Akzeptierende TURING-Maschine

Definition:

Eine akzeptierende TURING-Maschine M ist ein Sechstupel

$$M = (X, Z, z_0, Q, \delta, F)\,,$$

wobei (X, Z, z_0, Q, δ) eine TURING-Maschine ist und $F \subseteq Q$ gilt.
Die von M akzeptierte Sprache $T(M)$ wird durch

$$T(M) = \{w \mid w \in X^*,\ (\lambda, z_0, w) \models^* (v_1, q, v_2) \text{ für ein } q \in F\}$$

definiert.

Akzeptierende TURING-Maschine – Beispiel I

$$M_1' = (\{a, b\}, \{z_0, z_a, z_b, q_a, q_b\}, z_0, \{q_a, q_b\}, \delta', \{q_a\})$$

δ	z_0	z_a	z_b
$*$	$(q, *, N)$	(q_a, a, N)	(q_b, b, N)
a	$(z_a, *, R)$	(z_a, a, R)	(z_b, a, R)
b	$(z_b, *, R)$	(z_a, b, R)	(z_b, b, R)

$$T(M_1') = \{aw \mid w \in \{a, b\}^*\}\,.$$

Akzeptierende TURING-Maschine – Beispiel II

$$M_2' = (\{a, b\}, \{z_0, z_1, q\}, z_0, \{q\}, \delta, \{q\})$$

δ	z_0	z_1
$*$	$(z_0, *, N)$	$(q, *, N)$
a	(z_1, a, R)	(z_0, a, R)
b	(z_1, b, R)	(z_0, b, R)

$$T(M_2') = \{w \mid w \in \{a, b\}^*, |w| \; ungerade\}.$$

Eine Normalform für akzeptierende TURING-Maschinen

Lemma:
Zu jeder akzeptierenden TURING-Maschine M gibt es eine akzeptierende TURING-Maschine M', deren Menge der Stopzustände mit der Menge der akzeptierenden Zustände übereinstimmt und für die $T(M) = T(M')$ gilt. Dabei kann die Menge der Stopzustände von M' einelementig gewählt werden.

Satz:
Eine Sprache wird genau dann von einer TURING-Maschine akzeptiert, wenn sie Definitionsbereich einer TURING-berechenbaren Funktion ist.

Satz:
Eine Sprache wird genau dann von einer TURING-Maschine akzeptiert, wenn sie Wertevorrat einer TURING-berechenbaren Funktion ist.

Rekursive Sprachen

Definition: Eine Sprache $L \subseteq X^*$ heißt rekursiv, falls es eine akzeptierende TURING-Maschine $M = (X, Z, z_0, Q, \delta, F)$ gibt, die auf jeder Eingabe stoppt und L akzeptiert.

Satz: Eine Sprache $L \subseteq X^*$ ist genau dann rekursiv, wenn sowohl L als auch $X^* \setminus L$ von TURING-Maschinen akzeptiert werden

Satz: Für eine rekursive Menge L ist die charakteristische Funktion

$$\varphi_L(x) = \begin{cases} 0 & x \notin L \\ 1 & x \in L \end{cases}$$

von L algorithmisch berechenbar.

Satz: Die Menge der rekursiven Sprachen ist echt in der Menge der von TURING-Maschinen akzeptierbarten Sprachen enthalten.

TURING-Maschinen versus Regelgrammatiken

Lemma:
Zu jeder TURING-Maschine M gibt es eine Regelgrammatik G mit $L(G) = T(M)$.

Lemma:
Zu jeder Regelgrammatik G gibt es eine nichtdeterministische TURING-Maschine M mit $T(M) = L(G)$.

Satz:
Die folgenden Aussagen sind äquivalent:

i) L wird von einer Regelgrammatik erzeugt.
ii) L wird von einer deterministischen TURING-Maschine akzeptiert.
iii) L wird von einer nichtdeterministischen TURING-Maschine akzeptiert.

Nichtdeterministische TURING-Maschinen I

Definition: Eine nichtdeterministische TURING-Maschine M ist ein Quintupel

$$M = (X, Z, z_0, Q, \tau, F),$$

wobei X, Z, z_0, Q und F wie bei einer (akzeptierenden deterministischen) TURING-Maschine definiert sind und τ eine Funktion

$$\tau : (Z \setminus Q) \times (X \cup \{*\}) \to 2^{Z \times (X \cup \{*\}) \times \{R,N,L\}}$$

ist.

