Komplexitätstheorie

WiSe 2011/12 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Komplexitätstheorie Gesamtübersicht

- Organisatorisches / Einführung
  Motivation / Erinnerung / Fragestellungen
- Diskussion verschiedener Komplexitätsklassen:
  Zeitkomplexität
  Platzkomplexität
- zugehörige Reduktionsbegriffe
- vollständige Probleme
- Anpassung von Klassenbegriffen und Reduktionen

Organisatorisches

Vorlesung: Montags 10-12 Uhr, H 6; Vorschlag ab 2. SW: 10.05-11.35
Zusätzlich in der ersten SW:
Freitag 10-12 Uhr, HZ 201

Übungen (Henning Fernau): Freitags 10-12 Uhr, HZ 201;
Beginn 2. Semesterwoche

Meine Sprechstunde: DO, 13-14 Uhr

Kontakt: fernau@uni-trier.de

Hausaufgaben / Schein ?! n.V. (Master ?! $\rightarrow$ mündliche Prüfung)

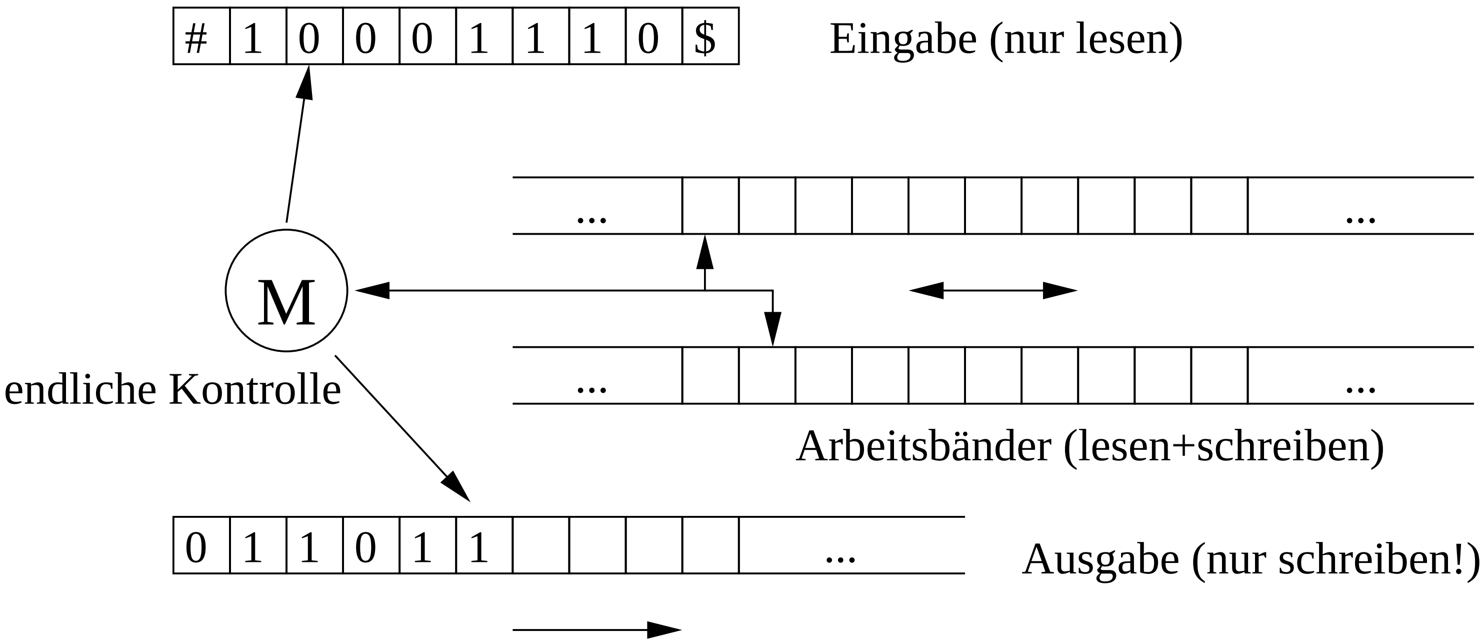
Turingmaschinen wichtigstes Modell der Komplexitätstheorie:
'Turingmaschine' (nach Alan M. Turing, 1936)

keine einheitliche Definition;
aber wesentliche Komponenten meist ähnlich

Resultate von Feinheiten der Definitionen meist unabhängig

Dennoch: exakte Angabe der Definition für spätere Beweise

Prinzipieller Aufbau einer Turingmaschine:



