Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Christian Ivicevic

Sommersemester 2015
Studentisches Tutorium
Übungsblatt 3
4. Mai 2015

Theoretische Informatik

– Myhill-Verfahren, Reguläre Ausdrücke, Thompson-Konstruktion, Sättigung –

Myhill-Verfahren (NFA $\rightarrow$ DFA)

Das Myhill-Verfahren bzw. die Potenzmengenkonstruktion wird genutzt um einen NFA in einen DFA umzuwandeln. Die Existenz und Korrektheit dieses Verfahrens sind übrigens (in Kombination mit den anderen Algorithmen dieser Woche) der Beweis dafür, dass ein DFA die gleichen Sprachen akzeptiert, wie ein NFA. Das wesentliche Konzept ist hierbei die Konstruktion von Potenzmengen um Ähnlichkeiten zwischen Zuständen festzustellen und diese dann miteinander in Relation zu setzen. Das Verfahren wird sogar als Grundlage bei der Minimierung von Automaten genutzt, weshalb es besonders wichtig ist, dieses zu verinnerlichen. Um diese Ähnlichkeiten zu finden und einen Automaten zu determinisieren simulieren wir quasi jeden Schritt des NFAs und halten fest, in welchen Zuständen er sich gleichzeitig befinden kann.

Satz. *Sei $N = (Q, \Sigma, \delta, \{q_0\}, F)$ ein NFA. Mit dem Myhill-Verfahren (bzw. der Potenzmengenkonstruktion) lässt sich ein äquivalenter DFA $M = (Q', \Sigma, \delta', q_0', F')$ konstruieren mit $L(M) = L(N)$. Der zugehörige Algorithmus lautet:*

Algorithmus 1 Myhill-Verfahren

1: $Q' = F' = \emptyset$
2: $q_0' = \{q_0\}, Q' = Q' \cup \{q_0'\}$
3: **für alle** $q' = \{q_{n_1}, \ldots, q_{n_k}\} \in Q'$:
4: **für alle** $s \in \Sigma$:
5: $q'' = \bigcup\{\delta(r, s) \mid r \in q'\} = \delta(q_{n_1}, s) \cup \ldots \cup \delta(q_{n_k}, s)$
6: $Q' = Q' \cup \{q''\}$
7: $\delta'(q', s) = q''$
8: **wiederhole** ab Zeile 3, bis Q' und δ' sich nicht mehr ändern
9: $F' = \{q' \in Q \mid q' \cap F \neq \emptyset\}$

Beispiel 1. Aufgrund der enormen Komplexität von $\mathcal{O}\left(2^{|Q|}\right)$ des Algorithmus werden wir in diesem Beispiel nur vier Zustände des DFA bestimmen, deren Herleitung die wichtigsten Fälle umfassen. Hierzu betrachten wir den NFA aus der Abbildung 1. Zu Beginn des Algorithmus findet ein Initialisierung in den Zeilen 1 und 2 statt, in der wir einen Startzustand des DFA aus dem Startzustand des NFA erzeugen (Abbildung 2). Nun iterieren wir in Zeile 3 über sämtliche vorhandenen Zustände und wählen $q' = \{q_0\}$ aus.

Für jedes Symbol im Alphabet (Zeile 4) werden wir Zustandsübergänge berechnen und diese im Automaten ergänzen. Wählen wir zunächst das Symbol $s = 1$ so berechnen wir einen neuen Zustand $q'' = \bigcup\{\delta(r, s) \mid r \in q'\} = \delta(q_0, 1) = \{q_1\}$ (Zeile 5) und fügen diesen dem Automaten hinzu (Zeile 6 und 7). Nun wiederholt man dies natürlich für $s = 0$, was wir an dieser Stelle abkürzen, da dies analog verläuft. Das aktuelle Zwischenergebnis ist Abbildung 3 zu entnehmen.

