2.2 Nichtdeterministische Endliche Automaten (NFA)

Motivation:

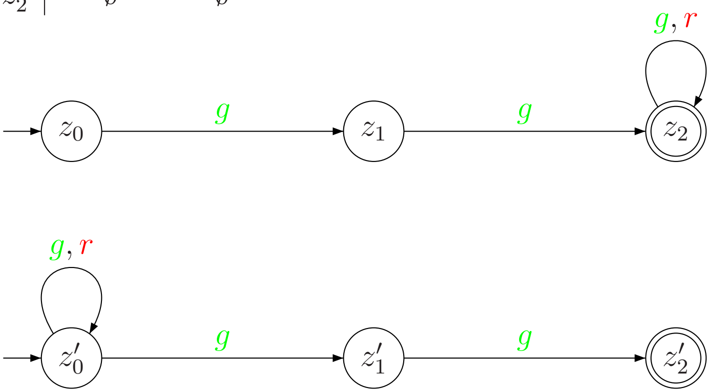
- Aktionen u.U. nicht vollständig beobachtbar bzw. unterscheidbar
- Vereinfachung der Konstruktionen

Definition 2.7. Ein *NFA* ist ein Tupel $M = (Z, \Sigma, I, \delta, F)$ mit folgenden Komponenten:

- Z ist eine endliche Menge (Elemente: „Zustände“)
- Σ ist ein Alphabet
- $I \subseteq Z$ („Initialzustände“)
- $\delta\colon Z \times \Sigma \to \mathcal{P}(Z)$ („Überführungsfunktion“)
- $F \subseteq Z$ („akzeptierende“ oder „Finalzustände“)

Beispiel 2.8. $Z = \{z_0, z_1, z_2, z_0', z_1', z_2'\}$, $\Sigma = \{g, r\}$, $I = \{z_0, z_0'\}$, $F = \{z_2, z_2'\}$

δ	g	r
z_0	$\{z_1\}$	$\emptyset$
z_1	$\{z_2\}$	$\emptyset$
z_2	$\{z_2\}$	$\{z_2\}$
z_0'	$\{z_0', z_1'\}$	$\{z_0'\}$
z_1'	$\{z_2'\}$	$\emptyset$
z_2'	$\emptyset$	$\emptyset$

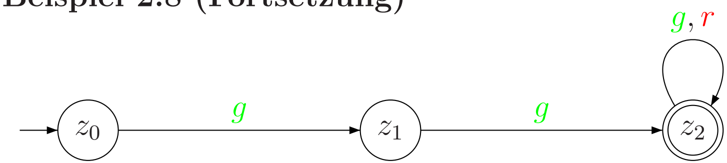


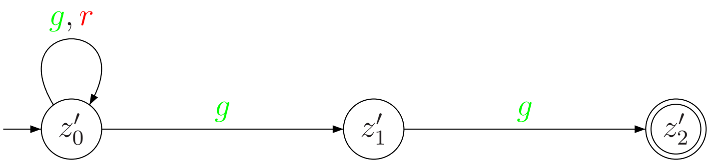
Definition 2.9. Sei $M = (Z, \Sigma, I, \delta, F)$ ein NFA.

1. Seien $z, z' \in Z$ und $w \in \Sigma^*$. Wir schreiben $z \xrightarrow{w} z'$, wenn es einen in z startenden w-beschrifteten Pfad gibt, der in z' endet

2. Die *von M akzeptierte Sprache* ist $L(M) = \{w \in \Sigma^* \mid \exists \iota \in I, f \in F\colon \iota \xrightarrow{w} f\}$.

