

FORMALE SYSTEME

4. Vorlesung: Nichtdeterministische Endliche Automaten

Hannes Straß

Folien: © Markus Krötzsch, <https://iccl.inf.tu-dresden.de/web/FS2020>, CC BY 3.0 DE

TU Dresden, 21. Oktober 2021

Wiederholung

- Grammatiken können Sprachen beschreiben und sie grob in Typen unterteilen.
- Typ-3-Grammatiken **generieren** reguläre Sprachen.
- Deterministische endliche Automaten **erkennen** reguläre Sprachen.
- Nichtdeterministische endliche Automaten verallgemeinern die Definition der Übergangsfunktion: Der Automat „rät“, welcher Übergang der richtige ist.

Hannes Straß, TU Dresden

Formale Systeme, VL 4

Folie 3 von 36

Wiederholung: NFA

Ein **nichtdeterministischer endlicher Automat** (international: „NFA“) \mathcal{M} ist ein Tupel $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ mit den folgenden Bestandteilen:

- Q : endliche Menge von **Zuständen**,
- Σ : Alphabet,
- δ : **Übergangsfunktion**, eine totale Funktion $Q \times \Sigma \rightarrow 2^Q$, wobei 2^Q die Potenzmenge von Q ist;
- Q_0 : Menge möglicher **Startzustände** $Q_0 \subseteq Q$,
- F : Menge von **Endzuständen** $F \subseteq Q$.

Notation: Wir schreiben statt $q' \in \delta(q, a)$ auch $q \xrightarrow{a} q'$.

Die Sprache eines NFA

Hannes Straß, TU Dresden

Formale Systeme, VL 4

Folie 4 von 36

Hannes Straß, TU Dresden

Formale Systeme, VL 4

Folie 6 von 36

Läufe eines NFA

Ein **Lauf** eines NFA $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ für ein Wort $w = \sigma_1 \dots \sigma_n$ ist eine Folge von Zuständen $q_0 \dots q_m$, so dass gilt:

- $q_0 \in Q_0$,
- $q_{i+1} \in \delta(q_i, \sigma_{i+1})$ für alle $0 \leq i < m$,
- (1) $m = |w| = n$ oder (2) $m < n$ und $\delta(q_m, \sigma_{m+1}) = \emptyset$.

Ein Lauf heißt **akzeptierend**, falls $m = n$ und $q_n \in F$.

Andernfalls heißt der Lauf **verwerfend**.

↪ Ein DFA hat genau einen Lauf für jedes Wort.

Er akzeptiert, wenn dieser Lauf akzeptierend ist.

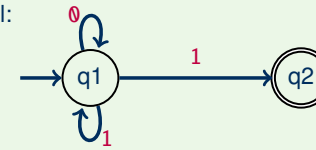
↪ Ein NFA kann für ein Wort mehrere Läufe haben.

Er akzeptiert, wenn einer dieser Läufe akzeptierend ist.

Sprache eines NFA

Die **Sprache eines NFA** $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ ist die Menge aller Wörter $w \in \Sigma^*$, für die \mathcal{M} einen akzeptierenden Lauf hat.

Beispiel:

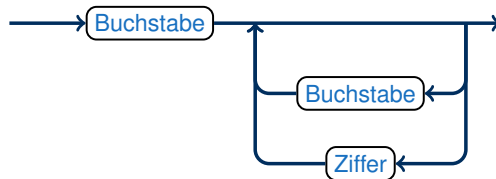


$\delta(q_1, 0) = \{q_1\}$
 $\delta(q_1, 1) = \{q_1, q_2\}$
 $\delta(q_2, 0) = \emptyset$
 $\delta(q_2, 1) = \emptyset$

Wort	Lauf	Ergebnis
011	$q_1 q_1 q_2$	verwerfend (zu kurz)
011	$q_1 q_1 q_1 q_2$	akzeptierend
011	$q_1 q_1 q_1 q_1$	verwerfend (kein Endzustand)

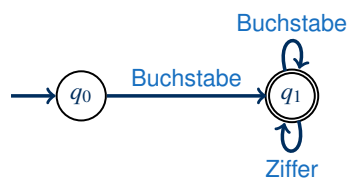
$$L(\mathcal{M}) = \{0, 1\}^* \circ \{1\}$$