Einzelheiten bei TM:
—Σ, Δ, Γ endliche Alphabete
—$\$, \#$ zwei beliebige Zeichen ('Endemarker'), nicht in Σ enthalten
—$B \in \Gamma$ ('Blank').
Dann ist eine *(nichtdeterministische) k-Band-Turingmaschine M*
mit Eingabealphabet Σ, Ausgabealphabet Δ und Arbeitsalphabet Γ
definiert als ein 7-Tupel

$$(Z, \Sigma, \Delta, \Gamma, \delta, z_0, Z_e)$$

wobei

- Z endliche Zustandsmenge
- $z_0 \in Z$ Anfangszustand
- Z_e Endzustandsmenge

- $\delta\colon Z \setminus Z_e \to \mathcal{P}(\texttt{BEFEHLE})$ *Übergangsfunktion*, wobei

$$
\begin{array}{rcll}
\texttt{BEFEHLE} & = & \{0\} \times \Delta \times Z & \text{(Ausgabe)} \\
& \cup & \{1\} \times (\Sigma \cup \{\#, \$\}) \times Z^2 & \text{(Test der Eingabe)} \\
& \cup & \{1\} \times \{L, R\} \times Z & \text{(Bewegung auf Eingabe)} \\
& \cup & \{2, \ldots, k{-}1\} \times \Gamma \times Z & \text{(Schreiben)} \\
& \cup & \{2, \ldots, k{-}1\} \times \Gamma \times Z^2 & \text{(Test)} \\
& \cup & \{2, \ldots, k{-}1\} \times \{L, R\} \times Z & \text{(Bewegung)}
\end{array}
$$

Enthält $\delta(z)$ für jeden Zustand z höchstens ein Element, so wird M auch als *deterministische Turingmaschine* bezeichnet.

Menge $\mathcal{K}$ der *Konfigurationen* von M = Menge aller Tupel

$$(z, w_0, w_{1,l}, a_1, w_{1,r}, w_{2,l}, a_2, w_{2,r}, \ldots, w_{k-1,l}, a_{k-1}, w_{k-1,r})$$

mit

$$z \in Z$$

$$w_0 \in W(\Delta)$$

$$w_{1,l}, w_{1,r} \in W(\Sigma \cup \{\#, \$\}), a_1 \in \Sigma \cup \{\#, \$\}$$

$$w_{i,l}, w_{i,r} \in W(\Gamma), a_i \in \Gamma \quad (2 \leq i \leq k-1)$$

Dabei

w_0	Inhalt des Ausgabebandes (Band 0)
$w_{1,l}a_1w_{1,r}$	Inhalt des Eingabebandes (Band 1)
$w_{i,l}a_iw_{i,r}$	Inhalt des Arbeitsbandes i
z	Zustand der Konfiguration K

Menge $\mathcal{K}_e$ der *Endkonfiguration*en:

$$\mathcal{K}_e := \{K \in \mathcal{K} \mid \text{ Zustand von } K \text{ ist aus Endzustandsmenge } Z_e\}$$

initiale oder Anfangskonfiguration $I_M(x)$ zu $x \in \mathcal{W}(\Sigma)$:

$$I_M(x) := (z_0, \lambda, \lambda, \#, x\$, \lambda, B, \lambda, \ldots, \lambda, B, \lambda)$$

Einzelschritt der Turingmaschine def. über Relation $\underset{M}{\rightarrow} \subseteq \mathcal{K} \times \mathcal{K}$:

Für zwei Konfigurationen

$$K = (z, w_0, w_{1,l}, a_1, w_{1,r}, w_{2,l}, a_2, w_{2,r}, \ldots, w_{k-1,l}, a_{k-1}, w_{k-1,r})$$

$$K' = (z', w'_0, w'_{1,l}, a'_1, w'_{1,r}, w'_{2,l}, a'_2, w'_{2,r}, \ldots, w'_{k-1,l}, a'_{k-1}, w'_{k-1,r})$$

gilt $K \underset{M}{\rightarrow} K'$, falls einer der folgenden Fälle eintritt:

(jeweils nicht angesprochene Bänder bleiben unverändert)

Es gibt $i \geq 2, x, z_1, z_2$ mit

- (Schreiben auf Ausgabeband)
  $(0, x, z_1) \in \delta(z)$ mit $w_0' = w_0 x$ und $z_1 = z'$

- (Testen auf Eingabeband und Verzweigen)
  $(1, x, z_1, z_2) \in \delta(z)$ mit $a_1 = x, z_1 = z'$ oder $a_1 \neq x, z_2 = z'$

- (Kopf auf Eingabeband nach links)
  $(1, L, z_1) \in \delta(z)$ mit $z_1 = z'$ und entweder $w_{1,l}' a_1' = w_{1,l}, w_{1,r}' = a_1 w_{1,r}$ oder $a_1' = a_1 = \#, w_{1,l}' = w_{1,l}, w_{1,r}' = w_{1,r}$

