Grundbegriffe

Σ^* Menge aller Wörter über Σ inkl. ε $\quad \Sigma^* A = \Sigma^*$

$\varnothing$ Sprache mit 0 Wörtern $\quad \{\varepsilon\}$ Sprache mit leerem Wort

$A^0 = \{\varepsilon\}$ $A^* = \bigcup_{n \in \mathbb{N}_0} A^n = A^0 \cup A^1 \cup A^2 \cup \ldots = A^0 \cup A^+$

$A^{n+1} = AA^n$ $A^+ = AA^* = \bigcup_{n \geq 1} A^n = A^1 \cup A^2 \cup \ldots$

konkatenieren AB $\{ab, b\}\{a, bb\} = \{aba, abbb, ba, bbb\}$

kartesische Produkt $A \times B$ $\{ab, b\} \times \{a, bb\} = \{(ab,a), (ab,bb), (b,a), (b,bb)\}$

$|w|_a = \#_a(w)$ Anzahl der a in w $\quad \#w\,\#100 = 4$ $\quad (w)_2\ (100)_2 = 4$

Rechenregeln $(A \setminus C)B \neq AB \setminus CB$ $\quad |AB| \neq |A||B|$

$\forall A: \varepsilon \in A^*$ $\quad \varnothing^* = \{\varepsilon\}$ $\quad \{\varepsilon\}A = A$ $\quad A(BC) = (AB)C$ $\quad A(B \cup C) = AB \cup AC$

$\varnothing A = \varnothing$ $\quad A \times \varnothing = \varnothing$ $\quad A^*A^* = A^*$ $\quad A^* = A^+ \Leftrightarrow \varepsilon \in A$ $\quad (A \cup B)C = AC \cup BC$

$\varnothing \setminus \alpha = \alpha$ $\quad \alpha\beta \neq \beta\alpha$ $\quad A(B \cap C) \neq AB \cap AC$ $\quad A \not\subseteq B \not\supseteq A^* \not\subseteq B^*$ $\quad |A| \leq |B| \not\leq |AB|$

Grammatik $G = (V, \Sigma, P, S)$ $\quad S \in V$ Startsymbol

V endliche Menge von Nichtterminale A,B,C ...

Σ endliche Menge von Terminalzeichen / Alphabet a,b,c ...

$P \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ Menge von Produktionen $X \to \alpha X b \mid \varepsilon$

Chomsky Hierarchie $L(\text{Typ } 3) \subset L(\text{Typ } 2) \subset L(\text{Typ } 1) \subset L(\text{Typ } 0)$

Typ 0: immer

Typ 1: Typ 0 #且 für jede Produktion $\alpha \to \beta$ außer $S \to \varepsilon$ gilt $|\alpha| \leq |\beta|$

ε-Produktion nur ausgehend von Startsymbol S erlaubt

und $S \to \varepsilon$: Startsymbol S darf nicht auf rechte Seite einer Produktion

Typ 2: Typ 1 #且 linke Seite hat nur 1 Nichtterminalzeichen

kontextfreie Grammatik erzeugt kontextfreie Sprachen

Typ 3: Typ 2 #且 rechte Seite hat genau ein Terminalzeichen und max. ein weiteres Nichtterminalzeichen, entweder $X \to a$ oder $X \to bY$

d.h. $T \to Tb$ oder $S \to T$ verletzt Typ 3

rechtslineare Grammatik erzeugt reguläre Sprachen (linkslinear auch regulär)

DFA Deterministisch Endlich Automat $M = (Q, \Sigma, \delta, q_0, F)$

Übergangsfunktion $\delta: Q \times \Sigma \to Q$ Zustand + Zeichen → Zustand

! von jedem Zustand aus jedes Symbol kommt genau 1 mal vor

$\delta(q,a)$ beschreibt den Zustand den von q aus mit Zeichen a erreicht

$\hat{\delta}(q,w)$ beschreibt den Zustand den von q aus mit Wort w erreicht

$\hat{\delta}(q,\varepsilon) = q$ $\quad \hat{\delta}(q,aw) = \hat{\delta}(\hat{\delta}(q,a),w)$ wobei $a \in \Sigma, w \in \Sigma^*$

Sprache regulär ⇔ von DFA akzeptiert

rechtslineare Grammatik G ⇔ DFA M mit $L(M) = L(G)$

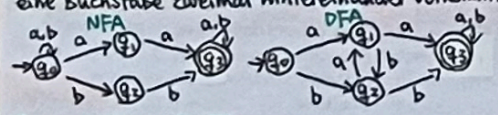
NFA Nichtdeterministisch Endlich Automat $N = (Q, \Sigma, \delta, q_0, F)$

! ein Wort von NFA akzeptiert gdw. mindestens eine dieser Zustandsfolgen zum Endzustand führt

rechtslineare Grammatik G ⇔ NFA N mit $L(N) = L(G)$

für jeden NFA N gibt ein DFA M mit $L(N) = L(M)$

NFA DFA Vergleich $\Sigma = \{a,b\}$ $B_i = \{w \in \Sigma^* \mid w_i = w_{|i+1|}\}$ eine Buchstabe zweimal hintereinander vorkommt

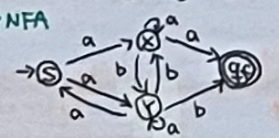


Grammatik G ⇔ NFA

$S \to aX \mid aY$

$X \to aX \mid bY \mid a$

$Y \to aS \mid bX \mid aY \mid b$



ε-NFA ⇒ NFA

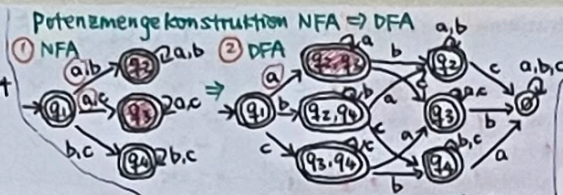
1) berechne für jeden Zustand q eine ε-Hülle

