Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin
Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Hausaufgabenblatt 2

- Die Hausaufgaben werden in folgender Reihenfolge korrigiert: H2.4, H2.5, H2.6, H2.3

AT-Aufgabe H2.1. (*Dem Tutor, dem die Studis vertrauen*) keine Korrektur

Bearbeiten Sie folgende Aufgaben in Automata Tutor (Aufgaben H2.1 a–f). Konstruieren Sie reguläre Ausdrücke für die folgenden Sprachen über dem Alphabet $\Sigma := \{a, b\}$. Beachten Sie dabei, dass die Bedingungen ab Aufgabenteil (b) sukzessiv kombiniert werden, die Sprache für (d) soll also beispielsweise die Bedingungen von (b) und (c) erfüllen.

(a) $\{(ab)^n a(ba)^n : n \in \mathbb{N}\}$.

(b) Die Sprache aller Wörter, die nicht mit einem a beginnen.

(c) Zusätzlich müssen die Wörter mit einem b enden.

(d) Zusätzlich müssen die Wörter eine ungerade Anzahl von as haben.

(e) Zusätzlich dürfen die Wörter bb nicht enthalten.

(f) Sei $L := L(ab^* \mid a^*b(a \mid b)^*)$. Geben Sie drei Wörter in L und drei Wörter nicht in L an.

Lösungsskizze.

(a) $(abab)^*a$

(b) $\epsilon \mid b(a \mid b)^*$

(c) $b \mid b(a \mid b)^*b$

(d) $bb^*a(ab^*a \mid b)^*b$

(e) $ba((b \mid \epsilon)a(b \mid \epsilon)a)^*b$

(f) $a, ab, abb \in L$, $\varepsilon, aa, aaa \notin L$

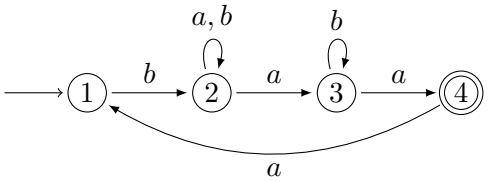
AT-Aufgabe H2.2. (*DFA-Konstruktion*) keine Korrektur

Bearbeiten Sie die Aufgaben 2(a)–(n) (außer (e)) auf Seite 88 des Buches Automata, Computability and Complexity: Theory and Applications in Automata Tutor. Die Beschreibungen der Sprachen auf Deutsch können Sie in Automata Tutor finden.

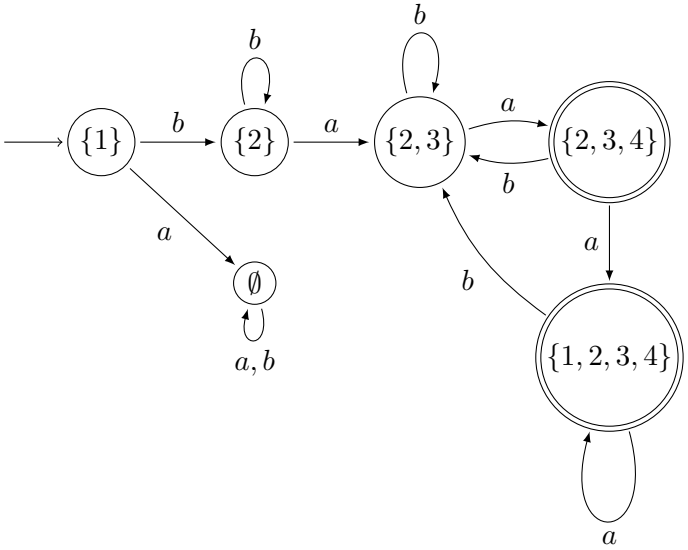
Aufgabe H2.3. (*Potenzmengenkonstruktion*)

Konvertieren Sie den folgenden NFA über dem Alphabet $\Sigma = \{a, b\}$ zu einem DFA, indem Sie die Potenzmengenkonstruktion aus der Vorlesung verwenden.

Hinweis: Es genügt, nur die vom Startzustand erreichbaren Zustände zu konstruieren.



Lösungsskizze.



Aufgabe H2.4. (*Sag a!*)

Sei $\Sigma = \{a, b\}$ und sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA über Σ. Geben Sie einen NFA N' für die Sprache $\{w \in L(N) \mid |w|_a \geq 1\}$ aller Wörter aus $L(N)$, die mindestens ein a enthalten, formal (als Tupel) an.