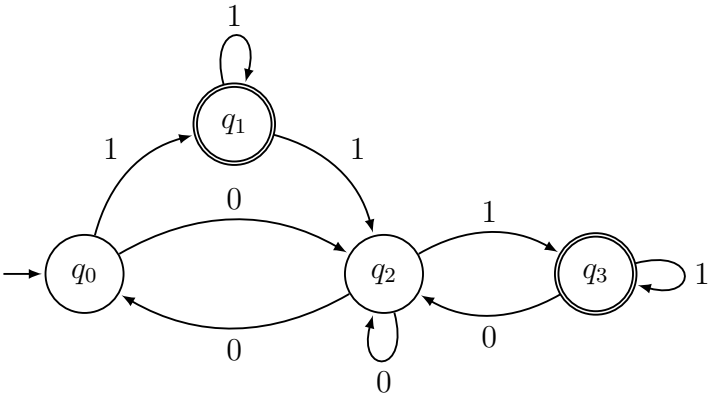


Abbildung 1: Willkürlicher NFA, der im Rahmen der Übung umgewandelt wird

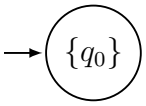


Abbildung 2: Schritt 1 des Myhill-Verfahren

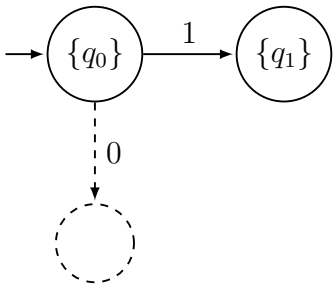


Abbildung 3: Schritt 2 des Myhill-Verfahren

Wir setzen das Verfahren nun mit $q' = \{q_1\}$ fort und betrachten zunächst $s = 0$ und stellen fest, dass $q'' = \bigcup\{\delta(r, s) \mid r \in q'\} = \delta(q_1, 0) = \emptyset$ gilt, da im ursprünglichen NFA ein Übergang in dem Fall undefiniert ist. Daher fügen wir $\emptyset$ als Fangzustand ein. Für $s = 1$ gilt $q'' = \bigcup\{\delta(r, s) \mid r \in q'\} = \delta(q_1, 1) = \{q_1, q_2\}$ und dieser Zustand wird ebenfalls dem Automaten hinzugefügt (Abbildung 4). Der letzte interessante Fall, der auftreten kann ist bei der Wahl von $q' = \{q_1, q_2\}$ mit $s = 1$. Zur Bestimmung von q'' müssen wir δ mehrmals für jeden Zustand in q' einzeln auswerten aus das Ergebnis zusammenfassen. Es gilt nun $q'' = \bigcup\{\delta(r, s) \mid r \in q'\} = \delta(q_1, 1) \cup \delta(q_2, 1) = \{q_1, q_2\} \cup \{q_3\} = \{q_1, q_2, q_3\}$ und wir erhalten den Automaten aus Abbildung 5. Die restlichen Zustände können analog hergeleitet

werden bis man schließlich einen DFA ohne Endzustände erhält. Zeile 9 des Algorithmus besagt hierzu, dass jeder Zustand $\{q_{n_1}, \ldots, q_{n_k}\}$ im DFA als Endzustand markiert werden soll, sofern mindestens ein $q_i \in F$. Konkret markieren wir hier alle Zustände q' für die $q_1 \in q'$ oder $q_3 \in q'$ gilt und erhalten schlussendlich den kompletten DFA aus Abbildung 6. Um sich viel Schreibarbeit zu sparen wird teils eine Alternative zur Mengenschreibweise genutzt. Hierbei wird beispielweise der Zustand $\{q_1, q_2, q_3\}$ zusammengefasst zu $q_1q_2q_3$ oder auch q_{123}. Es ist empfehlenswert stets in einem Satz zu erwähnen, wie genau man selber einen Zustand notiert, damit ein Korrektor keine Probleme hat.

