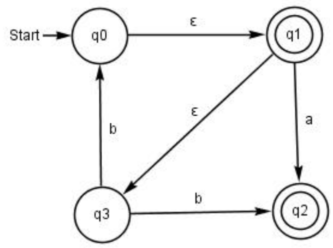
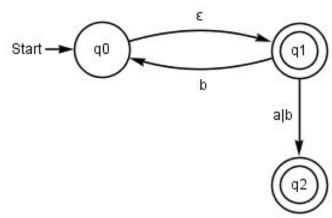
1 Eliminieren von ϵ-Übergängen

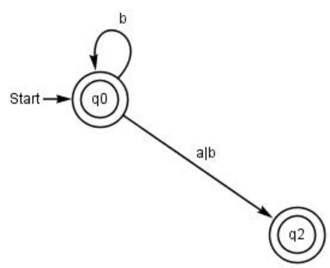
1.1 Beispiel 1



(a) Ausgangspunkt: Zwei ϵ-Übergänge



(b) Entfernung eines ϵ-Übergangs, Reduktion



(c) Entfernen eines ϵ-Übergangs, Reduktion

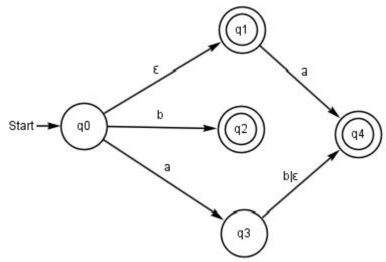
Abbildung 1: Elimination von ϵ-Übergängen, Beispiel 1

Der Automat in (1a) besitzt zwei ϵ-Übergänge, die es zu beseitigen gilt. In (1b) wurde dazu eine neue Kante b vom Zustand $q1$ zu $q0$ erstellt. Dadurch fallen die ϵ-Kante von $q1$ zu $q3$ und die b-Kante von $q3$ zu $q0$ weg. Danach erstellt man eine Kante b von $q1$ zu $q2$ und ersetzt damit die Kante b von $q3$ zu $q2$. Der Zustand $q3$ fällt weg, da er nicht mehr erreichbar ist

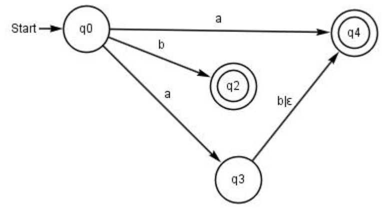
(1c) zeigt den Automaten nach Entfernung der letzten ϵ-Kante. Es wird eine Kante b von $q0$ auf sich selbst erstellt und der Zustand $q0$ wird zu einem Endzustand gemacht. Eine neue Kante $a|b$ von $q0$ zu $q2$ wird erstellt. Dadurch fällt die Kante ϵ von $q0$ zu $q1$, die Kante b von $q1$ zu $q0$ und die Kante $a|b$ von $q1$ zu $q2$ weg. Da $q1$ nicht mehr erreichbar ist fällt dieser Zustand auch weg.

1.2 Beispiel 2

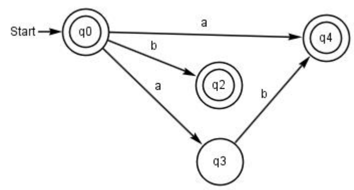
In Abbildung 2a sehen wir den Ausgangsautomaten. Erkannt werden ϵ, a, b und ab. Der erste ϵ-Übergang wurde in 2b enfernt. Der Zustand $q1$ wurde entfernt



(a) Ausgangspunkt: Zwei ϵ-Übergang



(b) Entfernen eines ϵ-Übergangs, Reduktion



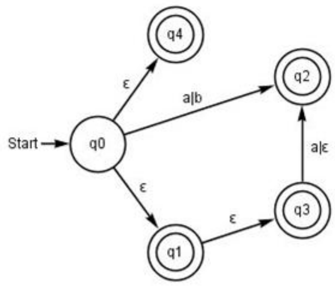
(c) Entfernen eines ϵ-Übergangs

Abbildung 2: Elimination von ϵ-Übergängen, Beispiel 2

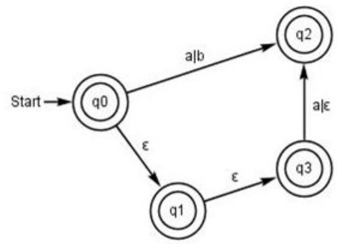
und die a-Kante zwischen $q0$ und $q4$ wurde ergänzt. Damit es sich immer noch um einen äquivalenten Automaten handelt, der auch noch ϵ lesen kann, muss der Startzustand $q0$ gleichzeitig zu einem Endzustand werden. Jetzt kann auch das letzte ϵ gelöscht werden und der Automat ist immer noch äquivalent zu dem ersten (2c).