Hinweis: Sie dürfen $(q, a, r) \in \delta$ statt $r \in \delta(q, a)$ schreiben.

Lösungsskizze. Idee: Wir haben zwei Kopien von N, wobei a-Übergänge aus der ersten Kopie in die zweite Kopie übergehen. Der Startzustand ist der Startzustand der ersten Kopie, die Endzustände sind die Endzustände der zweiten Kopie.

Formal: $N' := (Q', \Sigma, \delta', q_0', F')$ mit

$$
\begin{aligned}
Q' &:= \{q_1, q_2 : q \in Q\} \\
q_0' &:= q_{0_1} \\
F' &:= \{q_2 : q \in F\} \\
\delta' &:= \{(q_1, b, r_1), (q_2, b, r_2) : (q, b, r) \in \delta\} \\
&\quad \cup \{(q_1, a, r_2), (q_2, a, r_2) : (q, a, r) \in \delta\}
\end{aligned}
$$

Aufgabe H2.5. (*Union of the States*)

Sei $\Sigma := \{\texttt{a}, ..., \texttt{z}\}$ und $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA über Σ. Wir nennen M *synchronisierend*, wenn es einen Zustand $q \in Q$ gibt, sodass jeder Zustand nach Lesen des Wortes `theo` in q landet. Formal gilt also $\hat{\delta}(r, \texttt{theo}) = q$ für alle $r \in Q$.

(a) Sei $L := \Sigma^*\{\texttt{theo}\}$, also die Sprache aller Wörter, die auf `theo` enden. Beweisen Sie, dass jeder synchronisierende DFA M alle oder keine Wörter aus L akzeptiert, d.h. $L(M) \cap L \in \{\emptyset, L\}$. Gehen Sie dabei kleinschrittig vor und beziehen sich, wenn möglich, auf die entsprechenden Definitionen.

Wir betrachtet nun auch NFAs. Einen NFA $M = (Q, \Sigma, \delta, q_0, F)$ bezeichnen wir demnach als *synchronisierend*, wenn es ein $q \in Q$ gibt, sodass $\hat{\delta}(\{r\}, \texttt{theo}) = \{q\}$ für jedes $r \in Q$ gilt.

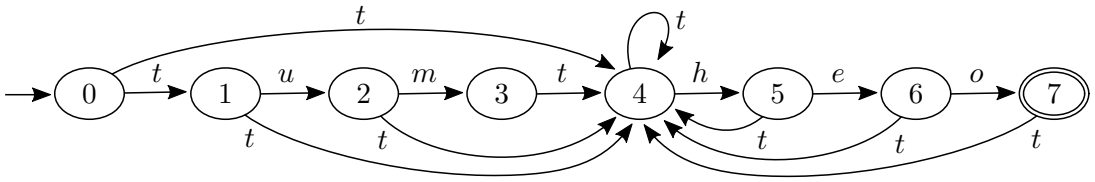
(b) Konstruieren Sie einen synchronisierenden NFA, der das Wort `tumtheo` akzeptiert, aber `lmutheo` nicht. Begründen Sie, wieso Ihr NFA synchronisierend ist.

Update: Sie können diese Aufgabe nun auch in apotheosis lösen.

Lösungsskizze. **(a)** Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein beliebiger synchronisierender DFA, und sei q der entsprechende Zustand, der immer nach Einlesen von `theo` erreicht wird. Sei $w \in L$ ein Wort. Wir zeigen nun, dass $w \in L(M)$ genau dann gilt, wenn $q \in F$. Da q unabhängig von w ist, werden entweder alle Wörter in L akzeptiert, oder keines.

Ein DFA akzeptiert ein Wort genau dann, wenn er nach dem Einlesen in einem Finalzustand ist. Es gilt also $w \in L(M) \Leftrightarrow \hat{\delta}(q_0, w) \in F$. Nach Definition von L können wir w als $w = v\texttt{theo}$ zerlegen. Nun gilt also $\hat{\delta}(q_0, w) = \hat{\delta}(\hat{\delta}(q_0, v), \texttt{theo}) = q$, wobei letzte Gleichung aus der Definition von synchronisierenden DFAs folgt. Damit ist $w \in L(M) \Leftrightarrow q \in F$ gezeigt.