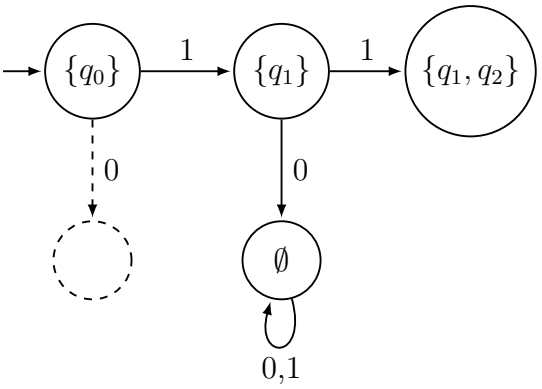


Abbildung 4: Schritt 3 des Myhill-Verfahren

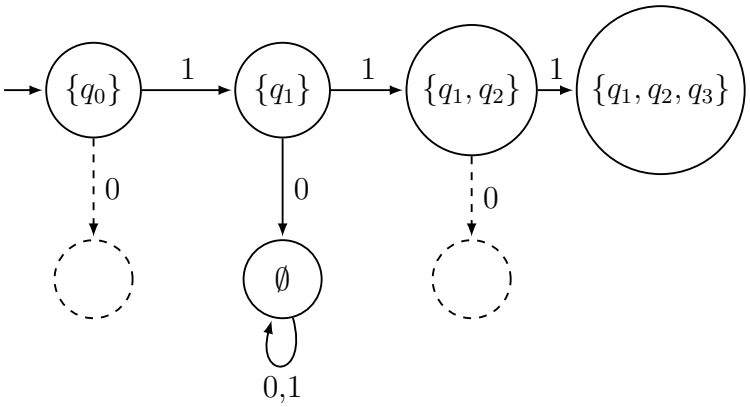


Abbildung 5: Schritt 4 des Myhill-Verfahren

Reguläre Ausdrücke

Wir führen an dieser Stelle das Konzept regulärer Ausdrücke ein, um die Struktur von Wörtern konkreter zu spezifizieren. Solche Ausdrücke werden meist beim Programmieren verwendet um gegebene Eingaben syntaktisch zu validieren. So kann man beispielweise bei Online-Registrierungen überprüfen, ob eine E-Mail-Adresse die korrekte Form hat um komplett unsinnige Eingaben zu umgehen. Ob es sich bei dieser Eingabe um eine aktive Adresse handelt lässt sich zwar nur mit Bestätigungs-Mails prüfen, jedoch kann man bereits vorher erste Vorkehrungen treffen um fehlerhafte Eingaben abzufangen.

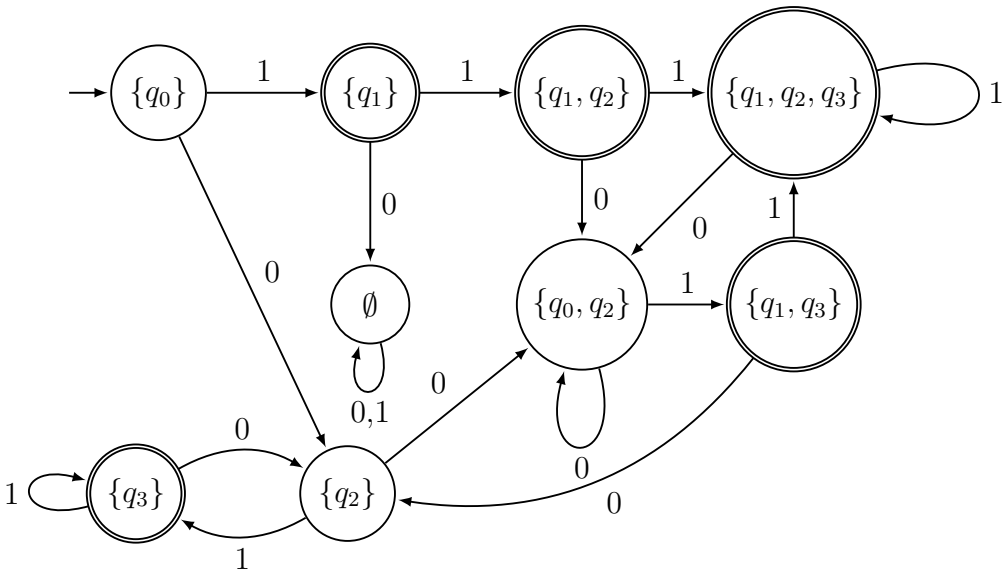


Abbildung 6: Ergebnis des Myhill-Verfahren

Definition (Regulärer Ausdruck)**.** *Wir bezeichnen die atomaren Objekte $\emptyset$, ε und $a \in \Sigma$ als* regulären Ausdruck *(RE). Sofern α und β reguläre Ausdrücke sind, dann ist auch die Konkatenation $\alpha\beta$, die Alternative $\alpha \mid \beta$, die Klammerung (α) sowie der Kleene-Star α^* als beliebige Wiederholung ein RE. Analog wird die von einem RE erzeugte Sprache induktiv definiert:*

$$L(\emptyset) = \emptyset \qquad L(\varepsilon) = \{\varepsilon\} \qquad L(a) = \{a\} \text{ mit } a \in \Sigma$$
$$L(\alpha\beta) = L(\alpha)L(\beta) \qquad L(\alpha \mid \beta) = L(\alpha) \cup L(\beta) \qquad L(\alpha^*) = L(\alpha)^*$$

Beispiel 2. Wir betrachten die Sprache $L = \{w \in \Sigma^* \mid w \text{ beginnt und endet mit } 0\}$ und finden den zugehörigen regulären Ausdruck $\alpha = 0 \mid 0(0 \mid 1)^*0$ für den $L(\alpha) = L$ gilt.