2) neue Übergangsfunktion $\delta(q,a) = \varepsilon$-Hülle aller Zustände, die von ε-Hülle(q) aus mit Symbol a erreichbar sind

3) Endzustände sind alle q_i deren ε-Hülle einen ursprünglichen Endzustand enthält

Potenzmengenkonstruktion NFA ⇒ DFA

① NFA ② DFA ⇒



DFA ⇒ rechtslineare Grammatik G

③ $S \to aB_1 \mid bB_2 \mid cB_3 \mid a \mid b \mid c \mid \varepsilon$ $\quad C_1 \to a \mid b \mid aC_1 \mid bC_1 \mid cD$

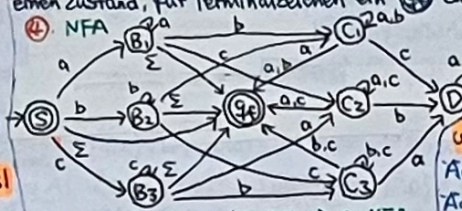
$B_1 \to a \mid b \mid c \mid aB_1 \mid bC_1 \mid cC_2$ $\quad C_2 \to a \mid c \mid aC_2 \mid cC_2 \mid bD$

$B_2 \to a \mid b \mid c \mid bB_2 \mid aC_1 \mid cC_3$ $\quad C_3 \to b \mid c \mid bC_3 \mid cC_3 \mid aD$

$B_3 \to a \mid b \mid c \mid cB_3 \mid aC_2 \mid bC_3$ $\quad D \to aD \mid bD \mid cD$

G in NFA N' übersetzen: erstellen für jedes Nichtterminal einen Zustand, für Terminalzeichen ein q_E Endzustand

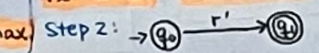
④ NFA



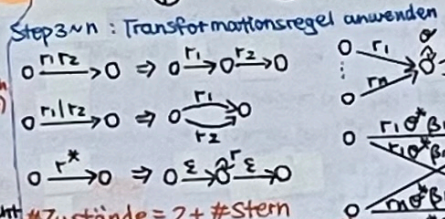
Regulär Ausdruck REGEX ⇒ ε-NFA

Step 1: eliminieren ∅ von r raus, dann r'

$r\varnothing = \varnothing = \varnothing r$ $\quad r \mid \varnothing = r = \varnothing \mid r$ $\quad \varnothing^* = \varepsilon$

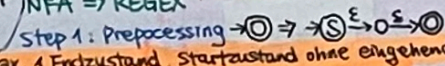
Step 2: 

Step 3 ∧ n: Transformationsregel anwenden

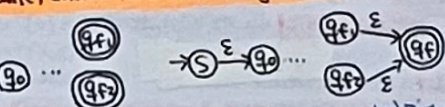


#Zustände = 2 + #Stern + #Konkatenation

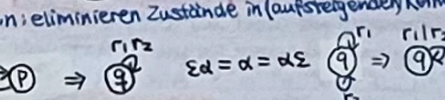
NFA ⇒ REGEX

Step 1: Prepocessing → ⓞ ⇒ 

max. 1 Endzustand, Startzustand ohne eingehende Kante, Endzustand ohne ausgehende Kante



Step 2 ∧ n: eliminieren Zustände in (aufsteigender) Reihenfolge

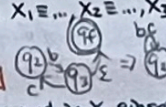


LGS mit Ardens Lemma lösen (NFA ⇒ REGEX)

Step 1: LGS laut NFA-Graph aufstellen $X_1 \equiv \ldots, X_2 \equiv \ldots, X_3 \equiv \ldots$

$X \equiv aY \mid \varepsilon$ falls X ist Endzustand

$Y \equiv bY \mid \varnothing$ falls Y ist kein Endzustand



Step 2: Gleichung nach X_2 bis X_n auflösen und in X_1 einsetzen

Step 3: verbleibende X_k von X_1-Gleichung auflösen

Step 4: in X_1 einsetzen und erhalten $X_1 \equiv \alpha \equiv \ldots$

Ardens Lemma: $X = \alpha X \mid \beta \to X = \alpha^* \beta$ nur wenn $\varepsilon \notin L(\alpha)$

$X = \alpha X \mid \varepsilon \to X = \alpha^*$ $\quad X = \alpha X \to X = \varnothing$

Rechenregel (reguläre Ausdruck) Bindungsstärke * > Konkatenation > |

$L(\varnothing) = \varnothing$ $\quad L(\varepsilon) = \{\varepsilon\}$ $\quad L(\alpha) = \{\alpha\}$ $\quad L(\alpha^*) = L(\alpha)^*$ $\quad L(\alpha\beta) = L(\alpha)L(\beta)$ $\quad L(\alpha|\beta) = L(\alpha) \cup L(\beta)$

$(\alpha|\beta)|\gamma \equiv \alpha|(\beta|\gamma)$ $\quad (\alpha\beta)\gamma \equiv \alpha(\beta\gamma)$ $\quad \alpha(\beta|\gamma) \equiv \alpha\beta|\alpha\gamma$ $\quad (\alpha|\beta)\gamma \equiv \alpha\gamma|\beta\gamma$ $\quad \alpha|\beta \equiv \beta|\alpha$ $\quad \alpha|\alpha \equiv \alpha$

$\varnothing|\alpha \equiv \alpha \equiv \alpha|\varnothing$ $\quad \varnothing\alpha \equiv \varnothing \equiv \alpha\varnothing$ $\quad \varepsilon\alpha \equiv \alpha \equiv \alpha\varepsilon$ $\quad \varepsilon^* \equiv \varepsilon$ $\quad \varnothing^* \equiv \varepsilon$ $\quad \varepsilon|\alpha\alpha^* \equiv \alpha^*$ $\quad \alpha^*\alpha \equiv \alpha\alpha^*$ $\quad (\alpha^*)^* \equiv \alpha^*$

