Grundlagen der Theoretischen Informatik

Turingmaschinen und rekursiv aufzählbare Sprachen

10.06.2021

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Bis jetzt

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

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

5. Turingmaschinen und rekursiv aufzählbare Sprachen

Inhalt von Teil 5

- Was ist eine **berechenbare** Funktion?
- Determinierte Turing-Maschinen (DTMs)?
- Modifikationen von DTMs:
  (mehrere) Halbbänder, zweiseitig unbeschränkte Bänder
- Indeterminierte Turing-Maschinen (NTMs)
- **Gödelisieren**: Programme als Wörter in Σ^*.
- **Aufzählbar** vs. **entscheidbar**
- **Unentscheidbarkeit**, Reduktionen von Problemen aufeinander.

Determinierte Turing-Maschinen (DTMs)

Grundlegende Fragen

- **Frage: Berechenbarkeit?**

  Betrachtet werden Abbildungen über den natürlichen Zahlen $\mathbb{N}$:

  Welche davon sollen berechenbar genannt werden?

Determinierte Turing-Maschinen (DTMs)

Grundlegende Fragen

- **Frage: Berechenbarkeit?**

  Betrachtet werden Abbildungen über den natürlichen Zahlen $\mathbb{N}$:

  Welche davon sollen berechenbar genannt werden?

- **Frage: Komplexität?**

  Um die Komplexität eines Algorithmus' zu messen braucht man ein Maschinenmodell zum Vergleich!

Determinierte Turing-Maschinen (DTMs)

Grundlegende Fragen

- **Frage: Berechenbarkeit?**

  Betrachtet werden Abbildungen über den natürlichen Zahlen $\mathbb{N}$:

  Welche davon sollen berechenbar genannt werden?

- **Frage: Komplexität?**

  Um die Komplexität eines Algorithmus' zu messen braucht man ein Maschinenmodell zum Vergleich!

Welches Modell wird gewählt?

Robustheit:
Das Modell soll nicht von einfachen Modifikationen abhängig sein.

Antwort: Turing-Maschinen

Alan Turing 1912–1954

- Mathematiker und Logiker
- Einer der Begründer der Informatik
- 1936: Definition des Berechenbarkeitsmodells "Turing-Maschine"
- 1938: Promotion bei Church in Princeton
- Während des zweiten Weltkriegs: Kriegsentscheidender Beitrag zur Entschlüsselung deutscher Funksprüche
- Dozent an der Universität Manchester
- Beiträge zur KI ("Turing-Test")
- Tragischer Tod: Strafverfolgung wegen Homosexualität; (vermutlich) Selbstmord
- Nach ihm benannt: Turing-Award

Turing machines

Alan Turing described a Turing machine (which he called "Logical Computing Machine"), as consisting of:

> " ... an unlimited memory capacity obtained in the form of an infinite tape marked out into squares, on each of which a symbol could be printed.
>
> At any moment there is one symbol in the machine; it is called the scanned symbol.
>
> The machine can alter the scanned symbol and its behavior is in part determined by that symbol, but the symbols on the tape elsewhere do not affect the behaviour of the machine.
>
> However, the tape can be moved back and forth through the machine, this being one of the elementary operations of the machine. Any symbol on the tape may therefore eventually have an innings."

Idee

Immer mächtigere Automaten

- Erinnerung: Pushdown-Automaten
  - akzeptieren kontextfreie Sprachen
  - Erster Speicher: der Zustand (endlich)
  - Zweiter Speicher: der **Keller**
    (unbeschränkte Größe, beschränkte Zugriffsart)
  - Das Eingabewort wird nur einmal gelesen, von links nach rechts.

Idee

Immer mächtigere Automaten

- Ausblick: Turing-Maschinen
  - akzeptieren Sprachen **vom Typ 0**.
  - Erster Speicher: der Zustand (endlich)
  - Zweiter Speicher: Band
    (unbeschränkte Größe, **Zugriff an beliebiger Stelle**)
  - Turing-Maschine hat einen Schreib-/Lesekopf, den sie über diesem Band in einem Rechenschritt um ein Feld nach rechts oder links bewegen kann.
  - Das Eingabewort steht (am Anfang) auf dem Band.
    Die Maschine kann es **beliebig oft lesen**.

