Weitere Begriffe

Definition: Sei L eine Sprache über Σ_0 mit $\{\#, N, Y\} \cap \Sigma_0 = \emptyset$, und sei $M = (K, \Sigma, \delta, s)$ eine Turing-Maschine mit $\Sigma_0 \subseteq \Sigma$.

- M **entscheidet** L, falls für alle $w \in \Sigma_0^*$:

$$s, \#w\underline{\#} \vdash_M^* \begin{cases} h, \#Y\underline{\#} & \text{falls } w \in L; \\ h, \#N\underline{\#} & \text{sonst;} \end{cases}$$

- M **akzeptiert** $w \in \Sigma_0^*$, falls M bei Input w hält.

  M akzeptiert L, falls für alle $w \in \Sigma_0^*$ gilt:

  – M akzeptiert $w \iff w \in L$.

- M **zählt** L **auf**, falls es ein $q_0 \in K$ gibt s.d.

$$L \quad = \quad \{w \in \Sigma_0^* \mid \exists u \text{ s.d. } s, \underline{\#} \vdash_M^* q_0, \#w\underline{\#}u\}$$

- L ist **entscheidbar** $\iff$ es gibt eine Turing-Maschine M die L entscheidet.
- L ist **akzeptierbar** (oder **semi-entscheidbar**) $\iff$ es gibt eine Turing-Maschine die L akzeptiert.
- L ist **rekursiv aufzählbar** (**r.e.**, "recursively enumerable") $\iff$ es gibt eine Turing-Maschine die L aufzählt.

Satz:

1. Jede entscheidbare Sprache ist akzeptierbar.
2. Das Komplement $\overline{L}$ einer entscheidbaren Sprache L ist entscheidbar.

Beweis:

1. Sei L entscheidbar und M eine Turing-Maschine die L entscheidet.

   $\Longrightarrow$ L wird von einer Maschine M' akzeptiert, die zunächst M simuliert und danach in eine Endlosschleife geht, falls M mit $h, \#N\underline{\#}$ endet.

2. Sei L entscheidbar und M eine Turing-Maschine die L entscheidet.

   Betrachte eine Maschine M' die genau wie M rechnet, außer daß M' die Antworten Y und N vertauscht.

   $\Longrightarrow$ M' entscheidet $\overline{L}$. ∎

Turing-Maschinen-Flußdiagramme

Übersichtliche Darstellung von TM statt Angabe der δ-Übergänge und Zustände.

$\Longrightarrow$ Nur Beschreibung der durchzuführenden Schritte und deren Ausführungsreihenfolge.

Konkrete Elemente:

- L (bzw. R) beschreibt eine TM die nach Start ein Feld nach links (bzw. rechts) geht und danach hält.
- a (für $a \in \Sigma$) beschreibt eine TM die den Buchstaben a auf das aktuelle Bandfeld druckt.
- Direkt aufeinanderfolgende Schritte werden direkt nebeneinander notiert oder mittels Pfeil verbunden:
  – $M_1 \longrightarrow M_2$ bzw. $M_1 M_2$ bezeichnet TM die zuerst wie M_1 arbeitet und dann, falls M_1 hält, wie M_2 weiterarbeitet.

- Der Startschritt wird mit $>$ bezeichnet.
- $M_1 \xrightarrow{a} M_2$ heißt, daß M_2 nur dann ausgeführt wird, wenn nach Beendigung von M_1 der aktuelle Bandbuchstabe a ist.

  Analog für $n > 1$ Maschinen:

$$\begin{array}{ll} & M_1 \\ & \uparrow a_1 \\ > & M_0 \xrightarrow{a_2} M_2 \\ & \downarrow a_n \qquad \ddots \\ & M_n \end{array}$$

  D.h. obige Maschine arbeitet zuerst wie M_0 und dann, falls M_0 mit a_i hält, wie M_i weiterarbeitet.

- σ steht für einen beliebigen Buchstaben aus Σ.

  Z.B., für $\Sigma = \{a_1, \ldots, a_n\}$, bezeichnet

$$M \xrightarrow{\sigma} \ldots \sigma$$

  folgende Maschine:

$$\begin{array}{ll} & \ldots a_1 \\ & \uparrow a_1 \\ > & M_0 \xrightarrow{a_2} \ldots a_2 \\ & \downarrow a_n \qquad \ddots \\ & \ldots a_n \end{array}$$

- Weitere Abkürzungen:
  - $\xrightarrow{\sigma \neq a}$ für $\sigma \in \Sigma \setminus \{a\}$;
  - $M_1 \xrightarrow{a,b} M_2$ bedeutet, daß nach Ausührung von M_1 sowohl für a als auch für b nach M_2 verzweigt werden soll.