$\alpha \equiv \beta \Leftrightarrow L(\alpha) \equiv L(\beta)$ $\quad L^{ab} = (L^a)^b$ Residualsprache einer leeren Sprache ist leere Sprache

DFA-Produktkonstruktion

beide DFA laufen synchron parallel (zuerst beide Startzustände im Paar, dann wie Potenzmengenkonstruktion), neue Endzustände je nach Schnitt / Vereinigung / Komplement markieren

Minimierung endlichen Automaten (DFA)

1) entfernen alle vom Startzustand nicht erreichbare Zustände

2) Berechnen äquivalente Zustände → $p \in F$ und $q \notin F \Rightarrow p,q$ unterscheidbar → $\delta(p,a)$ und $\delta(q,a)$ unterscheidbar, dann p,q unterscheidbar

3) äquivalente Zustände zusammenfassen, minimal DFA uptodate

Entscheidungsprobleme sind entscheidbar, wenn ein Algo gibt, der immer terminiert und die richtige (Ja/Nein) liefert. Sei D ein DFA, NFA, RE, rechtslineare Grammatik

- Wortproblem $w \in D$? DFA $O(|w||M|)$ NFA $O(|Q|^2|w| + |N|)$
- Leerheitsproblem $L(D) = \varnothing$? DFA $O(|Q||\Sigma|)$ NFA $O(|Q|^2|\Sigma|)$
- Endlichkeitsproblem $L(D)$ endlich? für DFAs u. NFAs entscheidbar
- Äquivalenzproblem $L(D_1) \equiv L(D_2)$? DFA $O(|Q_1||Q_2||\Sigma|)$ NFA $O(2^{|Q_1|+|Q_2|})$

Residualsprachen von L bzgl. w $L^w = \{z \in \Sigma^* \mid wz \in L\}$

Wörter äquivalent $u \equiv_L v \Leftrightarrow L^u = L^v$ gleiche Residualsprachen

Äquivalenzklasse $[a]_\approx = \{b \mid a \approx b\}$ $\quad [a]_\approx = [b]_\approx \Leftrightarrow a \approx b$

Äquivalenzrelation $\approx \subseteq A \times A$, falls Reflexivität $\forall a \in A.\ a \approx a$.

Symmetrie $\forall a,b \in A.\ a \approx b \Rightarrow b \approx a$. Transitivität $\forall a,b,c \in A.\ a \approx b \wedge b \approx c \Rightarrow a \approx c$

Quotientenmenge $A/\approx = \{[a]_\approx \mid a \in A\}$ Menge der Äquivalenzklasse

eine Sprache ist regulär ⇔ es hat endliche Residualsprachen

Anzahl der Äquivalenzklasse von $\equiv_M$ ist Anzahl der Sprachen, die von den Zuständen von M erkannt werden

Sprache nicht regulär: endliche Menge von Wörter mit paarweise disjunkt Residualsprachen

Bestimmen einer unendlichen Menge von Wörter mit unterschiedlichen Residualsprachen

$L_Z = \{a^i b^i c^i \mid i \in \mathbb{N}\}$ Seien $i,j \in \mathbb{N}$ verschieden. Dann $a^i b^i c^i \in L_Z$ aber $a^i b^i c^i \notin L_Z$ daher $L_Z^{a^i b^i} \neq L_Z^{a^j b^j}$ Somit sind Residualsprache der obigen Wörter unterschiedlich. L_Z ist keine reguläre Sprache.

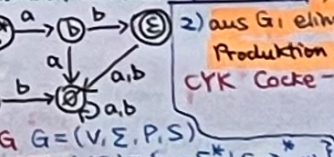
Kanonische Minimalautomat $M_L = (R_L, \Sigma, \delta_L, L, F_L)$

R_L: Menge der Residualsprachen von L

jede minimal DFA für reguläre Sprachen ist äquivalent zu M_L

Regulär Ausdruck ⇒ kanonisch Minimalautomat DFA

$r = ab \mid ba^*$

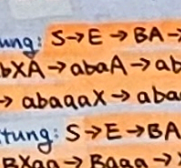


CFG mit n Produktion
max n nützlich Nichtterminal
max n erzeugend Nichtterminal

Kontextfreie Grammatik CFG $G = (V, \Sigma, P, S)$

Kontextfreie Sprachen CFL $L(G) = \{w \in \Sigma^* \mid S \to_G^* w\}$

Syntaxbaum



Linksableitung: $S \to E \to BA \to XBXA$

$\to aBXA \to abXA \to abaA \to abaXAX$

$\to abaaAX \to abaaaX \to abaaaa$

Rechtsableitung: $S \to E \to BA \to BXAX$

$\to BXAa \to BXaa \to Baaa \to XBXaaaa$

$\to XBaaaa \to Xbaaaa \to abaaaa$

mehrdeutige CFG ⇔ Es gibt $w \in L(G)$, das 2 verschiedene Syntaxbäume hat

inhärent-mehrdeutig CFL ⇔ jede CFG mit $L(G) = L$ mehrdeutig ist

für jede DCFL gibt eine nicht mehrdeutige Grammatik

Chomsky-Normalform CNF: ∀ Produktionen in Form $A \to a$ oder $A \to BC$

CNF-Konstruktion

1) $\beta \in \Sigma \cup V^*$ Terminale darf nur in Produktion der Länge 1 erzeugt

Terminal a durch X_a ersetzen, $X_a \to a$ hinzufügen

2) $|\beta| \leq 2$ Jede Produktion hat höchstens Länge 2

Entfernen lange Produktionen (Länge ≥ 3)