NFA zur Darstellung von Syntaxdiagrammen



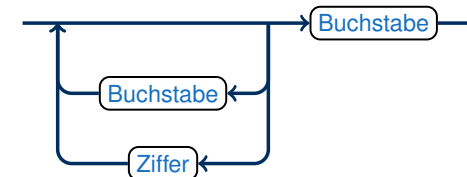
Übersetzung in NFA:

- zusammenhängende Linienbereiche werden Zustände
- Knoten werden Übergänge

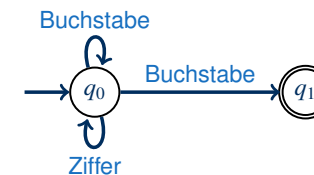


Syntaxdiagramme und Nichtdeterminismus

Das folgende Beispiel führt zu einem NFA, der kein DFA ist:



Entsprechender NFA:



Verallgemeinerte NFA-Übergangsfunktion

Wie beim DFA können wir auch bei einem NFA $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ eine erweiterte Übergangsfunktion definieren, die ganze Wörter einliest.

Zuerst erweitern wir δ auf Mengen von Zuständen:

Für eine Zustandsmenge $R \subseteq Q$ und ein Terminalsymbol a sei

$$\delta^{\cup}(R, a) = \bigcup_{q \in R} \delta(q, a).$$

Dann erweitern wir δ^{\cup} von einzelnen Symbolen zu beliebigen Wörtern:

Für eine Zustandsmenge $R \subseteq Q$ und ein Wort $w \in \Sigma^*$ sei $\delta^*(R, w)$ die Menge aller Zustände, die man erreichen kann, wenn man in einem Zustand aus R beginnt und das Wort w einliest. Formal:

- $\delta^*(R, \epsilon) = R$
- $\delta^*(R, av) = \delta^*(\delta^{\cup}(R, a), v)$

Wir notieren künftig wieder δ^* als δ , wenn die gemeinte Funktion aus dem Kontext klar ist.

Sprache eines NFA (2. Version)

Die erweiterte Übergangsfunktion hilft bei der Definition der Sprache, die ein NFA akzeptiert:

Die Sprache eines NFA $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ ist die Menge

$$L(\mathcal{M}) = \{w \in \Sigma^* \mid \delta(Q_0, w) \cap F \neq \emptyset\}.$$

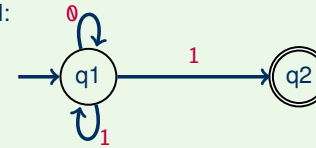
Die Bedingung „ $\delta(Q_0, w) \cap F \neq \emptyset$ “ bedeutet:

„Mindestens einer der Zustände, die man durch Einlesen von w von einem Startzustand aus erreichen kann, ist ein Endzustand.“

Behauptung: Diese Variante stimmt mit der vorherigen (mit akzeptierenden Läufen) überein.

Beispiel

Beispiel:



$$\begin{aligned} \delta(q_1, 0) &= \{q_1\} \\ \delta(q_1, 1) &= \{q_1, q_2\} \\ \delta(q_2, 0) &= \emptyset \\ \delta(q_2, 1) &= \emptyset \end{aligned}$$

Die Menge der Startzustände ist $Q_0 = \{q_1\}$.

Dann gilt:

$$\delta(Q_0, 0) = \delta(q_1, 0) = \{q_1\}$$

$$\delta(Q_0, 1) = \delta(q_1, 1) = \{q_1, q_2\}$$

$$\begin{aligned} \delta(Q_0, 10) &= \delta(\delta(Q_0, 1), 0) = \delta(\{q_1, q_2\}, 0) \\ &= \delta(q_1, 0) \cup \delta(q_2, 0) = \{q_1\} \cup \emptyset = \{q_1\} \end{aligned}$$

$$\delta(Q_0, 01) = \delta(\delta(Q_0, 0), 1) = \delta(\{q_1\}, 1) = \{q_1, q_2\}$$

Äquivalenz der Sprachdefinitionen für NFAs

Sei $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ ein NFA und $w = \sigma_1 \cdots \sigma_n \in \Sigma^*$ ein Wort.

Behauptung: Es gibt einen akzeptierenden Lauf für w genau dann, wenn $\delta(Q_0, w) \cap F \neq \emptyset$.

Wdh.: Wir beweisen $P \Leftrightarrow Q$ („ P gdw. Q “) durch Zeigen von (1) $P \Rightarrow Q$ und (2) $Q \Rightarrow P$.