Nichtdeterministische TURING-Maschinen II

$(w_1, z, xw_2) \models (w_1', z', w_2')$, falls $(z', x', r) \in \tau(z, x)$ existiert und
$(w_1, z, w_2) \models (w_1', z', w_2')$ bei einer (deterministischen) TURING-Maschine
mit $(z', x', r) = \delta(z, x)$ gilt

Definition: Es sei $M = (X, Z, z_0, Q, \delta)$ eine nichtdeterministische TURING-Maschine. Die von M akzeptierte Sprache $T(M)$ definieren wir durch

$$T(M) = \{w \mid w \in X^*,\ (\lambda, z_0, w) \models^* (v_1, q, v_2) \text{ für ein } q \in F\}.$$

Nichtdeterministische TURING-Maschinen – Beispiel

$$M = (\{a, b\}, \{z_0, z_{0,2}z_{1,2}, z_2, z_2', z_2'', q\}, z_0, \{q\}, \tau, \{q\})$$

$$\begin{aligned}
\tau(z_0, x) &= \{(z_2, x, N), (z_3, x, N)\} \quad \text{für } x \in \{a, b\}, \\
\tau(z_i, a) &= \{(z_i', a, R)\} \quad \text{für } i \in \{2, 3\}, \\
\tau(z_i', a) &= \{(z_i'', a, R)\} \quad \text{für } i \in \{2, 3\}, \\
\tau(z_i'', a) &= \{(z_i'', a, R)\}, \\
\tau(z_i, *) &= \{(z_i, *, N)\} \quad \text{für } i \in \{2, 3\}, \\
\tau(z_i', *) &= \{(q, *, N)\} \quad \text{für } i \in \{2, 3\}, \\
\tau(z_i'', *) &= \{(z_{0,i}, *, L)\} \quad \text{für } i \in \{2, 3\}
\end{aligned}$$

Nichtdet. TURING-Maschinen – Beispiel - Fortsetzung

$$
\begin{aligned}
\tau(z_{0,2}, a) &= \{(z_{1,2}, b, L)\}, \quad \tau(z_{1,2}, a) = \{(z_{0,2}, a, L)\},\\
\tau(z_{j,2}, b) &= \{(z_{i,2}, b, L)\} \quad \text{für } j \in \{0, 1\},\\
\tau(z_{0,3}, a) &= \{(z_{1,2}, b, L)\}, \quad \tau(z_{1,3}, a) = \{(z_{2,3}, b, L)\},\\
\tau(z_{2,3}, a) &= \{(z_{0,3}, a, L)\},\\
\tau(z_{j,3}, b) &= \{(z_{j,3}, b, L)\} \quad \text{für } j \in \{0, 1, 2\},\\
\tau(z_{0,i}, *) &= \{(z_i, *, R)\} \quad \text{für } i \in \{2, 3\},\\
\tau(z_{j,i}, *) &= \{(z_{j,i}, *, N)\} \quad \text{für } j \in \{1, 2\} \text{ , } i \in \{2, 3\}
\end{aligned}
$$

$$T(M) = \{a^m \mid m = 2^n \text{ oder } m = 3^n \text{ für ein } n \geq 0\}$$

Linear beschränkter Automat

Definition:
Ein linear beschränkter Automat ist eine nichtdeterministische TURING-Maschine $M = (X, Z, z_0, Q, \delta, F)$, deren Kopf sich während der Abarbeitung der Eingabe $w \in X^*$ höchstens über $|w| + 2$ verschiedenen Zellen befindet.

Satz:
Eine Sprache ist genau dann kontextabhängig, wenn sie von einem linear beschränkten Automaten akzeptiert werden kann.

Endlicher Automat – Definition

Definition: i) Ein endlicher Automat ist ein Quintupel

$$\mathcal{A} = (X, Z, z_0, F, \delta),$$

wobei
- X und Z Alphabete sind,
- $z_0 \in Z$ und $F \subseteq Z$ gelten,
- δ eine Funktion von $Z \times X$ in Z ist.

ii) Die Erweiterung δ^* von δ auf $Z \times X^*$ definieren wir durch

$\delta^*(z, \lambda) = z$ und $\delta^*(z, wx) = \delta(\delta^*(z, w), x)$ für $w \in X^*,\ x \in X$.