$A \to B_1 B_2 B_3 \ldots \Rightarrow A \to B_1 C_2$ $\quad C_2 \Rightarrow B_2 C_3$

eg. $A \to XYZ \Rightarrow A \to XX_{YZ}$ $\quad X_{YZ} \Rightarrow YZ$

3) $\beta \neq \varepsilon$ Es gibt keine ε-Produktionen

→ berechnen iterativ alle Variablen die ε erzeugen

→ erzeugen zusätzlich alle möglichen Produktion (wegen ε-Produktion) → entfernen alle ε-Produktionen

eg. $A \to B$ $\quad B \to b \mid \varepsilon$ $\quad E_0 = \{B\}$ $\quad E_1 = \{B, A\}$

4) $\beta \notin V$ Es gibt keine Kettenproduktionen

→ berechnen iterativ alle Kettenproduktionen

$T_0 = \{(X \to Y) \in P \mid X, Y \in V\} =$ direkt vorhanden eg. $\{(A \to B), (B \to C)\}$

$T_{k+1} = T_k \cup \{(X \to Z) \mid X, Y, Z \in V \wedge (X \to Y) \in T_k \wedge (Y \to Z) \in T_k\}$ eg. $\{(A \to C)\}$

→ entfernen T_0 von aktuelle Produktion und erhalten diese neue Produktion-Version ✌, dann T_{k+1} stückweise mithilfe dieser neue ✌ Version durchführen u. hinzufügen

Greibach-Normalform jede Produktion in $A \to aA_1 \ldots A_n$

Symbol $X \in V \cup \Sigma$

- erzeugend ⇔ $X \to_G^* w$ (vom Nichtterminal aus ein Wort produzieren)
- erreichbar ⇔ $S \to_G^* \alpha X \beta$ (von Startsymbol aus ein Nichtterminal produziert)
- nützlich ⇔ $S \to_G^* w$ (von Startsymbol zu einem Wort, in der das Nichtterminal verwendet)

nützlich Symbole sind erzeugend u. erreichbar, aber nicht notwendigerweise umgekehrt

S → AB|C A → aA|AS B → bS C → aCb|ε

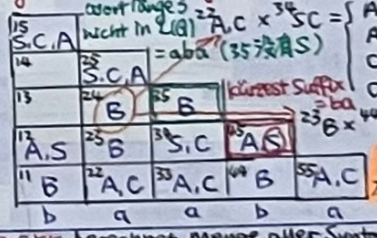
erzeugend: S,B,C erreichbar: S,A,B,C nützlich: S,C

Reduzieren G auf nützliche Nichtterminale

1) aus G eliminieren alle nicht erzeugenden Symbole 以及出现这些 Symbol 的 Produktion 都不 ⇒ 得到 G_1

2) aus G_1 eliminieren alle unerreichbare Symbole 以及出现这些 Symbol 的 Produktion 都不 ⇒ 得到最终 G nützlich

CYK Cocke-Younger-Kasami-Algo (entscheidet Wortproblem für CFL)



erweiterte CYK berechnet Menge aller Syntaxbäume

S → aT, T → Sb kann keine Terminal-Ableitungen erzeugen $L(G) = \varnothing$

$A \in NP \Leftrightarrow \exists$ polynomiell beschränkte Verifikator für A

1. Als Zertifikat definieren wir den gesuchten Object

2. Verifikator (Prüfung): um das Zertifikat zu verifizieren, prüfen wir die folgenden Eigenschaften in polynomieller Zeit: Bedingung 1 ∧ n auflisten u. jeweils prüfen.

Bedingung $|S| \geq k$ Prüfung: Wir zählen Element in S und vergleichen die Anzahl mit k.

S ist eine unabhängige Menge → Wir prüfen für jedes Knotenpaar (u,v) in S_i ob eine Kante (u,v) in E existiert erfordert einen einzigen Durchlauf durch Menge S, erfordert ein Suche in List der Elemente

3. Jeder dieser Prüfungen (zB. Schleifen über Eingabendaten, Zählen von Elemente, Überprüfung aller Knotenpaare) ist in polynomieller Zeit bezgl. der Eingabegröße durchführbar. Daher liegt A in NP.

P: Sprachen für $w \in L$ schnell entscheiden

NP: Sprachen, ein Zertifikat für $w \in L$ schnell verifiziert

PDA (nichtdeterministisch) Kellerautomat $M=(Q,\Sigma,\Gamma,q_0,Z_0,\delta,F)$ ≥ 2

a) akzeptieren mit Endzustand $L_F(M)=\{w \mid \exists f\in F, r\in\Gamma^*: (q_0,w,Z_0)\vdash_M^* (f,\varepsilon,r)\}$

b) akzeptieren mit leeren Keller $L_\varepsilon(M)=\{w \mid \exists q\in Q: (q_0,w,Z_0)\vdash_M^* (q,\varepsilon,\varepsilon)\}$ ≥ 1

push A $\delta(p,a,Z_0)=\{(p,AZ_0)\}$ $a, Z_0/AZ_0$ $|v|=|Q|\cdot|\Gamma|+1$

pop Z_0 $\delta(p,a,Z_0)=\{(p,\varepsilon)\}$ $a, Z_0/\varepsilon$ $L^\omega=\bigcup_{c\geq 0} L^{\leq c}\cup\begin{cases}\{\varepsilon\} & \text{wenn } w\in L\\ \varnothing & \text{sonst}\end{cases}$

ε-Übergang $\delta(p,\varepsilon,Z_0)=\{(q,Z_0)\}$ $p \xrightarrow{\varepsilon, Z_0/Z_0} q$ $L^b=\{a\}L^{ba}\cup\{b\}L^{bb}$

$a, Z_0/BBB$ Lauf: $(q,abbb,Z_0)\to(q,bbb,BBB)$ $=\varnothing\cup\{b\}L(a^*)=L(ba^*)$