Beweis „ \Rightarrow “: Angenommen, es gibt einen akzeptierenden Lauf $q_0 \dots q_n$ für w .

- Dann ist $q_n \in F$.
- Wir behaupten $q_i \in \delta(Q_0, \sigma_1 \cdots \sigma_i)$. (Damit folgt $\delta(Q_0, w) \cap F \neq \emptyset$.)
- Wir zeigen die stärkere Behauptung $q_i \in \delta(Q_0, \sigma_1 \cdots \sigma_i)$ für alle $0 \leq i \leq n$ mittels Induktion über i :
 - Induktionsanfang: Für $i = 0$ gilt $q_0 \in Q_0 = \delta(Q_0, \epsilon)$.
 - Induktionsvoraussetzung: Die Behauptung gelte für i .
 - Induktionsschritt: Für $i + 1$ gilt:
 - $q_i \in \delta(Q_0, \sigma_1 \cdots \sigma_i)$ (Induktionsvoraussetzung)
 - $q_{i+1} \in \delta(q_i, \sigma_{i+1})$ (laut Definition eines Laufs)
 - $q_{i+1} \in \delta(\delta(Q_0, \sigma_1 \cdots \sigma_i), \sigma_{i+1}) = \delta(Q_0, \sigma_1 \cdots \sigma_i \sigma_{i+1})$

Äquivalenz der Sprachdefinitionen für NFAs

Sei $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ NFA und $w = \sigma_1 \cdots \sigma_n \in \Sigma^*$ ein Wort.

Behauptung: Es gibt einen akzeptierenden Lauf für w genau dann, wenn $\delta(Q_0, w) \cap F \neq \emptyset$.

Beweis „ \Leftarrow “: Angenommen, $\delta(Q_0, w) \cap F \neq \emptyset$.

- Wir ermitteln einen akzeptierenden Lauf $q_0 \dots q_n$ für w .
- Dazu gehen wir rückwärts vor:
 - Wähle $q_n \in F \cap \delta(Q_0, w)$. (Existiert nach Voraussetzung.)
 - Für alle $i = n, \dots, 1$: (Annahme: $q_i \in \delta(Q_0, \sigma_1 \cdots \sigma_i)$ ist bereits gewählt.)
Wähle $q_{i-1} \in \delta(Q_0, \sigma_1 \cdots \sigma_{i-1})$, so dass $q_i \in \delta(q_{i-1}, \sigma_i)$.
(Solch ein q_{i-1} existiert immer:
 $\delta(Q_0, \sigma_1 \cdots \sigma_i) = \delta(\delta(\cdots \delta(\delta(Q_0, \sigma_1), \sigma_2) \cdots, \sigma_{i-1}), \sigma_i) = \delta(\delta(Q_0, \sigma_1 \cdots \sigma_{i-1}), \sigma_i)$.)
- Dies ist ein Lauf, da $q_0 \in \delta(Q_0, \epsilon) = Q_0$ und alle Übergänge erlaubt sind.
- Es ist ein akzeptierender Lauf, da $q_n \in F$. \square

NFA vs. DFA

Quiz: Sprache eines NFA

Die Sprache eines NFA $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ ist $\mathbf{L}(\mathcal{M}) = \{w \in \Sigma^* \mid \delta(Q_0, w) \cap F \neq \emptyset\}$.

Quiz: Wir betrachten den grafisch gegebenen NFA $\mathcal{M} = \langle Q, \Sigma, \delta, Q_0, F \rangle$ über $\Sigma = \{0, 1\}$:

...

Vergleich DFA – NFA

Ein endlicher Automat \mathcal{M} ist genau dann **äquivalent** zu einem endlichen Automaten \mathcal{M}' , wenn $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}')$ gilt.

Offensichtlich sind NFAs mindestens so allgemein wie DFAs:

Satz: Jeder DFA kann als NFA aufgefasst werden. Daher wird jede von einem DFA akzeptierbare Sprache auch von einem NFA akzeptiert.

Beweis: Für jeden DFA $\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle$ gibt es einen entsprechenden NFA $\mathcal{M}' = \langle Q, \Sigma, \delta_{\text{NFA}}, \{q_0\}, F \rangle$ mit $\delta_{\text{NFA}}(q, a) = \{\delta(q, a)\}$. Offensichtlich ist $\mathbf{L}(\mathcal{M}) = \mathbf{L}(\mathcal{M}')$. \square

Die Umkehrung dieses Satzes gilt allerdings auch:

Satz: Jede von einem NFA akzeptierbare Sprache wird auch von einem DFA akzeptiert.

