Berechenbarkeit und Komplexität

Vorlesung #2: Die Turingmaschine

Prof. Berthold Vöcking
präsentiert durch Prof. Joost-Pieter Katoen

21. Oktober 2008

Zentrale Fragestellung

Welche Funktionen sind durch einen Computer berechenbar?

bzw.

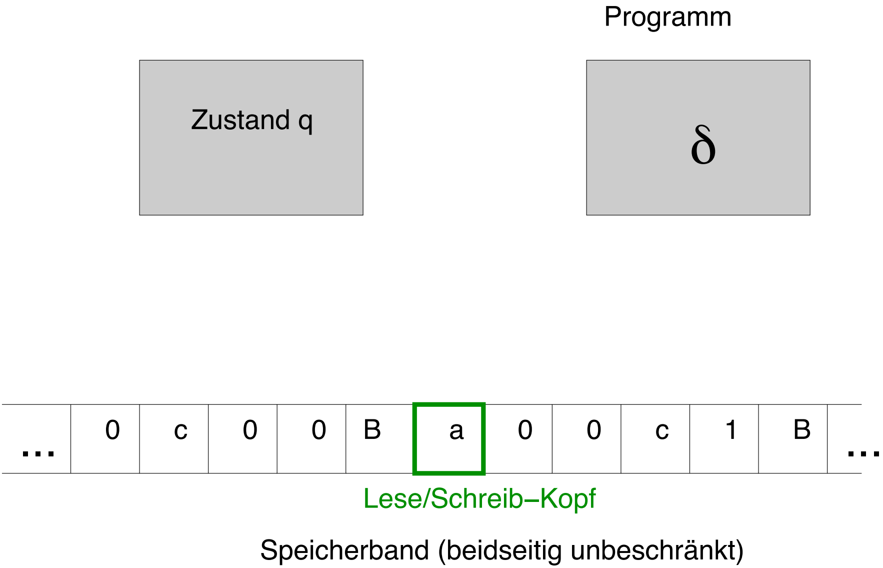
Welche Sprachen kann eine Computer entscheiden?

Um diese Fragen in einem mathematisch exakten Sinne klären zu können, müssen wir noch klären was eigentlich ein Computer ist.

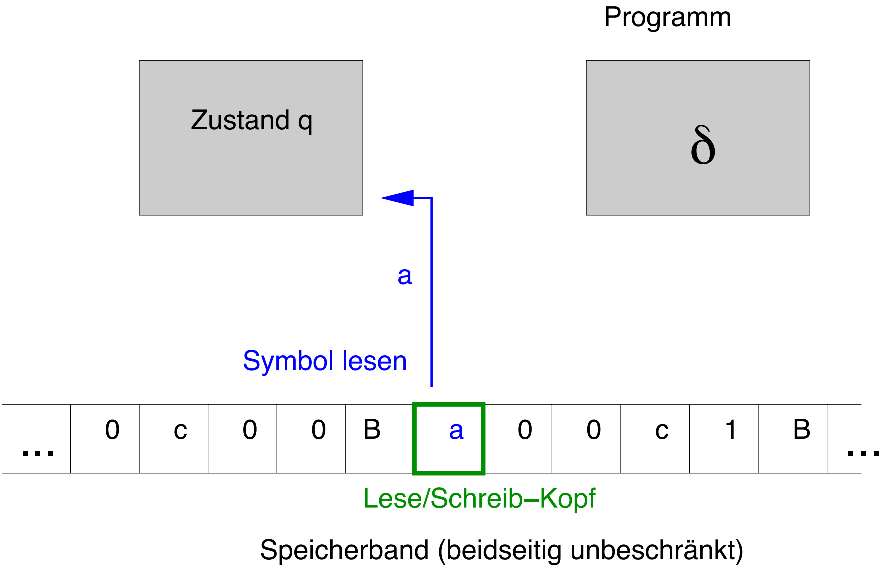
Wir benötigen ein mathematisches Rechnermodell.