Beispiel 2.8 (Fortsetzung)



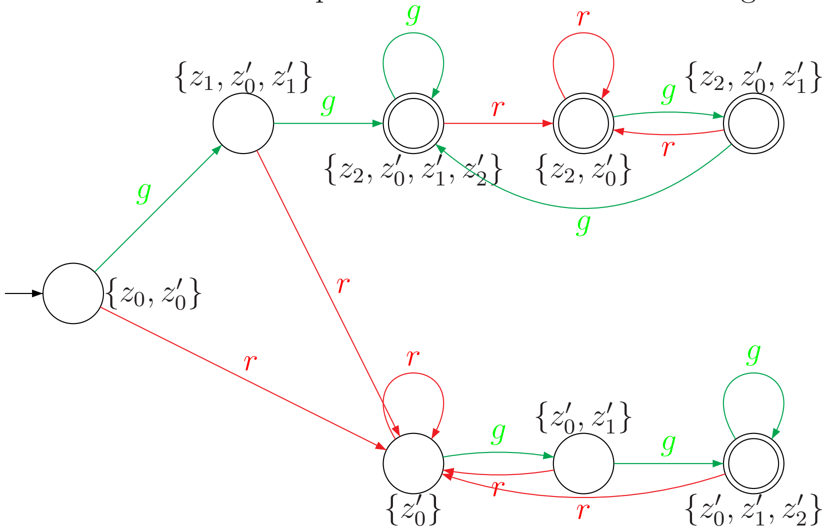


$z_0' \xrightarrow{r} z_0'$
$z_0' \xrightarrow{g} z_0'$ und $z_0' \xrightarrow{g} z_1'$
$z_0 \xrightarrow{w} z_2$ gdw. w beginnt mit gg
$z_0' \xrightarrow{w} z_2'$ gdw. w endet auf gg

Daher ist $L(M)$ die Menge der Wörter, die mit gg beginnen oder auf gg enden

Satz 2.10. *Sei M ein NFA. Dann ist $L(M)$ regulär.*

Beweis durch Beispiel:
Aus dem NFA in Beispiel 2.8 erhält man z.B. den folgenden äquivalenten DFA:



Der DFA M' akzeptiert $L(M)$, d.h. er „simuliert“ den NFA M.

Idee: Ein Pfad des DFA M' verfolgt alle Pfade des NFA M gleichzeitig, d.h. beim Einlesen eines Wortes w bestimmt er laufend die Menge derjenigen NFA-Zustände, in denen der NFA sein könnte. Zustände des DFA sind also Mengen von Zuständen des NFA d.h. $Z' \subseteq \mathcal{P}(Z)$. Daher sprechen wir von der *Potenzmengenkonstruktion*.

Algorithmus Für $N \subseteq Z$ und $a \in \Sigma$ definiere $N.a := \{z' \in Z \mid \exists z \in N\colon z' \in \delta(z, a)\}$.

Der DFA M' wird von folgendem Algorithmus aus dem NFA M berechnet:

setze $Z' = \{I\}$
solange es $N \in Z'$, $a \in \Sigma$ gibt mit $N.a \notin Z'$ setze $Z' := Z' \cup \{N.a\}$
setze $\iota = I$
für $N \in Z', a \in \Sigma$ setze $\delta'(N, a) = N.a$
setze $F' = \{N \in Z' \mid N \cap F \neq \emptyset\}$ $\square$

Um zu zeigen, daß eine Sprache regulär ist (d.h. von einem DFA akzeptiert wird), reicht es also, einen NFA anzugeben, der sie akzeptiert.

Wir werden jetzt NFAs verwenden, um zwei weitere Fakten *á la* Lemmas 2.4 und 2.5 und Folgerung 2.6 zu zeigen.

Definition. Sei Σ ein Alphabet.

- Sind $u, v \in \Sigma^*$, so definiere $u \cdot v := uv \in \Sigma^*$ mit $|uv| = |u| + |v|$.
- Sind $K, L \subseteq \Sigma^*$, so setze $K \cdot L = KL = \{uv \mid u \in K, v \in L\} \subseteq \Sigma^*$.