$L(a^*)=\{\varepsilon,a,aa,aaa,\dots\}$

CFG $\Rightarrow$ PDA

① $S\to ABa$, $A\to Sa\mid a$, $B\to Sb\mid b$

② $S\to ABX_a$, $A\to SX_a\mid a$, $B\to SX_b\mid b$, $X_a\to a$, $X_b\to b$

③ $\delta(q_0,\varepsilon,S)=\{(q_0,ABX_a)\}$, $\delta(q_0,\varepsilon,A)=\{(q_0,SX_a)\}$, $\delta(q_0,\varepsilon,B)=\{(q_0,SX_b)\}$, $\delta(q_0,a,A)=\{(q_0,\varepsilon)\}$, $\delta(q_0,b,B)=\{(q_0,\varepsilon)\}$, $\delta(q_0,a,X_a)=\{(q_0,\varepsilon)\}$, $\delta(q_0,b,X_b)=\{(q_0,\varepsilon)\}$

PDA $M=(\{q,p\},\{0,1\},\{X,Z\},\delta,q,Z)$ $\Rightarrow$ CFG $G=(V,\Sigma,P,S)$

$V=\{S,[qXq],[qXp],[pXp],[pXq],[qZq],[qZp],[pZq],[pZp]\}$

$S\to[q_0Z_0p]$ for all states p $S\to[qZq]$, $S\to[qZp]$

$\delta(q,a,X)=(r,Y_1Y_2\dots Y_k)$

$[qXr_k]\to a[rY_1r_1][r_1Y_2r_2]\dots[r_{k-1}Y_kr_k]$ for all states $r_1,r_2,\dots,r_k$

$\delta(q,1,Z)=(q,XZ)$

$[qZq]\to 1[qXq][qZq]$ $[qZp]\to 1[qXq][qZp]$

$\to 1[qXp][pZq]$ $\to 1[qXp][pZp]$

$\delta(q,a,X)=(r,\varepsilon)\Rightarrow[qXr]\to a$ $\delta(q,\varepsilon,X)=((r),\varepsilon)\Rightarrow[qXr]\to\varepsilon$

Endlich Sprachen $\subset$ REG $\subset$ DCFL $\subset$ CFL

alle REG, alle DCFL, alle CFL abzählbar

DPDA deterministischer Kellerautomat

$|\delta(q,a,z)+\delta(q,\varepsilon,z)|\leq 1$ nur 1 Transition/Nachfolgekonfig

DCFL deterministisch CFL $\Leftrightarrow$ von DPDA akzeptiert (alle reguläre Sprache $\Rightarrow$ DCFL)

$\exists$ DPDA für L sodass $L_\varepsilon(M)\Leftrightarrow\exists$ DPDA für L sodass $L_F(M)$ u. L erfüllt Präfix-Bedingung

Abschlusseigenschaften REG $\cap$ CFL $\subseteq$ CFL REG $\cap$ CFL $\not\subseteq$ REG

	$\bar{L}$	$\cup$	$\cap$	$L_1\setminus L_2$	$L_1 L_2$	L^*	L^R	$h(L)$	$h^{-1}(L)$	L^a	$L_1\times L_2$
regulär	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CFL	×	✓	×	×	✓	✓	✓	✓	✓	✓	✓
DCFL	✓	×	×	×	×	×	×	×	✓	✓	×

Entscheidbarkeit

	Wortproblem	Leerheit	Äquivalenz	Schnittproblem
DFA	$O(n)$	✓	✓	✓
DPDA	$O(n)$	✓	✓	×
CFG	$O(n^3)$	✓	×	×

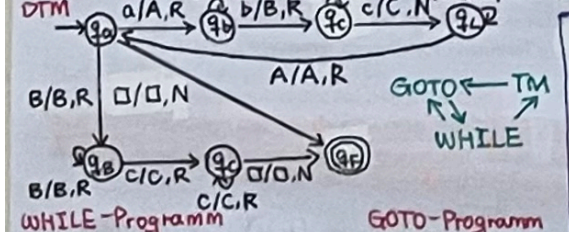
Turing-Maschine $M=\{Q,\Sigma,\Gamma,\delta,q_0,\square,F\}$

$\delta: Q\times\Gamma\to Q\times\Gamma\times\{L,R,N\}$

$\delta(q,a)\to(q',b,R)$ ersetzen a zu b, Zustand q zu q', rechts bewegen

$L(M)=\{a^nb^nc^n\mid n\in\mathbb{N}\}$ $M=(\{q_a,q_b,q_c,q_L,q_B,q_C,q_F\},\{a,b,c\},\Sigma\cup\{A,B,C,\square\},\delta,q_a,\square,\{q_F\})$

deterministisch



WHILE-Programm

P → X:=X+C
X:=X-C
P;P (sequentiell ausführen)
IF X=0 THEN P ELSE Q END
WHILE X≠0 DO P END

GOTO-Programm

$x_i:=x_j+n$
$x_i:=x_j-n$
GOTO M_i
IF $x_i=n$ GOTO M_j
HALT

Beweis, dass regulär: REGEX/NFA/DFA/ε-NFA konstruieren (begründen)

Pumping Lemma (beweisen nicht regulär)

$L=\{w\in\{0,1\}^*\mid w=w^R\}$ Wir führen einen Widerspruchsbeweis und nehmen an, dass L regulär wäre. Dann gilt die Aussage des Pumping-Lemmas für reguläre Sprachen. Sei n die PL-Zahl. Nun wählen wir Wort $z=0^n10^n\in L$ mit $|z|\geq n$. Nach PL existiert eine Zerlegung $z=uvw$ mit $|uv|\leq n$ und $v\neq\varepsilon$, sodass $uv^iw\in L$ für alle i gilt. Aus unserer Wahl folgt $uv=0^k$ für $0<k\leq n$. $u=0^i$, $v=0^j$ mit $i+j=k$ und $j>0$. Dann ist aber $uv^2w=0^i0^{2j}0^{n-k}1^n\notin L$ denn $i+2j+n-k=n+j>n$. Dies im Widerspruch zur Annahme, L nicht regulär