In diesem Sinne sind NFA nicht ausdrucksstärker als DFA – wie kann das sein?

NFAs als DFAs – Idee

Die verallgemeinerte NFA-Übergangsfunktion bildet **Mengen von Zuständen** auf **Mengen von Zuständen** ab:

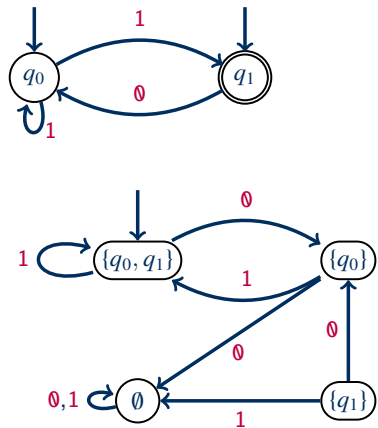
$$\delta(R, a) = \bigcup_{q \in R} \delta(q, a).$$

„Wenn der Automat in einem der Zustände R ist und a liest, so ist er anschließend in einem der Zustände der Menge $\delta(R, a)$.“

Dieser Übergang zwischen Mengen möglicher Zustände ist an sich deterministisch.

→ Wir können einen NFA deterministisch simulieren, indem wir die Menge der möglichen Zustände berechnen.

Beispiel Potenzmengenkonstruktion



$$\begin{aligned} \delta_{\text{DFA}}(\{q_0\}, 0) &= \emptyset \\ \delta_{\text{DFA}}(\{q_0\}, 1) &= \{q_0, q_1\} \\ \delta_{\text{DFA}}(\{q_1\}, 0) &= \{q_0\} \\ \delta_{\text{DFA}}(\{q_1\}, 1) &= \emptyset \\ \delta_{\text{DFA}}(\{q_0, q_1\}, 0) &= \{q_0\} \\ \delta_{\text{DFA}}(\{q_0, q_1\}, 1) &= \{q_0, q_1\} \\ \delta_{\text{DFA}}(\emptyset, 0) &= \emptyset \\ \delta_{\text{DFA}}(\emptyset, 1) &= \emptyset \end{aligned}$$

Erkannte Sprache:
 $\{1\}^* \circ (\{0\} \circ \{1\}^+)^*$

Die Potenzmengenkonstruktion

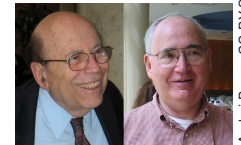
Für einen NFA $M = \langle Q, \Sigma, \delta, Q_0, F \rangle$ definieren wir den **Potenzmengen-DFA** $M_{\text{DFA}} = \langle Q_{\text{DFA}}, \Sigma, \delta_{\text{DFA}}, q_0, F_{\text{DFA}} \rangle$ wie folgt:

- $Q_{\text{DFA}} = 2^Q$ (Potenzmenge von Q)
- $\delta_{\text{DFA}}(R, a) = \bigcup_{q \in R} \delta(q, a)$
- $q_0 = Q_0$
- $F_{\text{DFA}} = \{R \in 2^Q \mid R \cap F \neq \emptyset\}$

Der resultierende DFA M_{DFA} ist äquivalent zu M :

Satz (Rabin/Scott): $L(M) = L(M_{\text{DFA}})$

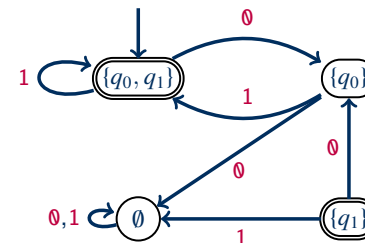
(Beweis später)



Michael Oser Rabin Dana Scott

Vereinfachung Potenzmengenkonstruktion

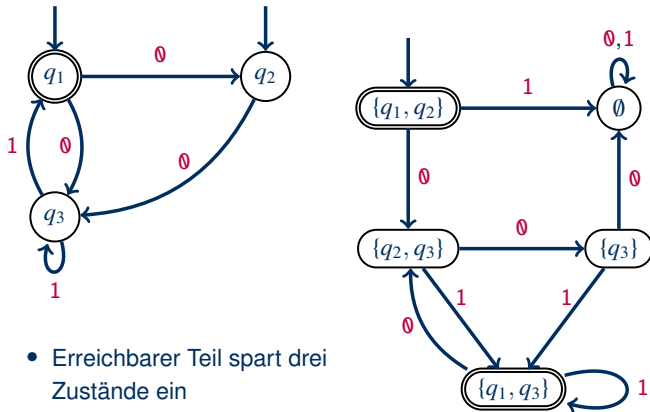
Der Automat aus dem vorherigen Beispiel kann vereinfacht werden:



- Zustand $\{q_1\}$ ist unerreichbar;
- Zustand \emptyset kann nicht verlassen werden (irrelevant für akzeptierende Läufe).

Potenzmengenkonstruktion „on the fly“

Vermeidung unnötiger Zustände durch schrittweise Konstruktion vom Startzustand:



- Erreichbarer Teil spart drei Zustände ein
- Zustand \emptyset wie zuvor unnötig

Potenzmengenkonstruktion: Korrektheit (2)

Satz (Rabin/Scott): $L(\mathcal{M}) = L(\mathcal{M}_{\text{DFA}})$

Beweis (Fortsetzung): Wir haben gezeigt: $\delta_{\text{DFA}}(R, w) = \delta(R, w)$. (Für alle $R \subseteq Q$, $w \in \Sigma^*$.)

Damit ergibt sich für beliebige Wörter $w \in \Sigma^*$:

- $w \in L(\mathcal{M})$ gdw. $\delta(Q_0, w) \cap F \neq \emptyset$
- gdw. $\delta_{\text{DFA}}(Q_0, w) \cap F \neq \emptyset$
- gdw. $\delta_{\text{DFA}}(Q_0, w) \in F_{\text{DFA}}$
- gdw. $w \in L(\mathcal{M}_{\text{DFA}})$

□

Die beiden Automatenmodelle DFA und NFA charakterisieren also die gleiche Klasse von Sprachen.

~> Benötigen wir den Nichtdeterminismus der NFAs überhaupt noch?

Potenzmengenkonstruktion: Korrektheit

Satz (Rabin/Scott): $L(\mathcal{M}) = L(\mathcal{M}_{\text{DFA}})$

Beweis: Wir nutzen die Korrespondenz der verallgemeinerten Übergangsfunktionen aus.

Zuerst zeigen wir, dass für jedes Wort $w \in \Sigma^*$ und jede Zustandsmenge R gilt:

$$\delta_{\text{DFA}}(R, w) = \delta(R, w). \quad (\text{Beweis per Induktion über die Länge von } w.)$$

Induktionsanfang:

- (1) $\delta_{\text{DFA}}(R, \epsilon) = R = \delta(R, \epsilon)$
- (2) $\delta_{\text{DFA}}(R, a) = \bigcup_{q \in R} \delta(q, a) = \delta(R, a)$

Induktionsvoraussetzung: Es gelte $\delta_{\text{DFA}}(R, v) = \delta(R, v)$ für alle Wörter v der Länge ℓ .

Induktionsschritt: Wir zeigen $\delta_{\text{DFA}}(R, av) = \delta(R, av)$ für ein beliebiges Wort av der Länge $\ell + 1$.

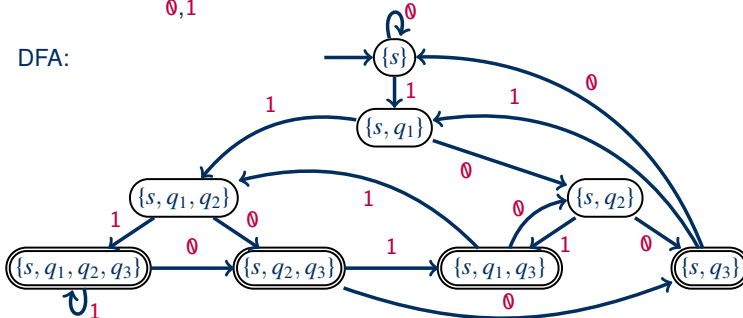
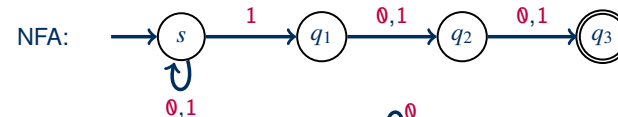
$$\begin{aligned} (3) \quad \delta_{\text{DFA}}(R, av) &= \delta_{\text{DFA}}(\delta_{\text{DFA}}(R, a), v) \\ &= \delta_{\text{DFA}}(\delta(R, a), v) && (\text{wegen (2)}) \\ &= \delta(\delta(R, a), v) && (\text{Induktionsvoraussetzung}) \\ &= \delta(R, av) \end{aligned}$$

Größenvergleich

Der DFA eines NFA hat $2^{|Q|}$ – also exponentiell viele – Zustände.

