

## Akzeptierende TURING-Maschine

### Definition:

Eine akzeptierende TURING-Maschine  $M$  ist ein Sechstupel

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

wobei  $(X, Z, z_0, Q, \delta)$  eine TURING-Maschine ist und  $F \subseteq Q$  gilt.

Die von  $M$  akzeptierte Sprache  $T(M)$  wird durch

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

definiert.

## Akzeptierende TURING-Maschine – Beispiel I

$$M'_1 = (\{a, b\}, \{z_0, z_a, z_b, q_a, q_b\}, z_0, \{q_a, q_b\}, \delta', \{q_a\})$$

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

$$T(M'_1) = \{aw \mid w \in \{a, b\}^*\}.$$

## Akzeptierende TURING-Maschine – Beispiel II

$$M'_2 = (\{a, b\}, \{z_0, z_1, q\}, z_0, \{q\}, \delta, \{q\})$$

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

$$T(M'_2) = \{w : w \in \{a, b\}^*, |w| \text{ ungerade}\}.$$

## 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.

## 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 akzeptierbaren 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

**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  $\tau$  eine Funktion

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

ist.

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

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

## Nichtdeterministische TURING-Maschinen – Beispiel

$$M = (\{a, b\}, \{z_0, z_{0,2}z_{1,2}, z_2, z'_2, z''_2, q\}, z_0, \{q\}, \tau, \{q\})$$

$$\tau(z_0, x) = \{(z_2, x, N), (z_3, x, N)\} \quad \text{für } x \in \{a, b\},$$

$$\tau(z_i, a) = \{(z'_i, a, R)\} \quad \text{für } i \in \{2, 3\},$$

$$\tau(z'_i, a) = \{(z''_i, a, R)\} \quad \text{für } i \in \{2, 3\},$$

$$\tau(z''_i, a) = \{(z''_i, a, R)\},$$

$$\tau(z_i, *) = \{(z_i, *, N)\} \quad \text{für } i \in \{2, 3\},$$

$$\tau(z'_i, *) = \{(q, *, N)\} \quad \text{für } i \in \{2, 3\},$$

$$\tau(z''_i, *) = \{(z_{0,i}, *, L)\} \quad \text{für } i \in \{2, 3\}$$



## Nichtdet. TURING-Maschinen – Beispiel - Fortsetzung

$$\tau(z_{0,2}, a) = \{(z_{1,2}, b, L)\}, \quad \tau(z_{1,2}, a) = \{(z_{0,2}, a, L)\},$$

$$\tau(z_{j,2}, b) = \{(z_{i,2}, b, L)\} \quad \text{für } j \in \{0, 1\},$$

$$\tau(z_{0,3}, a) = \{(z_{1,2}, b, L)\}, \quad \tau(z_{1,3}, a) = \{(z_{2,3}, b, L)\},$$

$$\tau(z_{2,3}, a) = \{(z_{0,3}, a, L)\},$$

$$\tau(z_{j,3}, b) = \{(z_{j,3}, b, L)\} \quad \text{für } j \in \{0, 1, 2\},$$

$$\tau(z_{0,i}, *) = \{(z_i, *, R)\} \quad \text{für } i \in \{2, 3\},$$

$$\tau(z_{j,i}, *) = \{(z_{j,i}, *, N)\} \quad \text{für } j \in \{1, 2\}, i \in \{2, 3\}$$

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

# 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,
- $\delta$  eine Funktion von  $Z \times X$  in  $Z$  ist.

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

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

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

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

## Endlicher Automat – Beispiel

$$\mathcal{A} = (X, Z, z_0, F, \delta) \text{ mit } X = \{a, b, c\}$$
$$Z = \{z_0, z_1, z_2, z_3\} ,$$
$$F = \{z_2\} ,$$
$$\delta(z, x) = \begin{cases} z_1 & \text{für } z = z_0, x = a \\ z_2 & \text{für } z = z_1, x = a \\ z_0 & \text{für } z \in \{z_0, z_2\}, x = c \\ z_3 & \text{sonst} \end{cases}$$

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

## 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  $\delta$  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

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

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 beiden folgenden Aussagen äquivalent.

