

Grundlagen der Theoretischen Informatik

Turingmaschinen und rekursiv aufzählbare Sprachen (III)

30.06.2016

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Übersicht

1. Motivation
2. Terminologie
3. Endliche Automaten und reguläre Sprachen
4. Kellerautomaten und kontextfreie Sprachen
5. Turingmaschinen und rekursiv aufzählbare Sprachen
6. Berechenbarkeit, (Un-)Entscheidbarkeit
7. Komplexitätsklassen P und NP

Turing-Maschine

Definition (Turing Machine (DTM))

Eine **determinierte Turing-Maschine (DTM)** \mathcal{M} ist ein Tupel

$$\mathcal{M} = (K, \Sigma, \delta, s)$$

Dabei ist

- K eine endliche Menge von Zuständen mit $h \notin K$,
(h ist der **Haltezustand**)
- Σ ein Alphabet mit $L, R \notin \Sigma$, $\# \in \Sigma$,
- $\delta : K \times \Sigma \rightarrow (K \cup \{h\}) \times (\Sigma \cup \{L, R\})$ eine **Übergangsfunktion**
- $s \in K$ ein Startzustand.

Wir erlauben auch, dass δ nicht überall definiert ist.

Falls die DTM dann in einen solchen nichtdefinierten Zustand kommt, sagen wir **die DTM hängt**. Sie **hält** also nicht.

Dies wird z.T. in der Literatur anders gehandhabt.

Turing-Maschine

Definition Konfiguration einer DTM $\mathcal{M} = (K, \Sigma, \delta, s)$: Wort der Form $C = q, w \underline{a} u$, wobei:

- $q \in K \cup \{h\}$ der aktuelle Zustand,
- $w \in \Sigma^*$ der Bandinhalt links des Kopfes,
- $a \in \Sigma$ das Bandzeichen unter der Schreib-/Lesekopf.
- $u \in \Sigma^* (\Sigma - \{\#\}) \cup \{\epsilon\}$ der Bandinhalt rechts des Kopfes.

Definition $C_1 \vdash_{\mathcal{M}} C_2$ (C_2 ist **Nachfolgekonfiguration** von C_1) falls:

- $C_i = q_i, w_i \underline{a}_i u_i$ für $i \in \{1, 2\}$, und
- es gibt einen Übergang $\delta(q_1, a_1) = (q_2, b)$ wie folgt:
 - Fall 1: $b \in \Sigma$. Dann ist $w_1 = w_2$, $u_1 = u_2$, $a_2 = b$.
 - Fall 2: $b = L$. Dann gilt für w_2 und a_2 : $w_1 = w_2 a_2$.
Für u_2 : Wenn $a_1 = \#$ und $u_1 = \epsilon$ ist, so ist $u_2 = \epsilon$, sonst ist $u_2 = a_1 u_1$.
 - Fall 3: $b = R$. Dann ist $w_2 = w_1 a_1$.
Für a_2 und u_2 gilt: Wenn $u_1 = \epsilon$ ist, dann ist $u_2 = \epsilon$ und $a_2 = \#$,
ansonsten ist $u_1 = a_2 u_2$.

Turing-Maschine

Definition (Eingabe) w heißt **Eingabe** (*input*) für \mathcal{M} , falls \mathcal{M} mit der **Startkonfiguration** $C_0 = s, \#w\#$ startet.

(w_1, \dots, w_n) heißt **Eingabe** für \mathcal{M} , falls \mathcal{M} mit der **Startkonfiguration**

$$C_0 = s, \#w_1\# \dots \#w_n\#$$

startet.

Definition (Halten, Hängen) Sei \mathcal{M} eine Turing-Maschine.

- \mathcal{M} **hält** in $C = q, w\underline{a}u$ gdw. $q = h$.
- \mathcal{M} **hängt** in $C = q, w\underline{a}u$ gdw. es keine Nachfolgekonfiguration gibt
Insbesondere: wenn $w = \epsilon \wedge \exists q' \delta(q, a) = (q', L)$.

