





**deterministischer endlicher Automat M**

- 5-Tupel  $M = (Z, \sum, z_0, \delta, E)$
- $Z$  eine endliche Menge von Zuständen
- $\sum$  das Eingabealphabet (mit  $Z \cap \sum = \emptyset$ )
- $z_0 \in Z$  der Startzustand
- $\delta : Z \times \sum \rightarrow Z$  die Übergangsfunktion
- $E \subseteq Z$  die Menge der Endzustände
- kurz: DFA (deterministic finite automaton)

**Turingmaschine**

Definition: Eine Turingmaschine (TM) ist ein 7-Tupel

$$M = (Z, \sum, \Phi, \delta, z_o, \square, E)$$

, wobei

- $\sum$  das Eingabealphabet
- $\Phi$  mit  $\Phi \supseteq \sum$  und  $\Phi \cap Z \neq \emptyset$  das Arbeits- oder Bandalphabet,
- $z_0 \in Z$  der Startzustand,
- $\delta : Z \times \Phi \rightarrow (Z \times \Phi \times \{L, N, R\})$  die Überföhrungsfunktion
- $\square \in \Phi / \sum$  das Leerzeichen oder Blank und
- $E \subseteq Z$  die Menge der Endzustände ist

**Linksableitung**

Eine Ableitung  $S = w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_r = w \in \sum^*$  heiÖt Linksableitung, wenn in jedem Schritt das am weitesten links stehende Nichtterminal ersetzt wird (Analog Rechtsableitung).

**CYK-Algorithmus**

Gehört ein gegebenes Wort zu  $L(G)$ ?

	$a_1$	$a_2$	$\dots$	$\dots$	$a_{n-1}$	$a_n$
$j = 1$	$V_{1,1}$	$V_{2,1}$	$\dots$	$\dots$	$V_{n-1,1}$	$V_{n,1}$
$j = 2$	$V_{1,2}$	$V_{2,2}$	$\dots$	$\dots$	$V_{n-1,2}$	
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$		
	$\dots$	$\dots$	$\dots$			
$j = n - 1$	$V_{1,n-1}$	$V_{2,n-1}$				
$j = n$	$V_{1,n}$					

Von oben nach unten, von links unten nach rechts oben

**Kellerautomaten**

Um ein Automatenmodell für Kontextfreie Sprachen zu erhalten führt man einen Keller-(Pushdown)-Speicher ein, auf dem sich eine beliebig lange Sequenz von Zeichen befinden darf. Beim Einlesen eines neuen Zeichens wird das oberste Zeichen des Kellers gelesen und durch eine (evtl. leere) Sequenz von Zeichen ersetzt. An anderen Stellen kann der Keller nicht gelesen/geändert werden

**die Greibach-Normalform**

Eine kontextfreie Grammatik G ist in Greibach Normalform falls alle Produktionen aus P folgende Form haben:  $A \rightarrow aB_1B_2\dots B_k$ , mit  $k \in \mathbb{N}$ ,  $A, B_1, \dots, B_k \in V$  und  $a \in \sum$ . Die Greibach Normalform garantiert, dass bei jedem Ableitungsschritt genau ein Alphabetsymbol entsteht.

**das Lemma von Ogden (William Ogden)**

Wenn L eine kontextfreie Sprache ist, dann gibt es  $n \geq 1$  derart, dass für alle  $z \in L$ , in denen  $n$  Positionen markiert sind, gilt: es gibt Wörter  $u, v, w, x, y \in \sum^*$  mit

1.  $z = uvwxy$
2. v oder x enthält wenigstens eine der Markierungen oder
3.  $uv^iwx^iy \in L$  für alle  $i \geq 0$

**Halteproblem**

allgemein: Das Halteproblem ist die Menge aller Paare  $(M, x)$ , wobei M eine TM ist und  $x \in \{0, 1\}^*$ , sodass M bei Eingabe von x hält.

**Reduktion**

Seien  $A \subseteq \sum^*, B \subseteq \Phi^*$ . Eine Reduktion von A auf B ist eine totale und berechenbare Funktion  $f : \sum^* \rightarrow \Phi^*$ , so dass für alle  $w \in \sum^*$  gilt:  $w \in A \leftrightarrow f(x) \in B$ . A heiÖt auf B reduzierbar (in Zeichen  $A \leq B$ ), falls es eine Reduktion von A auf B gibt.

**Satz von Rice**

Sei R die Klasse aller Turing-berechenbaren Funktionen  $\{0, 1\}^* \rightarrow \{0, 1\}^*$ ,  $\Omega$  die nirgendwo definierte Funktion und sei  $S \subseteq \mathbb{R}$  mit  $\Omega \in S$  und  $\neq \mathbb{R}$ . Dann ist die Sprache  $C(S) = \{w \in L_{TM} \mid \phi_w \in S\}$  unentscheidbar.

**Semi Entscheidbarkeit**

Auch wenn das Halteproblem bei leerer Eingabe  $H_0$  unentscheidbar ist, so kann doch nach endlicher Zeit festgestellt werden, daß die Maschine  $M_w$  bei leerer Eingabe anhält -  $H_0$  ist also "halb-öder Semi-entscheidbar".

Eine Sprache  $L \subseteq \sum^*$  heiÖt semi-entscheidbar, falls die "halbe" charakteristische Funktion von L, d.h. die partielle Funktion  $X'_L : \sum^* \rightarrow \{1\}$  mit  $x'_L = \begin{cases} 1 & \text{falls } w \in L \\ undef. & \text{falls } w \notin L \end{cases}$  berechenbar ist.

**Universelle Turing Maschine**

eine Turing-Maschine, die jede Turing-Maschine simulieren kann, wenn deren Kodierung gegeben ist. Buchstaben des Bandalphabets als Wörter über  $\{0, 1, 2\}$  mit  $\square = 2$  kodiert. Ab jetzt nehmen wir an, daß wir immer dieses Bandalphabet haben. Eine Turing Maschine U heiÖt universelle Turing Maschine, wenn sie die folgende partielle Funktion berechnet.

$$\{0, 1\}^* \rightarrow \{0, 1\}^*$$

$$y \rightarrow \begin{cases} \phi_w(x) & \text{falls } y = w000x, w \in L_{TM}, x \in \{0, 1\}^* \\ undef. & \text{sonst} \end{cases}$$

**Totale berechenbare Funktionen**

Gesucht ist  $C \subseteq TOT \subseteq L_{TM}$ , so dass  $\{\phi_w \mid w \in C\} = \{\phi_w \mid w \in TOT\}$  die Menge der totalen berechenbaren Funktionen ist.

- $TOT$  ist nicht einmal semi-entscheidbar
- indirekt nehmen wir an, dass C semi-entscheidbar ist
- dann existiert eine totale berechenbare Funktion  $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$  mit Bild C
- neue Funktion  $g : \{0, 1\}^* \rightarrow \{0, 1\}^* : w \mapsto 1\phi_{f(w)}(w)$

**Chomsky Normalform**

Eine kontextfreie Grammatik G ist in Chomsky Normalform, falls alle Produktionen

- die Form  $A \rightarrow AB$  oder  $A \rightarrow a$  haben,
- und  $S \rightarrow \epsilon$  und S nie auf der rechten Seite einer Produktion vorkommt