Grammatiken

G= (V, • , P, S) (Variablen, Terminale, Produktionen, Startsymbol)

TYP 0: Alle Grammatiken
TYP 1: Eine Grammatik ist kontextsensitiv, falls für alle Regeln w_1 à w_2 aus P $|w_1|$ • $|w_2|$ gilt.
TYP 2: Eine Grammatik ist kontextfrei, wenn sie von TYP 1 ist und für alle Regeln aus P gilt dass w_1 eine einzelne Variable ist.
TYP 3: Eine Grammatik ist regulär, wenn sie von Typ 2 ist und die rechte Seite der Regeln entweder ein einzelnes Terminalzeichen oder ein Terminalzeichen gefolgt von einer Variablen ist.
Jede L_{CF} über einem einelementigen • ist bereits regulär.

Reguläre Sprachen

DFA: (Z, • , • , Z_0, E), NFA mehrere Startzustände, NFA à DFA über Potenzmengen, beide Modelle sind gleichmächtig, sie erkennen die Menge der reg. Sprachen

Folgender Algorithmus liefert aus einem DFA den Minimalautomaten:
1. Stelle eine Tabelle aller Zustandspaare {z, z_0} mit z • z_0 auf.
2. Markiere alle Paare {z, z_0} mit z ∈ E und z_0 ∉ E (oder umgekehrt)
3. Für jedes noch unmarkierte Paar {z, z_0} und für jedes $a \in \sum$ teste, ob $\{d(z,a), d(z',a)\}$ bereits markiert ist. Wenn ja markiere auch {z, z_0}
4. Wiederhole den letzten Schritt, bis sich keine Änderungen mehr an der Tabelle ergeben
5. Alle jetzt noch unmarkierten Paare können zu jeweils einem Zustand verschmolzen werden.

Pumping Lemma:
Sei L eine reguläre Sprache. Dann gibt es eine Zahl n, so dass alle Worte x mit |x| • n zerlegen lassen in x = uvw so dass folgende Eigenschaften erfüllt sind:
|v| • 1; |uv| • n; für alle i ist uv^iw in L

Kontextfreie Sprachen

(D)PDA: (Z, • , • , • , Z_0, #), PDA akzeptiert mit leerem Keller, DPDA mit Endzustand, PDA und DPDA sind nicht gleichwertig, ein DPDA ist in einen PDA überführbar, andersherum jedoch nicht.
Konfiguration: (Zustand, noch zu lesende Eingabe, Kellerinhalt)

CNF: Alle Regeln in der Form A à BC oder A à a, zuerst e-frei machen, dann umformen.
CYK-Algorithmus: entscheidet in $O(n^3)$ ob ein Wort in L enthalten, benötigt Grammatik in CNF

Pumping Lemma:
Sei L eine kontextfreie Sprache. Dann gibt es eine Zahl n, so dass alle Worte x mit |x| • n zerlegen lassen in x = uvwxy so dass folgende Eigenschaften erfüllt sind:
|vx| • 1; |vwx| • n; für alle i ist uv^iwx^iy in L

Beispiel: Annahme $L = \{a^ib^jc^k | i,k > 0 \wedge j = \max\{i,k\}$ kontextfrei (z.Z. Gegenteil); Wähle das Wort $z = a^nb^nc^n$ der Länge $|z| = 3n \geq n$, wobei n die Pumping Lemma Zahl. Nach PL gibt es nun eine Zerlegung $z = uvwxy = a^nb^nc^n$. Fallunterscheidung: 1. vx nur a's (b's, c's), dann auch $q = uv^2wx^2y = a^{n+|vx|}b^nc^n \in L$, Widerspruch. 2. vx hat r a's und s b's, dann wäre auch $q = ^{n-r} b^{n-s}c^n \in L$, Widerspruch. 3. vx hat r a's und s c's, dann ist wegen $|vwx| \leq n$ vwx nicht lang genug, um alle b zu überbrücken, Widerspruch. 4. r b's und s c's läuft wie 2. 5. a's, b's und c's, nicht möglich, da in $|vwx| \leq n$ eben nicht alles drin sein kann, wegen der Längenbeschr. Also alles widersprüchlich, damit L nicht kontextfrei.

Einige Beispielsprachen:
$\{a^m b^m c^m\}$ ist nicht kontextfrei
$\{a^m b^m\}$ ist kontextfrei
$\{a^m b^n \mid m < m\}$ ist nicht regulär
$\{a^m b^n a^m b^n\}$ ist nicht kontextfrei
$\{a^m b^n a^n b^m\}$ ist kontextfrei
$\{a^n b^n a^n b^n\}$ ist nicht kontextfrei

Abschlusseigenschaften

	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

Typ 3	reguläre Grammatik DFA NFA reguläre Ausdrücke
Det. kf.	LR(k)-Grammatik
Typ 2	kontextfreie Grammatik Kellerautomat
Typ 1	kontextsensitive Grammatik linear beschränkter Automat
Typ 0	Typ 0 Grammatik Turinmaschine

nichtdet. Automat	deterministischer Automat	äquivalent?
NFA	DFA	Ja
PDA	DPDA	nein
LBA	DLBA	?
TM	DTM	Ja

Entscheidbarkeit

	Wortproblem	Leerheitsproblem	Äquivalenzproblem	Schnittproblem
Typ 3	ja	ja	ja	ja
det. kf	ja	ja	ja	nein
Typ 2	ja	ja	nein	nein
Typ 1	ja	nein	nein	nein
Typ 0	nein	nein	nein	nein