iii) Die durch $\mathcal{A}$ akzeptierte Wortmenge definieren wir durch

$$T(\mathcal{A}) = \{w \mid w \in X^*, \delta^*(z_0, w) \in F\}.$$

Endlicher Automat – Beispiel

$\mathcal{A} = (X, Z, z_0, F, \delta)$ mit $X = \{a, b, c\}$

$Z = \{z_0, z_1, z_2, z_3\}$,

$F = \{z_2\}$,

$$\delta(z,x) = \begin{cases} z_1 & \text{für } z = z_0, x = a \\ z_2 & \text{für } z = z_1, x = a \\ z_0 & \text{für } z \in \{z_0, z_2\}, x = c \\ z_3 & \text{sonst} \end{cases}$$

$$T(\mathcal{A}) = \{c^{n_1}aac^{n_1}aac^{n_2}aa \ldots c^{n_k}aa \mid k \geq 1, n_1 \geq 0, n_j \geq 1, 2 \leq j \leq k\}$$

Nichtdeterministischer endlicher Automat

Definition:

i) Ein nichtdeterministischer endlicher Automat ist ein Quintupel $\mathcal{A} = (X, Z, z_0, F, \delta)$, wobei für X, Z, z_0, F die Bedingungen wie beim endlichen Automaten gelten und δ eine Funktion von $Z \times X$ in die Menge der Teilmengen von Z ist.

ii) Wir definieren $\delta^*(z, \lambda) = \{z\}$ für $z \in Z$, und für $w \in X^*$, $x \in X$ und $z \in Z$ gelte $z' \in \delta^*(z, wx)$ genau dann, wenn es einen Zustand $z'' \in \delta^*(z, w)$ mit $z' \in \delta(z'', x)$ gibt.

iii) Die von $\mathcal{A}$ akzeptierte Wortmenge ist durch

$$T(\mathcal{A}) = \{w \mid \delta^*(z_0, w) \cap F \neq \emptyset\}$$

definiert.

Endliche Automaten versus reguläre Sprachen

Satz:
Die beiden folgenden Aussagen sind für eine Sprache L äquivalent:
i) L wird von einem (deterministischen) endlichen Automaten akzeptiert.
ii) L wird von einem nichtdeterministischen endlichen Automaten akzeptiert.

Satz:
Für eine Sprache L sind die folgenden Aussagen äquivalent.
i) L ist regulär.
ii) L wird von einem (deterministischen) endlichen Automaten akzeptiert.
iii) L wird von einem nichtdeterministischen endlichen Automaten akzeptiert.

Beispiel für Konstruktion des Automaten aus Grammatik I

$$
\begin{aligned}
G &= (\{S, A, B\}, \{a, b\}, P, S) \\
P &= \{S \to \lambda, S \to aA, S \to a, S \to b, S \to bB, A \to a, \\
&\qquad A \to b, A \to aA, A \to bB, B \to bB, B \to bB, B \to b\} \\
G' &= (\{S, A, B, \$\}, \{a, b\}, P', S) \\
P' &= \{S \to \lambda, S \to aA, S \to a\$, S \to b\$, S \to bB, A \to a\$, \\
&\qquad A \to b\$, A \to aA, A \to bB, B \to bB, B \to b\$, \$ \to \lambda\}.
\end{aligned}
$$

nichtdeterministischer endlicher Automat: $\mathcal{B} = (\{a, b\}, \{S, A, B, \$\}, S, \{S, \$\}, \delta)$

$\delta(S, a) = \delta(A, a) = \{A, \$\}$, $\quad \delta(S, b) = \delta(A, b) = \delta(B, b) = \{B, \$\}$,
$\delta(B, a) = \delta(\$, a) = \delta(\$, b) = \emptyset$.

Beispiel für Konstruktion des Automaten aus Grammatik II

deterministischer endlicher Automat: $\mathcal{B}' = (\{a,b\}, \{S,A,B,\$\}, \{S\}, \{S,\$\}, \delta')$

$$
\begin{aligned}
\{A,\$\} &= \delta'(\{S\},a) = \delta'(\{A\},a) = \delta'(\{S,A\},a) = \delta'(\{S,B\},a) \\
&= \delta'(\{A,B\},a) = \delta'(\{S,A,B\},a), \\
\emptyset &= \delta'(\{B\},a) = \delta'(\emptyset,a) = \delta'(\emptyset,b) \\
\{B,\$\} &= \delta'(\{S\},b) = \delta'(\{A\},b) = \delta'(\{B\},b) = \delta'(\{S,A\},b) \\
&= \delta'(\{S,B\},b) = \delta'(\{A,B\},b) = \delta'(\{S,A,B\},b), \\
\delta'(U \cup \{\$\},x) &= \delta'(U,x) \cup \{\$\} \quad \text{für } U \subseteq \{S,A,B\},\ x \in \{a,b\}
\end{aligned}
$$

Kellerautomat – Definition

Definition:
Ein Kellerautomat ist ein Sechstupel

$$\mathcal{M} = (X, Z, \Gamma, z_0, F, \delta),$$

wobei
- X das Eingabealphabet ist,
- Z die endliche Menge von Zuständen ist,
- Γ das Bandalphabet ist,
- $z_0 \in Z$ und $F \subseteq Z$ gelten,
- δ eine Funktion von $Z \times X \times (\Gamma \cup \{\#\})$ in die Menge der endlichen Teilmengen von $Z \times \{R, N\} \times \Gamma^*$ ist,
  wobei $\# \notin \Gamma$, R und N zusätzliche Symbole sind.