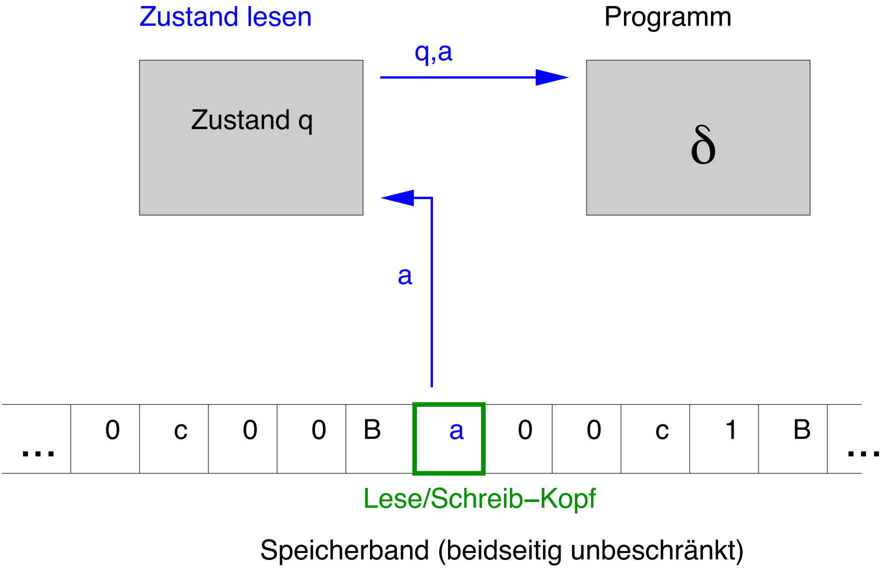
Dies ist die Turingmaschine.

Deterministische Turingmaschinen (TM bzw. DTM)

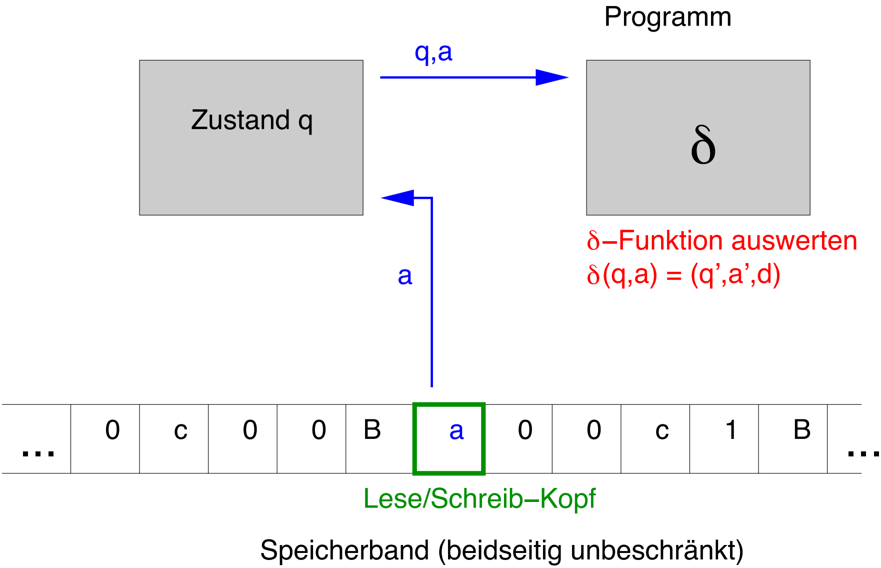


Deterministische Turingmaschinen (TM bzw. DTM)

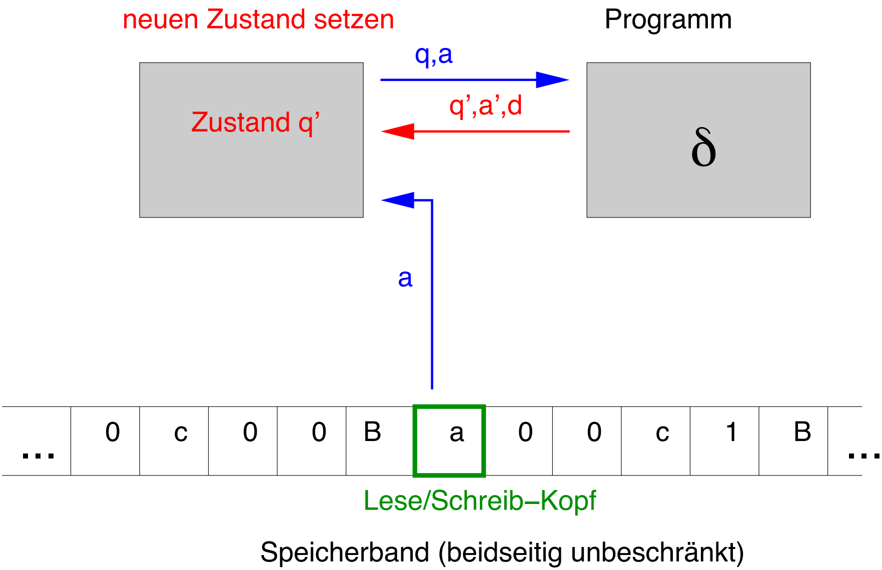



Zustand lesen
Programm
q,a
Zustand q
δ
a
... 0 c 0 0 B a 0 0 c 1 B ...
Lese/Schreib–Kopf
Speicherband (beidseitig unbeschränkt)