(b)



Nach Lesen von `t` wird entweder genau $\{4\}$ erreicht, oder $\{1, 4\}$. Aus Zustand 4 kommt man nach Lesen von `heo` genau nach $\{7\}$, während Zustand 1 bereits nach Einlesen des `h` keine Transition hat. Es gilt $\hat{\delta}(\{1\}, \texttt{heo}) = \emptyset$ und somit $\hat{\delta}(\{1, 4\}, \texttt{heo}) = \{7\}$.

(Es gibt eine kleinere Lösung mit nur einem Zustand.)

Aufgabe H2.6. (*Regex-Rekursion*)

In den Übungsaufgaben haben wir eine Funktion definiert, die für einen regulären Ausdruck r bestimmt, ob die Sprache $L(r)$ unendlich viele Wörter enthält. Nun wollen wir eine Funktion $\mathsf{contains}(\mathsf{a}, r)$ definieren, die für einen Buchstaben $\mathsf{a} \in \Sigma$ und regulären Ausdruck r berechnet, ob a in jedem Wort aus $L(r)$ vorkommt. Für alle $w \in L(r)$ soll also gelten, dass $(\exists u, v \in \Sigma^*.\ w = u\mathsf{a}v) =: P(w)$.

(a) Geben Sie eine rekursive Funktion an, indem Sie das folgende Gerüst vervollständigen.

- $\mathsf{contains}(\mathsf{a}, \emptyset) = \ldots$
- $\mathsf{contains}(\mathsf{a}, \epsilon) = \ldots$
- $\mathsf{contains}(\mathsf{a}, \mathsf{x}) = \ldots$
- $\mathsf{contains}(\mathsf{a}, r^*) = \ldots$
- $\mathsf{contains}(\mathsf{a}, r_1|r_2) = \ldots$
- $\mathsf{contains}(\mathsf{a}, r_1 r_2) = \ldots$

(b) Beweisen Sie die Korrektheit ihrer Funktion mit struktureller Induktion. Kennzeichnen Sie dabei die Induktionshypothesen und deren Anwendungen deutlich.

Lösungsskizze.

(a)
- $\mathsf{contains}(\mathsf{a}, \emptyset) = \mathsf{true}$
- $\mathsf{contains}(\mathsf{a}, \epsilon) = \mathsf{false}$

- $\mathsf{contains}(\mathsf{a}, \mathsf{x}) = (\mathsf{a} = \mathsf{x})$
- $\mathsf{contains}(\mathsf{a}, r_1 \mid r_2) = \mathsf{contains}(\mathsf{a}, r_1) \wedge \mathsf{contains}(\mathsf{a}, r_2)$
- $\mathsf{contains}(\mathsf{a}, r_1 r_2) = \mathsf{contains}(\mathsf{a}, r_1) \vee \mathsf{contains}(\mathsf{a}, r_2)$
- $\mathsf{contains}(\mathsf{a}, r^*) = \mathsf{false}$

(b) **Fall** $r = \emptyset$
Es gilt $\mathsf{contains}(\mathsf{a}, \emptyset) = \mathsf{true} \iff \forall w \in L(\emptyset).\ P(w)$, da $L(\emptyset) = \emptyset$.

Fall $r = \boldsymbol{\epsilon}$
Wir haben $L(r) = \{\varepsilon\}$ und $\mathsf{contains}(a, \boldsymbol{\epsilon}) = \mathsf{false} \iff P(\varepsilon) \iff \forall w \in L(\boldsymbol{\epsilon}).\ P(w)$.

Fall $r = \mathsf{x}$
Wenn $\mathsf{a} = \mathsf{x}$, dann gilt $L(\mathsf{x}) = \{\mathsf{a}\}$ und wir haben

$$\mathsf{contains}(\mathsf{a}, \mathsf{x}) = \mathsf{true} \iff P(\mathsf{a}) \iff \forall w \in L(\mathsf{x}).\ P(w).$$

Ansonsten haben wir

$$\mathsf{contains}(\mathsf{a}, \mathsf{x}) = \mathsf{false} \iff P(\mathsf{x}) \iff \forall w \in L(\mathsf{x}).\ P(w),$$

da $L(\mathsf{x}) = \{\mathsf{x}\}$.

Fall $r = s^*$
Dann gilt $\varepsilon \in L(s^*)$ und da $P(\varepsilon) = \mathsf{false}$, ist die Definition $\mathsf{contains}(a, r^*) = \mathsf{false}$ korrekt.

