Methodenblatt

1 Definitionen/Sätze/Lemma

Muss man formal korrekt wiedergeben können:

- Reguläre Ausdrücke
- Reguläre Pumping-Eigenschaft
- Reguläres Pumping-Lemma (mit Beweis)
- Rechtskongruenzrelation auf L ($\approx_L$)
- Rechtskongruenzrelation auf M ($\sim_M$)
- Satz von Myhill-Nerode (mit Beweis)
- Satz von Kleene
- Satz von Robin & Scott
- Kontextfreie Pumping-Eigenschaft
- Kontextfreies Pumping-Lemma

2 Reguläre Sprachen (REG)

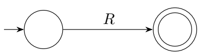
2.1 Beweise/Widerlege $L \in REG$

$L \in REG$	$L \notin REG$
• Regulären Ausdruck angeben	• keine reguläre Pumping-Eigenschaft (hat jede reguläre Sprache nach reg. Pumping-Lemma)
• NDEA konstruieren	• Unendlich viele Äquivalenzklassen bzgl. $\approx_L$ (reguläre Sprachen haben endlich viele ÄK nach Satz von Myhill-Nerode)

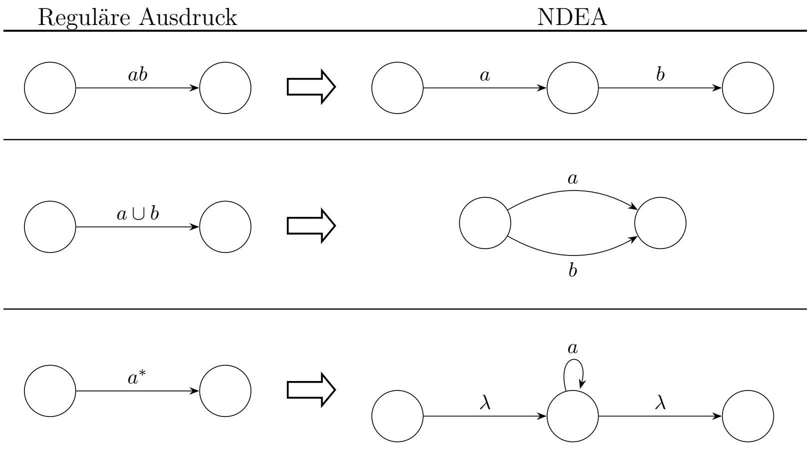
Abschluss: unter Vereinigung, Schnitt, Komplement, Differenz, Konkatenation, Kleene-Stern, Homomorphismus, Inverse Homomorphismus

2.2 NDEA $\leftrightarrow$ REG

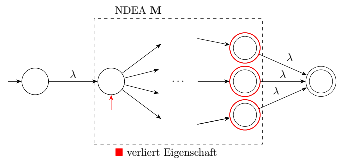
REG $\rightarrow$ NDEA



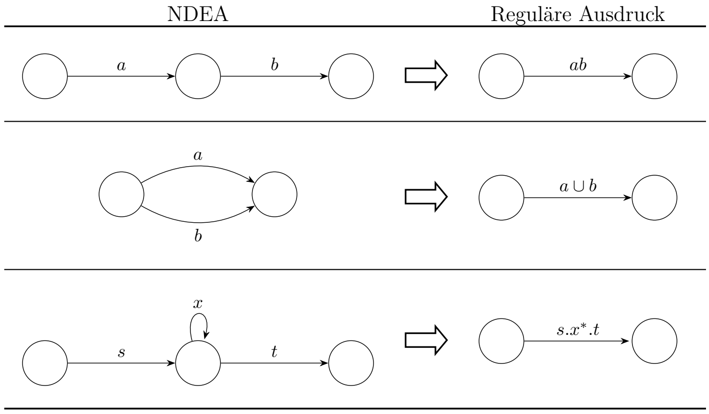
und dann



NDEA → REG



und dann



2.3 NDEA → DEA

Aus dem NDEA $M = (K, \Sigma, \Delta, s, F)$ konstruiert man den DEA (Potenzautomat) $M' = (K', \Sigma, \delta', s', F')$ mit:

- $K' = \mathcal{P}(K)$ (Potenzmenge)
- $s' = E(s) \in K'$
- $F' = \{Q \subset K \mid Q \cap F \neq \emptyset\}$
- $\delta'(Q, a) := \bigcup_{q \in Q} \{E(p) \mid p \in K \wedge (q, a, p) \in \Delta_M\}$

$E(q) := \{p \in K \mid (q, \lambda) \vdash_M^* (p, \lambda)\} \cup \{q\}$ (Alle von q mit λ-Übergänge erreichbaren Zustände)

