

Typ 3	reguläre Grammatik DEA NEA (mit und ohne ε) regulärer Ausdruck	z.B. a^*b^*
Det. KF	LR-Grammatik deterministischer Kellerautomat (DKellerA)	
Typ 2	kontextfreie Grammatik Kellerautomat (NKellerA)	$a^n b^n$
Typ 1	kontextsensitive Grammatik linear beschränkter Automat (NLBTM)	$a^n b^n c^n$
Typ 0	Typ 0 - Grammatik (rek. aufz. Spr.) Turingmaschine (TM)	\bar{L}^d

Tabelle 1: **Beschreibungsmittel**

Nichtdet. Automat	Determ. Automat	äquivalent ?
NEA	DEA	ja
NKellerA	DKellerA	nein
NLBTM	DLBTM	???
NTM	DTM	Ja

Tabelle 2: **Determinismus und Nichtdeterminismus**

	Schnitt	Vereinigung	Komplement	Produkt	Stern
Typ 3	ja	ja	ja	ja	ja
Det. KF	nein	nein	ja	nein	nein
Typ 2	nein	ja	nein	ja	ja
Typ 1	ja	ja	ja	ja	ja
Typ 0	ja	ja	nein	ja	ja
semient. Spr.	ja	ja	nein	nein	nein
ent. Spr.	ja	ja	ja	nein	nein

Tabelle 3: **Abschlusseigenschaften**

Die **Nerode-Relation** R_L zu einer Sprache ist def. durch:

$$R_L = \{(x, y) : xz \in L \text{ gdw. } yz \in L \forall z \in \Sigma^*\}$$

Die **Nerode-Relation** R_M zu einem Automaten ist def. durch:

$$R_M = \{(x, y) : \delta^*(s, x) = \delta^*(s, y)\}$$

Algorithmisch **entscheidbare Eigenschaften** von Automaten:

1. $L(A) = \{\}$
2. Endlichkeit von $L(A)$

Typ	Komp.
3	$O(n)$
Det. KF	$O(n)$
2	$O(n^3)$
1	$ \Sigma ^{O(n)}$, NP-Hart
0	semientscheidbar

Tabelle 4: **Komplexität des Wortproblems**

Typ	Wort	Leerheit	Äquivalenz	Schnitt
3	Ja	Ja	Ja	Ja
Det. KF	Ja	Ja	Ja	Nein
2	Ja	Ja	Nein	Nein
1	Ja	Nein	Nein	Nein
0	Nein	Nein	Nein	Nein

Tabelle 5: **Entscheidbarkeit**

3. $L(A) = \Sigma^*$

Entscheidbarkeit(rekursiv): Es ex. eine TM, die alle Wörter aus L akzeptiert und auf jede Eingabe hält.

Semi-Entscheidbarkeit(rekursiv aufzählbar): Es ex. eine TM, die alle Wörter aus L akzeptiert. Das Verhalten für Wörter $w \notin L$ ist undefiniert.

Kodierungsvorschrift **Gödelnummer**

1. Kodiere δ :

$$\text{code}(\delta(q_i, a_j) = (q_r, a_s, d_t)) = 0^i 10^j 10^r 10^s 10^t$$

mit $d_t \in \{d_1 = L, d_2 = R, d_3 = N\}$

2. Die TM wird dann kodiert durch:

$$111\text{code}_1 11\text{code}_2 11 \dots 11\text{code}_z 111$$

mit code_i für $i = 1, 2, \dots, z$ in bel. Reihenfolge.

Pumping Lemma

L regulär \Rightarrow

$$(\exists n \in \mathbf{N})(\forall z \in L, |z| \geq n)(\exists u, v, w)$$

$$[(z = uvw) \wedge (|v| \geq 1) \wedge (|uv| \leq n) \wedge (uv^i w \in L, \forall i \geq 0)]$$

L kontextfrei \Rightarrow

$$(\exists n \in \mathbf{N})(\forall z \in L, |z| \geq n)(\exists u, v, w, x, y)$$

$$[(z = uvwxy) \wedge (|vx| \geq 1) \wedge (|vwx| \leq n) \wedge (uv^iwx^iy \in L, \forall i \geq 0)]$$

Automatenminimierung:

1. nicht erreichbare Zustände entfernen.
2. Tabelle aller Zustandspaare $\{z, z'\}$ mit $z \neq z'$
($z_1 - z_k$ links, $z_0 - z_{k-1}$ unten)
3. Markieren der Zustandspaare mit $z \in F$ und $z \notin F$ oder umgekehrt.
4. Betrachte unmarkierte Paare $\{z, z'\}$.
Wenn $\{\delta(z, a), \delta(z', a)\}$ für mind. ein $a \in \Sigma$ bereits markiert, markiere $\{z, z'\}$.
5. (4) Wiederholen bis keine Änderung mehr.
6. Unmarkierte Paare können verschmolzen werden.