1.3 Beispiel 3

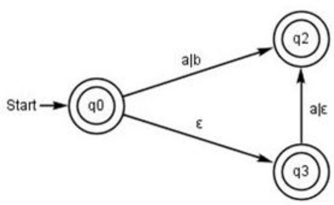
Der Automat in Abbildung 3a erkennt das leere Wort, das Wort a und das Wort b. In 3b wurde die ϵ-Kante von $q0$ nach $q4$ entfernt und q0 zum Endzustand gemacht. 3c zeigt den Automaten nach Entfernen der ϵ-Kante von $q1$ nach $q3$. Anschließend wird die ϵ-Kante von $q0$ nach $q3$ entfernt. Schlussendlich sehen wir in 3e den Automaten, nachdem die ϵ-Kante von $q0$ nach $q2$ entfernt wurde.



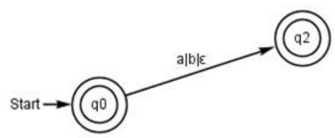
(a) Ausgangspunkt



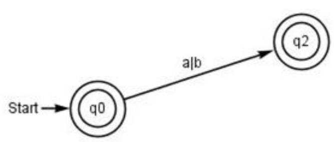
(b) Entfernen eines ϵ-Übergangs, Reduktion



(c) Entfernen eines ϵ-Übergangs, Reduktion



(d) Entfernen eines ϵ-Übergangs, Reduktion



(e) Entfernen eines ϵ-Übergangs

Abbildung 3: Eliminiation von ϵ-Übergängen, Beispiel 3

2 Abschlusseigenschaften regulärer Sprachen

Reguläre Sprachen gelten unter bestimmten Operationen als *abgeschlossen*. Wendet man eine dieser Operationen auf reguläre Sprachen an, so ist das Ergebnis wiederum eine reguläre Sprache. In den folgenden Abschnitten wollen wir zeigen, warum dies bei den Operationen Vereinigung, Verkettung, Stern, Komplementbildung und Schnitt der Fall ist.

2.1 Vereinigung ($X \cup Y$)

$L(R1) = L1$, $L(R2) = L2$; $L1 \cup L2 = L(R1 + R2)$

Die Klasse der regulären Sprachen ist gegen die (endliche) Vereinigung abgeschlossen, d.h. mit $L1$ und $L2$ ist auch die Vereinigung von $L1$ und $L2$ regulär. Eine Eingabe wird akzeptiert, wenn sie für mindestens eine der beiden Sprachen akzeptiert wird. Die regulären Sprachen $L1$ und $L2$ lassen sich durch die regulären Ausdrücke $R1$ und $R2$ beschreiben. Da nun $L1 \cup L2$ durch den regulären Ausdruck $R1 + R2$ beschrieben werden kann, ist das Ergebnis ebenfalls eine reguläre Sprache.

$L(R1 + R2) = L(R1) \cup L(R2)$

2.2 Verkettung ($X \circ Y$)

$L(a \bullet b) = L(a) \circ L(b)$

Die Verkettung zweier regulärer Sprachen wird auch Konkatenation genannt. Um zu zeigen, dass reguläre Sprachen unter Verkettung abgeschlossen sind, betrachtet man statt Symbolen aus dem Alphabet Wörter aus der Sprache. Wenn $L1$ und $L2$ regulär sind, dann auch $L1 \circ L2$. Beschreibt man die Sprachen durch Automaten, so kann man die Endzustände des Automaten für $L1$ mittels ϵ-Übergang zum Startzustand des Automaten für $L2$ verknüpfen. Endzustände dieses Automaten sind dann alle Endzustände von $L2$. Der entstandene Automat ist immer noch ein DEA, damit beschreibt er eine reguläre Sprache.

Beispiel:
$\Sigma = \{a, b\}$
$\{a\} \circ \{ab\} = \{aab\}$
$\{a, bb\} \circ \{aa, b\} = \{aaa, ab, bbaa, bbb\}$
$\{abb, bab\} \circ \{\epsilon, aab, bb\} = \{abb, bab, abbaab, babaab, abbbb, babbb\}$