Auch „on the fly“ lässt sich das im Allgemeinen nicht vermeiden.

Beispiel: „Wörter mit 1 an drittletzter Stelle“



Größenvergleich (2)

Allgemein kann man für jede Zahl $n \geq 1$ die Sprache $L_n = \{0, 1\}^* 1 \{0, 1\}^{n-1}$ betrachten („Wörter mit 1 an n -letzter Stelle“).

Es gilt:

- Es gibt einen NFA mit $n + 1$ Zuständen, der L_n erkennt.
- Jeder DFA, der L_n erkennt, hat mindestens 2^n Zustände.

Beweis (per Widerspruch): Angenommen, DFA $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ erkennt L_n und hat weniger als 2^n Zustände. Über $\{0, 1\}$ gibt es 2^n verschiedene Wörter der Länge n . Nach dem Schubfachprinzip gibt es $w_1, w_2 \in \{0, 1\}^n$ mit $w_1 \neq w_2$, sodass $\delta(q_0, w_1) = \delta(q_0, w_2)$ gilt. Da $w_1 \neq w_2$ gibt es ein $k \leq n$, so dass w_1 und w_2 sich an k -letzter Stelle unterscheiden. Wir definieren nun $v = 0^{n-k}$ und betrachten die Wörter w_1v und w_2v , die sich an n -letzter Stelle unterscheiden. Aus $\delta(q_0, w_1) = \delta(q_0, w_2)$ folgt $\delta(q_0, w_1v) = \delta(q_0, w_2v)$, also gilt $w_1v \in L(M)$ gdw. $w_2v \in L(M)$. Dies steht im Widerspruch dazu, dass M die Sprache L_n erkennt, da sich die Wörter an n -letzter Stelle unterscheiden. \square

Schlussfolgerung: NFAs können exponentiell kompakter sein als äquivalente DFAs.

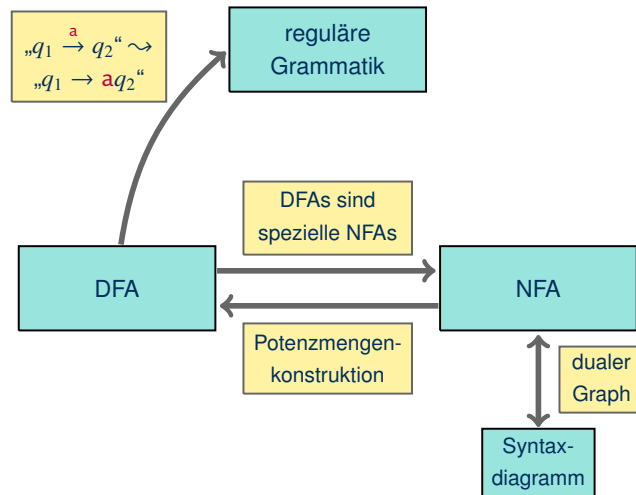
~ Nichtdeterminismus der NFAs trotzdem nützlich.

Quiz: NFA vs. DFA

Quiz: Sei M_N ein NFA und M_D ein DFA mit $L(M_N) = L(M_D)$.

...

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 M_G , welcher die selbe Sprache akzeptiert (d.h., $L(G) = L(M_G)$).

Für $G = \langle V, \Sigma, P, S \rangle$ ergibt sich $M_G = \langle Q, \Sigma, \delta, Q_0, F \rangle$ wie folgt:

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

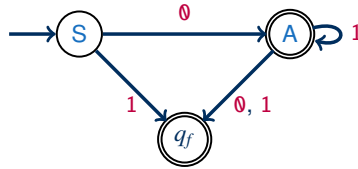
Beispiel

Wir betrachten eine reguläre Grammatik mit den folgenden sechs Regeln:

$$S \rightarrow 1 \mid 0A$$

$$A \rightarrow 0 \mid 1 \mid 1A \mid \epsilon$$

Entsprechender NFA:



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

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 $L(G) = L(M_G)$, d.h. für jedes Wort $w \in \Sigma^*$ soll gelten:
 $w \in L(G)$ gdw. $w \in L(M_G)$.

