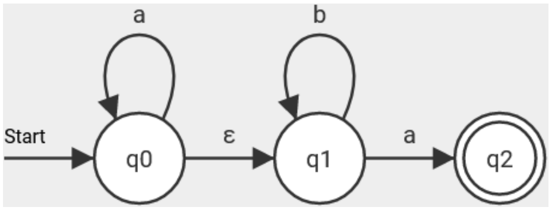
NEA mit Epsilon-Übergängen

In der theoretischen Informatik gibt es eine Erweiterung des Konzepts der nicht-deterministischen endlichen Automaten um sogenannte **Epsilon-Übergänge**. Ein solcher Übergang erlaubt dem Automaten, den Zustand zu wechseln, **ohne** ein Zeichen des Eingabewortes zu lesen.
D.h. für das aktuelle Zeichen kann zweimal der Zustand gewechselt werden, einmal mit dem Epsilon-Übergang (ohne das Zeichen zu verbrauchen), und ein zweites Mal, bei dem das Zeichen dann „gelesen“ wird und anschließend das nächste Zeichen des Eingabewortes an der Reihe ist.

Der abgebildete NEA erkennt z.B. die Sprache
L = { $a^n b^m a$ | $n \geq 0$, $m \geq 0$ }:

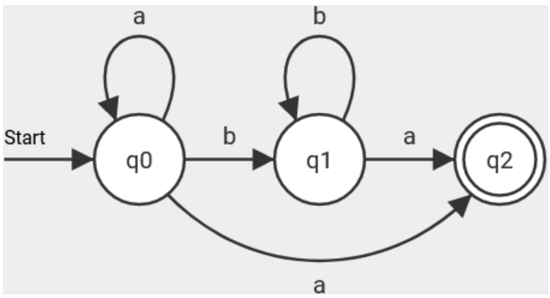


Bei Eingabe „aba“ würde der Automat zuerst a lesen und in q_0 bleiben.
Für den nächsten Buchstaben b gibt es von q_0 keinen Übergang.
Der ε-Übergang von q_0 nach q_1 erlaubt aber, nach q_1 zu wechseln, ohne b zu „lesen“, d.h. der Automat kann einfach nach q_1 springen. Dort wird dann b gelesen und man bleibt in q_1.
Mit dem nächsten Buchstaben a geht es dann wie gewohnt in q_2, und das Wort wird akzeptiert.

Endliche Automaten mit ε-Übergängen sind **immer nicht-deterministisch**, da es für jeden ε-Übergang grundsätzlich die Möglichkeit gibt, im Ausgangszustand zu bleiben, oder in den Zielzustand zu wechseln. Man hat also eine Wahl, die es bei einem deterministischen Automaten nicht geben dürfte.

Jeder NEA mit ε-Übergängen lässt sich leicht in einen **NEA ohne ε-Übergänge** überführen.
Man überlegt lediglich für den Ausgangszustand jedes ε-Übergangs, mit welchen Zeichen man vom Zielzustand weitergehen könnte, und fügt entsprechend Übergänge mit diesen Zeichen hinzu.

Der abgebildete NEA ohne ε-Übergänge erkennt die gleiche Sprache:



Welchen **Nutzen** haben dann NEA mit ε-Übergängen, wenn man für die gleiche Sprache auch einen NEA ohne ε-Übergänge angeben kann?

NEA mit ε-Übergängen sind für bestimmte **Beweise** im Gebiet der formalen Sprachen hilfreich, sie haben also nur einen theoretischen Nutzen. Auf der folgenden Seite ist ein solcher Beweis als Beispiel aufgeführt.

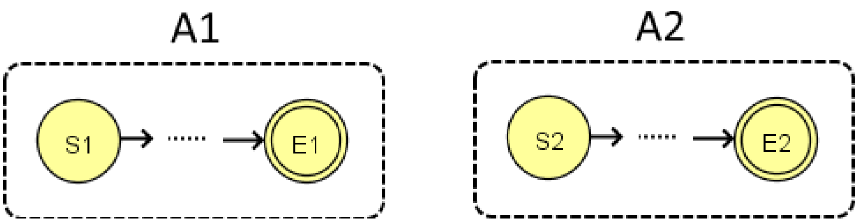
Beispiel: Beweis der Abgeschlossenheit über Vereinigung

Aus Sicht der theoretischen Informatik ist es interessant, bestimmte Eigenschaften von Sprachklassen zu zeigen. Dieses Bedürfnis entspringt den mathematischen Wurzeln der Informatik.

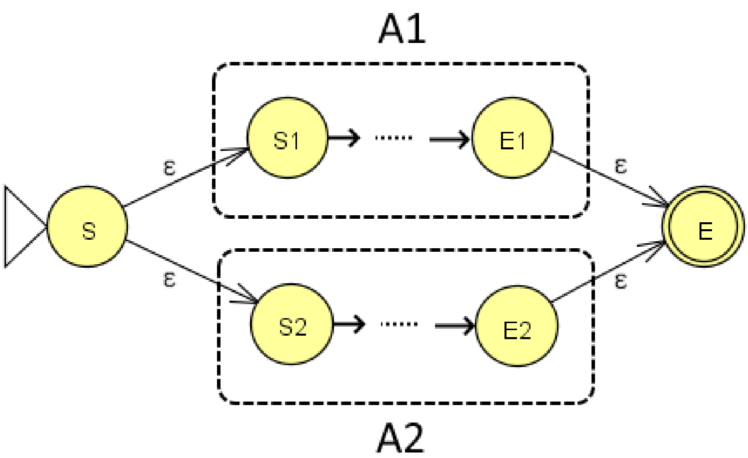
Ein typisches Beispiel wäre, zu zeigen, dass jede **Vereinigung** zweier regulärer Sprachen ebenfalls eine reguläre Sprache ist.
Wenn also z.B. die Sprache $L_1 = \{ (ab)^n \}$ regulär ist, und $L_2 = \{ a^m b^n \mid n, m > 0 \}$ regulär, dann ist auch $L_1 \cap L_2 = \{ w \mid w = (ab)^n \text{ oder } w = a^m b^n, n, m > 0 \}$ regulär.

Der Beweis, dass die Vereinigung zweier beliebiger regulärer Sprachen immer regulär ist, ist recht einfach. Eine hinreichende Bedingung für das Kriterium „regulär“ ist, dass es einen endlichen Automaten (deterministisch oder nicht-deterministisch) gibt, der die Sprache erkennt.

Wenn zwei beliebige Sprachen L_1 und L_2 regulär sind, gibt es also zwei endliche Automaten A_1 bzw. A_2., die diese Sprachen erkennen. Diese Automaten sind in der Abbildung schematisch dargestellt: Sie haben jeweils einen Startzustand und einen (oder mehrere) Endzustände, und dazwischen weitere Zustände sowie Übergänge.



Jetzt ist es leicht, einen Automaten A_V zu konstruieren, der die Vereinigung der beiden Sprachen L_1 und L_2 erkennt – indem man ε-Übergängen verwendet:



Vom Startzustand kann man, ohne ein Zeichen zu lesen, einen der beiden Startzustände von A_1 oder A_2 wählen, und dann den jeweiligen Automaten das Wort komplett bearbeiten lassen. Falls man am Ende des Wortes in einem Endzustand von A_1 bzw. A2 landet, geht man mit dem ε-Übergang in den Endzustand des gemeinsamen Automaten AV und akzeptiert das Wort.

Damit gibt es also einen NEA, der die Sprache $L_1 \cap L_2$ akzeptiert, und damit ist $L_1 \cap L_2$ regulär.