Beispiel: Betrachte TM M^+ die zwei natürliche Zahlen in Unärdarstellung addieren soll, d.h. M^+ rechnet

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

Methode: M^+ löscht letzten Strich von $|^m$ und schreibt ihn zwischen $|^n$ und $|^m$.

$\Longrightarrow M^+ = (\{s, q_1, q_2, q_3, q_4\}, \{|, \#\}, \delta, s)$ mit

$$\begin{array}{ll} \delta(s, \#) = (q_1, L) & \delta(q_2, \#) = (q_3, |) \\ \delta(q_1, \#) = (h, \#) & \delta(q_2, |) = (q_2, L) \\ \delta(q_1, |) = (q_2, L) & \delta(q_3, |) = (q_3, R) \end{array}$$

$$\delta(q_3, \#) = (q_4, L)$$
$$\delta(q_4, |) = (h, \#)$$

$\Longrightarrow$ Flußdiagramm:

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

Beispiel: Die TM $R_\#$ (bzw. $L_\#$) geht mindestens einmal nach rechts (bzw. links) und solange nach rechts (bzw. links) bis sie ein $\#$ liest.

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

Beispiel: Die TM *Copy* rechnet

$$s, \#w\underline{\#} \vdash^*_{Copy} h, \#w\#w\underline{\#}.$$

Arbeitsweise:

- *Copy* geht das Eingabewort von links nach rechts durch;
- merkt sich jeweils ein Zeichen σ von w;
- markiert die aktuelle Position (überschreiben von σ mit $\#$) und kopiert σ.

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

Variationen von Turing-Maschinen

Bisher:

- *deterministische* Turing-Maschinen
- einseitig unbeschränktes Band (“Halbband”)

$\Longrightarrow$ *Standard-Turing-Maschinen* (“Standard-TM”)

Erweiterungen:

- Turing-Maschinen mit **zweiseitig unbeschränktem** Band

  $\Longrightarrow$ solche Maschinen können nicht hängen!
- Turing-Maschinen mit mehreren Bändern.
- Nicht-deterministische Turing-Maschinen.

Wir werden zeigen: diese Erweiterungen ändern nicht die Klasse der berechenbaren Probleme.

Bemerkung:

Wir wissen: Für Standard-Turing-Maschinen gibt es zwei Möglichkeiten, nicht zu halten:

- sie bleiben hängen; oder
- sie laufen unendlich.

ObdA können wir annehmen, daß Standard-Turing-Maschinen **nie hängen**.

Beweis dazu: Sei M eine Standard-TM mit Input w. Wir konstruieren M' wie folgt:

- M' stellt zunächst fest, wo das Bandende ist, nämlich ein Zeichen links vom Eingabewort.
- M' verschiebt dann das Eingabewort um ein Zeichen nach rechts und druckt zwei Stellen links vom Eingabewort ein Sonderzeichen, α.

D.h. M' rechnet zunächst $s, \#w\underline{\#} \vdash^*_{M'} s', \alpha\#w\underline{\#}$.

Ab dann verhält sich M' wie M, nur wenn sie α erreicht, bleibt M' stehen und druckt immer wieder α neu.

$\Longrightarrow$ Es gilt: M' hält für $w \Longleftrightarrow M$ hält für w, und M' bleibt nie hängen.

Definition:

Eine **Turing-Maschine mit zweiseitig unbeschränktem Band (zw-TM)** ist eine TM mit folgendem Konfigurationsbegriff:

Eine **Konfiguration** einer zw-TM $M = (K, \Sigma, \delta, s)$ ist ein Wort der Form $C = q, w\underline{a}u$, mit

- $q \in K \cup \{h\}$ ist aktueller Zustand;
- $w \in (\Sigma \setminus \{\#\})\Sigma^* \cup \{\varepsilon\}$ ist Bandinhalt links vom Kopf;
- $a \in \Sigma$ ist Zeichen unter dem Kopf;
- $u \in \Sigma^*(\Sigma \setminus \{\#\}) \cup \{\varepsilon\}$ ist Bandinhalt rechts vom Kopf.

Bemerkungen:

1. Beachte: Bei Standard-TM war $w \in \Sigma^*$ definiert.

   $\Longrightarrow$ Jetzt enthält w alle Zeichen bis zum letzten nicht-Blank links vom Schreib-/Lesekopf.

2. $w = \varepsilon$ bzw. $u = \varepsilon$ bedeutet, daß links bzw. rechts vom Kopf nur noch Blanks stehen.

