TECHNISCHE UNIVERSITÄT DRESDEN

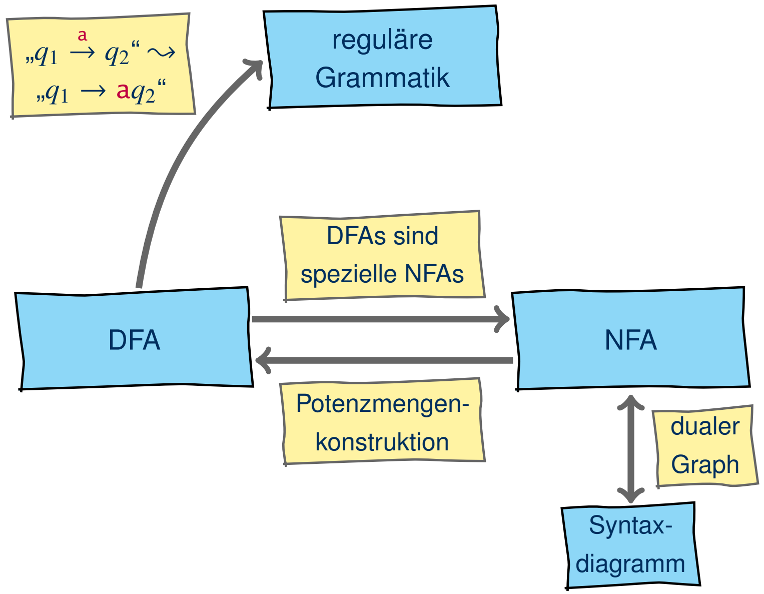
FORMALE SYSTEME

5. Vorlesung: Automaten & Grammatiken / Abschlusseigenschaften

Markus Krötzsch

TU Dresden, 24. Oktober 2016

Darstellungen von Typ-3-Sprachen



Von regulären Grammatiken zu NFAs

Satz: Die Klasse der Sprachen, die durch DFAs oder NFAs erkannt werden können, ist genau die Klasse der regulären Sprachen.

Beweis: Wir können nun die noch fehlende Richtung dieser Behauptung zeigen:

Für jede reguläre Grammatik G gibt es einen NFA $\mathcal{M}_G$, welcher die selbe Sprache akzeptiert (d.h., $\mathbf{L}(G) = \mathbf{L}(\mathcal{M}_G)$).

Für $G = \langle V, \Sigma, P, \mathsf{S}\rangle$ ergibt sich $\mathcal{M}_G = \langle Q, \Sigma, \delta, Q_0, F\rangle$ wie folgt:

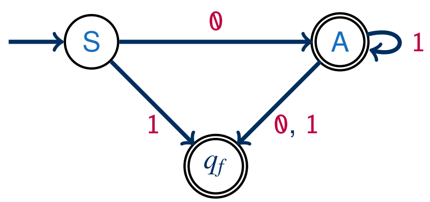
- $Q := V \cup \{q_f\}$
- $Q_0 := \{\mathsf{S}\}$
- $F := \{q_f\} \cup \{\mathsf{A} \in V \mid \mathsf{A} \to \epsilon \in P\}$
- $\delta(\mathsf{A}, \mathsf{c}) := \{\mathsf{B} \mid \mathsf{A} \to \mathsf{cB} \in P\} \cup \{q_f \mid \mathsf{A} \to \mathsf{c} \in P\}$

Beispiel

Wir betrachten eine reguläre Grammatik mit den folgenden sechs Regeln:

$$\mathsf{S} \to \mathtt{1} \mid \mathtt{0}\mathsf{A}$$
$$\mathsf{A} \to \mathtt{0} \mid \mathtt{1} \mid \mathtt{1}\mathsf{A} \mid \epsilon$$

Entsprechender NFA:



Dargestellte Sprache: $\{\mathtt{1}\} \cup (\{\mathtt{0}\} \circ \{\mathtt{1}\}^* \circ \{\epsilon, \mathtt{0}\})$

Korrektheit

Satz: Die Klasse der Sprachen, die durch DFAs oder NFAs erkannt werden können, ist genau die Klasse der regulären Sprachen.

Beweis: Wir behaupten, dass $\mathbf{L}(G) = \mathbf{L}(\mathcal{M}_G)$, d.h. für jedes Wort $w \in \Sigma^*$ soll gelten: $w \in \mathbf{L}(G)$ gdw. $w \in \mathbf{L}(\mathcal{M}_G)$.

Der Sonderfall $w = \epsilon$ ist ziemlich einfach:

$$\begin{aligned} \epsilon \in \mathbf{L}(G) \quad &\text{gdw.} \quad \mathsf{S} \to \epsilon \in P \\ &\text{gdw.} \quad \mathsf{S} \in F \\ &\text{gdw.} \quad \epsilon \in \mathbf{L}(\mathcal{M}_G) \end{aligned}$$

$\mathbf{L}(G) \subseteq \mathbf{L}(\mathcal{M}_G)$

Wir zeigen noch $w \in \mathbf{L}(G)$ gdw. $w \in \mathbf{L}(\mathcal{M}_G)$ für den Fall $|w| \geq 1$.