Chomsky-Normalform

1. $r \in V^* \cup \Sigma$
2. $|r| \leq 2$
3. ε -Produktionen entfernen
4. Kettenregeln ersetzen

Problem	Gegeben	Gesucht	polyn. red. von
SAT	aussagenlog. Formel	Wahrheitsbelegung	TM
3SAT	boolesche Formel in KNF mit 3 Literalen pro Klausel	Erfüllbarkeit	SAT
Set Cover	Mengensystem über endl. Grundmenge M , also $T_1, \dots, T_k \subseteq M$, Zahl $n \leq k$	Auswahl n Mengen T_{i_1}, \dots, T_{i_n} , in denen alle Elemente aus M vorkommen	3SAT
Steiner-Tree	Graph $G = (V, E)$ mit positiven Kantengewichten $c : E \rightarrow \mathbf{R}$, $V = R(\text{Pflichtknoten}) \cup F(\text{Steinerknoten})$	Baum $T \subseteq E$ der mit minimalen Kosten alle Pflichtknoten verbindet	3SAT
Clique	ungerichteter Graph $G = (V, E)$ und Zahl $k \in \mathbf{N}$	Clique $V' \subseteq V$, so dass $\forall i, j \in V', i \neq j$, gilt: $\{i, j\} \in E$, mit $ V' \geq k$	3SAT
Vertex Cover	ungerichteter Graph $G = (V, E)$ und Zahl $k \in \mathbf{N}$	überdeckende Knotenmenge $V' \subseteq V$ mit $ V' \geq k$, sodass $\forall \{u, v\} \in E: u \in V'$ oder $v \in V'$	Clique
Subset Sum (Rucksack)	Zahlen $a_1, \dots, a_k \in \mathbf{N}$ und $b \in \mathbf{N}$	Teilmenge $I \subseteq \{1, \dots, k\}$ mit $\sum_{i \in I} a_i = b$	3SAT
Partition	Zahlen $a_1, \dots, a_k \in \mathbf{N}$	Teilmenge $J \subseteq \{1, \dots, k\}$ mit $\sum_{i \in J} a_i = \sum_{i \notin J} a_i$	Subset Sum
Bin Packing	Behältergröße $b \in \mathbf{N}$, Behälteranzahl $k \in \mathbf{N}$, Objekte $a_1, \dots, a_k \leq b$	Abb. $f : \{1, \dots, n\} \rightarrow \{1, \dots, k\}$, sodass $\forall j = 1, \dots, k : \sum_{f(i)=j} a_i \leq b$	Partition
Knapsack	endl. Menge M , Gewichtsfkt. $w : M \rightarrow \mathbf{N}_0$, Kostenfkt. $c : M \rightarrow \mathbf{N}_0$, $W, C \in \mathbf{N}_0$	$M' \subseteq M$ mit $\sum_{a \in M'} w(a) \leq W$ und $\sum_{a \in M'} c(a) \geq C$	Subset Sum
ILP	?????????	?????????	Subset Sum
Directed Hamilton Circle	gerichteter Graph $G = (V, E)$	Hamiltonkreis: einfacher Kreis der jeden Knoten genau einmal enthält	3SAT
Hamilton Circle	ungerichteter Graph $G = (V, E)$	Hamiltonkreis	Directed Hamilton Circle
TSP	$n \times n$ -Matrix M_{ij} und Zahl k	Rundreise mit max. Länge k	Hamilton Circle
Coloring	ungerichteter Graph $G = (V, E)$ und Zahl $k \in \mathbf{N}$	Färbung der Knoten in V mit k versch. Farben, mit je 2 unterschiedlich gefärbten Nachbarn	3SAT

Tabelle 6: NP-Vollst. Probleme