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

$$\begin{aligned} \epsilon \in L(G) &\text{ gdw. } S \rightarrow \epsilon \in P \\ &\text{ gdw. } S \in F \\ &\text{ gdw. } \epsilon \in L(M_G) \end{aligned}$$

$L(G) \subseteq L(M_G)$

Wir zeigen noch $w \in L(G)$ gdw. $w \in L(M_G)$ für den Fall $|w| \geq 1$.

„ \Rightarrow “ Angenommen $w \in L(G)$ mit $w = a_1 \cdots a_n$ und $n \geq 1$.

Es gibt zwei mögliche Herleitungen für w :

$$(1) S \Rightarrow a_1 B_1 \Rightarrow \dots \Rightarrow a_1 \cdots a_{n-1} B_{n-1} \Rightarrow a_1 \cdots a_{n-1} a_n$$

$$(2) S \Rightarrow a_1 B_1 \Rightarrow \dots \Rightarrow a_1 \cdots a_{n-1} B_{n-1} \Rightarrow a_1 \cdots a_{n-1} a_n B_n \Rightarrow a_1 \cdots a_n$$

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

$$S \rightarrow a_1 B_1 \quad B_1 \rightarrow a_2 B_2 \quad \dots \quad B_{n-1} \rightarrow a_n$$

Also hat M_G die folgenden Übergänge:

$$S \xrightarrow{a_1} B_1 \quad B_1 \xrightarrow{a_2} B_2 \quad \dots \quad B_{n-1} \xrightarrow{a_n} q_f$$

Also ist $SB_1 B_2 \dots B_{n-1} q_f$ ein akzeptierender Lauf von M_G und M_G akzeptiert das Wort w .

Fall (2) ist ähnlich, wobei der Lauf auf B_n endet und $B_n \in F$.

$L(G) \supseteq L(M_G)$

Wir zeigen noch $w \in L(G)$ gdw. $w \in L(M_G)$ für den Fall $|w| \geq 1$.

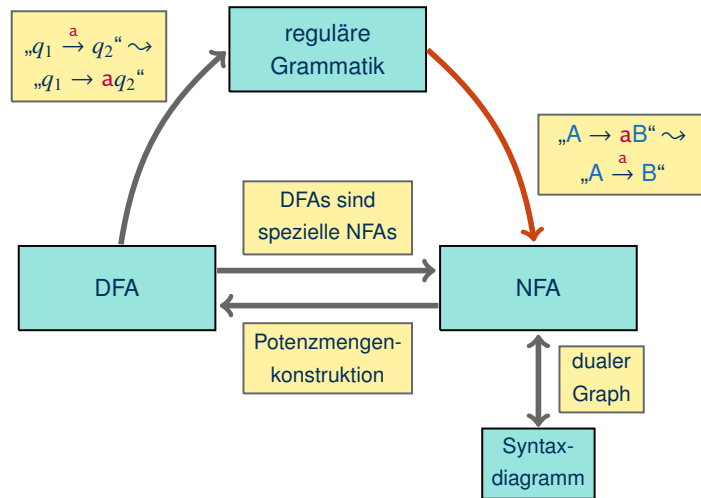
„ \Leftarrow “ Angenommen $w \in L(M_G)$ mit $w = \sigma_1 \cdots \sigma_n$ und $n \geq 1$.

Beweis analog zur vorangegangenen Richtung; grob skizziert:

- w hat einen akzeptierenden Lauf in 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



Zusammenfassung und Ausblick

Nichtdeterministische endliche Automaten (NFA) vereinfachen die Modellierung, z.B. die direkte Darstellung von Syntaxdiagrammen.

Rabin/Scott: DFAs und NFAs erkennen die selben Sprachen.
(Potenzmengenkonstruktion)

DFAs/NFAs erkennen genau die **regulären Sprachen**. (Grammatik \leftrightarrow NFA)

Offene Fragen:

- Gibt es noch mehr Darstellungsformen für reguläre Sprachen?
- Was kommt heraus, wenn man Operationen auf reguläre Sprache anwendet?
- Wir haben gesehen, dass man Automaten manchmal vereinfachen kann – geht das noch besser?