$C_2 = q_2, w_2\underline{a_2}u_2$ ist **Nachfolgekonfiguration** von $C_1 = q_1, w_1\underline{a_1}u_1$, symbolisch $C_1 \vdash_M C_2$, falls es einen Übergang $\delta(q_1, a_1) = (q_2, b)$ gibt s.d.:

1. Falls $b \in \Sigma$, dann $w_1 = w_2$, $u_1 = u_2$, $a_2 = b$;
2. falls $b = L$, dann

   $$u_2 = \begin{cases} \varepsilon & \text{falls } a_1 = \# \text{ und } u_1 = \varepsilon; \\ a_1 u_1 & \text{sonst;} \end{cases}$$

   und $w_2 = \varepsilon$, $a_2 = \#$ falls $w_1 = \varepsilon$, sonst $w_1 = w_2 a_2$;
3. falls $b = R$, dann

   $$w_2 = \begin{cases} \varepsilon & \text{falls } a_1 = \# \text{ und } w_1 = \varepsilon; \\ w_1 a_1 & \text{sonst;} \end{cases}$$

   und $u_2 = \varepsilon$, $a_2 = \#$ falls $u_1 = \varepsilon$, sonst $u_1 = a_2 u_2$.

Begriffe *TM-berechenbar*, *entscheiden*, *akzeptieren* und *aufzählen* analog wie für Standard-TM definiert.

Satz: Zu jeder zw-TM M, die eine Funktion f berechnet oder eine Sprache L akzeptiert, existiert eine Standard-TM M', die ebenfalls f berechnet oder L akzeptiert.

Beweis:

Sei $w = a_1 a_2 \ldots a_n$ Input für $M = (K, \Sigma, \delta, s)$.
$\Longrightarrow$ Das beidseitig unendliche Band sieht zu Beginn der Rechnung folgendermaßen aus:

$$\ldots \#\#\# a_1 a_2 \ldots a_n \#\#\# \ldots$$

Idee: wir simulieren M mittels M' derart, daß

- das beidseitig unendliche Band von M "gekippt" wird und
- auf einseitig unendlichem Band mit **zwei Spuren** dargestellt wird:

Spur 1: $\#\#\ldots\#\#$
Spur 2: $\#a_1\ldots a_n\#$ $\ldots$

Formal: $M' = (K', \Sigma', \delta', s')$, wo

- $\Sigma' \supseteq \Sigma \times \Sigma$;
- $K' \supseteq K \times \{1, 2\}$.

(q, i) bedeutet, daß die simulierte Maschine M in Zustand q ist und M' auf Spur i arbeitet.

Arbeitsweise von M':

1. M' legt zunächst eine zweite Spur an, d.h.

   $$s, \#a_1 \ldots a_n\underline{\#} \vdash^*_{M'} p, \$ {\# \atop \#} {\# \atop a_1} \cdots {\# \atop a_n} {\# \atop \underline{\#}} \# \ldots$$

   ("\$" markiert das Ende des Halbbandes.)

2. Anschließend simuliert M' die Maschine M. Dabei soll gelten:

   $$s, \# \vdots \#w\underline{\#} \vdash^*_M q, u_1 b \vdots a u_2 \iff$$
   $$p, \$ {\# \cdots \# \atop \# w \underline{\#}} \# \vdash^*_{M'} p', \$ {b u_1^R \atop a u_2} {\# \atop \#} \cdots {\# \atop \#} \#$$

   - "$\vdots$" markiert die Bandstelle wo das Band "umgeklappt" wird;
   - u_i^R bezeichnet das Reverse von u_i.

3. Wenn M hält muss M' ihr Band auf eine Spur heruntertransformieren und ebenfalls halten.

Einige Details:

Zu Schritt 2:

- Wenn M' das Zeichen \$ erreicht, wechselt sie die Spur.
- Wenn M rechts (bzw. links) geht, dann geht M'
  - rechts (bzw. links) auf Spur 2 und
  - links (bzw. rechts) auf Spur 1.
- Wenn M' ein $\#$ erreicht, schreibt sie ${\# \atop \#}$.

Für δ muss z.B. gelten: falls $\delta_M(q, a) = (q', L)$, dann

- $\delta_{M'}((q, 2), {x \atop a}) = ((q', 2), L)$, für alle x;
- $\delta_{M'}((q, 1), {a \atop x}) = ((q', 1), R)$, für alle x.

Weiters gilt immer:

- $\delta_{M'}((q, 1), \$) = ((q, 2), R)$ und
  $\delta_{M'}((q, 2), \$) = ((q, 1), R)$
  (Spurwechsel beim Überschreiten von \$);
- $\delta_{M'}((q, i), \#) = ((q, i), {\# \atop \#})$.
- etc.

Zu Schritt 3:

Wenn M mit $h, \#u\underline{\#}$ hält, dann erreicht M' eine der folgenden Konfigurationen:

(a) $(h, 1), \$ {\# \atop \#} \cdots {\overline{\#} \atop \#} {u^R \atop \# \cdots \#} {\# \atop \#} \cdots {\# \atop \#}$;

(b) $(h, 2), \$ {\# \atop \#} \cdots {\# \atop \#} {\# \cdots \# \atop u} {\# \atop \underline{\#}} \cdots {\# \atop \#}$;

(c) $(h, 2), \$ {u_1^R \atop u_2 \underline{\#}} {\# \atop \#} \cdots {\# \atop \#}$ mit $u = u_1 u_2$.