$L=\{w\in\{0,1\}^*\mid |w|_0\geq|w|_1\}$ $z=0^n1^n$, $u=0^i$, $v=0^j$ mit $j>0$

$uv^0w=0^i0^{n-i-j}1^n\notin L$, da $i+n-i-j=n-j<n$

$L=\{a^nb^na^mb^m\mid n,m\in\mathbb{N}\}$ $v=a^i$ mit $i>0$ $uv^0w=a^{n-i}b^na^nb^n\notin L$, da $n-i\neq n$

Nicht reguläre Sprachen nicht kontextfrei markiert 陈光震

$L=\{a^nb^n\mid n\geq 0\}$ $L=\{a^nb^m\mid n>m, n,m\in\mathbb{N}\}$ $L=\{a^nb^nc^n\mid n\geq 0\}$ $L=\{ab^nc^n\mid n\in\mathbb{N}+1\}$

$L=\{a^nb^mc^{n+m}\mid n,m\geq 0\}$ $L=\{a^ib^jc^k\mid i\leq j\leq k\}$ $L=\{a^nb^mc^\ell\mid n>m\wedge n>\ell\}$ $\{w^k\mid w\in L\}$

$L=\{ab^{2^i}cd^ie\mid i\in\mathbb{N}\}$ $L=\{w\in\{a,b\}^*\mid |w|_a=|w|_b\}$ $L=\{w\in\Sigma^*\mid |w|_a=z|w|_b\}$

$L=\{w\in\{0,1\}^*\mid |w|_0\geq|w|_1\}$ $L=\{ww\mid w\in\Sigma^*\}$ $L=\{w\in\Sigma^*\mid w=w^R\}$ $L=\{ww^R\mid w\in\{a,b\}^*\}$

$L=\{ww\mid w\in\{a,b\}^*\wedge w=w^R\}$ $L=\{a^{n^2}\mid n\geq 0\}$ $L=\{a^p\mid p \text{ primzahl}\}$ $L=\{a^{2^i}\mid i\geq 0\}$

$L=\{p\#w\mid p,w\in\Sigma^*\wedge p \text{ ist Präfix von } w\}$ $L=\{a^nb^na^mb^m\mid n,m\in\mathbb{N}\}$ $L=\{a^nb^ma^nb^m\mid n,m\in\mathbb{N}\}$

$L=\{a^nb^ma^nb^m\mid n,m\in\mathbb{N}\}$ $L=\{a^nb^mb^na^m\mid n,m\in\mathbb{N}\}$ $L=\{(ab)^n(ba)^n\mid n,m\in\mathbb{N}\}$

$L=\{a^ib^ja^k\mid (i=j\vee j=k)\wedge i,j,k\geq 0\}$ $L=\{a^ib^ja^k\mid j\geq i+k\wedge i,j,k\geq 0\}$ $\{a^nb^mc^m\mid n,m\in\mathbb{N}\}$

$L=\{a^ib^ja^k\mid j=\min(i,k)\}$ $L=\{a^{n_1}b^{n_2}a^{n_3}b^{n_4}\dots a^{n_{k-1}}b^{n_k}, k\in\mathbb{N}_+, n_1,\dots n_k\in\mathbb{N}, n_1<n_2<\dots<n_k\}$

Reguläre Sprachen $L=\Sigma^*$ $L=\varnothing$ $L=\{\varepsilon\}$ $L=\{a\}$ $L=a^*$ $L=a^+$ $L=\Sigma$

$L=\{a^{2i}\mid i\in\mathbb{N}\}$ $L=\{a^nb^n\mid n\leq 3\}$ $L=\{w\in\{a,b\}^*\mid |w|_a \text{ ist gerade}\}$

$L=\{w\in\{a,b\}^*\mid \text{jedes } a \text{ von ein } b \text{ gefolgt}\}$ $L=\{(a\mid b)^*\}$ $L=\{w\in\{a,b\}^*\mid w \text{ beginnt mit } a\}$

$L=\{w\in\{a,b\}^*\mid w \text{ endet mit } a\}$ $L=\{w\in\{a,b\}^*\mid |w|\leq 3\}$ $L=\{w\in\{a,b\}^*\mid |w| \text{ ist gerade}\}$

$L=\{w\in\{a,b\}^*\mid |w|=0 \pmod k\}$ $L=\{w\in\{a,b\}^*\mid w \text{ enthält } ab\}$ $L=\{a^n\mid n \text{gerade}\}$ $L=\{a^n\mid n\in\mathbb{N}^*\}$

Pumping-Lemma (beweisen nicht kontextfrei) $L=\{uv\mid u,v\in\Sigma^*\wedge v\neq\varepsilon\}$ $\{w\in L: |w|^3\leq 10|w|\}$

$\{w_1\dots w_k: w_1,\dots,w_k\in L\}$

$L=\{a^nb^mc^\ell\mid n,m,\ell\in\mathbb{N}\wedge n>m\wedge n>\ell\}$ Angenommen, L wäre kontextfrei. Dann können wir Pumping Lemma für kontextfreie Sprachen anwenden. Sei $n\in\mathbb{N}^+$ eine PL-Zahl für L. Es gilt $z=a^{n+1}b^nc^n\in L$ mit $|z|\geq n$. Es gibt für z eine Zerlegung $z=uvwxy$ mit ① $vx\neq\varepsilon$, ② $|vwx|\leq n$, ③ $\forall i\in\mathbb{N}_0: uv^iwx^iy\in L$. Fallunterscheidungen: falls $|vx|_a>0$. Aus ② und Wahl der z folgt $|vwx|_c=0$. Folglich liegt das Wort uv^0wx^0y nicht mehr in L, da es höchstens so viele a wie c enthält. Dies Widerspruch zu ③; falls $|vx|_a=0$, wegen ① gelten $|vx|_b>0$ oder $|vx|_c>0$. Folglich liegt Wort uv^3wx^3y nicht in L, da es entweder mehr b oder mehr c als a enthält. Dies Widerspruch zu ③, alle Fälle Widerspruch. L nicht kontextfrei

