



FORMALE SYSTEME

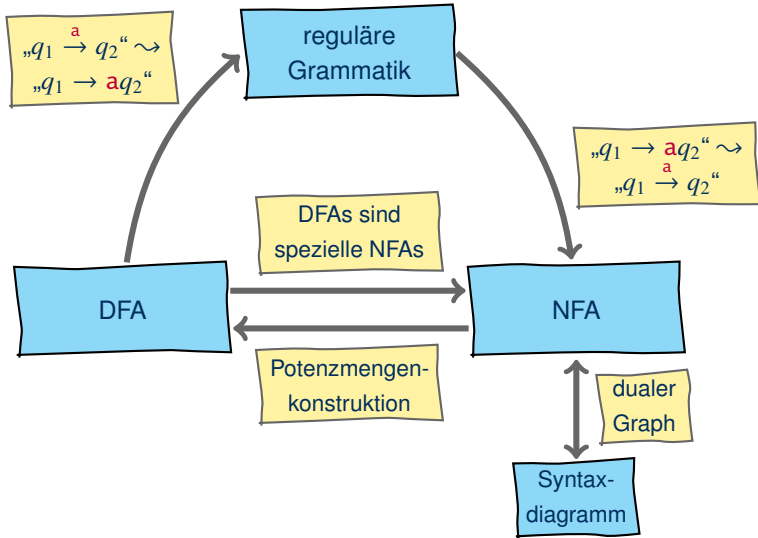
5. Vorlesung: Abschlusseigenschaften

Markus Krötzsch

Professur für Wissensbasierte Systeme

TU Dresden, 23. Oktober 2023

Darstellungen von Typ-3-Sprachen

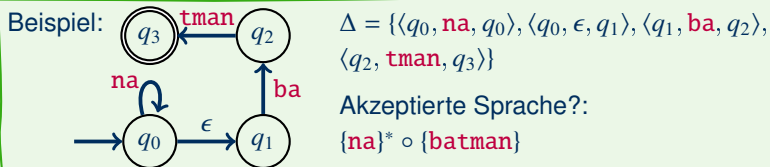


Automaten mit Wortübergängen

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.



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 $w = a_1 \cdots a_n$ und $n \geq 2$:

- Führe $n - 1$ neue Zustände p_1, \dots, p_{n-1} ein.
- Ersetze $q \xrightarrow{w} q'$ durch Übergänge $q \xrightarrow{a_1} p_1, p_1 \xrightarrow{a_2} p_2, \dots, p_{n-1} \xrightarrow{a_n} q'$.

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

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

ϵ -Übergänge machen NFA nicht stärker

Ein NFA mit Wortübergängen, bei dem alle Übergänge die Form $q_1 \xrightarrow{a} q_2$ 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.

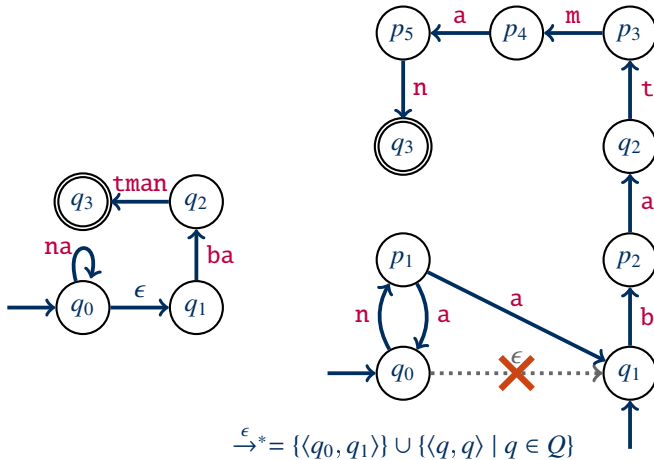
Beweis: Sei $\xrightarrow{\epsilon}^*$ der reflexive, transitive Abschluss von $\xrightarrow{\epsilon}$, d.h. die Menge aller Zustandspaare $\langle q, q' \rangle \in Q^2$ für die es Übergänge $q = p_0 \xrightarrow{\epsilon} p_1 \xrightarrow{\epsilon} \dots \xrightarrow{\epsilon} p_n = q'$ gibt, mit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