Fall $r = r_1 \mid r_2$
Als Induktionshypothesen erhalten wir $\mathsf{contains}(\mathsf{a}, r_1) \iff \forall w \in L(r_1).\ P(w)$ und $\mathsf{contains}(\mathsf{a}, r_2) \iff \forall w \in L(r_2).\ P(w)$. Damit haben wir

$$\begin{aligned}
\mathsf{contains}(\mathsf{a}, r_1 \mid r_2) &\iff \mathsf{contains}(\mathsf{a}, r_1) \wedge \mathsf{contains}(\mathsf{a}, r_2) \\
&\overset{IH}{\iff} (\forall w \in L(r_1).\ P(w)) \wedge (\forall w \in L(r_2).\ P(w)) \\
&\iff \forall w \in L(r_1) \cup L(r_2).\ P(w) \\
&\iff \forall w \in L(r_1 \mid r_2).\ P(w).
\end{aligned}$$

Fall $r = r_1 r_2$
Als Induktionshypothesen erhalten wir $\mathsf{contains}(\mathsf{a}, r_1) \iff \forall w \in L(r_1).\ P(w)$ und $\mathsf{contains}(\mathsf{a}, r_2) \iff \forall w \in L(r_2).\ P(w)$.

Es gilt

$$\begin{aligned}
\mathsf{contains}(\mathsf{a}, r_1 r_2) &\iff \mathsf{contains}(\mathsf{a}, r_1) \vee \mathsf{contains}(\mathsf{a}, r_2) \\
&\overset{IH}{\iff} (\forall u \in L(r_1).\ P(u)) \vee (\forall v \in L(r_2).\ P(v)) \\
&\iff \forall u \in L(r_1), v \in L(r_2).\ P(u) \vee P(v) \\
&\overset{(*)}{\iff} \forall u \in L(r_1), v \in L(r_2).\ P(uv) \\
&\iff \forall w \in L(r_1 r_2).\ P(w)
\end{aligned}$$

Nun zeigen wir noch, dass $(*)$ gilt (dies könnte man auch als offensichtlich annehmen). Seien $u, v \in \Sigma^*$ beliebig. Wir zeigen nun $P(u) \vee P(v) \Leftrightarrow P(uv)$.

„⇒“ Falls $P(u)$ gilt, folgern wir

$$(\exists u_1, u_2.\ u = u_1 \mathsf{a} u_2) \Rightarrow (\exists u_1, u_2.\ uv = u_1 \mathsf{a} u_2 v) \Rightarrow P(uv)$$

Falls $P(v)$ gilt:

$$(\exists v_1, v_2.\ v = v_1 \mathsf{a} v_2) \Rightarrow (\exists v_1, v_2.\ uv = uv_1 \mathsf{a} v_2) \Rightarrow P(uv)$$

„⇐“ Es gilt $P(uv) \Rightarrow (\exists w_1, w_2.\ uv = w_1 \mathsf{a} w_2)$, wir wählen also w_1, w_2 entsprechend. Es gilt $|u| + |v| = |w_1| + |w_2| + 1 \Rightarrow |w_1| \geq |u| \vee |w_2| \geq |v|$. Falls $|w_1| \geq |u|$, dann folgt

$$uv = w_1 \mathsf{a} w_2 \Rightarrow (\exists x.\ uv = ux \mathsf{a} w_2) \Rightarrow (\exists x.\ v = x \mathsf{a} w_2) \Rightarrow P(v)$$

Falls $|w_2| \geq |v|$, dann folgt

$$uv = w_1 \mathsf{a} w_2 \Rightarrow (\exists x.\ uv = w_1 \mathsf{a} xv) \Rightarrow (\exists x.\ u = w_1 \mathsf{a} x) \Rightarrow P(u)$$

Quizaufgabe H2.7. (*Sparmaßnahmen*) keine Korrektur

Die Universitätsleitung zeigt sich entsetzt über die große Menge an Tinte, die in die fettgedruckten Symbole $\boldsymbol{\epsilon}$ und $\boldsymbol{\emptyset}$ in regulären Ausdrücken fließt. Deshalb wird angeordnet, dass diese Symbole künftig – wenn möglich – vermieden werden. Jeder reguläre Ausdruck soll in eine der folgenden Formen gebracht werden:

(F1) r (F2) $r \mid \boldsymbol{\epsilon}$ (F3) $\boldsymbol{\epsilon}$ (F4) $\boldsymbol{\emptyset}$

Hierfür ist r ein beliebiger regulärer Ausdruck, der weder $\boldsymbol{\epsilon}$ noch $\boldsymbol{\emptyset}$ enthält.