$L=\{ww\mid w\in\{a,b\}^*\wedge w=w^R\}$ $z=a^nb^na^{2n}b^na^n\in L$ mit $|z|\geq n$. Betrachten wir Wort uv^0wx^0y

Es hat eine der folgenden Formen, abhängig davon, wo vwx in z beginnt und endet. Es gilt $i>0$ oder $j>0$, $a^{n-i}b^{n-j}a^{2n}b^na^n$, $a^nb^{n-i}a^{2n-j}b^na^n$, $a^nb^na^{2n-i}b^{n-j}a^n$ $a^na^nb^na^{2n}b^{n-i}a^{n-j}$ Alle Fälle $uv^0wx^0y\notin L$ und Widerspruch zu ③. L nicht kontextfrei

$L(G)=L$ formal beweisen alle regulär, alle CFL, alle DCFL entscheidbar

$L(G)\subseteq L$ (jede von G generiertes Wort hat Form a^kb^k)

sei $w\in L(G)$, damit $S\to^*w$. Wir zeigen $\exists k\geq 0\ w=a^kb^k$ mittels Induktion über Ableitung von w

Induktionsbasis: $S\to\varepsilon=a^0b^0\in L$

Induktionsschritt: angenommen $S\to^n w$. Induktionshypothese gilt $\forall m\leq n\ \forall w': S\to^m w'\Rightarrow\exists k\geq 0\ w=a^kb^k$

Wir betrachten den ersten Schritt der Ableitung: Es gilt $S\to aSb\to^n aub$ mit $w=aub$ und $S\to^n u$

dann gibt es nach IH ein $k\geq 0$ mit $u=a^kb^k$, damit gilt $w=aa^kb^kb=a^{k+1}b^{k+1}$ mit $k+1\geq 0$. $w\in L$ bewiesen

$L\subseteq L(G)$ (jede Wort a^kb^k durch G generierbar)

sei $w\in L$, also $w=a^kb^k$ für $k\geq 0$. Wir zeigen $S\to^*a^kb^k$ per Induktion über k

Induktionsbasis: $w=a^0b^0=\varepsilon$ $S\to\varepsilon$ $w\in L(G)$

Induktionsschritt: von k nach k+1. Induktionshypothese $S\to^*a^kb^k$

Zu zeigen: $S\to^*a^{k+1}b^{k+1}$ Es gilt $S\to aSb\overset{IH}{\to^*}aa^kb^kb=a^{k+1}b^{k+1}$ somit $w\in L(G)$ bewiesen

Satisfiability Problem SAT ist NP-vollständig. für jedes Problem A in P gilt $A\leq_p$ SAT (alle Probleme in NP lassen sich auf NP-hart reduzieren)

Sprache nicht in NP: H_0, totale, nicht berechenbare Funktion; charakteristische Funktion von H_0, $H_0\cap\bar{H}_0$ sind unentscheidbar. P=NP folgt 3-SAT $\leq_p\{0^n1^n0^n\mid n\in\mathbb{N}\}$

	$\bar{L}$	$\cup$	$\cap$	$L_1\setminus L_2$	$L_1 L_2$	L^*	L^R	$h(L)$	$h^{-1}(L)$	L^a	$L_1\times L_2$	effektive $\bigcup_i L_i$	Projektion $\exists$ $\forall$
E	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	×	× ×
S-E	×	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓ ×
UN	✓	×	×	×	×	×	✓	×	×	×	×	×	× ×
P	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓		× ×
NP	?	✓	✓	?	✓	✓	✓	×	✓	✓	✓		✓ ×

Entscheidbarkeit: A entscheidbar, falls $\chi_A(x)=\begin{cases}1 & \text{falls } x\in A\\ 0 & \text{falls } x\notin A\end{cases}$ berechenbar ist

$f_{TM}(x)=\begin{cases}1 & \text{falls M auf allen Eingabe hält}\\ 0 & \text{sonst}\end{cases}$ für jede TM M berechenbar

A entscheidbar, existiert eine TM M mit $L(M)=A$ jede entscheidbar auch semi-entscheidbar

Sei M ein TM mit $L(M)=A$, gilt A ist berechenbar

$\{ww\mid w\in\Sigma^*\}$ $\{w\in\Sigma^*\mid g_w \text{ berechenbar}\}$ $\{w\in\Sigma^*\mid L(M_w) \text{ semi-entscheidbar}\}$

$g_w:\Sigma^*\to\Sigma^*$ von M_w berechnete Funktion $\{w\in\Sigma^*\mid g_w(v)=0^2\}$ $\{w\in\Sigma^*\mid g_w(w)=0^2\}$

Es gibt TM M, deren Halteproblem $\{w\mid M[w]_\downarrow\}$ entscheidbar $\Rightarrow$ M nie hält

Es gibt TM M, deren Halteproblem $\{w\mid M[w]_\downarrow\}$ unentscheidbar $\Rightarrow$ TM (PCP s-e)

$f:\mathbb{N}\to\mathbb{N}$, f berechenbar, all Sprachen in NTIME(f(n)) entscheidbar

$A=\{w\mid g_w=\chi_{H_0}\}$ $\{(G_1,G_2)\in K\times K\mid \min(L(G_1))\leq L(G_2)\}$

TM entscheidbar: in n Schritten hält, mindestens n-mal besucht

semi-entscheidbar = rekursive aufzählbar

A semi-entscheidbar gdw $\chi_A'(x)=\begin{cases}1 & \text{falls } x\in A\\ \perp \text{ hält nie falls } x\notin A\end{cases}$ berechenbar

A entscheidbar gdw A u. $\bar{A}$ semi-entscheidbar

$A\leq B$ und B semi-entscheidbar, dann auch A semi-entscheidbar

$\{w\in\Sigma^*\mid L(M_w)=\Sigma^*\}\cap L$ $\{w\in\Sigma^*\mid L(M_w)\cap L=L\}$ $\{w\in\Sigma^*\mid L(M_w)\cap L=\Sigma^*\}$

für $v\in\Sigma^*$ $L_v=\{w\in\Sigma^*: ww\in L(M_v)\}$, $\text{Graph}(\vec{f})=\{(w,f(w))\mid w\in\Sigma^*, f(w)\neq\perp\}$

