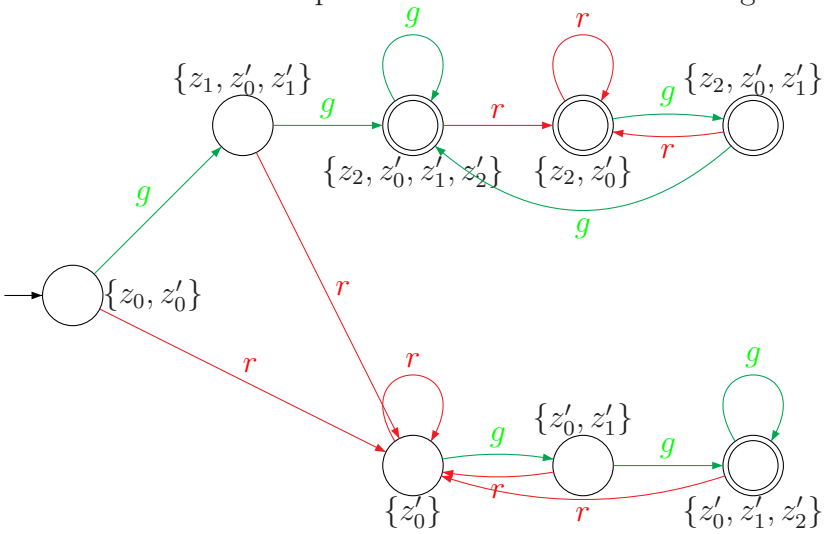
Satz 2.10. *Sei M ein NFA. Dann ist $L(M)$ regulär.*

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 $F' = \{N \in Z' \mid N \cap F \neq \emptyset\}$ □

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.

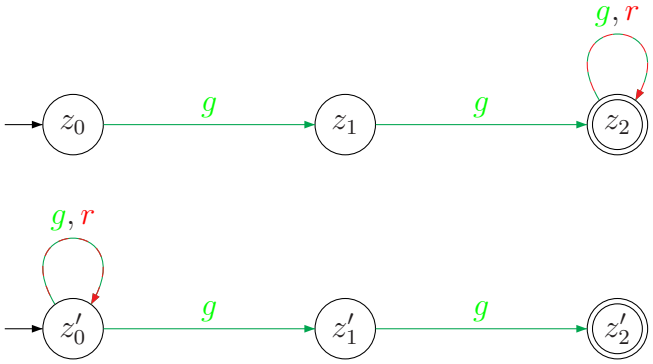
- Sind $u, v \in \Sigma^*$, so definiere $u \cdot v := uv \in \Sigma^*$ mit $|uv| = |u| + |v|$.
- Sind $K, L \subseteq \Sigma^*$, so setze $K \cdot L = KL = \{uv \mid u \in K, v \in L\} \subseteq \Sigma^*$.

Beispiel.

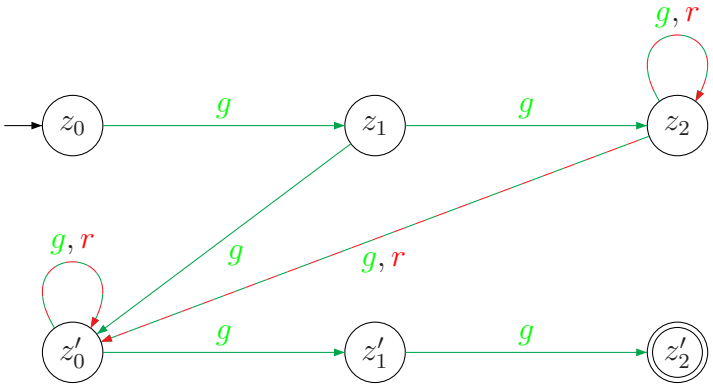
- $aabba \cdot bba = aabbabba$, $aab \cdot \varepsilon = aab$, $\varepsilon \cdot \varepsilon = \varepsilon$
- $\{aa, \varepsilon\} \cdot \{bb, \varepsilon\} = \{aabb, aa, bb, \varepsilon\}$ $L \cdot \{\varepsilon\} = L$ und $L \cdot \emptyset = \emptyset$
  Achtung: $L \cdot L \neq \{uu \mid u \in L\}$

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 $w \in \{g, r\}^*$. (Vorsicht: $gg \notin L_1 \cdot L_2$.)
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

□

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
- $L^+ = \bigcup_{n \geq 1} L^n = \{w_1 w_2 \cdots w_n \mid n \geq 1, w_1, w_2, \ldots, w_n \in L\}$ heißt „positive Kleene-Iteration“ oder „Kleene-Plus“ von L