Beruhigen Sie ihre verzweifelten Kollegen, indem Sie beweisen, dass jeder Ausdruck in diese Form gebracht werden kann. Zeigen Sie, dass zu jedem regulären Ausdruck r' ein äquivalenter Ausdruck r existiert, der in einer der obigen Formen ist.

Hinweise: Sie können strukturelle Induktion verwenden. Sie dürfen ohne Beweis verwenden, dass $(r \mid \boldsymbol{\epsilon})^* \equiv r^*$ für jeden regulären Ausdruck r gilt.

Lösungsskizze. Wir zeigen die Aussage mithilfe von struktureller Induktion. Als Induktionsbasis zeigen wir die Aussage also für alle atomaren regulären Ausdrücke (d.h. für $\boldsymbol{\emptyset}, \boldsymbol{\epsilon}$ und jedes $c \in \Sigma$); für den Induktionsschritt nehmen wir an, dass die Aussage für zwei beliebige reguläre Ausdrücke r, s gilt, und zeigen, dass sie auch für r^*, $r \mid s$ und rs gilt.
Induktionsbasis. $\boldsymbol{\emptyset}$ ist bereits in (F4), $\boldsymbol{\epsilon}$ in (F3), und jedes Zeichen $c \in \Sigma$ ist in (F1).
Induktionsschritt. Seien r', s' beliebige reguläre Ausdrücke, die jeweils einen äquivalent regulären Ausdruck r, s in einer der obigen Formen besitzen.

- r^*: Wenn $r = \boldsymbol{\epsilon}$ oder $r = \boldsymbol{\emptyset}$ (also (F3) oder (F4)), dann ist $r^* \equiv \boldsymbol{\epsilon}$. Wenn r in (F1) ist, so ist es auch r^*. Wenn r in (F2) ist, dann gilt $r = t \mid \boldsymbol{\epsilon}$ für einen regulären Ausdruck t in (F1). Nach Hinweis gilt $r^* \equiv t^*$; und t^* ist in (F1).
- rs: Wenn $r = \boldsymbol{\emptyset}$ oder $s = \boldsymbol{\emptyset}$, dann ist $rs \equiv \boldsymbol{\emptyset}$. Wenn $r = \boldsymbol{\epsilon}$, dann ist $rs \equiv s$, und s ist bereits in (F1), (F2) oder (F3). Analog für $s = \boldsymbol{\epsilon}$. Es bleiben 4 Fälle:
  1. r und s in (F1): Dann ist rs auch in (F1).
  2. r in (F1), s in (F2): Sei $s = u \mid \boldsymbol{\epsilon}$. Es gilt $rs = r(u \mid \boldsymbol{\epsilon}) \equiv ru \mid r$, was in (F1) ist.
  3. r in (F2), s in (F1): Analog zum vorherigen Fall.

4. r und s in (F2): Seien $r = t \mid \boldsymbol{\epsilon}$ und $s = u \mid \boldsymbol{\epsilon}$. Dann gilt $rs = (t \mid \boldsymbol{\epsilon})(u \mid \boldsymbol{\epsilon}) \equiv (tu \mid t \mid u) \mid \boldsymbol{\epsilon}$, was in (F2) ist.

$r \mid s$: Da Vereinigung kommutativ ist, können wir ohne Beschränkung der Allgemeingültigkeit annehmen, dass r und s nach Formnummer sortiert sind. (Wenn z.B. r in (F4) ist und s in (F3), dann tauschen wir sie; r hat also nie eine größere Form als s.) Wenn $s = \boldsymbol{\emptyset}$ (also (F4)) dann ist $r \mid s \equiv r$, und r ist in einer der 4 Formen. Folgende Fälle bleiben übrig:

1. r und s in (F1): Dann ist $r \mid s$ auch in (F1).
2. r in (F1), s in (F2): Sei $s = u \mid \boldsymbol{\epsilon}$. Es gilt $r \mid s \equiv (r \mid u) \mid \boldsymbol{\epsilon}$, was in (F2) ist.
3. r in (F1), s in (F3): Dann ist $r \mid s$ bereits in (F2).
4. r und s in (F2): Seien $r = t \mid \boldsymbol{\epsilon}$ und $s = u \mid \boldsymbol{\epsilon}$. Dann gilt $r \mid s = (t \mid \boldsymbol{\epsilon}) \mid (u \mid \boldsymbol{\epsilon}) \equiv (t \mid u) \mid \boldsymbol{\epsilon}$, was in (F2) ist.
5. r in (F2) oder (F3), s in (F3): Dann ist $r \mid s \equiv r$, und r ist bereits in (F2) oder (F3).