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 endliches Alphabet mit $L, R \notin \Sigma$, $\# \in \Sigma$,
- $\delta : K \times \Sigma \to (K \cup \{h\}) \times (\Sigma \cup \{L, R\})$ eine Übergangs**funktion**
- $s \in K$ ein Startzustand.

Anzahl der Zustände: $|K| - 1$
(Startzustand wird nicht mitgezählt).

Turing-Maschine

Arbeitsschritt einer Turing-Maschine

Übergang

$$\delta(q, a) = (q', x)$$

bedeutet:

In Abhängigkeit

- vom aktuellen Zustand $q \in K$
- von dem Zeichen $a \in \Sigma$, das unter dem Schreib-/Lesekopf steht

geschieht folgendes:

- entweder ein **Schritt nach links**, falls $x = L$ ist
- oder ein **Schritt nach rechts**, falls $x = R$ ist
- oder **das Zeichen** a, das momentan unter dem Schreib-/Lesekopf steht, wird **durch $b \in \Sigma$ überschreiben**, falls $x = b \in \Sigma$
- der **Zustand** wird zu $q' \in K \cup \{h\}$ **geändert**,

Turing-Maschine

Leerzeichen

Das spezielle Zeichen $\#$ (*blank*) ist das Leerzeichen.

Es ist nie Teil des Eingabeworts; man kann es u.a. dazu benutzen, Wörter voneinander abzugrenzen.

Begrenzung des Bandes

Das Band einer DTM ist **einseitig unbeschränkt**:

- Nach rechts ist es unendlich lang.
- Nach links hat es ein Ende.
- Wenn eine DTM versucht, das Ende zu überschreiten, bleibt sie „hängen“.

  In diesem Fall hält sie nicht.

Turing-Maschine

Anfangskonfiguration

- Ganz links auf dem Band steht ein Blank
- Direkt rechts davon steht das Eingabewort
- Wenn eine DTM mehrere Eingabewörter hintereinander bekommt, sind sie durch Blanks getrennt.
- Rechts vom letzten Eingabewort stehen nur noch Blanks.
- Der Schreib-/Lesekopf der DTM steht auf dem Blank direkt rechts neben dem (letzten) Eingabewort.

Merke:

Das Band enthält immer nur endlich viele Symbole, die keine Blanks sind.

Turing-Maschine

Beispiel 1: $\mathscr{R}(a)$: a's durch b's ersetzen

Die folgende Turing-Maschine $\mathscr{R}(a)$ erwartet *ein* Eingabewort.
Sie liest es von rechts nach links einmal durch und macht dabei jedes a zu einem b.

Es ist

$$\mathscr{R}(a) = (\ \{q_0, q_1\}, \{a, b, \#\}, \delta, q_0\ )$$

mit folgender δ-Funktion:

$$\begin{array}{rclcrcl}
q_0, \# & \mapsto & q_1, L & \quad & q_1, \# & \mapsto & h, \# \\
q_0, a & \mapsto & q_0, a & & q_1, a & \mapsto & q_1, b \\
q_0, b & \mapsto & q_0, b & & q_1, b & \mapsto & q_1, L
\end{array}$$

Turing-Maschine

Beispiel 2 ($L_{\#}$)

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

Es ist $L_{\#} = (\ \{q_0, q_1\}, \{a, b, \#\}, \delta, q_0\ )$ mit folgender δ-Funktion:

$$\begin{array}{rcl@{\qquad}rcl}
q_0, \# & \mapsto & q_1, L & q_1, \# & \mapsto & h, \# \\
q_0, a & \mapsto & q_1, L & q_1, a & \mapsto & q_1, L \\
q_0, b & \mapsto & q_1, L & q_1, b & \mapsto & q_1, L
\end{array}$$

q_0: Anfangsposition
q_1: Anfangsposition verlassen

Turing-Maschine

Beispiel 3 ($\mathscr{C}$) Die folgende DTM $\mathscr{C}$ erhält als Eingabe einen String Einsen.