Turing-Maschine

Definition (Rechnung) Sei \mathcal{M} eine Turing-Maschine. Man schreibt $C \vdash_{\mathcal{M}}^* C'$ gdw.:
es gibt eine Reihe von Konfigurationen C_0, C_1, \dots, C_n ($n \geq 0$) so dass
 $C = C_0$, $C' = C_n$ und für alle $i < n$ gilt: $C_i \vdash_{\mathcal{M}} C_{i+1}$

Dann heißt C_0, C_1, \dots, C_n eine **Rechnung** der Länge n von C_0 nach C_n .

Definition (TM-berechenbare Funktion) Sei Σ_0 ein Alphabet mit $\# \notin \Sigma_0$.

Eine (partielle) Funktion $f : (\Sigma_0^*)^m \rightarrow (\Sigma_0^*)^n$ heißt **DTM-berechenbar**, falls:

Es existiert eine determinierte Turing-Maschine $\mathcal{M} = (K, \Sigma, \delta, s)$

- mit $\Sigma_0 \subseteq \Sigma$,
- so dass für alle $w_1, \dots, w_m, u_1, \dots, u_n \in \Sigma_0^*$ gilt:
 - $f(w_1, \dots, w_m) = (u_1, \dots, u_n)$ gdw $s, \#w_1\# \dots \#w_m\# \vdash_{\mathcal{M}}^* h, \#u_1\# \dots \#u_n\#$
 - $f(w_1, \dots, w_m)$ ist undefiniert gdw

\mathcal{M} gestartet mit $s, \#w_1\# \dots \#w_m\#$ hält nicht (läuft unendlich oder hängt)

Berechnete Funktion/Akzeptierte Sprache

Bei Turing-Maschinen untersuchen wir,

- **welche Sprachen sie akzeptieren** und
- **welche Funktionen sie berechnen.**

Akzeptieren ist Spezialfall von Berechnen

Definition (Von einer DTM akzeptierte Sprache)

Ein Wort w wird **akzeptiert von einer DTM \mathcal{M}** ,

falls \mathcal{M} auf Eingabe von w hält

(wobei am Ende der Kopf auf dem ersten Blank rechts von w steht).

Eine Sprache $L \subseteq \Sigma^*$ **wird akzeptiert von einer DTM \mathcal{M}** , wenn genau die Wörter aus L aus \mathcal{M} und keine anderen akzeptiert werden.

Achtung

Bei nicht akzeptierten Wörtern muss die DTM nicht halten

Sie darf es sogar nicht!

TM-Flussdiagramme

Graphische Darstellung der Übergangsfunktion einer DTM: mit einem Flußdiagramm.

- Die Zustandsnamen werden nicht genannt.
- Nur die Schritte und die Ausführungsreihenfolge werden beschrieben.

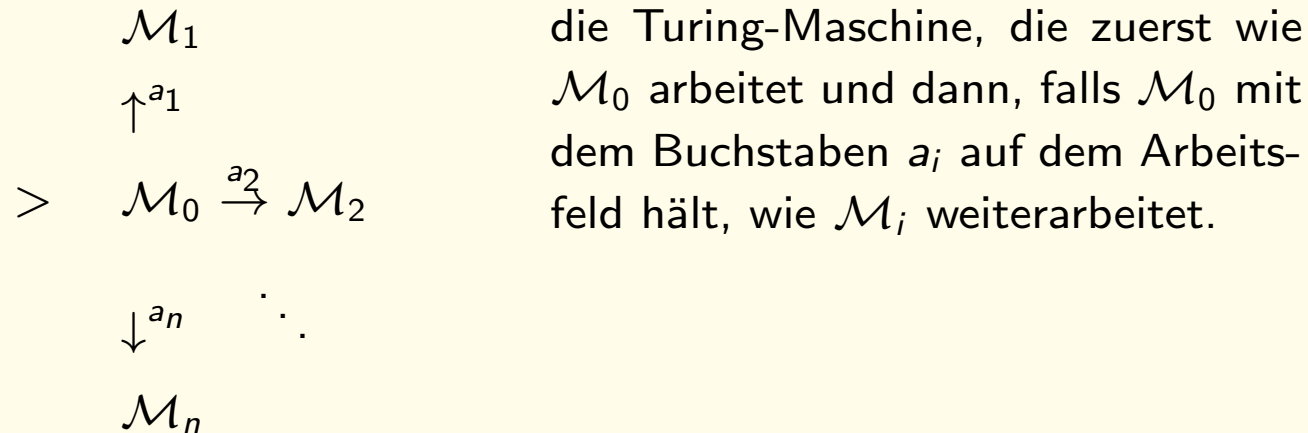
TM-Flussdiagramme

Folgende Elemente stehen zur Verfügung:

- L : eine DTM, die nach dem Starten ein Feld nach links geht und danach hält.
- R : eine DTM, die nach dem Starten ein Feld nach rechts geht und danach hält.
- a : TM, die a auf dem Band schreibt und danach hält.
- Startschritt: mit einer Pfeilspitze $>$ bezeichnet
- $M_1 \longrightarrow M_2$ oder abgekürzt $M_1 M_2$ (falls $\mathcal{M}_1, \mathcal{M}_2$ die Flußdiagramme zweier DTM sind):
eine DTM die zuerst wie M_1 arbeitet und dann, falls M_1 hält, wie M_2 weiterarbeitet.
- $M_1 \xrightarrow{a} M_2$: M_2 ist nur dann aufgeführt, wenn nach der Beendigung von M_1 der aktuelle Bandbuchstabe a ist.

TM-Flussdiagramme

Sind $\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_n$ Turing-Maschinen, $a_i \in \Sigma$ für $1 \leq i \leq n$, so ist



- Weitere Schreibabkürzungen sind:

$\xrightarrow{\sigma \neq a}$ für $\sigma \in \Sigma - \{a\}$

$M_1 \xrightarrow{\sigma \neq a} M_2$: M_2 ist nur dann aufgeführt, wenn M_1 hält, und Lesekopf auf Buchstabe, die nicht a ist positioniert ist.

$M_1 \xrightarrow{a,b} M_2$ falls nach der Ausführung von M_1 sowohl für den Bandbuchstaben a als auch für b nach M_2 verzweigt werden soll.

TM-Flussdiagramme

Beispiel:

Die DTM $\mathcal{M}^+ = (\{s, q_1, q_2, q_3, q_4\}, \{ |, \# \}, \delta, s)$ addiert zwei natürliche Zahlen in Unärdarstellung.

Sie rechnet

$$s, \# |^n \# |^m \underline{\#} \vdash_{\mathcal{M}^+}^* h, \# |^{n+m} \underline{\#}$$

Der Trick: Sie löscht den letzten Strich von $|^m$ und schreibt ihn in den Zwischenraum zwischen $|^n$ und $|^m$.

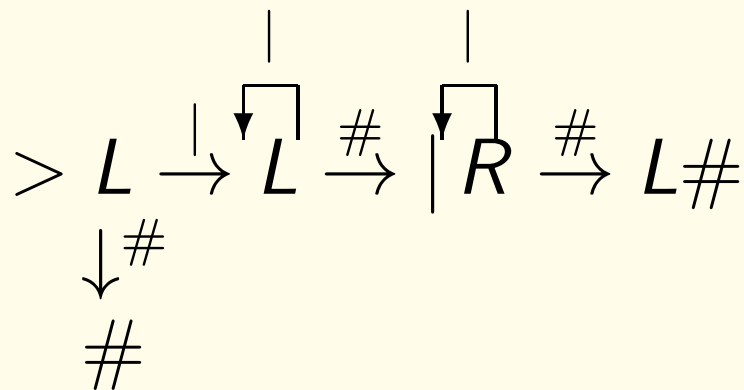
TM-Flussdiagramme

Beispiel: Hier ist zunächst die δ -Funktion:

$$\begin{array}{llll}
 s, \# & \mapsto & q_1, L & \quad q_2, \# & \mapsto & q_3, | & \quad q_3, \# & \mapsto & q_4, L \\
 q_1, \# & \mapsto & h, \# & \quad q_2, | & \mapsto & q_2, L & \quad q_4, | & \mapsto & h, \# \\
 q_1, | & \mapsto & q_2, L & \quad q_3, | & \mapsto & q_3, R & & &
 \end{array}$$

Für $\delta(s, |)$ und $\delta(q_4, \#)$ haben wir keine Werte angegeben; sie sind beliebig, weil \mathcal{M}^+ sie nie benötigt.