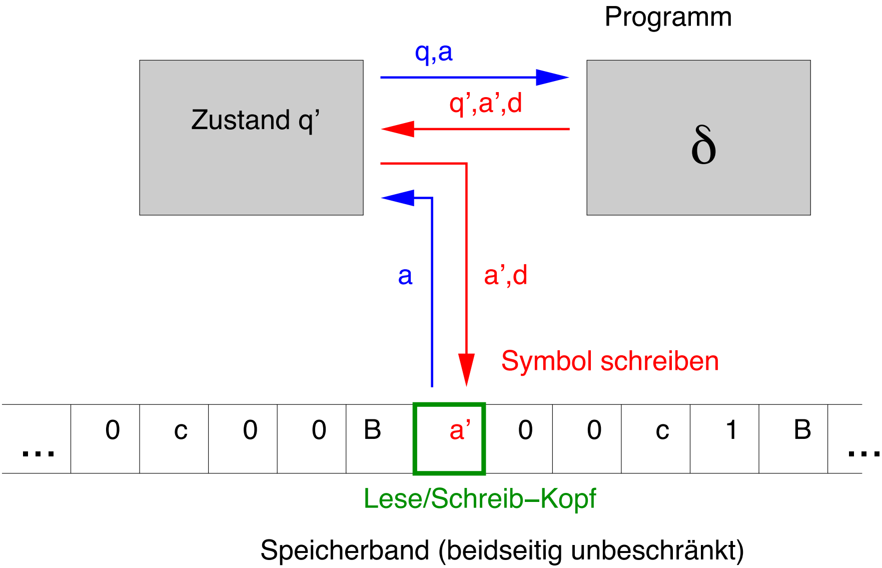
Deterministische Turingmaschinen (TM bzw. DTM)



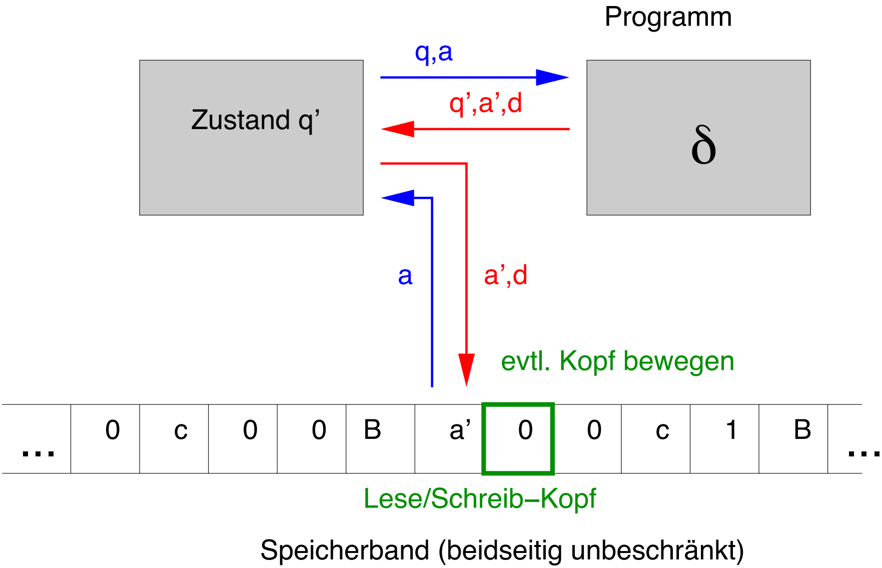
Deterministische Turingmaschinen (TM bzw. DTM)



Deterministische Turingmaschinen (TM bzw. DTM)



Deterministische Turingmaschinen (TM bzw. DTM)



Komponenten der TM

- Q, die endliche Zustandsmenge
- Σ, das endliche Eingabealphabet
- $\Gamma \supset \Sigma$, das endliche Bandalphabet
- $B \in \Gamma \setminus \Sigma$, das Leerzeichen (Blank)
- $q_0 \in Q$, der Anfangszustand
- $\bar{q} \in Q$, der Endzustand
- $\delta : (Q \setminus \bar{q}) \times \Gamma \to Q \times \Gamma \times \{R, L, N\}$, die Zustandsüberführungsfunktion

Eine TM ist definiert durch das 7-Tupel $(Q, \Sigma, \Gamma, B, q_0, \bar{q}, \delta)$.