Kellerautomat – Definition – Fortsetzung

Definition: Eine Konfiguration K des Kellerautomaten $\mathcal{M}$ ist ein Tripel $(w, z, \alpha\#)$ mit $w \in X^*$, $z \in Z$ und $\alpha \in \Gamma^*$.

Der Übergang von einer Konfiguration K_1 in die nachfolgende Konfiguration K_2 (geschrieben als $K_1 \models K_2$) wird wie folgt beschreiben: Für $x \in X, v \in X^*, z \in Z$, $z' \in Z, \gamma \in \Gamma, \beta \in \Gamma^* \alpha \in \Gamma^*$ gilt

$$\begin{array}{lll}
(xv, z, \gamma\alpha\#) \models (v, z', \beta\alpha\#), & \text{falls} & (z', R, \beta) \in \delta(z, x, \gamma), \\
(xv, z, \gamma\alpha\#) \models (xv, z', \beta\alpha\#), & \text{falls} & (z', N, \beta) \in \delta(z, x, \gamma), \\
(xv, z, \#) \models (v, z', \beta\#), & \text{falls} & (z', R, \beta) \in \delta(z, x, \#), \\
(xv, z, \#) \models (xv, z', \beta\#), & \text{falls} & (z', N, \beta) \in \delta(z, x, \#).
\end{array}$$

Definition: Sei $\mathcal{M}$ ein Kellerautomat. Die von $\mathcal{M}$ akzeptierte Sprache definieren wir durch

$$T(\mathcal{M}) = \{w \mid (w, z_0, \#) \models^* (\lambda, q, \#) \text{ für ein } q \in F\}.$$

Kellerautomat – Beispiele

$\mathcal{M} = (X, Z, \Gamma, z_0, F, \delta)$

$X = \{a, b\},\ \Gamma = \{a\}$,
$Z = \{z_0, z_1, z_2\},\ F = \{z_1\}$,

$\delta(z_0, a, \#) = \{(z_0, R, aa)\}$,
$\delta(z_0, a, a) = \{(z_0, R, aaa)\}$,
$\delta(z_0, b, a) = \{(z_1, R, \lambda)$,
$\delta(q, b, a) = \{(z_1, R, \lambda)\}$,
$\delta(z, x, \gamma) = \{(z_2, R, \gamma)\}$
in allen anderen Fällen

$T(\mathcal{M}) = \{a^n b^{2n} \mid n \geq 1\}$

$\mathcal{M}' = (X, Z', \Gamma', z_0', F', \delta')$

$X = (\{a, b\},\ \Gamma' = \{S, a, b\}$,
$Z' = \{z_0', z_1', z_2'\},\ F' = \{z_1'\}$

$\delta'(z_0', x, \#) = \{(z_1', N, S)\}$ für $x \in \{a, b\}$,
$\delta'(z_1', x, S) = \{(z_1', N, aSbb), (z_1', Nabb)\}$
für $x \in \{a, b\}$,
$\delta'(z_1', x, x) = \{(z_1', R, \lambda)\}$ für $x \in \{a, b\}$
$\delta'(z, x, \gamma) = \{(z_2', R, \lambda)\}$
in allen weiteren Fällen

$T(\mathcal{M}') = \{a^n b^{2n} \mid n \geq 1\}$

Kellerautomaten versus kontextfreie Spachen

Lemma:
Für jede kontextfreie Sprache L gibt es einen Kellerautomaten $\mathcal{M}$ mit $T(\mathcal{M}) = L$.

Lemma:
Zu jedem Kellerautomaten $\mathcal{M}$ gibt es eine kontextfreie Grammatik G mit $L(G) = T(\mathcal{M})$.

Satz:
Die beiden folgenden Aussagen sind für eine Sprache L äquivalent:
i) L ist eine kontextfreie Sprache.
ii) $L = T(\mathcal{M})$ gilt für einen Kellerautomaten $\mathcal{M}$.