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

- $\delta(q, a) = \{q' \mid q \xrightarrow{a} r \xrightarrow{\epsilon}^* q' \text{ für ein } r \in Q\}$
- $Q'_0 = \{q \mid q_0 \xrightarrow{\epsilon}^* q \text{ für ein } q_0 \in Q_0\}$

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

Beispiel: Eliminierung von Wortübergängen



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

$$p_0 \xrightarrow{\mathbf{a}_1} p_1 \xrightarrow{\mathbf{a}_2} \dots \xrightarrow{\mathbf{a}_n} p_n$$

- Dann gilt in \mathcal{M} :

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

- 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 $q_0 \dots p_0 q_1 \dots p_1 \dots q_n \dots p_n$.

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 $w = a_1 \cdots a_n \in \mathbf{L}(\mathcal{M})$.
- Dann gibt es einen akzeptierenden Lauf mit Übergängen der Form

$$q_0 \xrightarrow{\epsilon^*} p_0 \xrightarrow{a_1} q_1 \xrightarrow{\epsilon^*} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_n \xrightarrow{\epsilon^*} p_n$$

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

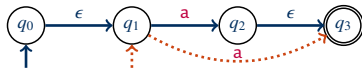
$$p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} p_n$$

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

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

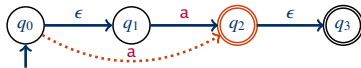
ϵ -NFA: Variationen

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

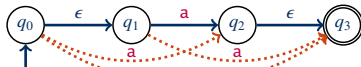


Man kann alternative Konstruktionen für den Beweis angeben:

- „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 , L_1 und L_2 reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

- (1) $L_1 \cup L_2$ (Abschluss unter Vereinigung)
- (2) $L_1 \cap L_2$ (Abschluss unter Schnitt)
- (3) \bar{L} (Abschluss unter Komplement)
- (4) $L_1 \circ L_2$ (Abschluss unter Konkatenation)
- (5) 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

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

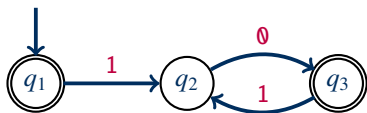
Das folgende Ergebnis ist leicht zu sehen:

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

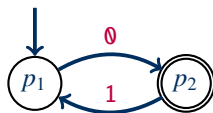
Also sind reguläre Sprachen unter Vereinigung abgeschlossen.

Beispiel

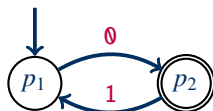
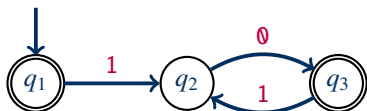
Die Vereinigung der NFAs



und



ergibt den NFA ...



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

Abschluss unter Schnitt

Satz: Wenn L , L_1 und L_2 reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

- (1) $L_1 \cup L_2$ (Abschluss unter Vereinigung)
- (2) $L_1 \cap L_2$ **Abschluss unter Schnitt**
- (3) \bar{L} (Abschluss unter Komplement)
- (4) $L_1 \circ L_2$ (Abschluss unter Konkatenation)
- (5) 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

↪ 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

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

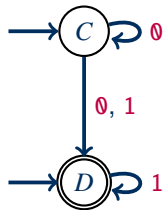
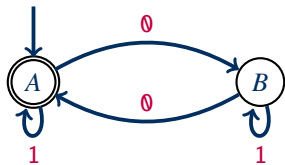
Wir werden zeigen, dass dies der gesuchte Automat ist:

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.