„⇒“ Angenommen $w \in \mathbf{L}(G)$ mit $w = \mathsf{a}_1 \cdots \mathsf{a}_n$ und $n \geq 1$.

Es gibt zwei mögliche Herleitungen für w:

(1) $\mathsf{S} \Rightarrow \mathsf{a}_1\mathsf{B}_1 \Rightarrow \ldots \Rightarrow \mathsf{a}_1 \cdots \mathsf{a}_{n-1}\mathsf{B}_{n\text{-}1} \Rightarrow \mathsf{a}_1 \cdots \mathsf{a}_{n-1}\mathsf{a}_n$

(2) $\mathsf{S} \Rightarrow \mathsf{a}_1\mathsf{B}_1 \Rightarrow \ldots \Rightarrow \mathsf{a}_1 \cdots \mathsf{a}_{n-1}\mathsf{B}_{n\text{-}1} \Rightarrow \mathsf{a}_1 \cdots \mathsf{a}_{n-1}\mathsf{a}_n\mathsf{B}_n \Rightarrow \mathsf{a}_1 \cdots \mathsf{a}_n$

In Fall (1) wurden Regeln der folgenden Form angewendet:

$$\mathsf{S} \to \mathsf{a}_1\mathsf{B}_1 \qquad \mathsf{B}_1 \to \mathsf{a}_2\mathsf{B}_2 \qquad \ldots \qquad \mathsf{B}_{n\text{-}1} \to \mathsf{a}_n$$

Also hat $\mathcal{M}_G$ die folgenden Übergänge:

$$\mathsf{S} \xrightarrow{\mathsf{a}_1} \mathsf{B}_1 \qquad \mathsf{B}_1 \xrightarrow{\mathsf{a}_2} \mathsf{B}_2 \qquad \ldots \qquad \mathsf{B}_{n\text{-}1} \xrightarrow{\mathsf{a}_n} q_f$$

Also ist $\mathsf{S}\mathsf{B}_1\mathsf{B}_2 \ldots \mathsf{B}_{n\text{-}1}q_f$ ein akzeptierender Lauf von $\mathcal{M}_G$ und $\mathcal{M}_G$ akzeptiert das Wort w.

Fall (2) ist ähnlich, wobei der Lauf auf B_n endet und $\mathsf{B}_n \in F$.

$\mathbf{L}(G) \supseteq \mathbf{L}(\mathcal{M}_G)$

Wir zeigen noch $w \in \mathbf{L}(G)$ gdw. $w \in \mathbf{L}(\mathcal{M}_G)$ für den Fall $|w| \geq 1$.

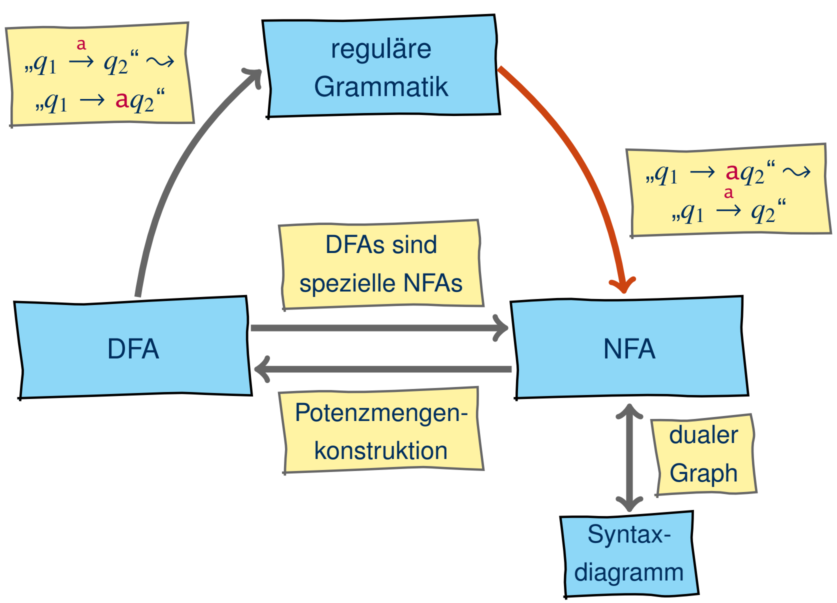
„⇐“ Angenommen $w \in \mathbf{L}(\mathcal{M}_G)$ mit $w = \sigma_1 \cdots \sigma_n$ und $n \geq 1$.