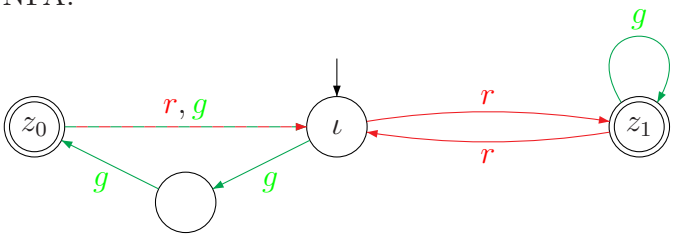
Beispiel. Mit $L = \{aa, ab, ba\}$ gelten

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

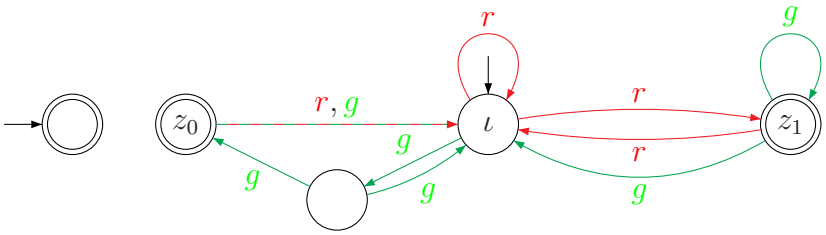
Insbes. gilt $\varepsilon \in L^*$ für jede Sprache L (sogar für $L = \emptyset$).

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)

□

Bemerkung. Auch wenn der Automat M ein DFAs ist, ist M' kein DFA, sondern ein NFA (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 *.

2.3 Reguläre Ausdrücke

DFAs und NFAs sind Beschreibungsmöglichkeiten für reguläre Sprachen. Gezeichnet sind sie sehr intuitiv, aber für Algorithmen schwer zu verwenden. Als Tupel angegeben sind sie für Menschen schwer verständlich.

Ziel: Eine textuelle Beschreibung regulärer Sprachen.

Definition. Sei Σ ein Alphabet.

1. Die Zeichenketten $\varnothing$, λ und a für $a \in \Sigma$ sind *reguläre Ausdrücke über* Σ.
2. Sind r_1 und r_2 reguläre Ausdrücke über Σ, so auch die Zeichenketten $(r_1 + r_2)$, $(r_1 r_2)$ und (r_1^*).
3. Nichts ist ein regulärer Ausdruck über Σ, was sich nicht in endlich vielen Schritten nach 1 und 2 erzeugen läßt.

Beispiel.
Sind $0, 1 \in \Sigma$, so sind $((((0 + 1)^*)0)1)$ und $((\varnothing + \varnothing)^*)$ reguläre Ausdrücke über Σ.

Jeder reguläre Ausdruck r ist also ein Wort über dem Alphabet

$$\Sigma \cup \{\varnothing, \lambda, +, ^*, (,)\}.$$

Er bezeichnet eine Sprache $L(r)$ über Σ:

Definition. 1. $L(\varnothing) = \emptyset$, $L(\lambda) = \{\varepsilon\}$ und $L(a) = \{a\}$ für alle $a \in \Sigma$.

2. $L\big((r_1 + r_2)\big) = L(r_1) \cup L(r_2)$, $L\big((r_1 r_2)\big) = L(r_1) \cdot L(r_2)$ und $L\big((r_1^*)\big) = L(r_1)^*$.

Damit ist ein regulärer Ausdruck r ein „Name“ für die Sprache $L(r)$.

Beispiel.

$$\begin{aligned} L(((((0+1)^*)0)1)) &= \{0,1\}^* \cdot \{01\} \\ L(((\varnothing + \varnothing)^*)) &= L((\varnothing + \varnothing))^* \\ &= (L(\varnothing) \cup L(\varnothing))^* \\ &= (\emptyset \cup \emptyset)^* \\ &= \emptyset^* = \{\varepsilon\} = L(\lambda) \end{aligned}$$

Fragen:

(1) Welche Sprachen haben einen Namen, d.h. für welche Sprachen L existiert ein regulärer Ausdruck r mit $L = L(r)$?

(2) Gibt es Algorithmen mit

   (a) Eingabe: regulärer Ausdruck r und $u \in \Sigma^*$
   Frage: gilt $u \in L(r)$?

   (b) Eingabe: reguläre Ausdrücke r_1 und r_2
   Frage: Gilt $L(r_1) \subseteq L(r_2)$?

(3) Gibt es zu jedem regulären Ausdruck r einen regulären Ausdruck r' mit $L(r') = \Sigma^* \setminus L(r)$? Kann man ggf. r' aus r berechnen?

Satz 2.13 (Antwort auf Frage (1))**.** *Sei $L \subseteq \Sigma^*$. Die Sprache L ist genau dann regulär, wenn es einen regulären Ausdruck r gibt mit $L = L(r)$.*

Insbes. kann aus einem regulären Ausdruck r ein NFA M berechnet werden mit $L(r) = L(M)$.