Jeder reguläre Ausdruck lässt sich in einen Automaten umwandeln mithlife der *Thompson-Konstruktion*. Diese Konstruktion beschreibt induktiv wie ein regulärer Ausdruck mit einem Teil-NFA, einem Baustein, beschrieben werden kann, der mit weiteren verknüpft werden kann. Hierzu führen wir ein Hilfssymbol ein.

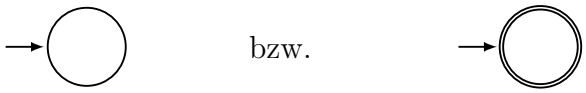
Definition (ε-NFA)**.** *Ein ε-NFA $N = (Q, \Sigma, \delta, S, F)$ ist ein NFA für dessen Zustandsübergangsrelation zusätzlich $\delta : Q \times \Sigma \cup \{\varepsilon\} \to \mathcal{P}(Q)$ gilt, also die Übergänge mit einem ε als Hilfssymbol markiert sein können.*

Thompson-Konstruktion (RE $\to$ ε-NFA)

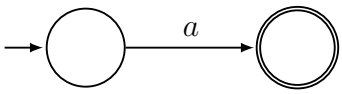
Satz (Kleene)**.** *Eine Sprache $L \subset \Sigma^*$ ist genau dann durch einen regulären Ausdruck darstellbar, wenn sie regulär ist.*

Beweis. In diesem Beweis wollen wir ausschließlich die Hinrichtung zeigen, das heißt, dass jeder RE eine reguläre Sprache beschreibt indem wir die angesprochenen Teil-NFAs induktiv angeben. Hat man einen nun einen komplexen RE gegeben so kann dieser zerlegt werden und die untenstehenden Bausteine zu einem größeren ε-NFA zusammengebaut werden.

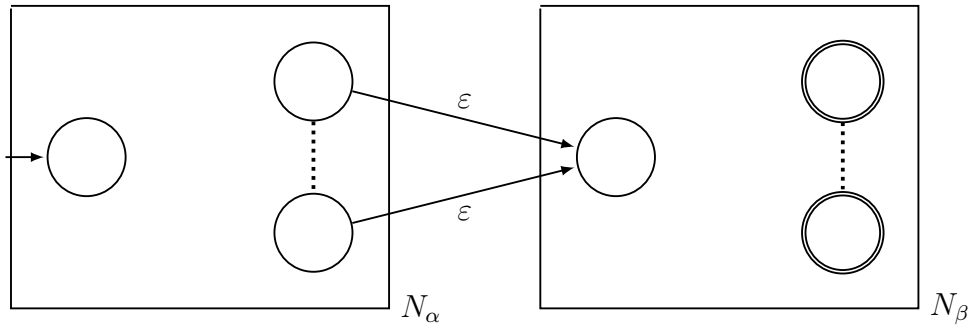
- Sei $\gamma = \emptyset$ bzw. $\gamma = \varepsilon$, dann ist der zugehörige ε-NFA N_γ



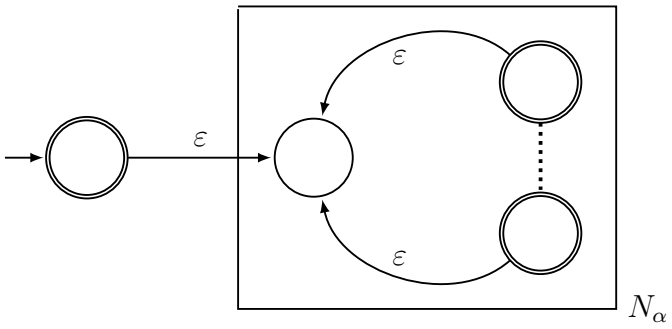
- Sei $\gamma = a \in \Sigma$, dann ist der zugehörige ε-NFA N_γ