Beweis analog zur vorangegangenen Richtung; grob skizziert:

- w hat einen akzeptierenden Lauf in $\mathcal{M}_G$
- wir betrachten die möglichen Formen solcher Läufe
- in jedem Fall finden wir entsprechende NFA-Übergänge
- daraus ergeben sich geeignete Grammatikregeln, um w abzuleiten

□

Darstellungen von Typ-3-Sprachen



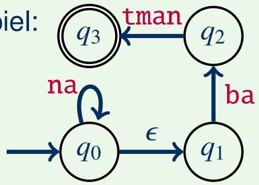
Automaten mit Wortübergängen

Ein nichtdeterministischer endlicher Automat mit Wortübergängen $\mathcal{M}$ ist ein Tupel $\mathcal{M} = \langle Q, \Sigma, \Delta, Q_0, F\rangle$ mit den folgenden Teilen:

- Q: endliche Menge von Zuständen
- Σ: Alphabet
- Δ: Übergangsrelation, eine endliche Relation $\Delta \subseteq Q \times \Sigma^* \times Q$
- Q_0: Menge möglicher Startzustände $Q_0 \subseteq Q$
- F: Menge von Endzuständen $F \subseteq Q$

Die Sprache solcher Automaten wird wie bei NFAs definiert, nur dass in einem Schritt beliebige Wörter gelesen werden können.

Beispiel:



$\Delta = \{\langle q_0, \mathtt{na}, q_0\rangle, \langle q_0, \epsilon, q_1\rangle, \langle q_1, \mathtt{ba}, q_2\rangle, \langle q_2, \mathtt{tman}, q_3\rangle\}$

Akzeptierte Sprache?: $\{\mathtt{na}\}^* \circ \{\mathtt{batman}\}$

Wortübergänge machen NFAs nicht stärker

Satz: Jede von einem NFA mit Wortübergängen akzeptierte Sprache, wird auch von einem normalen NFA akzeptiert.

Beweis: Wir eliminieren zunächst Wortübergänge für Wörter $w \neq \epsilon$.

Für jeden Übergang $q \xrightarrow{w} q'$ mit $w = \mathtt{a}_1 \cdots \mathtt{a}_n$ und $n \geq 2$:

- Führe $n-1$ neue Zustände $p_1, \ldots, p_{n-1}$ ein.
- Ersetze $q \xrightarrow{w} q'$ durch Übergänge $q \xrightarrow{\mathtt{a}_1} p_1, p_1 \xrightarrow{\mathtt{a}_2} p_2, \ldots, p_{n-1} \xrightarrow{\mathtt{a}_n} q'$.

Es ist leicht zu sehen, dass diese Umformung korrekt ist.

Dadurch erhalten wir einen NFA, in dem nur noch zwei Formen von Übergängen vorkommen: $q_1 \xrightarrow{\mathtt{a}} q_2$ und $q_1 \xrightarrow{\epsilon} q_2$.

ϵ-Übergänge machen NFA nicht stärker

Ein NFA mit Wortübergängen, bei dem alle Übergänge die Form $q_1 \xrightarrow{\mathtt{a}} q_2$ oder $q_1 \xrightarrow{\epsilon} q_2$ haben, wird ϵ-NFA genannt.

Um den vorigen Beweis abzuschließen müssen wir noch zeigen:

Satz: Jede von einem ϵ-NFA akzeptierte Sprache, wird auch von einem normalen NFA akzeptiert.

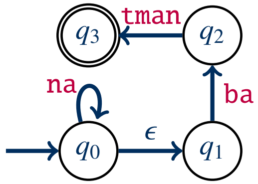
Beweis: Sei $\xrightarrow{\epsilon}^*$ der reflexive, transitive Abschluss von $\xrightarrow{\epsilon}$, d.h. die Menge aller Zustandspaare $\langle q, q'\rangle \in Q^2$ für die es Übergänge $q = p_0 \xrightarrow{\epsilon} p_1 \xrightarrow{\epsilon} \ldots \xrightarrow{\epsilon} p_n = q'$ gibt, mit $n \geq 0$.

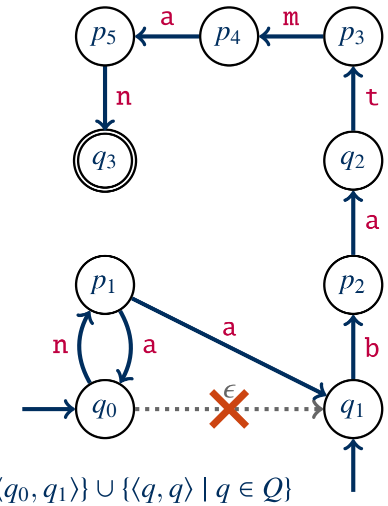
Für einen ϵ-NFA $\mathcal{M} = \langle Q, \Sigma, \Delta, Q_0, F\rangle$ definieren wir einen NFA $\mathcal{M}' = \langle Q, \Sigma, \delta, Q_0', F\rangle$ wobei

- $\delta(q, \mathtt{a}) = \{q' \mid q \xrightarrow{\mathtt{a}} r \xrightarrow{\epsilon}^* q' \text{ für ein } r \in Q\}$
- $Q_0' = \{q \mid q_0 \xrightarrow{\epsilon}^* q \text{ für ein } q_0 \in Q\}$

Wir behaupten $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}')$.

