktiv definiert:

- 1 Für jedes Register  $x_i$  sind die folgenden Instruktionen sowohl Befehle wie Programme:  $x_i := x_i + 1$  und  $x_i := x_i - 1$
- 2 Wenn  $P_1$  und  $P_2$  Programme sind, dann ist  $P_1; P_2$  ein Programm.
- 3 Wenn  $P_1$  ein Programm und  $x_i$  ein Register, dann ist  
**while  $x_i \neq 0$  do  $P_1$  end**

sowohl ein Befehl als auch ein Programm.

## Semantik von Registermaschinen

- 1 Zu Beginn der Berechnung steht die **Eingabe** (in Form natürlicher Zahlen) in den Registern.
- 2 Die Befehle
  - $x_i := x_i + 1$
  - $x_i := x_i - 1$
 bedeuten, dass der Inhalt des Register  $x_i$  um 1 erhöht bzw. erniedrigt wird (falls möglich).
- 3  $P_1; P_2$  bedeutet, dass zunächst das Programm  $P_1$  und dann das Programm  $P_2$  ausgeführt wird.
- 4 Der Befehl (und das Programm)

**while  $x_i \neq 0$  do  $P_1$  end**

bedeutet, der Schleifenrumpf  $P_1$  wird ausgeführt, bis die Bedingung  $x_i \neq 0$  falsch ist.

- 5 RM  $R$  **hält**, wenn kein auszuführender Befehl mehr existiert.

## Beispiel (Addition)

Sei  $R = ((x_i)_{1 \leq i \leq 3}, P)$  eine RM mit folgendem Programm:

```
while  $x_1 \neq 0$  do
   $x_1 := x_1 - 1$ ;
   $x_3 := x_3 + 1$ 
end;
while  $x_2 \neq 0$  do
   $x_2 := x_2 - 1$ ;
   $x_3 := x_3 + 1$ 
end
```

Startet  $R$  mit  $(m, n, 0)$ , dann hält  $R$  mit  $(0, 0, m + n)$ .

## Proseminar

Zuweisungen der Form  $x_i := x_j$  können (mit einem Hilfsregister) als Programm einer Registermaschine formuliert werden.

## Beispiel (Multiplikation)

Sei  $R = ((x_i)_{1 \leq i \leq 5}, P)$  eine RM mit folgendem Programm:

```

 $x_3 := 0;$ 
while  $x_1 \neq 0$  do
   $x_1 := x_1 - 1;$ 
   $x_4 := x_2;$ 
  while  $x_4 \neq 0$  do
     $x_4 := x_4 - 1;$ 
     $x_3 := x_3 + 1$ 
  end
end
end

```

Startet  $R$  mit  $(m, n, 0, 0, 0)$ , dann hält  $R$  mit  $(0, n, m \times n, 0, 0)$ .

## Definition (Berechenbarkeit mit einer RM)

- Gegeben RM  $R = ((x_i)_{1 \leq i \leq n}, P)$ .  
Eine partielle Funktion  $f: \mathbb{N}^k \rightarrow \mathbb{N}$  heißt  **$R$ -berechenbar**, wenn:  
 $f(n_1, \dots, n_k) = m$  gdw.  $R$  mit  $n_i$  in den Registern  $x_i$  für  $1 \leq i \leq k$  startet und  $R$  mit  $n_i$  in den Registern  $x_i$  für  $1 \leq i \leq k$  und  $m$  im Register  $x_{k+1}$  hält.
- Eine partielle Funktion  $f: \mathbb{N}^k \rightarrow \mathbb{N}$  heißt **berechenbar auf einer RM**, wenn eine RM  $R$  existiert, sodass  $f$   $R$ -berechenbar.

## Satz

Jede partielle Funktion  $f: \mathbb{N}^k \rightarrow \mathbb{N}$ , die berechenbar auf einer RM ist, ist auf einer TM berechenbar und umgekehrt.