M' muß die zwei Spuren zu einer Spur heruntertransformieren, um $h, \#u\underline{\#}$ zu erreichen.

Dies geht folgendermaßen:

- M' macht zunächst alle ${\# \atop \#}$ rechts vom beschriebenen Band zu $\#$. Im Fall (a) und (b) löscht M' auch die ${\# \atop \#}$ links von u^R bzw. u.
- Im Fall (b) ersetzt M' ${\# \cdots \# \atop u}$ dann durch u und im Fall (c) schiebt M' die untere Spur nach rechts bis $q, \$ {u_1^R \atop \# \cdots \#} u_2 \underline{\#}$ erreicht wird.
- Für (a) und (c) wird u^R bzw. u_1^R auf eine Spur transformiert und zugleich invertiert.
- Abschließend wird \$ gelöscht.

$\Longrightarrow$ M' simuliert die Arbeitsweise von M. ■

Definition: Eine **Turing-Maschine mit k Halbbändern (k-TM)** ist eine Turing-Maschine der Form $M = (K, \Sigma_1, \ldots, \Sigma_k, \delta, s)$, mit Übergangsfunktion

$$\delta : K \times \Sigma_1 \times \cdots \times \Sigma_k \to (K \cup \{h\}) \times (\Sigma_1 \cup \{L, R\} \times \cdots \times (\Sigma_k \cup \{L, R\})$$

Eine **Konfiguration** einer k-TM hat die Form

$$C = q, w_1 \underline{a_1} u_1, \ldots, w_k \underline{a_k} u_k$$

Beachte:

- Die Köpfe einer k-TM können sich **unabhängig** voneinander bewegen! (Sonst hätten wir eine 1-Band-TM mit k Spuren.)

Festlegung:

- Bei der Berechnung von Funktionen sollen Eingabe- und Ausgabewörter stets am ersten Band stehen.

Satz: Zu jeder k-TM, die eine Funktion f oder eine Sprache L akzeptiert, existiert eine Standard-TM, die ebenfalls f berechnet oder L akzeptiert.

Beweis: Sei M eine k-TM. Wir simulieren M mit einer TM M' mit $2k$ Spuren:

- die Spuren mit ungerader Nummer enthalten die Inhalte der k Bänder von M;
- die Spuren mit gerader Nummer markieren die Positionen der k Köpfe von M:
  - für jedes i gilt ($1 \leq i \leq n$):
    die $2i$-te Spur enthält an genau einer Stelle ein $*$, nämlich da, wo M gerade seinen i-ten Kopf hätte, ansonsten lauter Blanks.

M' arbeitet wie folgt:

- Zunächst kodiert M' die Eingabe:
  - die linkeste Bandposition wird mit α markiert; die rechteste mit ω.
  - Zwischen α und ω werden $2k$ Spuren angelegt.
  - Die oberste Spur enthält die Eingabe; die restlichen Spuren mit ungerader Nummer sind leer; die geraden Spuren enthalten die Kopfpositions-Markierungen $*$.
- M' simuliert einen Schritt von M in zwei Durchgängen:
  - Zunächst läuft M' von ω nach α und liest für jede Spur das Zeichen a_i unter dem Kopf von M (= Stelle der $(2i-1)$-ten Spur wo die $2i$-te Spur ein $*$ hat).
  - Anschliessend läuft M' von α nach ω und führt für jedes simulierte Band lokal $\delta_M(q, a_1, \ldots, a_k)$ aus. Wenn nötig wird ω nach rechts verschoben.
- Am Ende der Rechnung wird Bandteil zwischen α und ω auf eine Spur transformiert. ■

Definition: Eine **Nicht-deterministische Turing-Maschine (NTM)** ist ein Tupel $M = (K, \Sigma, \Delta, s)$, wo

- K, Σ, s sind wie bei determinierten TM definiert;
- $\Delta \subseteq (K \times \Sigma) \times ((K \cup \{h\}) \times (\Sigma \cup \{L, R\}))$ ist die **Übergangsrelation**.

Wir schreiben Δ in Infix-Form, d.h. $(q, a)\Delta(q', b)$ bedeutet $((q, a), (q', b)) \in \Delta$.

$\Longrightarrow$ Wir können Δ auch als mehrwertige Funktion auffassen:

$$\Delta : K \times \Sigma \to 2^{((K \cup \{h\}) \times (\Sigma \cup \{L, R\}))}$$

Damit Schreibweise $(q', b) \in \Delta(q, a)$ möglich.

Begriff der **Konfiguration** wie bei Standard-Turing-Maschine definiert.