Beispiel: Eliminierung von Wortübergängen





$\xrightarrow{\epsilon}^* = \{\langle q_0, q_1\rangle\} \cup \{\langle q, q\rangle \mid q \in Q\}$

Korrektheit ϵ-NFA $\rightsquigarrow$ NFA (1)

Wir behaupten $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}')$.

Beweis: Richtung $\mathbf{L}(\mathcal{M}) \supseteq \mathbf{L}(\mathcal{M}')$.

- Sei $w = \mathtt{a}_1 \cdots \mathtt{a}_n \in \mathbf{L}(\mathcal{M}')$ und $p_0 p_1 \ldots p_n$ ein akzeptierender Lauf für w in $\mathcal{M}'$
- Dann gibt es in $\mathcal{M}'$ Übergänge:

$$p_0 \xrightarrow{\mathtt{a}_1} p_1 \xrightarrow{\mathtt{a}_2} \ldots \xrightarrow{\mathtt{a}_n} p_n$$

- Dann gilt in $\mathcal{M}$:

$$q_0 \xrightarrow{\epsilon}{}^{*} p_0 \xrightarrow{\mathtt{a}_1} q_1 \xrightarrow{\epsilon}{}^{*} p_1 \xrightarrow{\mathtt{a}_2} \ldots \xrightarrow{\mathtt{a}_n} q_n \xrightarrow{\epsilon}{}^{*} p_n$$

- Es ist $q_0 \in Q_0$ (da $p_0 \in Q_0'$) und $p_n \in F$, also hat $\mathcal{M}$ einen akzeptierenden Lauf der Form $q_0 \ldots p_0 q_1 \ldots p_1 \ldots q_n \ldots p_n$.

Also ist $w \in \mathbf{L}(\mathcal{M})$.

Korrektheit ϵ-NFA $\rightsquigarrow$ NFA (2)

Wir behaupten $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}')$.

Beweis: Richtung $\mathbf{L}(\mathcal{M}) \subseteq \mathbf{L}(\mathcal{M}')$.

- Sei $w = \mathtt{a}_1 \cdots \mathtt{a}_n \in \mathbf{L}(\mathcal{M})$.
- Dann gibt es einen akzeptierenden Lauf mit Übergängen der Form

$$q_0 \xrightarrow{\epsilon}{}^{*} p_0 \xrightarrow{\mathtt{a}_1} q_1 \xrightarrow{\epsilon}{}^{*} p_1 \xrightarrow{\mathtt{a}_2} \ldots \xrightarrow{\mathtt{a}_n} q_n \xrightarrow{\epsilon}{}^{*} p_n$$

- Dann gibt es in $\mathcal{M}'$ Übergänge:

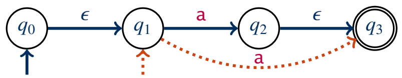
$$p_0 \xrightarrow{\mathtt{a}_1} p_1 \xrightarrow{\mathtt{a}_2} \ldots \xrightarrow{\mathtt{a}_n} p_n$$

- Dies führt zu einem akzeptierenden Lauf in $\mathcal{M}'$.

Also ist $w \in \mathbf{L}(\mathcal{M}')$. $\square$

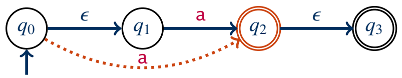
ϵ-NFA: Variationen

Die Konstruktion im Beweis „verlängert“ normale Übergänge „nach rechts“ durch Anhängen von ϵ-Transitionen.

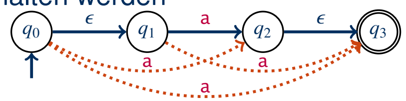


Man kann alternative Konstruktionen für den Beweis angeben:

- „Verlängerung nach links“: ϵ-Transitionen vor normalen Übergängen; Anfangszustände werden beibehalten; dafür werden Endzustände mit ϵ-Transitionen erweitert



- „Verlängerung in beide Richtungen“: ϵ-Transitionen vor und nach normalen Übergängen; Anfangs- und Endzustände können beibehalten werden*



*) Ausnahme: Anfangszustände mit ϵ-Pfad zu einem Endzustand werden Endzustand

Abschlusseigenschaften

Abschlusseigenschaften regulärer Sprachen

Reguläre Sprachen sind unter zahlreichen Operationen abgeschlossen:

Satz: Wenn **L**, $\mathbf{L_1}$ und $\mathbf{L_2}$ reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

(1) $\mathbf{L_1} \cup \mathbf{L_2}$ (Abschluss unter Vereinigung)

(2) $\mathbf{L_1} \cap \mathbf{L_2}$ (Abschluss unter Schnitt)

(3) $\overline{\mathbf{L}}$ (Abschluss unter Komplement)

(4) $\mathbf{L_1} \circ \mathbf{L_2}$ (Abschluss unter Konkatenation)

(5) $\mathbf{L}^*$ (Abschluss unter Kleene-Stern)

Beweisidee: Für jede Operation auf Sprachen entwickeln wir eine entsprechende Operation auf Automaten. Dadurch konstruieren wir Automaten, welche die gesuchte Sprache erkennen (daher muss die Sprache regulär sein).