2.4 2 DEA → DEA

Zum DEA $M_1 = (K_1, \Sigma, \delta, s_1, F_1)$ mit $L(M_1) = L_1$, und DEA $M_2 = (K_2, \Sigma, \delta, s_2, F_2)$ mit $L(M_2) = L_2$ konstruiert man den DEA (Produktautomat) $M = (K, \Sigma, \delta, s, F)$ mit

- $K = K_1 \times K_2$
- $s = (s_1, s_2)$
- $F = \begin{cases} F_1 \times F_2 & \text{für } L_1 \cap L_2 \\ K_1 \times F_2 \cup K_2 \times F_1 & \text{für } L_1 \cup L_2 \end{cases}$
- $\delta : ((q_1, q_2), a) \rightarrow (\delta_1(q_1, a), \delta_2(q_2, a))$

2.5 DEA minimieren

- Nicht erreichbare Zustände entfernen
- $\equiv_0$ bilden: F und $K \setminus F$
- $\equiv_{n+1}$ bilden: $p \equiv_{n+1} q \Leftrightarrow \begin{array}{l} 1)\ q \equiv_n p \\ 2)\ \forall a \in \Sigma : \delta(q, a) \equiv_n \delta(p, a) \end{array}$
- Gilt $\equiv_{n+1} = \equiv_n \rightarrow$ Fertig ($\equiv_n = \equiv_M$)
- Minimalen Zustandsautomaten konstruieren: Jede Klasse ein Zustand, Startzustand wird die Klasse mit dem ehemaligen Startzustand, Endzustände sind die Klassen,

die aus F hervorgingen, für die Transitionen einen Zustand q^* der Klasse wählen, und $\delta(q^*, a)\ \forall a \in \Sigma$ betrachten.

2.6 ÄK bzgl. $\sim_M \rightarrow$ REG

Zur Äquivalenzklasse $A = [a]$ bestimmt wie folgt den regulären Ausdruck:

- Zu M den minimalen Automaten M' bestimmen
- Bestimme Zustand $q \rightarrow$ Zustand wo die Rechnung von M' auf a endet
- Der reguläre Ausdruck α für A ist dann:

$$\alpha = \left(\bigcup\{\text{Pfad von } s \text{ zu } q\}\right) . \left(\bigcup\{\text{Zyklus von } q \text{ zu } q\}\right)^*$$

3 Kontextfreie Sprachen (CFL)

3.1 Beweise/Widerlege $L \in CFL$

$L \in CFL$

- kfr. Grammatik angeben
- PDA konstruieren
- regulären Ausdruck angeben ($REG \subset CFL$)
- NDEA konstruieren

$L \notin CFL$

- keine kfr. Pumping-Eigenschaft (hat jede kfr. Sprache nach kfr. Pumping-Lemma)

Abschluss: unter Vereinigung, Konkatenation, Kleene-Stern, Homomorphismus, Inverse Homomorphismus, Schnitt mit einer regulären Sprache

!kein! Abschluss: unter Schnitt, Komplement, Differenz

3.2 CFL $\leftrightarrow$ NDEA

CFL $\rightarrow$ NDEA

- $K := V$
- $s := S$
- $A \rightarrow wB$ wird zu $(A, w_1, A_1), \ldots, (A_{n-1}, w_n, B) \in \Delta$
- $A \rightarrow w$ wird zu $(A, w_1, A_1), \ldots,$ $(A_{n-1}, w_n, A_n) \in \Delta$ und $A_n \in F$

  (Für $A \rightarrow \lambda$ nur $A \in F$)

NDEA $\rightarrow$ CFL

- $V := K$
- $S := s$
- $\delta(q, a) = p$ wird zu $Q \rightarrow aP$
- für alle $q \in F$ wird $Q \rightarrow \lambda$ hinzugefügt

3.3 CFL $\leftrightarrow$ PDA

CFL $\rightarrow$ PDA

$$
\begin{aligned}
K :=& \{p, q\} \\
\Gamma :=& V \cup \Sigma \\
s :=& p \\
F :=& \{q\} \\
\Delta :=& \{ (p, \lambda, \lambda) \rightarrow (q, S) \\
& \quad (q, \lambda, A) \rightarrow (q, B) \quad \forall\, (A \rightarrow B) \in R \\
& \quad (q, a, a) \rightarrow (q, \lambda) \quad \forall\, a \in \Sigma \\
& \}
\end{aligned}
$$

PDA $\rightarrow$ CFL

1) $S \rightarrow [s, z_0, q] \quad \forall q \in K$
2) $[q, A, q_{m+1}] \rightarrow a[q_1, B_1, q_2][q_2, B_2, q_3] \ldots [q_{m-1}, B_{m-1}, q_m][q_m, B_m, q_{m+1}]$ für jede Transition $(q, a, A) \underset{M}{\longrightarrow} (q_1, B_1 \ldots B_m)$ in M und jede Wahl von $q_2, \ldots, q_{m+1} \in K$
3) $[q, A, q_1] \rightarrow a$ für $a \in \Sigma \cup \{\lambda\}$, falls $(q, a, A) \underset{M}{\longrightarrow} (q_1, \lambda)$ in M

