1 Varianten von Turingmaschinen

Es gibt weitere Definitionen für Turingmaschinen. Diese haben sich aber alle als äquivalent herausgestellt. Ein wiederkehrendes Element der Vorlesung: Äquivalenz von Formalismen zeigen (z.B. durch konstruktive Beweise).
Bsp.: DEA = NEA = ε-NEA = RegEx = RLIN, PDA = CFG, 2-PDA = QA = TM = CH0-G

(a) TMs mit k Bändern: Ein- und Ausgabe auf Band 1, k Lese-/Schreibköpfe, ein paralleler Schritt für alle Köpfe

Simulation:

- Schreibe alle Bandinhalte durch Trennsymbole getrennt hintereinander
- Markiere Kopfpositionen durch neue Symbole
- laufe von links nach rechts und sammle die Symbole ein (nur endlich viele Kombinationen möglich)
- laufe über das Band und führe nacheinander alle Änderungen durch (endlich viele)

Sollte mal ein Band „voll werden“, so unterbrich die aktuelle Arbeit, verschiebe alle Symbole um ein Feld, und fahre dann fort. Am Ende muss man das Band entsprechend leeren.

(b) TMs mit k Köpfen

(c) TMs mit zweidimensionalem Band

(d) TMs mit einseitig unendlichem Band

1.1 TMs mit Endzuständen

nicht äquivalent, da nur Sprachakzeptor (also keine allgemeine Ausgabe mehr)
Def. 1.12

(1) Eine Konfiguration $K = uqv$ von τ heißt *akzeptierend*, falls $q \in F$ ist.

(2) Sei $v \in \Sigma^*$. Die TM τ *akzeptiert* v, falls es eine akzeptierende Endkonfiguration K mit $\alpha(v) \vdash_\tau^* K$ gibt.

(3) Die von τ *akzeptierte Sprache* ist

$$L(\tau) = \{v \in \Sigma^* \mid \tau \text{ akzeptiert } v\}.$$

Eine Sprache $L \subseteq \Sigma^*$ heißt *Turing-akzeptierbar*, falls es eine TM τ mit Endzuständen gibt, für die $L = L(\tau)$ gilt.

(4) Mit $\mathcal{T}$ bezeichnen wir in diesem Zusammenhang die von Turingmaschinen akzeptierten Sprachen.

Satz 1.13 Sei $L \subseteq \Sigma^*$ und $\overline{L} = \Sigma^* \setminus L$.
L und $\overline{L}$ sind Turing-akzeptierbar $\Leftrightarrow$ L ist Turing-entscheidbar.

Bew.: „$\Leftarrow$“: Sei L durch τ entscheidbar. Durch Hinzufügen von Endzustandsübergängen erhält man akzeptierende TM für L und $\overline{L}$.
„$\Rightarrow$“: L werde durch τ_1 und $\overline{L}$ durch τ_2 akzeptiert. Konstruiere nun τ mit 2 Bändern so, dass *ein Schritt von τ_1 auf Band 1 und gleichzeitig ein Schritt von τ_2 auf Band 2* ausgeführt wird. Akzeptiert τ_1 das vorgegebene Wort, so gibt τ den Wert 1 aus. Sonst akzeptiert τ_2 das vorgegebene Wort, und τ gibt den Wert 0 aus. Somit entscheidet τ die Sprache L.

Bem.: Wenn nur L Turing-akzeptierbar ist, so folgt noch nicht, dass L auch Turing-entscheidbar ist. Die akzeptierende TM könnte nämlich für Wörter aus $\overline{L}$ *nicht anhalten.*

1.2 Nichtdeterministische TMs

Nur sinnvoll für Sprachakzeptanz; später und im nächsten Kapitel werden wir den Nichtdeterminismus als Hilfsmittel zum Raten verwenden.
Idee: In jedem Schritt gibt es nur endlich viele Möglichkeiten. Laufe in Breitensuche durch den Suchbaum und zähle alle Möglichkeiten auf. Simuliere dann nacheinander alle Möglichkeiten systematisch.
Sei r der höchste Grad an Nichtdeterminismus. Wir konstruieren eine TM mit 3 Bändern, die auf Band 1 das Eingabewort v unverändert speichert. Auf Band 2 werden alle Wörter über $\Sigma = \{1, \ldots, r\}$ der Länge nach und bei gleicher Länge in lexikografischer Reihenfolge erzeugt, d.h. ε, 1, 2, …, r, $(1,1)$, $(1,2)$, $(1,r)$, $(2,1)$, …, (r,r), … . Auf Band 3 wird jeweils die Simulation der aktuellen Auswahl durchgeführt.
Wenn es eine gültige Konfigurationenfolge gibt, die in einem Endzustand hält, wird das Wort irgendwann akzeptiert. Wenn nicht, gibt es zwei Möglichkeiten:

- Der Suchbaum ist endlich, d.h. jede mögliche Reihenfolge führt irgendwann zur Terminierung. Dann terminiert der Algorithmus.
- Es gibt nicht-terminierende Läufe. Dann terminiert der Algorithmus nicht, sondern zählt immer längere Reihenfolgen auf.

Bem.: Es werden u.U. Reihenfolgen aufgezählt, die nicht durchführbar sind. Das kann aber erkannt werden; in diesem Fall wird die Simulation einfach übersprungen. Der Suchaufwand ist exponentiell (Suchbaum mit Breite r).

2 Grammatiken

Def. 2.1 $CH0$-Grammatik in Kurzform:
Für Regeln $u \to v$ in P gilt: $u = u_1 A u_2$ für $u_1 u_2 \in (T \cup N)^*, A \in N$

Def. 2.2 Eine Sprache $L \subseteq T^*$ heißt Chomsky-0-Sprache, wenn es eine $CH0$-Grammatik G mit $L(G) = L$ gibt.

Satz 2.3 Jede Turing-akzeptierbare Sprache $L \subseteq T^*$ ist eine Chomsky-0-Sprache ($\mathcal{T} \subseteq CH0$).

Beweis: Gegeben TM τ mit o.B.d.A. $F = \{q_e\}$ und in jeder Endkonfiguration ist der Lesekopf auf dem ersten Zeichen des Bandes. Wir konstruieren eine Grammatik G in vier Schritten: (1) $G_1 = (N_1, T, P_1, S)$: Doppel-Anfangskonfiguration erzeugen: Für alle $w \in T^*$ gilt $S \vdash^* w\#\alpha(w)\$$
Wähle $N_1 = \{S, \#, \$, q_0, \sqcup, A, B\} \cup \{C_a \mid a \in T\}$ und P_1 wie folgt:

$$
\begin{array}{lrll}
P_1 = & \{ & S \to \#q_0\sqcup\$\} & \text{Start} \\
\cup & \{ & S \to aA\#C_a\$ \mid a \in T\} & \text{setze erstes Symbol} \\
\cup & \{ & C_a b \to bC_a \mid a, b \in T\} & \text{bewege } C_a \text{ nach rechts} \\
\cup & \{ & C_a\$ \to Ba\$ \mid a \in T\} & \text{ersetze } C_a \text{ ganz rechts} \\
\cup & \{ & bB \to Bb \mid b \in T\} & \text{bewege } B \text{ nach links} \\
\cup & \{ & A\#B \to \#q_0\} & \text{terminiere} \\
\cup & \{ & A\#B \to aA\#C_a \mid a \in T\} & \text{füge ein neues Symbol ein}
\end{array}
$$

(2) $G_2 = (N_2, T, P_2, S)$: Transitionsrelation $\vdash_\tau$ simulieren: $N_2 = \{S, \#, \$\} \cup Q \cup \Gamma$, sodass für alle $w \in T^*, v \in \Gamma^*$ gilt:

$$\alpha(w) \vdash_\tau{}^* q_e v \quad \Longleftrightarrow \quad w\#\alpha(w)\$ \vdash_{G_2}^* w\#q_e v\$.$$

Wir benutzen die Übergangsfunktion δ:

$$
\begin{aligned}
P_2 = \quad & \{qa \to q'a' && | q, q' \in Q \text{ und } a, a' \in \Gamma \text{ und } \delta(q,a) = (q', a', S) && \} \\
\cup\ & \{qab \to a'q'b && | q, q' \in Q \text{ und } a, a', b \in \Gamma \text{ und } \delta(q,a) = (q', a', R) && \} \\
\cup\ & \{ \underbrace{qa\$}_{\text{TM steht vor rechtem Rand}} \to a'q'\sqcup\$ && | q, q' \in Q \text{ und } a, a' \in \Gamma \text{ nd } \delta(q,a) = (q', a', R) && \} \\
\cup\ & \{bqa \to q'ba' && | q, q' \in Q \text{ und } a, a', b \in \Gamma \text{ und } \delta(q,a) = (q', a', L) && \} \\
\cup\ & \{ \underbrace{\#qa}_{\text{TM steht vor linkem Rand}} \to \#q'\sqcup a' && | q, q' \in Q \text{ und } a, a' \in \Gamma \text{ und } \delta(q,a) = (q', a', L) && \}
\end{aligned}
$$