L entscheidbar und M ein NTM, dann $L(M)\setminus L$ rekursiv aufzählbar

Wertebereich $\{y\in\Sigma^*\mid\exists x\in\Sigma^*\ y=f(x)\}$ einer berechenbare Funktion f

M ein DTM, sodass Eingabe $w\in\Sigma^*$ gibt, auf der M unendlich viele Schritte macht

$\hookrightarrow w\notin L(M)$, $L(M)$ immer semi-entscheidbar

Es gibt ein abzählbar unendliche Familie von entscheidbare Sprachen $\{L_i\}_{i=1}^\infty$ über demselben Alphabet, sodass $\bigcup_{i=1}^\infty L_i$ entscheidbar/unentscheidbar / nicht s-e ist

unentscheidbar nicht semi-entscheidbar $\Rightarrow$ unentscheidbar

allgemein Halteproblem $H=\{w\#w\in\{0,1\}^*\mid M_w[x]_\downarrow\}$ unentscheidbar

Halteproblem auf leeren Band $H_0=\{w\in\{0,1\}^*\mid M_w[\varepsilon]_\downarrow\}$ unentscheidbar

$\{w\in\Sigma^*\mid w\in L(M_w)\}$ $\{w\mid L(M_w)=\varnothing\}$ $\{w\in\Sigma^*\mid L(M_w) \text{ nicht regulär}\}$ $\{w\mid g_w \text{ total}\}$

$\{w\in\Sigma^*\mid |L(M_w)|\leq|w|\}$ $\{w\mid g_w \text{ Loop-berechenbar}\}$ ebenbar auch WHILE-berechbar

$\{w\mid g_w \text{ WHILE-berechenbar}\}$ entscheidbar (g_w immer berechenbar, alle berechenbar

TM M_1, TM M_2 $L(M_1)\subseteq L\subseteq L(M_2)$ dann L nicht s-e (⇒ unentscheidbar)

$L\subseteq\{0\}^*$ unentscheidbar. $f(n)=\min\{k\in\mathbb{N}: 0^k\in L\}$ berechenbar (BB-konstant)

$L\subseteq\{0,1\}^*$ entscheidbar. dann $L'=\{w\in\{0,1\}^*\mid L(M_w)=L\}$ unentscheidbar ⇒ unendlich

jede unentscheidbare Sprache hat ein unentscheidbar echte Teilmenge total berechenbar

Reduktion: Menge A^* auf Menge B^* reduzierbar gdw $\exists f:\Sigma^*\to\Gamma^*$ mit $\forall w\in\sigma^*: w\in A\Leftrightarrow f(w)\in B$. Schreibweise $A\leq B$ (Sei $A\leq B$ mittels f. dann $f^{-1}(B)=A$)

$A\leq B$ und B entscheidbar, A auch entscheidbar | B mindestens so schwer zu lösen wie A

$A\leq B$ und A nicht entscheidbar, B auch nicht entscheidbar

$\forall A\leq\Sigma^*: A\leq\Sigma^*$ false, $\forall A,B\leq\Sigma^*: A\leq B\Leftrightarrow\bar{A}\leq\bar{B}$ true, $\forall A,B,C\leq\Sigma^*: A\leq B\wedge B\leq C\Rightarrow A\leq C$ true

$\forall A,B\leq\Sigma^*, A\leq B\wedge B\leq A$, dann $A=B$ false, A reduzierbar auf B und B semi-entscheidbar, dann A auch s-e true

Satz von Rice: sei F eine Menge berechenbarer Funktionen. Es gelte weder $F=\varnothing$ noch $F=$ {"alle berechenbare Funktionen"}, also F nicht trivial. Dann ist unentscheidbar, ob die von einer gegebenen TM M_w berechnete Funktion Element ist, d.h. ob $g_w\in F$

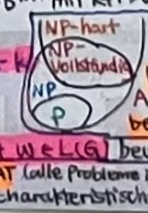
alle nicht-triviale semantische Eigenschaften von Programmen sind unentscheidbar

P genau von DTM in polynomieller Zeit lösbare Probleme, NP ... von NTM ... $P\subseteq NP$

alle endliche Sprache in P und NP, auch unendlich (aber nicht alle) in P und NP

alle reguläre, kontextfreie in P und NP, entscheidbar

semi-entscheidbar und unentscheidbar nicht in P und NP



Sprache L heißt NP-hart gdw $\forall A\in NP\ A\leq_p L$; NP-vollständig gdw $L\in NP$ und L ist NP-hart

NP-vollständige Probleme sind schwerste Probleme in NP, alle Problem in NP polynomiell reduzierbar

$P=NP\Leftrightarrow\exists$ NP-vollständiges Problem $\in P$) $\Sigma=\{a,b\}$ $L\leq\Sigma^*$. $L^a\in NP$, $L^b\in NP$, dann $L\in P$

A polynomiell reduzierbar auf B, $A\leq_p B$, gdw $f:\Sigma^*\to\Gamma^*$ gibt, mit f total, f von DTM in polynomieller zeit berechenbar, $\forall w\in\Sigma^*. w\in A\Leftrightarrow f(w)\in B$ Eigenschaften: $\leq_p$ transitiv; $A\leq_p B\wedge B\in P\Rightarrow A\in P$;

$A\leq_p B\wedge A\in NP\Rightarrow B\in NP$; $A\leq_p B$ gdw $\bar{A}\leq_p\bar{B}$. wenn $A\in NP$, dann $A\leq_p H$ allgem. HalteP

$A,B\leq\Sigma^*$ $A\leq_p SAT\leq_p B\Rightarrow A\in NP$, B NP-hart, $A\leq_p B$