Beweis durch Beispiel:
(nur Richtung „$\Leftarrow$“)
Wir betrachten den regulären Ausdruck $r = ((((0 + 1)^*)0)1))$.

Die Sprachen $L(0) = \{0\}$ und $L(1) = \{1\}$ sind regulär

$\Longrightarrow L((0 + 1)) = L(0) \cup L(1) = \{0, 1\}$ regulär (Lemma 2.5)

$\Longrightarrow L((0 + 1)^*) = L((0 + 1))^*$ regulär (Lemma 2.12)

$\Longrightarrow L(((0 + 1)^*)0) = L((0 + 1)^*) \cdot L(0)$ regulär (Lemma 2.11)

$\Longrightarrow L((((0 + 1)^*)0)1) = L((0 + 1)^*0) \cdot L(1)$ regulär (Lemma 2.11) □

Satz 2.14 (Antwort auf Frage (2))**.** *Es gibt Algorithmen mit*

(a) Eingabe: regulärer Ausdruck r und $u \in \Sigma^*$
Frage: gilt $u \in L(r)$?

(b) Eingabe: reguläre Ausdrücke r_1 und r_2
Frage: Gilt $L(r_1) \subseteq L(r_2)$?

BewIdee:

(a) berechne aus r einen NFA M mit $L(r) = L(M)$ (verwendet Satz 2.13) und dann einen DFA M' mit $L(M') = L(M) = L(r)$ (verwendet Satz 2.10). Teste, ob $u \in L(M')$.

(b) Berechne wie in (a) DFAs M_1 und M_2 mit $L(M_1) = L(r_1)$ und $L(M_2) = L(r_2)$.

Berechne DFA M_{12} mit $L(M_{12}) = L(M_1) \setminus L(M_2) = L(M_1) \cap \Sigma^* \setminus L(M_2)$ (verwendet Lemmas 2.4 und 2.6)

Teste, ob $L(M_{12}) \neq \emptyset$, d.h. ob in M_{12} ein Finalzustand vom Initialzustand aus erreichbar ist (z.B. durch Breitensuche). Wenn ja, so gilt $L(M_1) \not\subseteq L(M_2)$, ansonsten gilt $L(M_1) \subseteq L(M_2)$

□

Satz 2.15 (Antwort auf Frage (3))**.** *Sei r ein regulärer Ausdruck über Σ. Dann existiert ein regulärer Ausdruck r' mit $L(r') = \Sigma^* \setminus L(r)$.*

BewIdee:
r regulärer Ausdruck

$\Longrightarrow$ es gibt NFA M mit $L(M) = L(r)$ (Satz 2.13)

$\Longrightarrow$ es gibt DFA M' mit $L(M') = L(M)$ (Satz 2.10)

$\Longrightarrow$ es gibt DFA M'' mit $L(M'') = \Sigma^* \setminus L(M')$ (Lemma 2.4)

$\Longrightarrow$ es gibt regulären Ausdruck r' mit $L(r') = L(M'')$ (Satz 2.13)

Also gilt $L(r') = L(M'') = \Sigma^* \setminus L(M') = \Sigma^* \setminus L(M) = \Sigma^* \setminus L(r)$. □

Zusammenfassung Für eine Sprache L sind äquivalent:

- L ist regulär
- L ist Sprache eines DFA
- L ist Sprache eines NFA
- L ist Sprache eines regulären Ausdrucks

2.4 Nicht-reguläre Sprachen

Fragen:

(1) Existiert eine Sprache, die nicht regulär ist?

(2) Wir haben viele Möglichkeiten, die Regularität einer Sprache zu zeigen (DFA, NFA, regulärer Ausdruck, Komplement einer bekanntermaßen regulären Sprache ...).

Aber wie kann man zeigen, daß eine Sprache nicht regulär ist?

Definition. für $w \in \Sigma^*$ und $n \in \mathbb{N}$ ist $w^n = \underbrace{w\,w\cdots w}_{n \text{ mal}}$, insbes. $w^0 = \varepsilon$ und $w^1 = w$

Satz 2.16 (Pumpinglemma für reguläre Sprachen)**.** *Sei $L \subseteq \Sigma^*$ regulär. Dann existiert $n > 0$, so daß für alle $z \in L$ mit $|z| \geq n$ gilt: es existieren Wörter $u, v, w \in \Sigma^*$ mit*