(3) $G_3 = (N_3, T, P_3, S)$: Endkonfiguration löschen: $N_3 = \{S, \#, \$, q_e, \sqcup, D\}$, sodass für alle $w \in T^*, v \in \Gamma^*$ gilt:

$$w\#q_e v\$ \vdash_{G_3}^* w.$$

Wähle P_3 wie folgt:

$$
\begin{aligned}
P_3 = \{ \quad & \#q_e \to D, \\
& Da \to D \quad \text{für alle } a \in \Gamma, \\
& D\$ \to \varepsilon \quad \}
\end{aligned}
$$

(4) Abschluss: Definiere $G = (N, T, P, S)$ wie folgt:

$$
\begin{aligned}
& N = N_1 \cup N_2 \cup N_3, \\
& P = P_1 \dot\cup P_2 \dot\cup P_3 \qquad \text{(disjunkte Vereinigung)}.
\end{aligned}
$$

Dann gilt für alle $w \in T^*, v \in \Gamma^*$:

$$
\begin{aligned}
\alpha(w) \vdash_\tau{}^* q_e v \quad \Longleftrightarrow \quad & S \vdash_G^* w\#\alpha(w)\$ && \text{(Regeln } P_1\text{)} \\
& \vdash_G^* w\#q_e v\$ && \text{(Regeln } P_2\text{)} \\
& \vdash_G^* w && \text{(Regeln } P_3\text{)}
\end{aligned}
$$

Für „$\Leftarrow$“ beachte man, dass auf ein vorgegebenes Wort über $N \cup T$ höchstens aus einer der drei Regelmengen P_1, P_2 oder P_3 eine Regel angewandt werden kann.
Insgesamt erhalten wir $L(G) = L$. $\square$

Korollar 2.4 Sei die Funktion $f : T^* \xrightarrow{part} T^*$ Turing-berechenbar. Dann ist der *Graph von* f, d.h. die Menge

$$L = \{w\#f(w) \mid w \in T^* \text{ und } f(w) \text{ ist definiert}\},$$

eine $CH0$-Sprache.

Bew.: Wenn f von einer TM τ berechnet wird, so gibt es eine 2-Band TM τ', die L akzeptiert. Die TM τ' arbeitet wie folgt:

(1) τ' belässt ein vorgegebenes Eingabewort der Form $w\#v$ unverändert auf dem 1. Band.

(2) τ' kopiert den Anteil w auf das anfangs leere 2. Band und simuliert dann τ auf diesem Band.

(3) Falls τ hält, wird das Ergebnis $f(w)$ der Endkonfiguration mit dem Anteil v des 1. Bandes verglichen. Falls $f(w) = v$ gilt, akzeptiert τ' die Eingabe $w\#v$ auf dem 1. Band. Anderenfalls akzeptiert τ' die Eingabe $w\#v$ auf dem 1. Band nicht.

Damit ist L Turing-akzeptierbar, also nach Satz 2.3 eine $CH0$-Sprache. □

Satz 2.5 Jede $CH0$-Sprache $L \subseteq T^*$ ist Turing-akzeptierbar ($CH0 \subseteq \mathcal{T}$).

Bew.: L werde von einer $CH0$-Grammatik G erzeugt, also $L(G) = L$.
Wir konstruieren eine nichtdeterministische TM τ mit 2 Bändern, die L akzeptiert. Die TM τ arbeitet wie folgt:

(1) τ belässt ein vorgegebenes Eingabewort $w \in T^*$ unverändert auf dem 1. Band.

(2) Auf dem anfangs leeren 2. Band erzeugt τ schrittweise Wörter über $N \cup T$ gemäß den Regeln aus P, beginnend mit dem Startwort S. In jedem Schritt wählt τ nichtdeterministisch ein Teilwort u aus dem zuletzt erzeugten Wort und eine Regel $u \to v$ aus P und ersetzt dann u durch v. Entsteht in irgendeinem Schritt das Eingabewort w, so wird w akzeptiert.

Damit gilt: τ akzeptiert L. □