Vereinigung von NFAs

Ein NFA zur Vereinigung von zwei NFAs ist leicht zu finden:

Gegeben seien zwei NFAs $\mathcal{M}_1 = \langle Q_1, \Sigma, \delta_1, Q_{0,1}, F_1\rangle$ und $\mathcal{M}_2 = \langle Q_2, \Sigma, \delta_2, Q_{0,2}, F_2\rangle$ mit $Q_1 \cap Q_2 = \emptyset$ (o.B.d.A.).

Der Vereinigungsautomat $\mathcal{M}_1 \oplus \mathcal{M}_2$ ist gegeben durch $\langle Q_1 \cup Q_2, \Sigma, \delta_{12}, Q_{0,1} \cup Q_{0,2}, F_1 \cup F_2\rangle$ wobei

$$\delta_{12}(q, \mathtt{a}) = \begin{cases} \delta_1(q, \mathtt{a}) & \text{falls } q \in Q_1 \\ \delta_2(q, \mathtt{a}) & \text{falls } q \in Q_2 \end{cases}$$

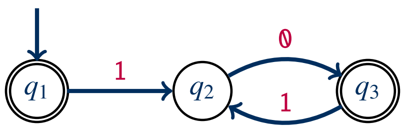
Das folgende Ergebnis ist leicht zu sehen:

Satz: $\mathbf{L}(\mathcal{M}_1 \oplus \mathcal{M}_2) = \mathbf{L}(\mathcal{M}_1) \cup \mathbf{L}(\mathcal{M}_2)$.

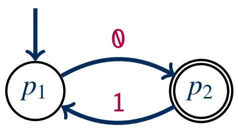
Also sind reguläre Sprachen unter Vereinigung abgeschlossen.

Beispiel

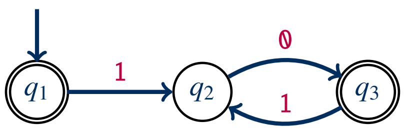
Die Vereinigung der NFAs

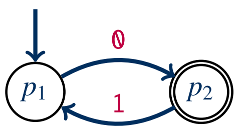


und



ergibt den NFA ...





Akzeptierte Sprache: $\{\mathtt{10}\}^* \cup (\{\mathtt{01}\}^* \circ \{\mathtt{0}\}) = \{\epsilon, \mathtt{0}\} \circ \{\mathtt{10}\}^*$

Abschluss unter Schnitt

Satz: Wenn **L**, $\mathbf{L_1}$ und $\mathbf{L_2}$ reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

(1) $\mathbf{L_1} \cup \mathbf{L_2}$ (Abschluss unter Vereinigung)

(2) $\mathbf{L_1} \cap \mathbf{L_2}$ Abschluss unter Schnitt

(3) $\overline{\mathbf{L}}$ (Abschluss unter Komplement)

(4) $\mathbf{L_1} \circ \mathbf{L_2}$ (Abschluss unter Konkatenation)

(5) $\mathbf{L}^*$ (Abschluss unter Kleene-Stern)

Schnitte mit Automaten

Gegeben: Automaten $\mathcal{M}_1$ und $\mathcal{M}_2$

Gesucht: Automat $\mathcal{M}$ mit $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}_1) \cap \mathbf{L}(\mathcal{M}_2)$

Idee:

- Verarbeite Eingabe gleichzeitig synchron auf beiden Automaten
- Akzeptiere, wenn beide Automaten akzeptieren

$\rightsquigarrow$ Produktautomat

Der Produktautomat

Erinnerung: Gegeben Mengen A und B bezeichnet $A \times B$ die Menge aller möglichen Paare von Elementen aus A und B, d.h. $A \times B = \{\langle a, b\rangle \mid a \in A, b \in B\}$.

Gegeben seien zwei NFAs $\mathcal{M}_1 = \langle Q_1, \Sigma, \delta_1, Q_{0,1}, F_1\rangle$ und $\mathcal{M}_2 = \langle Q_2, \Sigma, \delta_2, Q_{0,2}, F_2\rangle$.