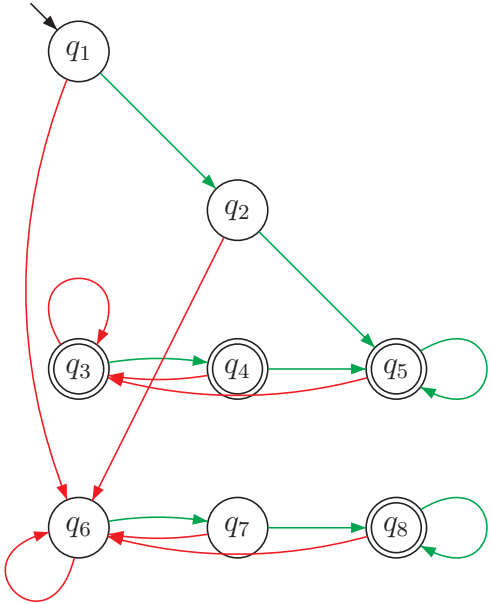
(i) $z = uvw$,

(ii) $|uv| \leq n$,

(iii) $|v| > 0$ *und*

(iv) *für alle* $m \in \mathbb{N}$ *gilt* $uv^m w \in L$.

Beweis durch Beispiel:
Sei L die Sprache des folgenden DFA mit 8 Zuständen (er akzeptiert die Menge aller Wörter, die mit gg beginnen oder mit gg enden):



zu zeigen ist:

$$\exists n \geq 1\, \forall z \in L \text{ mit } |z| \geq n\, \exists u, v, w \in \Sigma^* \colon z = uvw, |uv| \leq n, |v| > 0, \forall m \colon uv^m w \in L$$

Setze $n = 8$ (Anzahl der Zustände von M).
Sei $z \in L(M)$ beliebig mit $|z| \geq n$, z.B. $z = rggrgrggrrgg$. Dann existiert $q \in F = \{q_3, q_4, q_5, q_8\}$ mit $\iota \xrightarrow{z} q$ (im Beispiel $q = q_8$). Auf diesem Pfad werden $|z| + 1 > 8$ Zustände besucht (im Beispiel 13), also gibt es Zustand $p \in \{q_1, \cdots, q_8\}$ (im Beispiel $p = q_6$), der sich unter den ersten 9 Zuständen wiederholt.
Also existieren Wörter u, v, w mit

$$z = uvw\,,\ \iota \xrightarrow{u} p \xrightarrow{v} p \xrightarrow{w} q \in F\,,\ |uv| \leq 8 \text{ und } |v| > 0\,.$$

(im Beispiel $u = r$, $v = ggr$ und $w = grggrrgg$)
Sei nun $m \geq 0$ beliebig. Dann gilt

$$\iota \xrightarrow{u} p \xrightarrow{v^m} p \xrightarrow{w} q \in F\,,$$

d.h. $uv^m w \in L(M)$. □

Mit dem Pumpinglemma für reguläre Sprachen kann gezeigt werden, daß bestimmte Sprachen nicht regulär sind.

Beispiel. Sei $\Sigma = \{a, b\}$ und $L = \{a^k b^k \mid k \geq 0\}$. Die Sprache L ist nicht regulär.

Beweis:
Angenommen, L wäre regulär. Nach dem Pumpinglemma existiert dann $n \geq 1$, so daß für alle $z \in L$ mit $|z| \geq n$ Wörter $u, v, w \in \Sigma^*$ existieren mit den Eigenschaften (i)-(iv) aus dem Pumpinglemma.
Betrachte das Wort $z = a^n b^n \in L$. Wegen $|z| = 2n \geq n$ existieren $u, v, w \in \Sigma^*$ mit (i)-(iv). Damit können wir folgern:

- $uvw = z = a^n b^n$ (d.h. (i)) und $|uv| \leq n$ (d.h. (ii)) $\Longrightarrow |uv|_b = 0 \Longrightarrow |v|_b = 0$
- $|v| > 0$ (d.h. (iii)) $\Longrightarrow |v|_a > 0$

Mit $m = 0$ gelten also $|uv^m w|_b = |uw|_b = |uvw|_b = n$ und $|uv^m w|_a = |uw|_a < |uvw|_a = n$. Damit folgt aber $uv^m w \notin L$, im Widerspruch zu (iv).
Also ist L nicht regulär. □

Beispiel. Ähnlich kann man zeigen, daß die folgenden Sprachen nicht regulär sind:

- $\{a^i b a^j b a^{i+j} \mid i, j \geq 0\}$
- $\{a^{2^i} \mid i \geq 0\}$
- $\{a^p \mid p \text{ ist Primzahl}\}$
- $\cdots$

Beispiel. Die Sprache $\{a^i b^j c^k \mid i = 0, j, k \text{ beliebig oder } i > 0, j = k\}$ ist nicht regulär, dies kann aber nicht mit dem Pumpinglemma gezeigt werden.

also Das Pumpinglemma gibt ein notwendiges, aber kein hinreichendes Kriterium für die Regularität einer Sprache an. Es kann daher nur genutzt werden um zu zeigen, daß eine Sprache nicht regulär ist.