Funktionsweise der TM

Ausgangssituation

- auf dem Band steht die Eingabe $w \in \Sigma^*$ eingerahmt von Blanks
- der initiale Zustand ist q_0
- der Kopf steht über dem ersten Symbol von w

Nummerierung der Zellen des Bandes

- die initiale Kopfposition wird als Position 0 bezeichnet
- bewegt sich der Kopf einen Schritt „nach rechts" erhöht sich die Position um 1
- bewegt sich der Kopf um einen Schritt „nach links" erniedrigt sich die Position um 1

Funktionsweise der TM

Durchführung eines Rechenschrittes

- $a \in \Gamma$ bezeichne das gelesene Symbol
- $q \in Q \setminus \bar{q}$ bezeichne den aktuellen Zustand
- es sei $\delta(q, a) = (q', a', d)$, für $q' \in Q, a' \in \Gamma, d \in \{R, L, N\}$
- dann wird der Zustand auf q' gesetzt
- an der Kopfposition wird das Symbol a' geschrieben
- der Kopf

$$\text{bewegt sich} \begin{cases} \text{um eine Position nach rechts} & \text{falls } d = R \\ \text{um eine Position nach links} & \text{falls } d = L \\ \text{nicht} & \text{falls } d = N \end{cases}$$

Ende der Rechnung

- die TM stoppt, wenn sie den Endzustand $\bar{q}$ erreicht
- das Ausgabewort $w' \in \Sigma^*$ kann dann vom Band abgelesen werden: w' beginnt an der Kopfposition und endet unmittelbar vor dem ersten Symbol aus $\Gamma \setminus \Sigma$
- *Spezialfall:* wenn wir es mit Entscheidungsproblemen zu tun haben, wird die Antwort wie folgt als JA oder NEIN interpretiert:
  - die TM *akzeptiert* das Eingabewort, wenn sie terminiert und das Ausgabewort mit einer 1 beginnt
  - die TM *verwirft* das Eingabewort, wenn sie terminiert und das Ausgabewort nicht mit einer 1 beginnt
- Beachte, es gibt die Möglichkeit, dass die Rechnung nicht terminiert.

Bemerkungen

- Beachte, es gibt die Möglichkeit, dass die TM den Endzustand niemals erreicht. Wir sagen dann, die *Rechnung terminiert nicht*.
- Laufzeit = Anzahl der Zustandsübergänge bis zur Terminierung
- Speicherplatz = Anzahl der Bandzellen, die während der Rechnung besucht werden

Funktionsweise der TM am Beispiel

Sei $L = \{w1 | w \in \{0,1\}^*\}$.

L wird *entschieden* durch die TM $M = (Q, \Sigma, \Gamma, B, q_0, \bar{q}, \delta)$ mit

- $Q = \{q_0, q_1, \bar{q}\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, B\}$
- δ gemäß Tabelle

δ	0	1	B
q_0	(q_0, B, R)	(q_1, B, R)	reject
q_1	(q_0, B, R)	(q_1, B, R)	accept

„accept“ steht als Abkürzung für $\bar{q}, 1, N$.
„reject“ steht als Abkürzung für $\bar{q}, 0, N$.

Die Übergangsfunktion entspricht dem *Programm der TM*.

Funktionsweise der TM am Beispiel

Beschreibung des Programms als Tabelle:

δ	0	1	B
q_0	(q_0, B, R)	(q_1, B, R)	reject
q_1	(q_0, B, R)	(q_1, B, R)	accept

Verbale Beschreibung des Programms:

- Solange ein Symbol aus $\{0, 1\}$ gelesen wird
  - überschreibe das Symbol mit B,
  - bewege den Kopf nach rechts, und
  - gehe in den Zustand q_0, wenn das Symbol eine 0 war, sonst in den Zustand q_1
- Sobald ein Blank gelesen wird, so
  - akzeptiere die Eingabe, falls der aktuelle Zustand q_1 ist, und
  - verwirf die Eingabe ansonsten.

Definition des Begriffes *berechenbar*

Definition

Eine Funktion $f : \Sigma^* \rightarrow \Sigma^*$ heit *rekursiv (berechenbar)*, wenn es eine TM gibt, die aus der Eingabe x den Funktionswert $f(x)$ berechnet.

Definition

Eine Sprache $L \subseteq \Sigma^*$ heit *rekursiv (entscheidbar)*, wenn es eine TM gibt, die auf allen Eingaben stoppt und die Eingabe w genau dann akzeptiert, wenn $w \in L$ ist.