Der Produktautomat $\mathcal{M}_1 \otimes \mathcal{M}_2$ ist gegeben durch $\langle Q_1 \times Q_2, \Sigma, \delta, Q_{0,1} \times Q_{0,2}, F_1 \times F_2\rangle$ wobei

$$\delta(\langle q_1, q_2\rangle, \mathtt{a}) = \delta_1(q_1, \mathtt{a}) \times \delta_2(q_2, \mathtt{a})$$

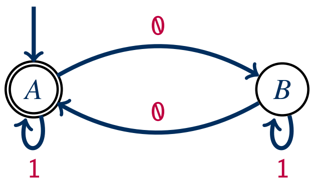
Wir werden zeigen, dass dies der gesuchte Automat ist:

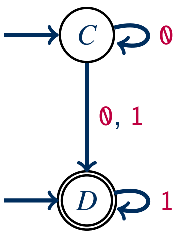
Satz: $\mathbf{L}(\mathcal{M}_1 \otimes \mathcal{M}_2) = \mathbf{L}(\mathcal{M}_1) \cap \mathbf{L}(\mathcal{M}_2)$.

Also sind reguläre Sprachen unter Schnitten abgeschlossen.

Beobachtung: Wenn $|A| = |B| = 1$, dann ist $|A \times B| = 1$. Also ist der Produktautomat von DFAs ebenfalls ein DFA.

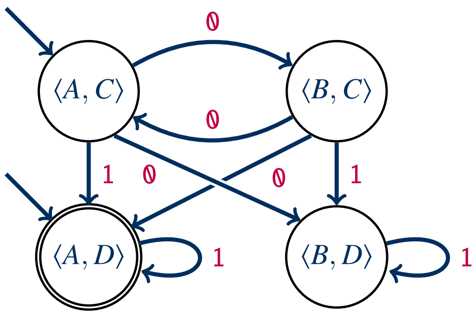
Beispiel





Sprachen?

„Wörter mit gerader Anzahl 0“ „Wörter ohne Infix 10“



Beweis der Korrektheit des Produktautomaten

„$\mathbf{L}(\mathcal{M}_1 \otimes \mathcal{M}_2) \subseteq \mathbf{L}(\mathcal{M}_1) \cap \mathbf{L}(\mathcal{M}_2)$“ Sei $w = \mathtt{a}_1 \cdots \mathtt{a}_n \in \mathbf{L}(\mathcal{M}_1 \otimes \mathcal{M}_2)$.

- Also gibt es einen akzeptierenden Lauf $\langle q_1^0, q_2^0\rangle\langle q_1^1, q_2^1\rangle \ldots \langle q_1^n, q_2^n\rangle$
- Dann gilt:
  - $\langle q_1^0, q_2^0\rangle \in Q_{0,1} \times Q_{0,2}$, also $q_1^0 \in Q_{0,1}$ und $q_2^0 \in Q_{0,2}$
  - $\langle q_1^n, q_2^n\rangle \in F_1 \times F_2$, also $q_1^n \in F_1$ und $q_2^n \in F_2$
  - $\langle q_1^i, q_2^i\rangle \in \delta(\langle q_1^{i-1}, q_2^{i-1}\rangle, \mathtt{a}_i)$, also $q_1^i \in \delta_1(q_1^{i-1}, \mathtt{a}_i)$ und $q_2^i \in \delta_2(q_2^{i-1}, \mathtt{a}_i)$
- Daher sind $q_1^0 q_1^1 \ldots q_1^n$ und $q_2^0 q_2^1 \ldots q_2^n$ akzeptierende Läufe von $\mathcal{M}_1$ und $\mathcal{M}_1$.

Also ist $w \in \mathbf{L}(\mathcal{M}_1)$ und $w \in \mathbf{L}(\mathcal{M}_2)$.

„$\mathbf{L}(\mathcal{M}_1 \otimes \mathcal{M}_2) \supseteq \mathbf{L}(\mathcal{M}_1) \cap \mathbf{L}(\mathcal{M}_2)$“ Analoge Schlussfolgerungen in entgegengesetzter Richtung. $\square$

Abschluss unter Komplement

Satz: Wenn $\mathbf{L}$, $\mathbf{L_1}$ und $\mathbf{L_2}$ reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

(1) $\mathbf{L_1} \cup \mathbf{L_2}$ (Abschluss unter Vereinigung)

(2) $\mathbf{L_1} \cap \mathbf{L_2}$ (Abschluss unter Schnitt)

(3) $\overline{\mathbf{L}}$ Abschluss unter Komplement

(4) $\mathbf{L_1} \circ \mathbf{L_2}$ (Abschluss unter Konkatenation)

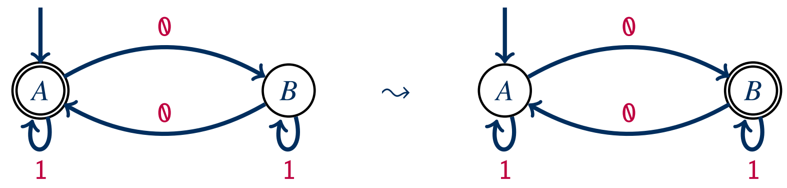
(5) $\mathbf{L}^*$ (Abschluss unter Kleene-Stern)

Komplementoperator für DFAs

Idee: Wir können DFA komplementieren, indem wir akzeptierende und verwerfende Zustände vertauschen.

Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F\rangle$ sei $\overline{\mathcal{M}}$ der DFA $\langle Q, \Sigma, \delta, q_0, Q \setminus F\rangle$.

Beispiel:



„Wörter mit gerader Anzahl $\mathtt{0}$" „Wörter mit ungerader Anzahl $\mathtt{0}$"

Korrektheitsbeweis Komplementierung

Satz: Für jeden DFA gilt $\mathbf{L}(\overline{\mathcal{M}}) = \overline{\mathbf{L}(\mathcal{M})}$.

Beweis: „$\mathbf{L}(\overline{\mathcal{M}}) \subseteq \overline{\mathbf{L}(\mathcal{M})}$" Sei $w = \mathtt{a}_1 \cdots \mathtt{a}_n \in \mathbf{L}(\overline{\mathcal{M}})$.

- Dann gibt es einen akzeptierenden Lauf $q_0 q_1 \ldots q_n$ für w in $\overline{\mathcal{M}}$.
- Dann ist $q_n \in Q \setminus F$.
- Dann ist $q_0 q_1 \ldots q_n$ ein verwerfender Lauf für w in $\mathcal{M}$.

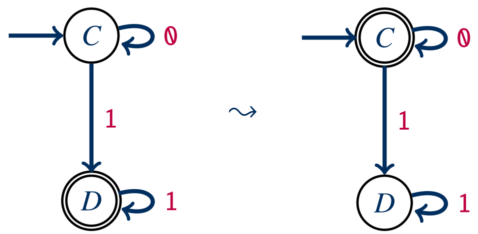
Also ist $w \notin \mathbf{L}(\mathcal{M})$, d.h. $w \in \overline{\mathbf{L}(\mathcal{M})}$.

„$\mathbf{L}(\overline{\mathcal{M}}) \supseteq \overline{\mathbf{L}(\mathcal{M})}$" Sei $w = \mathtt{a}_1 \cdots \mathtt{a}_n \in \overline{\mathbf{L}(\mathcal{M})}$

- Dann hat $\mathcal{M}$ einen verwerfenden Lauf $p_0 p_1 \ldots p_m$ für w
- Dann ist (1) $p_m \notin F$ oder (2) $m < |w|$
- In Fall (1) ist $p_0 p_1 \ldots p_m$ ein akzeptierender Lauf für w in $\overline{\mathcal{M}}$
- In Fall (2) . . . ?

Der Satz ist falsch!

Gegenbeispiel



Wörter der Form $\mathtt{0}^*\mathtt{1}^+$ Wörter der Form $\mathtt{0}^*$

Das Wort $\mathtt{010}$ zum Beispiel wird von keinem der beiden Automaten erkannt.

⇝ keine komplementären Sprachen

Korrekte Komplementierung

Satz: Für jeden DFA mit totaler Übergangsfunktion gilt $\mathbf{L}(\overline{\mathcal{M}}) = \overline{\mathbf{L}(\mathcal{M})}$.

Beweis: Wie vorher für den falschen Satz, aber diesmal kann der Problemfall (2) der Rückrichtung nicht auftreten.

Daraus folgt der Abschluss regulärer Sprachen unter Komplement, da wir jeden DFA mit totalen Übergängen ausstatten können.

Auch NFAs dürfen nicht direkt komplementiert werden:

- DFAs sind NFAs: unser Gegenbeispiel trifft weiterhin zu
- Selbst NFAs, in denen jeder Zustand für jedes Symbol mindestens einen Folgezustand hat, können nicht so einfach komplementiert werden (Übung)

Abschluss unter Konkatenation

Satz: Wenn $\mathbf{L}$, $\mathbf{L_1}$ und $\mathbf{L_2}$ reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

(1) $\mathbf{L_1} \cup \mathbf{L_2}$ (Abschluss unter Vereinigung)

(2) $\mathbf{L_1} \cap \mathbf{L_2}$ (Abschluss unter Schnitt)

(3) $\overline{\mathbf{L}}$ (Abschluss unter Komplement)

(4) $\mathbf{L_1} \circ \mathbf{L_2}$ Abschluss unter Konkatenation

(5) $\mathbf{L}^*$ (Abschluss unter Kleene-Stern)

Konkatenation von NFAs

Idee: Wir können Automaten hintereinander hängen, indem wir von Endzuständen des ersten zu Startzuständen des zweiten wechseln.
$\rightsquigarrow$ besonders elegant geht das mit ϵ-Transitionen

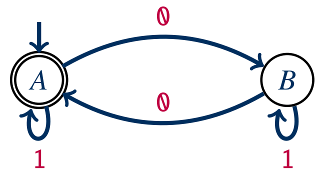
Gegeben seien zwei NFAs $\mathcal{M}_1 = \langle Q_1, \Sigma, \delta_1, Q_{0,1}, F_1\rangle$ und $\mathcal{M}_2 = \langle Q_2, \Sigma, \delta_2, Q_{0,2}, F_2\rangle$ mit $Q_1 \cap Q_2 = \emptyset$ (o.B.d.A.).