- (Kopf auf Eingabeband nach rechts)
  $(1, R, z_1) \in \delta(z)$ mit $z_1 = z'$ und entweder $w_{1,l}' = w_{1,l} a_1, w_{1,r} = a_1' w_{1,r}'$ oder $a_1' = a_1 = \$, w_{1,l}' = w_{1,l}, w_{1,r}' = w_{1,r}$

- (Schreiben auf Arbeitsband i)
  $(i, x, z_1) \in \delta(z)$ mit $z_1 = z', a_i' = x, w_{i,l}' = w_{i,l}, w_{i,r}' = w_{i,r}$

- (Testen auf Arbeitsband i und Verzweigen)
  $(i, x, z_1, z_2) \in \delta(z)$ mit entweder $a_i = x, z_1 = z'$ oder $a_i \neq x, z_2 = z'$, stets jedoch $w_{i,l}' = w_{i,l}, w_{i,r}' = w_{i,r}$

- (Kopf auf Arbeitsband i nach links)
  $(i, L, z_1) \in \delta(z)$ mit $z_1 = z'$ und und im Falle $w_{i,l} \neq \lambda$: $w_{i,l}' a_i' = w_{i,l}, w_{i,r}' = a_i w_{i,r}$ und im Falle $w_{i,l} = \lambda$: $w_{i,l}' = \lambda, a_i' = B, w_{i,r}' = a_i w_{i,r}$

- (Kopf auf Arbeitsband i nach rechts)
  $(i, R, z_1) \in \delta(z)$ mit $z_1 = z'$ und im Falle $w_{i,r} \neq \lambda$: $a_i' w_{i,r}' = w_{i,r}, w_{i,l}' = w_{i,l} a_i$ und im Falle $w_{i,r} = \lambda$: $w_{i,r}' = \lambda, a_i' = B, w_{i,l}' = w_{i,r} a_i$

Anmerkungen:

—nichtdet. TM: $\delta(z)$ evtl. mehrere Elemente
$\Rightarrow$ zu K mehrere Folgekonfigurationen $K', K'', K''' \ldots$ mit $K \underset{M}{\rightarrow} K'$, $K \underset{M}{\rightarrow} K''$, $K \underset{M}{\rightarrow} K''$ möglich

—det. TM: stets höchstens eine Folgekonfiguration

—Endkonfiguration oder $\delta(z) = \emptyset$: keine Folgekonfiguration!

—Eingabeband: Bandendemarkierungen # und $\$$ erreichbar/testbar,
Bandende nicht berschreitbar, Eingabe read-only!

—Arbeitsbänder: bei Bedarf neue Leerzeichen hinzugefügt
$\Rightarrow$ TM 'sieht' unendlich lange Bänder
aber: benutzter Speicher bleibt erkennbar!

—initiale Konfiguration: leeres Ausgabeband, leere Arbeitsbänder, Kopf der Eingabe auf dem linken Bandende

Mehrere Schritte einer TM

—n-fache Iteration $\xrightarrow[M]{n} \subseteq \mathcal{K} \times \mathcal{K}$ der Einzelschritt-Relation:

Es gilt $K \xrightarrow[M]{n} K'$, wenn Konfigurationen $K_0, \ldots, K_n$ existieren mit

$K_0 = K$, $K_{i-1} \xrightarrow[M]{} K_i$ für $1 \leq i \leq n$ sowie $K_n = K'$.

—n-fache Iteration entspricht n Schritten der Maschine M.

—$K \xrightarrow[M]{0} K' \iff K = K'$

—$\xrightarrow[M]{*} \subseteq \mathcal{K} \times \mathcal{K}$: reflexive und transitive Hülle der Relation, d.h.

$$K \xrightarrow[M]{*} K' \iff (\exists n) K \xrightarrow[M]{n} K'$$

deterministische Turingmaschinen:

—Relationen $\xrightarrow[M]{n}$ sind (partielle) Funktionen $\xrightarrow[M]{n}: \mathcal{K} \dashrightarrow \mathcal{K}$

—exakt festgelegt, wie Konfiguration nach n Schritten aussieht

—δ für Endzustände nicht definiert:
bei det. TM zu jeder Anfangskonfiguration $I_M(x)$ (maximal) eine Endkonfiguration K mit

$$I_M(x) \xrightarrow[M]{*} K$$

Die von einer det. TM M *berechnete Funktion*

$$f_M: W(\Sigma) \dashrightarrow W(\Delta)$$

ist definiert durch:Es gilt $f_M(x) = y$ genau dann, wenn eine Endkonfiguration K existiert, die als Inhalt des Ausgabebandes das Wort y hat und für die $I_M(x) \xrightarrow[M]{*} K$ gilt.