Beispiel: Wir entwickeln eine TM für die Sprache

$$L = \{0^n 1^n | n \geq 1\} \ .$$

Sei $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, B\}$, $Q = \{q_0, \ldots, q_6, \bar{q}\}$.

Unsere TM arbeitet in zwei Phasen:

- **Phase 1:** Teste, ob das Eingabewort von der Form $0^i 1^j$ für $i \geq 0$ und $j \geq 1$ ist.
- **Phase 2:** Test, ob $i = j$ gilt.

Phase 1 verwendet q_0 und q_1 und wechselt bei Erfolg zu q_2.
Phase 2 verwendet q_2 und q_6 und akzeptiert bei Erfolg.

Programmierung der TM am Beispiel - Phase 1

δ	0	1	B
q_0	$(q_0, 0, R)$	$(q_1, 1, R)$	reject
q_1	reject	$(q_1, 1, R)$	(q_2, B, L)

q_0 : Laufe von links nach rechts über die Eingabe bis ein Zeichen ungleich 0 gefunden wird.

- Falls dieses Zeichen eine 1 ist, gehe über in Zustand q_1.
- Sonst ist dieses Zeichen ein Blank. Verwirf die Eingabe.

q_1 : Gehe weiter nach rechts bis zum ersten Zeichen ungleich 1.

- Falls dieses Zeichen eine 0 ist, verwirf die Eingabe.
- Sonst ist das gefundene Zeichen ein Blank. Bewege den Kopf um eine Position nach links auf die letzte gelesene 1. Wechsel in den Zustand q_2, Phase 2 beginnt.

Programmierung der TM am Beispiel - Phase 2

δ	0	1	B
q_2	reject	(q_3, B, L)	reject
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	(q_4, B, R)
q_4	(q_5, B, R)	reject	reject
q_5	$(q_6, 0, R)$	$(q_6, 1, R)$	accept
q_6	$(q_6, 0, R)$	$(q_6, 1, R)$	(q_2, B, L)

q_2 : Kopf steht auf dem letzten Nichtblank. Falls dieses Zeichen eine 1 ist, so lösche es, gehe nach links, und wechsel in den Zustand q_3. Sonst verwirf die Eingabe.

q_3 : Bewege den Kopf auf das erste Nichtblank. Dann q_4.

q_4 : Falls das gelesene Zeichen eine 0 ist, ersetze es durch ein Blank und gehe nach q_5, sonst verwirf die Eingabe.

q_5 : Wir haben jetzt die linkeste 0 und die rechteste 1 gelscht. Falls Restwort leer, dann $\bar{q}$ (akzeptiere), sonst q_6.

q_6 : Laufe wieder zum letzten Nichtblank und starte erneut in q_2.

Konfigurationen und (direkte) Nachfolgekonfigurationen

Definition

i) Eine *Konfiguration* einer TM ist ein String $\alpha q \beta$, fr $q \in Q$ und $\alpha, \beta \in \Gamma^*$. Bedeutung: auf dem Band steht $\alpha\beta$ eingerahmt von Blanks, der Zustand ist q, und der Kopf steht unter dem ersten Zeichen von β.

ii) $\alpha' q' \beta'$ ist *direkte Nachfolgekonfiguration* von $\alpha q \beta$, falls $\alpha' q' \beta'$ in einem Rechenschritt aus $\alpha q \beta$ ensteht. Wir schreiben $\alpha q \beta \vdash \alpha' q' \beta'$.

iii) $\alpha'' q'' \beta''$ ist *Nachfolgekonfiguration* von $\alpha q \beta$, falls $\alpha'' q'' \beta''$ in endlich vielen Rechenschritten aus $\alpha q \beta$ ensteht. Wir schreiben $\alpha q \beta \vdash^* \alpha'' q'' \beta''$.

Bemerkung: insbesondere gilt $\alpha q \beta \vdash^* \alpha q \beta$.

Beispiel zum Umgang mit Konfigurationen

Die für die Sprache $L = \{0^n 1^n | n \geq 1\}$ beschriebene TM liefert in Phase 1 auf die Eingabe 0011 die folgende Konfigurationsfolge.

Phase 1:

$$q_0 0011 \vdash 0 q_0 011 \vdash 00 q_0 11 \vdash 001 q_1 1 \vdash 0011 q_1 B \vdash 001 q_2 1$$

Beobachtung: abgesehen von Blanks am Anfang und Ende des Strings sind die Konfigurationsbeschreibungen eindeutig.