Das Flußdiagramm zur gleichen DTM ist erheblich leichter zu lesen:



TM-Flussdiagramme

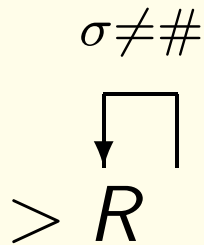
Die folgenden Turing-Maschinen werden wir als Bestandteile von komplexeren Turing-Maschinen später noch häufig benutzen.

Beispiel: ($R_{\#}$)

$R_{\#}$ bewegt nur den Kopf, ohne zu schreiben:

- Sie geht mindestens einmal nach rechts.
- Dann geht sie solange weiter nach rechts, bis sie ein $\#$ liest.

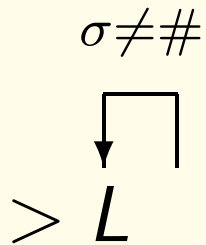
Die folgende Turing-Maschine $R_{\#}$ läuft zum ersten Blank rechts von der momentanen Position.



TM-Flussdiagramme

Beispiel: ($L_{\#}$) Analog funktioniert die DTM $L_{\#}$:

- Sie geht mindestens einmal nach links.
- Dann geht sie solange weiter nach links, bis sie ein $\#$ liest.

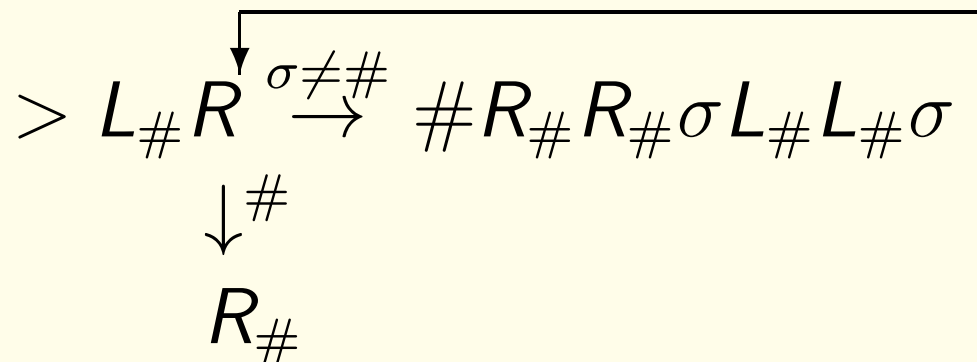


TM-Flussdiagramme

Beispiel: (\mathcal{C}) Die folgende DTM \mathcal{C} erhält als Eingabe einen String Einsen.

Sie rechnet so:

- Sie bewegt sich nach links auf das Blank vor das Eingabewort,
- geht das Eingabewort von links nach rechts durch,
- merkt sich jeweils ein Zeichen σ von w , markiert die aktuelle Position, indem sie σ mit $\#$ überschreibt,
- und kopiert das Zeichen σ .
- Sie verwendet dabei die Maschinen $L_{\#}$ und $R_{\#}$, die wir schon definiert haben.



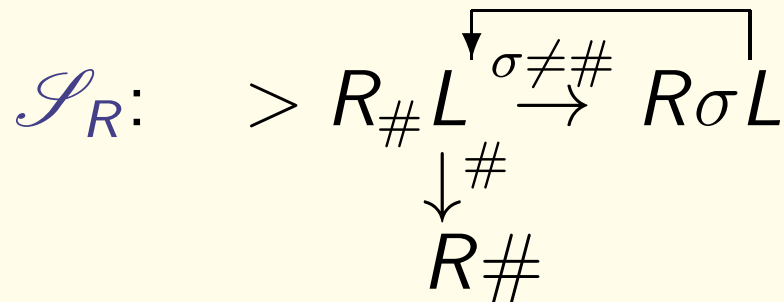
TM-Flussdiagramme

Beispiel: Die DTM \mathcal{S}_R bewirkt eine “Verschiebung nach rechts”, das heißt, wenn \mathcal{S}_R das Alphabet Σ besitzt, rechnet sie

$$s, w_1 \underline{\#} w_2 \# w_3 \vdash_{\mathcal{S}_R}^* h, w_1 \# \underline{\#} w_2 w_3$$

für alle Wörter $w_1, w_3 \in \Sigma^*$ und $w_2 \in (\Sigma - \{\#\})^*$. (Entgegen der sonstigen Konvention startet sie zwischen zwei Eingabewörtern.)