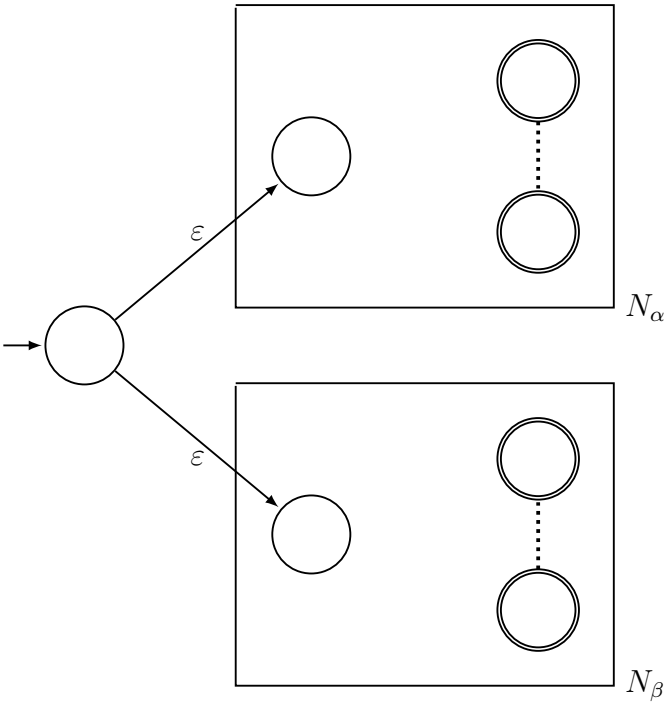
- Sei $\gamma = \alpha\beta$, dann setzt sich der zugehörige ε-NFA N_γ aus den Automaten N_α und N_β zu den Teilausdrücken α und β zusammen. Jeder Endzustand von N_α wird zu einem normalen Zustand und erhält eine ε-Kante zum ursprünglichen Startzustand von N_β



- Sei $\gamma = \alpha^*$, dann setzt sich der zugehörige ε-NFA N_γ aus dem Automaten N_α zu dem Teilausdruck α und weiteren Kanten zusammen. Er wird ein zusätzlicher Zustand hinzugefügt und je eine ε-Kante von jedem Endzustand zum ursprünglichen Startzustand



- Sei $\gamma = \alpha \mid \beta$, dann setzt sich der zugehörige ε-NFA N_γ aus den Automaten N_α und N_β zu den Teilausdrücken α und β zusammen. Es wird ein neuer Zustand eingeführt, der mit ε-Kanten in je einen von beiden Automaten führen kann



Da wir nun für jeden möglichen regulären Ausdruck einen NFA gefunden haben, der diesen korrekt beschreibt ist die Hinrichtung somit bewiesen. □

Beispiel 3. Wir betrachten den RE $\gamma = (0\,(1 \mid 2))^*$ und konstruieren mit der oben beschriebenen Thompson-Konstruktion einen zugehörigen ε-NFA. Dieser Automat hat als grundlegende Struktur den Automaten zum Ausdruck α^* aus dem Beweis von oben. Im Automaten selber ist eine Konkatenation zweier Automaten wobei der zweite Automat sich aus einer Alternative zusammensetzt. Der Automat ist in Abbildung 7 zu sehen.

Beispiel 4. Wir betrachten nun den RE $\gamma = (1\,(0 \mid 1)^*) \mid 0$ und leiten analog den Automaten aus Abbildung 8 her. Erwähnenswert ist der Zustand q_2, der leicht vergessen werden kann[1], obwohl der Baustein für den Kleene-Star diesen vorgibt!

Sättigungsalgorithmus (ε-NFA $\rightarrow$ NFA)

Nachdem man zu einem RE den zugehörigen ε-NFA konstruiert hat kann dieser weiter vereinfacht werden. In diesem Teil schauen wir uns den Algorithmus an um die Hilfssymbole aus dem Automaten entfernen zu können. Die grundlegende Idee ist es, die ε-Kanten unnötig zu machen, damit diese entfernt werden können. Dies erreichen wir, indem wir alle Wege (genauer gesagt Kantenzüge) auf denen ε vorkommt abkürzen durch eine manuelle Simulation. Die drei nötigen Schritte für diesen Algorithmus lauten wie folgt:

[1]Vor einigen Jahren hatte die Übungsleitung diesen vergessen, daher ist noch der Zustand q_2' im Automaten zu sehen als kleiner Beweis, dass wirklich jeder Fehler machen kann.