- i)  $L$  ist regulär.
- ii)  $L$  wird von einem (nichtdeterministischen) endlichen Automaten akzeptiert.

## Beispiel für Konstruktion des Automaten aus Grammatik I

$$G = (\{S, A, B\}, \{a, b\}, P, S)$$

$$P = \{S \rightarrow \lambda, S \rightarrow aA, S \rightarrow a, S \rightarrow b, S \rightarrow bB, A \rightarrow a, \\ A \rightarrow b, A \rightarrow aA, A \rightarrow bB, B \rightarrow bB, B \rightarrow bB, B \rightarrow b\}$$

$$G' = (\{S, A, B, \$\}, \{a, b\}, P', S)$$

$$P' = \{S \rightarrow \lambda, S \rightarrow aA, S \rightarrow a$, $S \rightarrow b$, $S \rightarrow bB, A \rightarrow a$, $, \\ A \rightarrow b$, $, A \rightarrow aA, A \rightarrow bB, B \rightarrow bB, B \rightarrow b$, $, \$ \rightarrow \lambda\}.$$

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

$$\begin{aligned}\{A, \$\} &= \delta'(\{S\}, a) = \delta'(\{A\}, a) = \delta'(\{S, A\}, a) = \delta'(\{S, B\}, a) \\ &= \delta'(\{A, B\}, a) = \delta'(\{S, A, B\}, a),\end{aligned}$$

$$\emptyset = \delta'(\{B\}, a) = \delta'(\emptyset, a) = \delta'(\emptyset, b)$$

$$\begin{aligned}\{B, \$\} &= \delta'(\{S\}, b) = \delta'(\{A\}, b) = \delta'(\{B\}, b) = \delta'(\{S, A\}, b) \\ &= \delta'(\{S, B\}, b) = \delta'(\{A, B\}, b) = \delta'(\{S, A, B\}, b),\end{aligned}$$

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



# Kellerautomat – Definition 1

## 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,
- $\Gamma$  das Bandalphabet ist,
- $z_0 \in Z$  und  $F \subseteq Z$  gelten,
- $\delta$  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 2

**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 beschrieben: Für  $x \in X, v \in X^*, z \in Z, z' \in Z, \gamma \in \Gamma, \beta \in \Gamma^*, \alpha \in \Gamma^*$  gilt

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

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

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

## Kellerautomat – Beispiele

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

$$X = \{a, b\}, \Gamma = \{a\},$$

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

$$\delta(z_0, a, \#) = \{(z_0, R, aa)\},$$

$$\delta(z_0, a, a) = \{(z_0, R, aaa)\},$$

$$\delta(z_0, b, a) = \{(z_1, R, \lambda)\},$$

$$\delta(z_1, b, a) = \{(z_1, R, \lambda)\},$$

$$\delta(z, x, \gamma) = \{(z_2, R, \gamma)\}$$

in allen anderen Fällen

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

$$\mathcal{M}' = (X, Z', \Gamma', z'_0, F', \delta')$$

$$X = (\{a, b\}, \Gamma' = \{S, a, b\},$$

$$Z' = \{z'_0, z'_1, z'_2\}, F' = \{z'_1\}$$

$$\delta'(z'_0, x, \#) = \{(z'_1, N, S)\} \text{ für } x \in \{a, b\},$$

$$\delta'(z'_1, x, S) = \{(z'_1, N, aSbb), (z'_1, N, abb)\}$$

für  $x \in \{a, b\}$ ,

$$\delta'(z'_1, x, x) = \{(z'_1, R, \lambda)\} \text{ für } x \in \{a, b\}$$

$$\delta'(z, x, \gamma) = \{(z'_2, R, \lambda)\}$$

in allen weiteren Fällen

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

## Kellerautomaten versus kontextfreie Sprachen

**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  $\mathcal{M}$ .