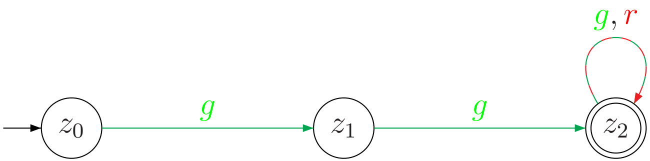
Beispiel.

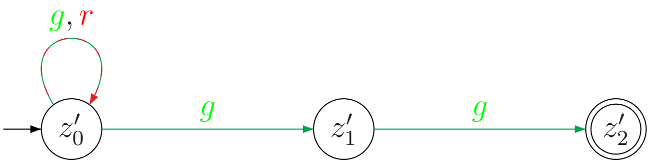
- $aabba \cdot bba = aabbabba$, $aab \cdot \varepsilon = aab$, $\varepsilon \cdot \varepsilon = \varepsilon$
- $\{aa, \varepsilon\} \cdot \{bb, \varepsilon\} = \{aabb, aa, bb, \varepsilon\}$ $L \cdot \{\varepsilon\} = L$ und $L \cdot \emptyset = \emptyset$

  Achtung: $L \cdot L \neq \{uu \mid u \in L\}$

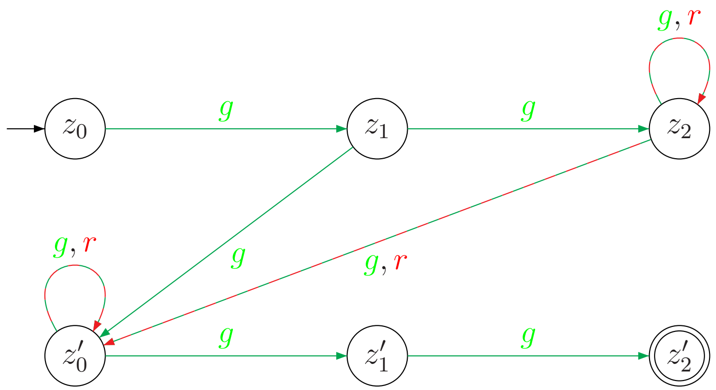
Lemma 2.11. *Sind L_1 und L_2 regulär, so auch $L_1 \cdot L_2$.*

Beweis durch Beispiel:
Sei $L_1 = \{ggw \mid w \in \{g, r\}^*\}$ die Menge der Wörter, die mit gg beginnen und $L_2 = \{wgg \mid w \in \{g, r\}^*\}$ die Menge der Wörter, die mit gg enden. Dann ist $L_1 \cdot L_2$ die Menge der Wörter $ggwgg$ mit $w \in \{g, r\}^*$.
Die Sprachen L_1 und L_2 werden von den folgenden NFAs M_1 und M_2 akzeptiert (M_1 hat die Zustände $\{z_0, z_1, z_2\}$ und M_2 die Zusände $\{z'_0, z'_1, z'_2\}$).





Die Sprache $L(M_1) \cdot L(M_2)$ wird vom folgenden NFA M akzeptiert:



Es gilt $L(M) = L(M_1) \cdot L(M_2) = L_1 \cdot L_2$, d.h. die Sprache $L_1 \cdot L_2$ wird von einem NFA akzeptiert. Nach Satz 2.10 ist sie also regulär.

Wir haben den NFA M konstruiert, indem wir in die Vereinigung der NFAs M_1 und M_2 zusätzliche Kanten eingefügt und die initialen und finalen Zustände angepaßt haben:

- Kanten, die in Zustände aus F_1 führen, wurden geklont und zu allen Zuständen aus I_2 gelenkt
- Ist $I_1 \cap F_1 = \emptyset$, so sind die initialen Zustände von M diejenigen aus I_1, ansonsten diejenigen aus $I_1 \cup I_2$
- die akzeptierenden Zustände von M sind die Zustände aus F_2

$\square$

Bemerkung. Auch wenn die Automaten M_1 und M_2 DFAs sind, ist M kein DFA, sondern ein NFA (da Kanten geklont werden).