Dieser String wird kopiert:
Falls n Einsen auf dem Band stehen, stehen nach Ausführung von $\mathscr{C}$ $2n$ Einsen auf dem Band (getrennt durch ein Blank $\#$).

state	$\#$	1	c
q_0	$\langle q_1, c\rangle$	—	—
q_1	$\langle q_2, R\rangle$	$\langle q_1, L\rangle$	$\langle q_1, L\rangle$
q_2	—	$\langle q_3, \#\rangle$	$\langle q_7, \#\rangle$
q_3	$\langle q_4, R\rangle$	—	—
q_4	$\langle q_5, 1\rangle$	$\langle q_4, R\rangle$	$\langle q_4, R\rangle$
q_5	$\langle q_6, 1\rangle$	$\langle q_5, L\rangle$	$\langle q_5, L\rangle$
q_6	—	$\langle q_2, R\rangle$	—
q_7	$\langle q_8, R\rangle$	—	—
q_8	$\langle h, \#\rangle$	$\langle q_8, R\rangle$	—

Turing-Maschine

Übergangsfunktion δ nicht Überall definiert

Wir erlauben ab jetzt 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

Beispiel 4 (Print n)

Für jedes $n \in \mathbb{N}$ konstruieren wir eine Maschine, die genau n Einsen auf das leere Band schreibt (**mit möglichst wenig Zuständen**):

1. schreibe $\lfloor \frac{n}{2} \rfloor$ viele Einsen auf das Band
   (höchstens $\lfloor \frac{n}{2} \rfloor$ Zustände)
2. kopiere diesen String
   (8 Zustände)
3. ersetze das trennende $\#$ durch eine 1
4. falls n gerade ist, ersetzen die letzte 1 durch $\#$ (2 neue Zustände)

Insgesamt können wir n Einsen mit höchstens $\lfloor \frac{n}{2} \rfloor + 10$ Zuständen konstruieren

Turing-Maschine

Begriff der Konfigurationen

- **Konfiguration** beschreibt die *komplette* aktuelle Situation der Maschine in einer Rechnung.
- Eine **Rechnung** ist eine Folge von Konfigurationen, wobei immer von einer Konfiguration zu einer Nachfolgekonfiguration übergegangen wird.

Besteht aus 4 Elementen:

- das aktuellen Zustand q,
- das Wort w links vom Schreib-/Lesekopf,
- das Zeichen a, auf dem der Kopf gerade steht,
- das Wort u rechts von der aktuellen Kopfposition.

Turing-Maschine

Konfigurationen sind endlich

- w enthält das Anfangsstück des Bandes vom linken Ende bis zur aktuellen Kopfposition.
- **Links ist das Band endlich!**
  $w = \varepsilon$ bedeutet, dass der Kopf ganz links steht
- u enthält den Bandinhalt rechts vom Schreib-/Lesekopf bis zum letzten Zeichen, das kein Blank ist.
- **Nach rechts ist das Band unendlich, aber es enthält nach rechts von einer bestimmten Bandposition an nur noch Blanks**.
  $u = \epsilon$ bedeutet, dass rechts vom Schreib-/Lesekopf nur noch Blanks stehen.

Turing-Maschine

Definition (Konfiguration einer DTM)
Eine **Konfiguration** C einer DTM $\mathcal{M} = (\ K, \Sigma, \delta, s\ )$ ist ein Wort der Form $C = q, w\underline{a}u$. Dabei ist

- $q \in K \cup \{h\}$ der aktuelle Zustand,
- $w \in \Sigma^*$ der Bandinhalt links des Kopfes,
- $a \in \Sigma$ das Bandzeichen unter der Schreib-/Lesekopf.
  Notation: Die Position des Schreib-/Lesekopfes ist durch einen Unterstrich gekennzeichnet.
- $u \in \Sigma^*(\Sigma - \{\#\}) \cup \{\epsilon\}$ der Bandinhalt rechts des Kopfes.

Turing-Maschine

Definition (Nachfolgekonfiguration)

Eine Konfiguration C_2 heißt **Nachfolgekonfiguration** von C_1, in Zeichen

$$C_1 \vdash_{\mathcal{M}} C_2$$

falls gilt:

- $C_i = q_i, w_i \underline{a_i} u_i$ 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 gilt: 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**

$$C_0 = s, \#w\underline{\#}$$

startet.

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

$$C_0 = s, \#w_1\# \ldots \#w_n\underline{\#}$$

startet.