Sie arbeitet so:



TM-Flussdiagramme

Beispiel: Dazu invers arbeitet die Maschine \mathcal{S}_L , die einen “shift nach links” bewirkt. Sie rechnet

$$s, w_1 \# w_2 \# w_3 \vdash_{\mathcal{S}_L}^* h, w_1 w_2 \# \# w_3$$

für alle $w_1, w_3 \in \Sigma^*$, $w_2 \in (\Sigma - \{\#\})^*$. Sie ist definiert als

$$\mathcal{S}_L: \quad > \begin{array}{c} \begin{array}{ccc} & \overbrace{} & \\ & \downarrow & \\ L\#R & \xrightarrow{\sigma \neq \#} & L\sigma R \\ & \downarrow \# & \\ & L\# & \end{array} \end{array}$$

Varianten von Turing-Maschinen

Variationen von Turing-Maschinen

Standard-DTM

Die Turing-Maschine, die wir bisher kennen, ...

- ist determiniert
- hat ein einseitig unbeschränktes Band (**Halbband**).

Ab jetzt nennen wir sie auch:

Standard-Turing-Maschine (Standard-DTM oder kurz DTM)

Variationen von Turing-Maschinen

Turing-Maschinen, die nie hängen

Gegeben:

Eine Turing-Maschine \mathcal{M} , mit Eingabe $\#w\#$

Daraus konstruieren wir eine DTM \mathcal{M}' , die

- dasselbe berechnet wie \mathcal{M}
- **nie hängt.**

Variationen von Turing-Maschinen

Konstruktion der TM, die nie hängt

Das Bandende ist am Anfang ein Zeichen links vom Eingabewort

DTM \mathcal{M}' rechnet so:

- Sie verschiebt die Eingabe ein Zeichen nach rechts.
- Dann druckt sie ganz links ein **Sonderzeichen** α , das das **Bandende** anzeigt.
- Ab dann rechnet sie wie \mathcal{M} .
- Aber:
Wenn sie α erreicht, bleibt sie dort stehen und druckt immer wieder α .

Variationen von Turing-Maschinen

Eigenschaften der nicht-hängenden DTM

- \mathcal{M}' hält für Eingabe w gdw \mathcal{M} hält für Eingabe w .
- \mathcal{M}' hängt nie.
Wenn \mathcal{M} hängt, rechnet \mathcal{M}' unendlich lang.

O.B.d.A. sollen alle Turing-Maschinen, die wir von jetzt an betrachten, nie hängen.

DTM mit zweiseitig unbeschränktem Band

DTM mit zweiseitig unbeschränktem Band (zw-DTM)

- Die Definition der Maschine bleibt gleich.
- Die Definition der **Konfiguration** ändert sich.
- Sie hat immer noch die Form $q, w\underline{a}u$, aber:
 - w umfasst analog zu u alle Zeichen bis zum letzten nicht-Blank links vom Schreib-/Lesekopf.
 - $w = \epsilon$ bzw. $u = \epsilon$ bedeutet, dass links bzw. rechts vom Schreib-/Lesekopf nur noch Blanks stehen.

DTM mit zweiseitig unbeschränktem Band

Definition (DTM mit zweiseitig unbeschränktem Band, zw-DTM)

Eine **Turing-Maschine mit zweiseitig unbeschränktem Band (zw-DTM)** ist eine DTM, für die die Begriffe der Konfiguration und der Nachfolgekonfiguration wie folgt definiert sind:

Eine **Konfiguration** C einer zw-DTM $\mathcal{M} = (K, \Sigma, \delta, s)$ ist von der Form

$$C = q, w\underline{a}u$$

Dabei ist

- $q \in K \cup \{h\}$ der aktuelle Zustand,
- $w \in (\Sigma - \{\#\})\Sigma^* \cup \{\epsilon\}$ der Bandinhalt links des Kopfes,
- $a \in \Sigma$ das Zeichen unter dem Kopf, und
- $u \in \Sigma^*(\Sigma - \{\#\}) \cup \{\epsilon\}$ der Bandinhalt rechts des Kopfes.