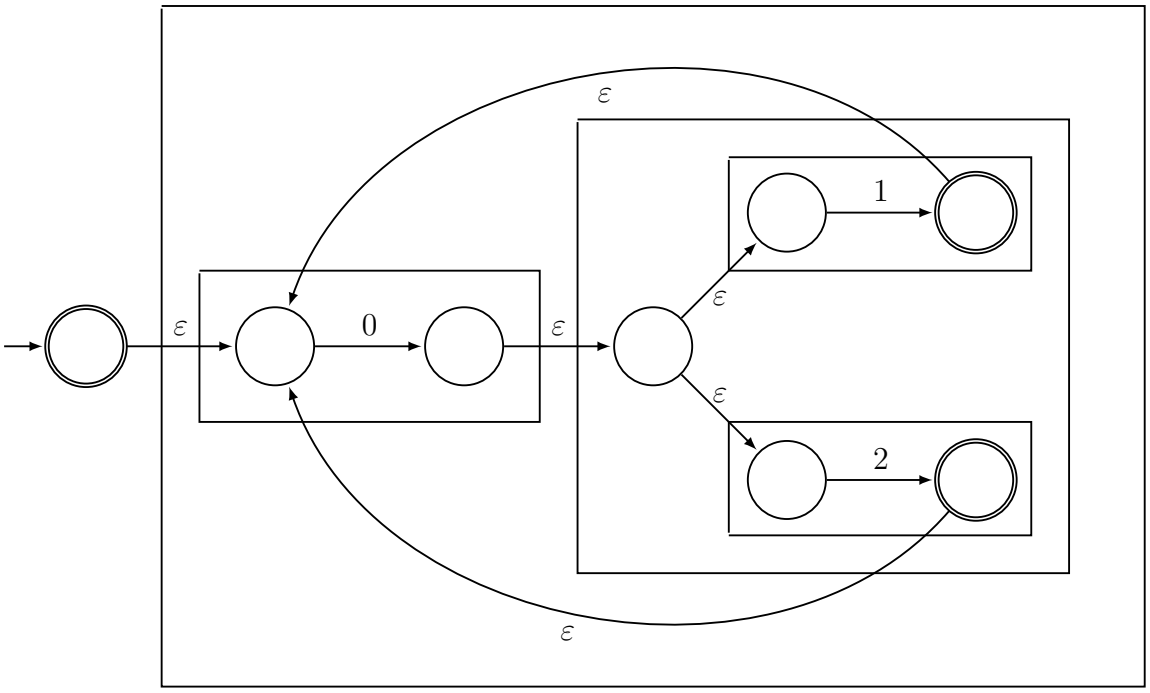


Abbildung 7: ε-NFA zu $(0\,(1\,|\,2))^*$

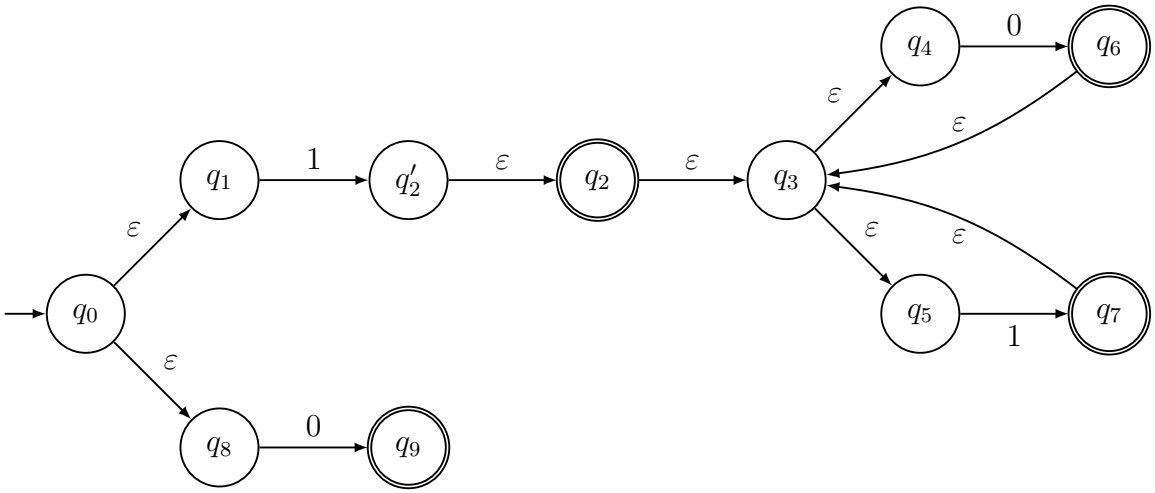


Abbildung 8: ε-NFA zu $(1\,(0\,|\,1)^*)\,|\,0$

1) Entferne überflüssige Knoten, die **KEINE** Endzustände sind und genau eine eingehende und genau eine ausgehende Kante besitzen von denen mindestens eine mit ε markiert ist.