Achtung, drei Fälle möglich:

(1) von $I_M(x)$ wird in einer gewissen Zahl von Schritten eine Endkonfiguration erreicht,

(2) nach einer gewissen Zahl von Schritten wird Konfiguration erreicht, für deren Zustand z die Menge $\delta(z)$ der möglichen Befehle leer ist (und daher keine Folgekonfigurationen mehr existieren)

(3) zu jeder erreichten Konfiguration existiert Folgekonfiguration, aber es tritt dabei nie Fall (1) oder (2) ein, d.h. die Maschine kommt nie zu einem Halt.

Nur im Fall (1) ist $f_M(x)$ definiert!

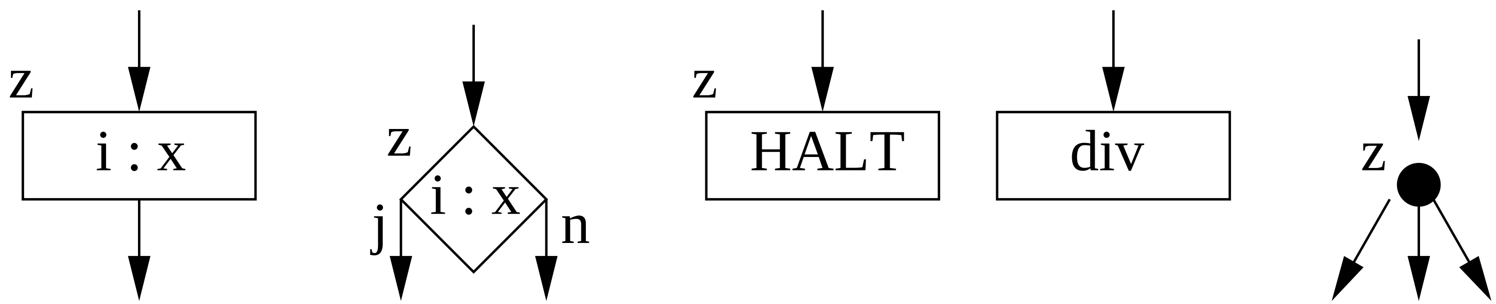
Angabe von Turingmaschinen z.B. graphisch mit Flussdiagrammen:

—Für jedem Zustand der Maschine ein Knoten im Diagramm

—Bezeichnung der einzelnen Zustände meist unwichtig

—Zahl der Arbeitsbänder / Arbeitsalphabet gesondert oder implizit

—im Flussdiagramm 5 Bausteine möglich:



Einschub:

Maschine, die zu $\Sigma = \Delta = \{0, 1\}$

die Funktion $f_M(x) = xx^r$ berechnet.

nichtdeterministische Turingmaschinen:

—aus initialer Konf. evtl. mehrere Endkonfigurationen erreichbar

—'berechnete Funktion' nicht definiert

Die von einer (n.det.) TM M *akzeptierte (oder berechnete) Sprache*

$$L_M \subseteq W(\Sigma)$$

ist definiert durch:

Es gilt $x \in L_M$ genau dann, wenn eine Endkonfiguration K existiert, für die gilt:

$$I_M(x) \xrightarrow[M]{*} K$$

(Dann wird das Wort x von M 'akzeptiert' oder 'erkannt')

Ein Wort wird von einer n.det. TM also akzeptiert, wenn diese die *Möglichkeit* hat, sich ‘richtig’ zu verhalten.

Akzeptanz bei det. TM: als Spezialfall der Definition

Erreichbarkeit einer Endkonfiguration: unabhängig von Ausgabe
$\Rightarrow$ für Akzeptanz sind Δ und Ausgabeband unwichtig!

Einschub:

Maschine, die zu $\Sigma = \{0, 1\}$

die Menge $\{xx^r \mid x \in W(\Sigma)\}$ akzeptiert.

Anmerkungen:

—Definition der ‘Akzeptanz’ bei $x \in L_M$ gibt keine Auskunft darüber, was bei $x \notin L_M$ geschieht.

—entspricht üblicher Definition bei *rekursiv aufzählbaren* oder *beweisbaren* Mengen

—Nach unserer Definition:
bei $x \notin L_M$ hält M mit Fehler oder läuft endlos weiter

—Wenn ‘Anzeige’ der Nichtakzeptanz gewünscht:
Untersuche *totale* (d.h. für alle Argumente definierte) Funktionen f mit der Eigenschaft $x \in L \Leftrightarrow f(x) = 1$

—f (mit TM) berechenbar $\Rightarrow$ L ist *rekursiv* oder *entscheidbar*

—Entscheidbarkeit $\Rightarrow$ Beweisbarkeit

—Beweisbarkeit $\not\Rightarrow$ Entscheidbarkeit

—Literatur: bei det. TM ‘Akzeptanz’ als Entscheidbarkeit definiert...
$\Rightarrow$ Vorsicht!!!!