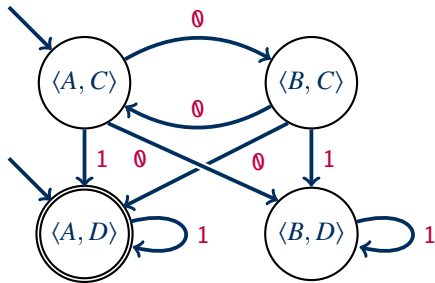
Beispiel



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 $w = \mathbf{a}_1 \cdots \mathbf{a}_n \in \mathbf{L}(\mathcal{M}_1 \otimes \mathcal{M}_2)$.

- Also gibt es einen akzeptierenden Lauf $\langle q_1^0, q_2^0 \rangle \langle q_1^1, q_2^1 \rangle \dots \langle q_1^n, q_2^n \rangle$
- 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 $q_2^n \in F_2$
 - $\langle q_1^i, q_2^i \rangle \in \delta(\langle q_1^{i-1}, q_2^{i-1} \rangle, \mathbf{a}_i)$,
also $q_1^i \in \delta_1(q_1^{i-1}, \mathbf{a}_i)$ und $q_2^i \in \delta_2(q_2^{i-1}, \mathbf{a}_i)$
- Daher sind $q_1^0 q_1^1 \dots q_1^n$ und $q_2^0 q_2^1 \dots q_2^n$ akzeptierende Läufe von \mathcal{M}_1 und \mathcal{M}_2 .

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. □

Abschluss unter Komplement

Satz: Wenn L , L_1 und L_2 reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

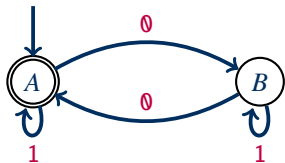
- (1) $L_1 \cup L_2$ (Abschluss unter Vereinigung)
- (2) $L_1 \cap L_2$ (Abschluss unter Schnitt)
- (3) \bar{L} **Abschluss unter Komplement**
- (4) $L_1 \circ L_2$ (Abschluss unter Konkatenation)
- (5) L^* (Abschluss unter Kleene-Stern)

Komplementoperator für DFAs

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

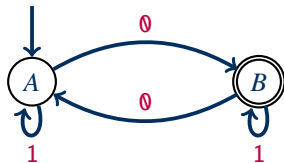
Für einen DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle$ sei $\overline{\mathcal{M}}$ der DFA $\langle Q, \Sigma, \delta, q_0, Q \setminus F \rangle$.

Beispiel:



„Wörter mit gerader Anzahl 0“

\sim



„Wörter mit ungerader Anzahl 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 $w = a_1 \cdots a_n \in \mathbf{L}(\overline{\mathcal{M}})$.

- Dann gibt es einen akzeptierenden Lauf $q_0q_1 \dots q_n$ für w in $\overline{\mathcal{M}}$.
- Dann ist $q_n \in Q \setminus F$.
- Dann ist $q_0q_1 \dots q_n$ ein verwerfender Lauf für w in \mathcal{M} .

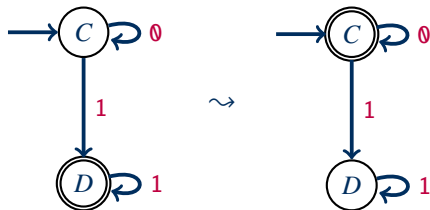
Also ist $w \notin \mathbf{L}(\mathcal{M})$, d.h. $w \in \overline{\mathbf{L}(\mathcal{M})}$.

„ $\mathbf{L}(\overline{\mathcal{M}}) \supseteq \overline{\mathbf{L}(\mathcal{M})}$ “ Sei $w = a_1 \cdots a_n \in \overline{\mathbf{L}(\mathcal{M})}$

- Dann hat \mathcal{M} einen verwerfenden Lauf $p_0p_1 \dots p_m$ für w
- Dann ist (1) $p_m \notin F$ oder (2) $m < |w|$
- In Fall (1) ist $p_0p_1 \dots p_m$ ein akzeptierender Lauf für w in $\overline{\mathcal{M}}$
- In Fall (2) ... ?
... vielleicht gar kein akzeptierender Lauf für w in $\overline{\mathcal{M}}$...