Turing-Maschine

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

$$C_0, C_1, \ldots, C_n \qquad (n \geq 0)$$

so dass

- $C = C_0$ und $C' = C_n$
- für alle $i < n$ gilt: $C_i \vdash_{\mathcal{M}} C_{i+1}$

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

Turing-Maschine können Funktionen berechnen

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 $w_1, \ldots, w_m,\ u_1, \ldots, u_n \in {\Sigma_0}^*$ gilt:
  - $f(w_1, \ldots, w_m) = (u_1, \ldots, u_n)$ <u>gdw</u>

    $s, \#w_1\# \ldots \#w_m\underline{\#} \vdash^*_{\mathcal{M}} h, \#u_1\# \ldots \#u_n\underline{\#}$

  - $f(w_1, \ldots, w_m)$ ist undefiniert <u>gdw</u>

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

Turing-Maschine können Funktionen berechnen

Vorsicht

Wir betrachten Turing-Maschinen hier unter einem anderen Aspekt als alle bisherigen Automaten:

- Bei endlichen Automaten und Pushdown-Automaten haben wir untersucht, **welche Sprachen sie akzeptieren**.

- Bei Turing-Maschinen untersuchen wir,
  - **welche Sprachen sie akzeptieren** und
  - **welche Funktionen sie berechnen**.

Akzeptieren ist Spezialfall von Berechnen

Turing-Maschine: Akzeptierte Sprache

Definition (Von einer DTM akzeptierte Sprache)
Ein Wort w wird **akzeptiert von einer DTM $\mathcal{M}$**,
falls $\mathcal{M}$ auf Eingabe von w hält
(Manchmal auch verlangt: 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: Funktionen auf natürlichen Zahlen

Funktionen auf natürlichen Zahlen

- Wir verwenden die **Unärdarstellung**

  Eine Zahl n wird auf dem Band der Maschine durch n senkrechte Striche dargestellt.

- Eine Turing-Maschine $\mathcal{M}$ berechnet eine Funktion

  $$f:\ \mathbb{N}^k \to \mathbb{N}^n$$

  in Unärdarstellung wie folgt:

  – Wenn $f(i_1, \ldots, i_k) = (j_1, \ldots, j_n)$ ist, dann rechnet $\mathcal{M}$

  $$s, \#|^{i_1}\# \ldots \#|^{i_k}\underline{\#} \ \vdash^*_{\mathcal{M}}\ h, \#|^{j_1}\# \ldots \#|^{j_n}\underline{\#}$$

  – Ist $f(i_1, \ldots, i_k)$ undefiniert, dann hält $\mathcal{M}$ bei Input $\#|^{i_1}\# \ldots \#|^{i_k}\#$ nicht.

TM: Funktionen auf natürlichen Zahlen

Definition

- $\mathbf{TM}^{part}$ ist die Menge der partiellen TM-berechenbaren Funktionen $f : \mathbb{N}^k \to \mathbb{N}$
- **TM** ist die Menge der totalen TM-berechenbaren Funktionen $f : \mathbb{N}^k \to \mathbb{N}$

Achtung: Einschränkung

In der Definition von TM und TM^{part} haben wir uns eingeschränkt:

- nur Funktionen über natürliche Zahlen
- nur Funktionen mit einstelligem Wertebereich

Das ist keine echte Einschränkung

Elemente (Wörter) aus anderen Definitions- und Wertebereiche können als natürliche Zahlen kodiert werden.

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

TM-Flussdiagramme

- $\mathcal{M}_1 \longrightarrow \mathcal{M}_2$ oder abgekürzt $\mathcal{M}_1\mathcal{M}_2$ (falls $\mathcal{M}_1$, $\mathcal{M}_2$ die Flußdiagramme zweier DTM sind):
  eine DTM die zuerst wie $\mathcal{M}_1$ arbeitet und dann, falls $\mathcal{M}_1$ hält, wie $\mathcal{M}_2$ weiterarbeitet.

  Direkt aufeinanderfolgende Schritte werden also

  – entweder direkt nebeneinander notiert

  – oder durch einen Pfeil verbunden.

  Im Gegensatz zu der Maschine $\mathcal{M}_1$ gilt also für $\mathcal{M}_1\mathcal{M}_2$: **Nachdem $\mathcal{M}_1$ seine Arbeit beendet hat, ist $\mathcal{M}_1\mathcal{M}_2$ nicht im Haltezustand, sondern im Startzustand von $\mathcal{M}_2$.**

TM-Flussdiagramme

- $\mathcal{M}_1 \xrightarrow{a} \mathcal{M}_2$: $\mathcal{M}_2$ ist nur dann aufgeführt, wenn nach der Beendigung von $\mathcal{M}_1$ der aktuelle Bandbuchstabe a ist.