2) Füge *sättigende Kanten* für Kantenzüge $\varepsilon^* a \varepsilon^*$ ein, die Quelle und Senke des Kantenzugs direkt mit $a \in \Sigma$ verbinden.

3) Entferne sämtliche ε-Kanten und mache den Startzustand zu einem Endzustand, sofern ursprünglich das leere Wort ε akzeptiert wurde.

Beispiel 5. Wir werden den beschriebenen Algorithmus auf den Automaten aus Abbildung 8 anwenden. Im ersten Schritt werden die Knoten q_1, q_2', q_4, q_5, q_8 gelöscht und der Automat kollabiert. Das Zwischenergebnis ist Abbildung 9 zu entnehmen. Nun müssen

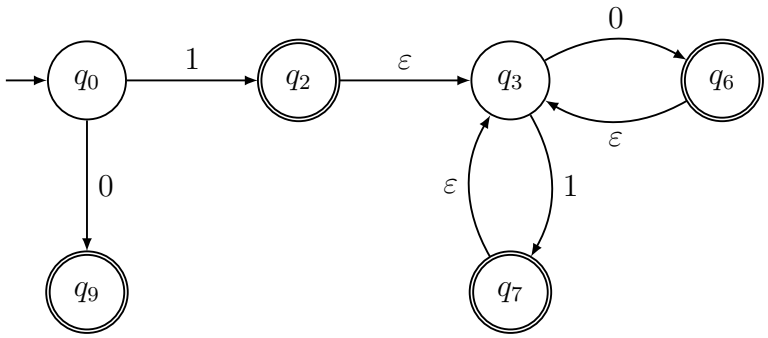


Abbildung 9: ε-NFA zu $(1\,(0\,|\,1)^*)\,|\,0$

längere Kantenzüge verkürzt werden durch die namensgebenden sättigenden Kanten. So erkennt man, dass man von q_0 mit 1ε über q_2 nach q_3 kommt und dies auch direkt können muss, weshalb eine 1-Kante von q_0 nach q_3 hinzugefügt wird. Nach diesem Schema können durch Ausprobieren ganze 11 Kanten hinzugefügt werden (Abbildung 10) und abschließend alle ε-Kanten entfernt werden.

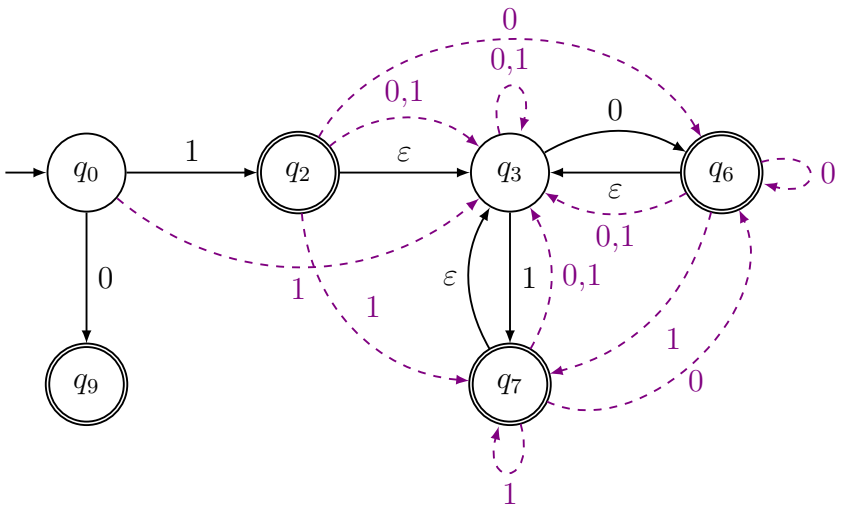


Abbildung 10: ε-NFA zu $(1\,(0\,|\,1)^*)\,|\,0$