3.4 CFL $\rightarrow$ CNF

1) Falls $\lambda \in L(G)$: füge $S_{neu} \rightarrow \lambda$ und $S_{neu} \rightarrow S$ hinzu

2) Entferne Regeln $B \rightarrow \lambda$, indem für alle Regeln mit B auf der rechten Seite, bspw. $A \rightarrow \alpha B \beta B \gamma$, zusätzlich die Regeln $A \rightarrow \alpha \beta B \gamma$, $A \rightarrow \alpha B \beta \gamma$ sowie $A \rightarrow \alpha \beta \gamma$ hinzugefügt werden, und $B \rightarrow \lambda$ entfernt wird

3) In Regeln $A \rightarrow w_1 w_2 \ldots w_k$ mit $k \geq 2$ ersetze alle $w_i \in \Sigma$ durch T_{w_i} (neue NTS) und füge die Regeln $T_{w_i} \rightarrow w_i$ hinzu

4) Elimination von $B \rightarrow C$ durch Untersuchung von Ableitungssequenzen von C aus:
   - $C \xrightarrow[G]{*} a \in \Sigma^*$: Füge $B \rightarrow a$ hinzu
   - $C \xrightarrow[G]{*} E_1 E_2 \ldots E_k$ $(k \geq 2)$: Füge $B \rightarrow E_1 E_2 \ldots E_k$ hinzu

   Lösche $B \rightarrow C$

5) Elimination von Regeln $A \rightarrow B_1 B_2 \ldots B_k$ $k \geq 3$ mit neuen NTS $N_1, \ldots, N_{k-2}$

$$
\begin{aligned}
A &\rightarrow B_1 N_1 \\
N_1 &\rightarrow B_2 N_2 \\
&\vdots \\
N_{k-2} &\rightarrow B_{k-1} B_k
\end{aligned}
$$

3.5 Prüfe $w \in L(G)$ (CYK-Alg.)

G muss in CNF sein!

- Erstelle eine Matrix $\mathcal{N}$ der Größe $n+1 \times n+1$ mit $n := |w|$, die Einträge sind Teilmengen von V mit:
$$A \in \mathcal{N}[i, j] \Leftrightarrow A \xrightarrow[G]{*} w_i w_{i+1} \ldots w_{j-1}, i < j$$
$\rightarrow$ alle Einträge unterhalb der Diagonalen (eingeschlossen) sind $\emptyset$ (da $i \geq j$)

- Berechne erste Nebendiagonale: $A \in \mathcal{N}[i, i+1] \overset{!}{\Leftrightarrow} A \underset{G}{\rightarrow} w_i$ muss Regel in G sein

- Berechne die restlichen Nebendiagonalen $\mathcal{N}[i,j]$ nacheinander ($j = i+2,\ldots,n+1$):

$$\mathcal{N}[i,j] = \bigcup_{k=i+1}^{j-1} \mathcal{N}[i,k] \odot \mathcal{N}[k,j]$$
$$\text{mit } M_1 \odot M_2 := \{A \in V \mid A \underset{G}{\rightarrow} BC \land B \in M_1, C \in M_2\}$$

- $w \in L(G) \Leftrightarrow S \in \mathcal{N}[1,n+1]$

4 Rekursiv aufzählbare Sprachen (RE)

$L \in RE$

- TM konstruieren, die L semi-entscheidet o. entscheidet
- Eine Funktion bzw. TM angeben, die L aufzählt
- Many-One Reduktion $L \leq_{mo} A$ angeben, wobei A rekursiv aufzählbar ist

$L \notin RE$

- Many-One Reduktion von $A \leq_{mo} L$ angeben, wobei A nicht rekursiv aufzählbar ist

Abschluss: unter Vereinigung, Schnitt, Konkatenation, Kleene-Stern, Homomorphismus, Inverse Homomorphismus

!kein! Abschluss: unter Komplement, Differenz

5 Entscheidbare Sprachen (REC)

$L \in REC$

- TM konstruieren, die L entscheidet
- TM konstruieren, die L semi-entscheidet, und eine zweite, die $\overline{L}$ semi-entscheidet
- Many-One Reduktion $L \leq_{mo} A$ angeben, wobei A entscheidbar ist

$L \notin REC$

- Many-One Reduktion $A \leq_{mo} L$ angeben, wobei A nicht entscheidbar ist (z.B. Halteproblem H)
- Zeigen, dass L oder $\overline{L}$ nicht rekursiv aufzählbar ist

Abschluss: unter Vereinigung, Schnitt, Komplement, Differenz, Konkatenation, Kleene-Stern, Homomorphismus, Inverse Homomorphismus