DTM mit zweiseitig unbeschränktem Band

Definition (Forts.)

$C_2 = q_2, w_2 \underline{a_2} u_2$ heißt **Nachfolgekonfiguration** von $C_1 = q_1, w_1 \underline{a_1} u_1$,

in Zeichen $C_1 \vdash_{\mathcal{M}} C_2$, falls es einen Übergang $\delta(q_1, a_1) = (q_2, b)$ gibt, mit:

Fall 1: $b \in \Sigma$. Dann $w_1 = w_2, u_1 = u_2$ und $a_2 = b$.

Fall 2: $b = L$. Für u_2 : Wenn $a_1 = \#$ und $u_1 = \epsilon$ ist, dann $u_2 = \epsilon$, sonst $u_2 = a_1 u_1$.

Für a_2 und w_2 : Wenn $w_1 = \epsilon$ ist, dann $w_2 = \epsilon$ und $a_2 = \#$; sonst $w_1 = w_2 a_2$.

Fall 3: $b = R$. Für w_2 : Wenn $a_1 = \#$ und $w_1 = \epsilon$ ist, dann $w_2 = \epsilon$, sonst $w_2 = w_1 a_1$.

Für a_2 und u_2 : Wenn $u_1 = \epsilon$ ist, dann $u_2 = \epsilon$ und $a_2 = \#$; ansonsten $u_1 = a_2 u_2$.

DTM mit zweiseitig unbeschränktem Band

Theorem (Simulation von zw-DTM durch DTM)

Zu jeder zw-DTM \mathcal{M} , die eine Funktion f berechnet oder eine Sprache L akzeptiert, existiert eine Standard-DTM \mathcal{M}' , die ebenfalls f berechnet bzw. L akzeptiert.

DTM mit zweiseitig unbeschränktem Band

Theorem (Simulation von zw-DTM durch DTM)

Zu jeder zw-DTM \mathcal{M} , die eine Funktion f berechnet oder eine Sprache L akzeptiert, existiert eine Standard-DTM \mathcal{M}' , die ebenfalls f berechnet bzw. L akzeptiert.

Beweis Sei $w = a_1 \dots a_n$ die Eingabe für $\mathcal{M} = (K, \Sigma, \delta, s)$.

Dann sieht das beidseitig unendliche Band zu Beginn der Rechnung so aus:

$\dots \#\#\#a_1 \dots a_n\#\#\dots$

DTM mit zweiseitig unbeschränktem Band

Beweis (Forts.) **Idee:**

- \mathcal{M} hat quasi zwei unendlich lange Halbbänder.
- Ziel ist, den Inhalt beider Halbbänder von \mathcal{M} auf einem unterzubringen.
- Dazu: Den Teil des Bandes, der zwei Zeichen links vom Input w beginnt, umklappen:

$$\begin{array}{l} \text{Spur 1 } \# \# \dots \# \# \\ \text{Spur 2 } \# a_1 \dots a_n \underline{\#} \dots \end{array}$$

- Die DTM \mathcal{M}' hat zwei **Spuren**, d.h. zwei Reihen von Zeichen, die auf demselben Band untergebracht (kodiert) sind.
- Das Bandalphabet von \mathcal{M}' ist $\Sigma' \supseteq \Sigma \times \Sigma$.

DTM mit zweiseitig unbeschränktem Band

Beweis (Forts.) Sei $\mathcal{M}' = (K', \Sigma', \delta', s)$. \mathcal{M}' rechnet so:

- \mathcal{M}' legt zunächst eine zweite Spur an,
- simuliert dann die Arbeit von \mathcal{M} , und
- transformiert dann das Ergebnis wieder auf nur eine Spur herunter.