Der Konkatenationsautomat $\mathcal{M}_1 \odot \mathcal{M}_2$ ist ein ϵ-NFA gegeben durch $\langle Q_1 \cup Q_2, \Sigma, \delta, Q_{0,1}, F_2\rangle$ wobei

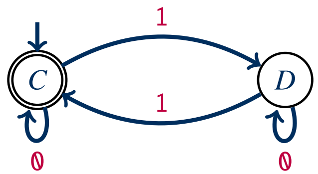
$$\delta(q, \mathtt{a}) = \begin{cases} \delta_1(q, \mathtt{a}) & \text{falls } q \in Q_1 \\ \delta_2(q, \mathtt{a}) & \text{falls } q \in Q_2 \end{cases} \qquad \delta(q, \epsilon) = \begin{cases} Q_{0,2} & \text{falls } q \in F_1 \\ \emptyset & \text{andernfalls} \end{cases}$$

$\mathcal{M}_1 \odot \mathcal{M}_2$ simuliert also zunächst $\mathcal{M}_1$. In jedem Endzustand aus F_1 entscheidet $\mathcal{M}_1 \odot \mathcal{M}_2$ nichtdeterministisch, diese Simulation fortzusetzen oder mit der Simulation von $\mathcal{M}_2$ zu beginnen.

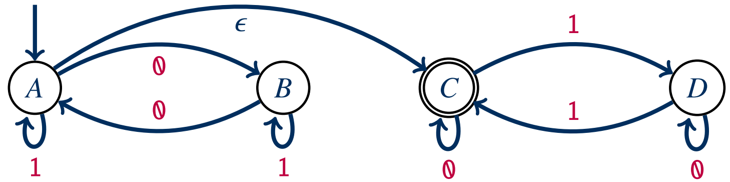
Beispiel Konkatenation



„Wörter mit gerader Anzahl 0“



„Wörter mit gerader Anzahl 1“



(Übung: Welche Wörter akzeptiert dieser Automat nicht?)

Satz: Für alle NFA $\mathcal{M}_1$ und $\mathcal{M}_2$ gilt $\mathbf{L}(\mathcal{M}_1 \odot \mathcal{M}_2) = \mathbf{L}(\mathcal{M}_1) \circ \mathbf{L}(\mathcal{M}_2)$.

Der Beweis ist einfach und analog zu den bisher gezeigten.

Abschluss unter Kleene-Stern

Satz: Wenn **L**, **L₁** und **L₂** reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

(1) $\mathbf{L_1} \cup \mathbf{L_2}$ (Abschluss unter Vereinigung)

(2) $\mathbf{L_1} \cap \mathbf{L_2}$ (Abschluss unter Schnitt)

(3) $\overline{\mathbf{L}}$ (Abschluss unter Komplement)

(4) $\mathbf{L_1} \circ \mathbf{L_2}$ (Abschluss unter Konkatenation)

(5) $\mathbf{L}^*$ Abschluss unter Kleene-Stern

Abschluss unter Kleene-Stern

Idee: Der Kleene-Stern ist eine verallgemeinerte Konkatenation, bei der ein Automat rekursiv hinter sich selbst gehängt wird.

Gegeben sei ein NFA $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F\rangle$. Der Automat $\mathcal{M}^*$ ist ein ϵ-NFA gegeben durch $\langle Q', \Sigma, \delta', Q'_0, F'\rangle$ wobei

- $Q' = Q \cup \{q_\epsilon\}$ (wobei $q_\epsilon \notin Q$ o.B.d.A.)
- $\delta'(q, \mathtt{a}) = \delta(q, \mathtt{a})$ für alle $q \in Q$, $\mathtt{a} \in \Sigma$ und $\delta'(q_f, \epsilon) = Q_0$ für alle $q_f \in F$
- $Q'_0 = Q_0 \cup \{q_\epsilon\}$
- $F' = F \cup \{q_\epsilon\}$

Satz: $\mathbf{L}(\mathcal{M}^*) = \mathbf{L}(\mathcal{M})^*$.

Der Beweis ist einfach und analog zu den bisher gezeigten.

Zusammenfassung und Ausblick

NFAs, DFAs, ϵ-NFAs, und NFAs mit Wortübergängen beschreiben die selbe Klasse der regulären Sprachen.

Reguläre Sprachen sind abgeschlossen unter $\cap$, $\cup$, -, $\circ$, * und allen davon ableitbaren Operatoren.

Den Sprachoperationen entsprechen Operationen auf Automaten. Manche erfordern bestimmte Typen von Automaten.

Offene Fragen:

- Gibt es noch mehr Darstellungsformen für reguläre Sprachen?
- Welche Sprachen sind nicht regulär?
- Wir haben gesehen, dass man Automaten manchmal vereinfachen kann – geht das noch besser?