Der Satz ist falsch!

Gegenbeispiel



Wörter der Form $\{0\}^*\{1\}^+$

Wörter der Form $\{0\}^*$

Das Wort 010 zum Beispiel wird von keinem der beiden Automaten erkannt.

~ keine komplementären Sprachen

Korrekte Komplementierung

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

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: Finden Sie ein Gegenbeispiel)

Abschluss unter Konkatenation

Satz: Wenn L , L_1 und L_2 reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

- (1) $L_1 \cup L_2$ (Abschluss unter Vereinigung)
- (2) $L_1 \cap L_2$ (Abschluss unter Schnitt)
- (3) \bar{L} (Abschluss unter Komplement)
- (4) $L_1 \circ L_2$ **Abschluss unter Konkatenation**
- (5) 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.

→ besonders elegant geht das mit ϵ -Transitionen

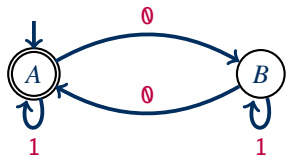
Gegeben seien zwei NFAs $\mathcal{M}_1 = \langle Q_1, \Sigma, \delta_1, Q_{0,1}, F_1 \rangle$ und $\mathcal{M}_2 = \langle Q_2, \Sigma, \delta_2, Q_{0,2}, F_2 \rangle$ mit $Q_1 \cap Q_2 = \emptyset$ (o.B.d.A.).

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

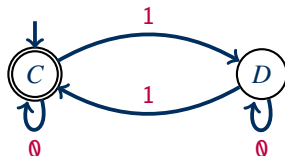
$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{falls } q \in Q_1 \\ \delta_2(q, a) & \text{falls } q \in Q_2 \end{cases} \quad \delta(q, \epsilon) = \begin{cases} Q_{0,2} & \text{falls } q \in F_1 \\ \emptyset & \text{andernfalls} \end{cases}$$

$\mathcal{M}_1 \odot \mathcal{M}_2$ 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.

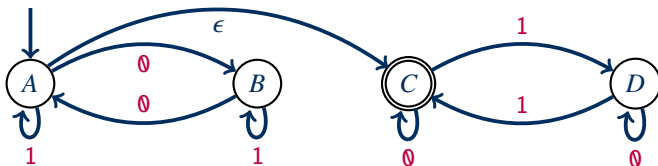
Beispiel Konkatenation



„Wörter mit gerader Anzahl 0“



„Wörter mit gerader Anzahl 1“



(Übung: Welche Wörter akzeptiert dieser Automat nicht?)

Satz: Für alle NFA M_1 und M_2 gilt $\mathbf{L}(M_1 \circ M_2) = \mathbf{L}(M_1) \circ \mathbf{L}(M_2)$.

Der Beweis ist einfach und analog zu den bisher gezeigten.

Abschluss unter Kleene-Stern

Satz: Wenn L , L_1 und L_2 reguläre Sprachen sind, dann beschreiben auch die folgenden Ausdrücke reguläre Sprachen:

- (1) $L_1 \cup L_2$ (Abschluss unter Vereinigung)
- (2) $L_1 \cap L_2$ (Abschluss unter Schnitt)
- (3) \bar{L} (Abschluss unter Komplement)
- (4) $L_1 \circ L_2$ (Abschluss unter Konkatenation)
- (5) 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.)
- $\delta'(q, a) = \delta(q, a)$ für alle $q \in Q, a \in \Sigma$ 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: $L(\mathcal{M}^*) = L(\mathcal{M})^*$.

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?