Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 1

Vorlesung Theoretische Grundlagen der Informatik im WS 19/20

Ausgabe: 17. Oktober 2019
Abgabe: 5. November 2019, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Aufgabe 1

(1 + 1 + 1 + 1 + 1 + 1 = 6 Punkte)

Gegeben seien zwei Sprachen $L_1, L_2 \subseteq \Sigma^*$ über dem Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Dabei sei L_1 die Sprache der Wörter, die das Teilwort bb enthalten, und L_2 die Sprache der Wörter, in denen kein b auf ein a folgt. Geben Sie reguläre Ausdrücke für folgende Sprachen an.

(a) $L_1 \cup L_2$

(b) $L_1 \cap L_2$

(c) L_1/L_2

(d) L_2/L_1

(e) L_1^c

(f) L_2^c

Lösung:

(a) $L_1 \cup L_2 = L((\mathsf{a} \cup \mathsf{b})^*\mathsf{bb}(\mathsf{a} \cup \mathsf{b})^* \cup \mathsf{b}^*\mathsf{a}^*)$

(b) $L_1 \cap L_2 = L(\mathsf{bb}^+\mathsf{a}^*)$

(c) $L_1/L_2 = \Sigma^*$

(d) $L_2/L_1 = L(\mathsf{b}^*)$

(e) $L_1^c = L(\mathsf{a}^*(\mathsf{ba}^+)^*(\mathsf{b} \cup \varepsilon))$

(f) $L_2^c = L((\mathsf{a} \cup \mathsf{b})^*\mathsf{ab}(\mathsf{a} \cup \mathsf{b})^*)$

Aufgabe 2

(3 + 2 + 1 = 6 Punkte)

Der ebenso geniale wie auch vergessliche Wissenschaftler und Superbösewicht Doktor Meta ist im Baufieber. Der Hauptzugang zu seinem unterirdischen Geheimlabor soll von einer unüberwindbaren Stahltür geschützt werden, die sich nur durch Eingabe eines gültigen Passworts öffnen lässt. Bei einer falschen Eingabe wird der Eindringling stattdessen durch eine Falltür im Boden den Haien vorgeworfen. Da sich Doktor Meta aufgrund seiner Vergesslichkeit nur schlecht an Passwörter erinnern kann, will er die Tür so einstellen, dass Sie jedes Wort aus einer zuvor festgelegten regulären Sprache L über dem Alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$ akzeptiert. Dieses – sicherheitstechnisch fragwürdige, aber für Doktor Meta sehr praktische – Verfahren soll intern durch einen deterministischen endlichen Automaten realisiert werden. Sie wurden von Doktor Meta als neuer Sicherheitsexperte eingestellt und sollen ihm bei den Details helfen:

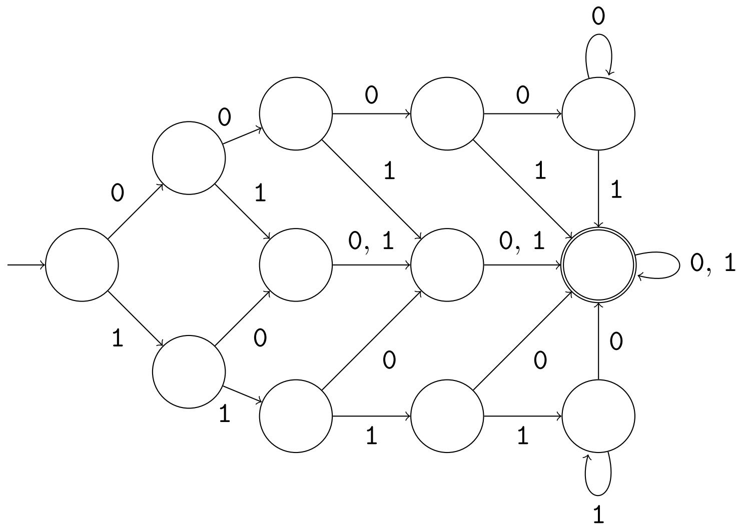
(a) Geben Sie einen deterministischen Automaten für die Sprache $L = \{w \in \Sigma^* \mid |w| \geq 4, |w|_\mathtt{0} \geq 1, |w|_\mathtt{1} \geq 1\}$ an. Dabei bezeichnet $|w|_a$ die Anzahl der Vorkommen des Symbols a in w. Also enthält L genau die Wörter mit mindestens 4 Zeichen, die mindestens eine `0` und mindestens eine `1` enthalten.