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

$$\begin{array}{cl} & \mathcal{M}_1 \\ & \uparrow^{a_1} \\ > & \mathcal{M}_0 \overset{a_2}{\to} \mathcal{M}_2 \\ & \downarrow^{a_n} \quad \ddots \\ & \mathcal{M}_n \end{array}$$

die Turing-Maschine, die zuerst wie $\mathcal{M}_0$ arbeitet und dann, falls $\mathcal{M}_0$ mit dem Buchstaben a_i auf dem Arbeitsfeld hält, wie $\mathcal{M}_i$ weiterarbeitet.

TM-Flussdiagramme

$\boldsymbol{\sigma}$ — eine Schreibabkürzung für einen beliebigen Buchstaben aus Σ.

- Die Maschine $\mathcal{M} \xrightarrow{\sigma} \ldots \sigma$ zum Beispiel ist eine Abkürzung für

$$
\begin{array}{ll}
 & \ldots a_1 \\
 & \uparrow^{a_1} \\
> & \mathcal{M} \xrightarrow{a_2} \ldots a_2 \\
 & \downarrow^{a_n} \qquad \ddots \\
 & \ldots a_n
\end{array}
$$

falls $\Sigma = \{a_1, \ldots, a_n\}$ ist.

TM-Flussdiagramme

Die Maschine $> L \stackrel{\sigma}{\rightarrow} R\sigma R$ für $\Sigma = \{\#, |\}$ macht also zuerst einen Schritt nach links; steht hier ein $\#$ (bzw. ein $|$), so geht sie einen Schritt nach rechts, druckt $\#$ (bzw. $|$) und geht ein weiteres Feld nach rechts.

- Weitere Schreibabkürzungen sind:

  $\stackrel{\boldsymbol{\sigma} \neq a}{\rightarrow}$ 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.

  $\boldsymbol{\mathcal{M}}_1 \stackrel{a,b}{\rightarrow} \boldsymbol{\mathcal{M}}_2$ falls nach der Ausführung von $\mathcal{M}_1$ sowohl für den Bandbuchstaben a als auch für b nach $\mathcal{M}_2$ 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:

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

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:

$$
\begin{tikzcd}
> L \arrow[r, "|"] \arrow[d, "\#"] & L \arrow[loop above, "|"] \arrow[r, "\#"] & {|}R \arrow[loop above, "|"] \arrow[r, "\#"] & L\# \\
\#
\end{tikzcd}
$$

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.

$$\begin{tikzcd} > R \arrow[loop above, "\sigma \neq \#"] \end{tikzcd}$$

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.

$$\sigma \neq \#$$

$$> L$$

TM-Flussdiagramme

Beispiel: ($\mathscr{C}$) Die folgende DTM $\mathscr{C}$ 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.

$$\begin{tikzcd}
> L_\# R \arrow[r, "\sigma\neq\#"] \arrow[d, "\#"] & \# R_\# R_\# \sigma L_\# L_\# \sigma \arrow[l, bend right] \\
R_\#
\end{tikzcd}$$

TM-Flussdiagramme

Beispiel: Die DTM $\mathscr{S}_R$ bewirkt eine "Verschiebung nach rechts", das heißt, wenn $\mathscr{S}_R$ das Alphabet Σ besitzt, rechnet sie

$$s, w_1\underline{\#}w_2\#w_3 \vdash^*_{\mathscr{S}_R} h, w_1\#\underline{\#}w_2w_3$$

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:

$$\mathscr{S}_R: \quad \begin{tikzcd} > R_{\#} L \arrow[r, "\sigma \neq \#"] \arrow[d, "\#"] & R\sigma L \arrow[l, bend right] \\ R\# \end{tikzcd}$$

TM-Flussdiagramme

Beispiel: Dazu invers arbeitet die Maschine $\mathscr{S}_L$, die einen "shift nach links" bewirkt. Sie rechnet

$$s, w_1\#w_2\underline{\#}w_3 \vdash^*_{\mathscr{S}_L} h, w_1w_2\underline{\#}\#w_3$$

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

$$\mathscr{S}_L: \quad \begin{tikzcd} > L_\# R \arrow[r, "\sigma \neq \#"] \arrow[d, "\#"] & L\sigma R \arrow[l, bend right] \\ L\# & \end{tikzcd}$$

Übersicht

- Determinierte Turing-Maschinen (DTMs)
- Eingabe, Konfiguration, Rechnung
- TM-berechenbare Funktion
- Von einer DTM akzeptierte Sprache
- Flussdiagramme