2.3 Stern ($X^\star$)

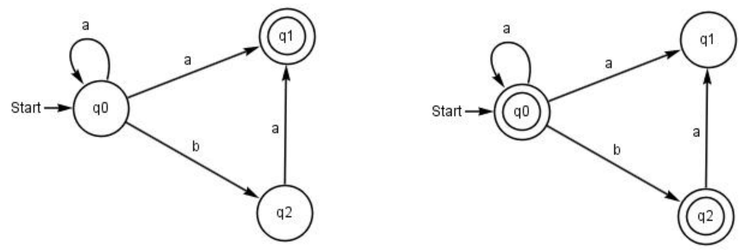
Der Kleen'sche Stern ist eine beliebig häufige Verkettung einer Sprache L mit sich selbst. Es genügt eine Kopie eines Automaten für L, in der wir in allen akzeptierenden Zuständen eine ϵ-Bewegung zum Anfangszustand erlauben. Immer wenn eine derartige ϵ-Bewegung ausgeführt wird, ist ein Teilwort aus L gelesen. Es werden also genau die Wörter aus L^+ akzeptiert. Für $L*$ führen wir einen neuen Startzustand ein, der zugleich Endzustand ist und mittels ϵ-Übergang in den ursprünglichen Startzustand wechselt. Folglich: Ist L regulär, dann ist auch $L^\star$ regulär.

2.4 Komplement ($\overline{X}$)

Um zu zeigen, dass reguläre Sprachen unter Komplementbildung abgeschlossen sind, ist es sehr praktisch, auf die Äquivalenz zwischen regulären Sprachen und endlichen Automaten (EA) zurückzugreifen. Fassen wir einen EA als die Menge an Wörtern auf, die von ihm erkannt wird, dann ist sein Komplement genau die Menge von Wörtern, die *nicht* von ihm erkannt wird. In beiden Fällen liegt den Wörtern das identische Alphabet zugrunde.

Ein EA erkennt ein Wort immer genau dann, wenn er sich nach der Verarbeitung des letzten Symbols in einem Endzustand befindet. Äquivalent dazu erkennt er genau diejenigen Wörter *nicht*, nach deren Verarbeitung er sich in *keinem* Endzustand befindet. Um einen EA X zu erzeugen, der alle und nur die Wörter erkennt, die ein EA Y nicht erkennt, reicht es daher scheinbar, das Komplement der Menge an Endzuständen von Y als Menge von Endzuständen von X zu wählen.

Der endliche Automat Y muss hierbei allerdings zwei wichtige Bedingungen erfüllen. Zunächst muss er deterministisch sein, damit wir beim "Umdrehen" der Endzustände keine Wörter erkennen, die vorher auch erkannt wurden. Ein Beispiel hierfür zeigt Abbildung 4. Beide Automaten, 4a und 4b, erkennen das Wort a.



(a) Ausgangspunkt (b) Vertauschte Endzustände

Abbildung 4: Vertauschen der Endzustände - a wird noch erkannt

Die nächste Bedingung, die erfüllt sein muss, lässt sich intuitiv erschließen: Der ursprüngliche Automat muss in der Lage sein, alle Ketten aus dem Alphabet zu verarbeiten, d.h. die Übergangsfunktion δ muss vollständig sein. Entfernt man "unnötige" Zustände, definiert man gleichzeitig Wörter, die nicht erkannt werden können. Genau diese muss aber unser Komplement-Automat erkennen. Ist δ nicht vollständig, kann er dies aber im Allgemeinen nicht, da ihm dazu notwendige Übergänge fehlen.

Die Vervollständigung von δ lässt sich am einfachsten dann bewerkstelligen, wenn man sie tabellarisch darstellt. Wir führen einen neuen Zustand q_{Neu} ein und tragen diesen in alle leeren Zellen der Tabelle ein. Abbildung 5 zeigt dies beispielhaft.

	a	b
q_1	q_2	
q_2	q_3	q_2
q_3		

$\xrightarrow{q_{Neu}}$

	a	b
q_1	q_2	q_{Neu}
q_2	q_3	q_2
q_3	q_{Neu}	q_{Neu}
q_{Neu}	q_{Neu}	q_{Neu}

Abbildung 5: Vervollständigen der Übergangsfunktion für $ab^{\star}a$

2.5 Schnitt $(X \cap Y)$

Glücklicherweise lässt sich für $\cap$ sehr einfach zeigen, dass reguläre Sprachen darunter abgeschlossen sind. Wir nutzen die deMorgan'schen Gesetze für Mengen aus und erkennen, dass $\overline{A \cap B} \Leftrightarrow \overline{A} \cup \overline{B}$. Damit ist $A \cap B \Leftrightarrow \overline{(\bar{A} \cup \bar{B})}$. Intuitiv heißt dies: Wir ermitteln den Schnitt zweier Sprachen A und B, indem wir alle Wörter nehmen, außer denen, die nicht in A oder nicht in B sind.