Knobelaufgabe H2.8. (*Auf den Kopf gefallen*)

Sei $\Sigma \neq \emptyset$ und $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA, sodass der Automat $M' = (Q, \Sigma, \delta', q_0, F)$, den man erhält, wenn man alle Transitionen in M umdreht, auch ein DFA ist. Formal gilt also $\delta'(r, c) = q$ für alle $\delta(q, c) = r$. Zeigen Sie, dass $L(M)$ unendlich oder leer ist.

Lösungsskizze. Wenn $L(M) = \emptyset$ sind wir bereits fertig. Wir können also ein Wort $w \in L(M)$ fixieren. Sei nun $c \in \Sigma$ beliebig und $q_1 := \delta(q_0, wc)$ der Zustand, der nach Einlesen von wc erreicht wird. (Wir hängen c an, damit $wc \neq \varepsilon$.) Wir betrachten nun die Menge $S := \{\hat{\delta}(q_1, v) : v \in \Sigma^*\}$ der Zustände, die von q_1 aus erreicht werden können. Nach Definition muss also jede ausgehende Transition eines Zustands in S wieder zu einem Zustand in S führen. Es gibt $|S|$ Zustände in S, von denen jeder genau $|\Sigma|$ ausgehende Transitionen hat. Insgesamt haben Zustände in S also mindestens $|S| \cdot |\Sigma|$ eingehende Transitionen.

Wenn $q_0 \notin S$ gelten würde, muss es aber noch eine zusätzliche Transition von einem Zustand $q \notin S$ zu einem Zustand $q' \in S$ geben (da q_0 nach Einlesen von w einen Zustand in S erreicht). Dann hätten Zustände in S also mindestens $|S| \cdot |\Sigma| + 1$ eingehende Transitionen; demzufolge gäbe es einen Zustand $r \in S$ mit mehr als $|\Sigma|$ eingehenden Transitionen. Nach Umdrehen der Transitionen wäre M dann kein DFA mehr, ein Widerspruch.

Also gilt $q_0 \in S$. Nach Definition von S ist q_0 somit von q_1 aus erreichbar, es gibt also ein Wort v mit $\hat{\delta}(q_1, v) = q_0$ und somit $\hat{\delta}(q_0, wcv) = q_0$. Da $\hat{\delta}(q_0, w) \in F$, folgt $\{wcv\}^*\{w\} \subseteq L$; mithilfe von $wcv \neq \varepsilon$ ist L unendlich.

Alternative Lösung: Angenommen $0 < |L(M)| < \infty$. Sei $w \in L(M)$ mit $|w|$ maximal und sei $p_0 := \hat{\delta}(q_0, w)$. Somit gilt $p_0 \in F$. Wir fixieren ein $a \in \Sigma$ und betrachten die Folge $(p_i)_{i \in \mathbb{N}}$, wobei $p_{i+1} := \delta(p_i, a)$ für $i \in \mathbb{N}$. Da Q endlich ist, gibt es $i < j$ mit $p_i = p_j$. Wir wählen nun solche i, j wobei i minimal ist. Angenommen, $i > 0$. Dann gilt $\delta(p_{i-1}, a) = p_i = p_j = \delta(p_{j-1}, a)$. Dies ist ein Widerspruch: entweder $p_{i-1} \neq p_{j-1}$ und M' wäre kein DFA, oder $p_{i-1} = p_{j-1}$ und i wäre nicht minimal. Also gilt $i = 0$. Es folgt $\hat{\delta}(q_0, wa^j) = \hat{\delta}(p_0, a^j) = p_j = p_0 \in F \Rightarrow wa^j \in L(M)$, und offensichtlich $|wa^j| > |w|$ da $j > i = 0$. Dies ist ein Widerspruch zur Wahl von w und $L(M) \in \{0, \infty\}$.