Definition. Sei Σ ein Alphabet und $L \subseteq \Sigma^*$.

- $L^0 = \{\varepsilon\}$ und $L^{n+1} = L \cdot L^n$
- $L^* = \bigcup_{n \geq 0} L^n = \{w_1 w_2 \cdots w_n \mid n \geq 0, w_1, w_2, \ldots, w_n \in L\}$ heißt „Kleene-Iteration“ oder „Kleene-Stern“ von L

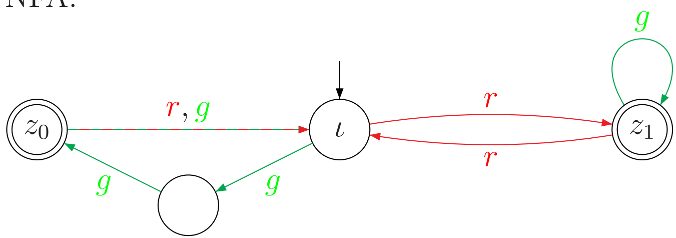
Beispiel. Mit $L = \{aa, ab, ba\}$ gelten

$$\begin{aligned}
L^0 &= \{\varepsilon\} \\
L^1 &= L \cdot L^0 = L \\
L^2 &= L \cdot L = \{w_1 w_2 \mid w_1, w_2 \in L\} \\
&= \{aa\,aa, aa\,ab, aa\,ba, ab\,aa, ab\,ab, ab\,ba, ba\,aa, ba\,ab, ba\,ba\} \\
L^3 &= L \cdot L^2 = \{w_1 w_2 w_3 \mid w_1, w_2, w_3 \in L\} \\
L^* &= \{w_1 w_2 \cdots w_n \mid n \geq 0, w_1, w_2, \ldots, w_n \in L\} \ni \varepsilon
\end{aligned}$$

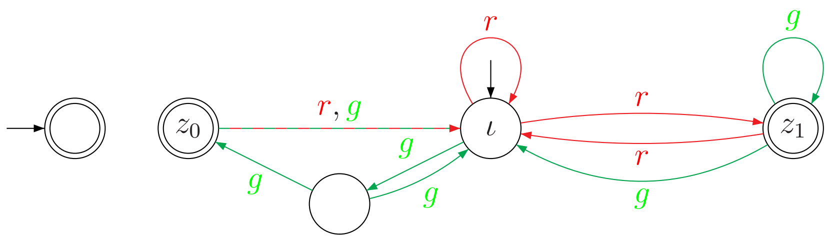
Insbes. gilt $\varepsilon \in L^*$ für jede Sprache L (sogar für $L = \emptyset$)

Lemma 2.12. *Ist L regulär, so auch L^*.*

Beweis durch Beispiel:
Sei M der folgende NFA:



Die Sprache $L(M)^*$ wird von dem folgenden NFA M' akzeptiert:



Wir haben den NFA M' konstruiert, indem wir den NFA M wie folgt erweitern:

- Kanten, die in Zustände aus F führen, werden geklont und zu allen Zuständen aus I gelenkt
- ein neuer Zustand wird hinzugefügt, der initial und final und an keiner Kante beteiligt ist (um ε zu akzeptieren)

$\square$

Bemerkung. Auch wenn der Automat M ein <u>D</u>FAs ist, ist M' kein <u>D</u>FA, sondern ein <u>N</u>FA (denn er hat ≥ 2 Initialzustände).

Zusammenfassung

- Für eine Sprache L sind äquivalent:
  - L ist regulär
  - L ist Sprache eines DFA
  - L ist Sprache eines NFA
- Alle endlichen Sprachen sind regulär (Übung)
- Die Klasse der regulären Sprachen ist abgeschlossen unter den Booleschen Operationen Vereinigung $\cup$, Schnitt $\cap$ und Komplement $\Sigma^*\setminus$ und unter Multiplikation $\cdot$ und Kleene-Iteration *.