(b) Nun sei L eine beliebige reguläre Sprache über Σ. Doktor Meta möchte einige seiner Mitarbeiter unauffällig verschwinden lassen. Dafür hat er die Passwörter, die sie normalerweise eingeben, studiert und Muster[1] darin gefunden. Er beauftragt Sie nun, die Kontrolleinheit so umzubauen, dass gerade die Muster der Leute, die er loswerden möchte, von der Kontrolleinheit abgelehnt werden. Sie bekommen also eine endliche Teilmenge $L' \subseteq L$ gegeben und sollen den Automaten so umbauen, dass er $L \setminus L'$ erkennt. Beweisen Sie, dass dies für jede endliche Teilmenge L' möglich ist, also dass $L \setminus L'$ regulär ist.

(c) Ist dies auch noch möglich, wenn Doktor Meta unendlich viele Mitarbeiter loswerden will, d.h. wenn L' unendlich viele Wörter enthält?

Lösung:

(a)



(b) L' ist eine endliche Sprache und damit wie in der Übung gezeigt regulär. Es gilt $L \setminus L' = L \cap L'^c$. Reguläre Sprachen sind unter Komplement- und Schnittbildung abgeschlossen, also ist auch $L \setminus L'$ regulär.

(c) Nein, denn dann muss L' nicht mehr unbedingt regulär sein. Zum Beispiel ist $\{\mathtt{0}^i\mathtt{1}^i \mid i \in \mathbb{N}\}$ eine nichtreguläre Teilmenge der regulären Sprache $\mathtt{0}^+\mathtt{1}^+$.

[1] Muster sind beliebige Mengen von Wörtern über dem Alphabet Σ.

Aufgabe 3 (1 + 1 + 1 = 3 Punkte)

Gegeben seien drei Grammatiken $G_i = (\Sigma, V, S, R_i), i = 1, 2, 3$ mit $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}, V = \{S, A, B\}$ und folgenden Produktionsregeln

(a) $R_1 = \{S \to \mathsf{b}S \mid A, A \to \mathsf{a}B\mathsf{a}, B \to \mathsf{a}A \mid \mathsf{a}\}$

(b) $R_2 = \{S \to \mathsf{b}S \mid A, A \to \mathsf{a}B\mathsf{a}, B \to \mathsf{c}A \mid \mathsf{c}\}$

(c) $R_3 = \{S \to \mathsf{a}S \mid A, A \to \mathsf{b}B, B \to \mathsf{c}B \mid A \mid \varepsilon\}$

Ist $L(G_i)$ eine reguläre Sprache? Wenn ja, geben Sie einen regulären Ausdruck an. Wenn nicht, begründen Sie dies kurz (informell) und geben sie eine formale Definition der Sprache in Mengenschreibweise an.

Lösung:

(a) $\mathsf{b}^*(\mathsf{aaa})^+$

(b) $L(G_2) = \{\mathsf{b}^i(\mathsf{ac})^j\mathsf{a}^j \mid i \in \mathbb{N}_0, j \in \mathbb{N}\}$. Nein, DEA können nicht zählen.

(c) $\mathsf{a}^*(\mathsf{bc}^*)^+$

Aufgabe 4 (1 + 2 = 3 Punkte)

(a) Offensichtlich lässt sich jeder deterministische endliche Automat (DEA) durch einen nichtdeterministischen endlichen Automaten (NEA) simulieren, indem der Nichtdeterminismus in der Überführungsfunktion nicht ausgenutzt wird. Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ ein NEA, der wie ein DEA arbeitet. Welche Bedingungen müssen für die Überführungsfunktion $\delta\colon Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ gelten? Formulieren Sie die Bedingungen als prädikatenlogische Formel.

(b) Für NEA wurde in der Vorlesung die erweiterte Überführungsfunktion $\bar{\delta}\colon Q \times \Sigma^* \to 2^Q$ eingeführt, die einen Zustand q und ein Wort w auf die Menge der Zustände abbildet, die von q nach der Abarbeitung von w erreichbar sind. Mithilfe der Simulation aus dem vorigen Aufgabenteil lässt sich $\bar{\delta}$ auch bei DEA anwenden. Kreuzen Sie in der folgenden Tabelle an, welche Aussage für *alle* NEA bzw. DEA $\mathcal{A}$ mit $L(\mathcal{A}) \neq \emptyset$ und $L(\mathcal{A}) \neq \Sigma^*$ gelten.

Aussage	NEA	DEA
$\exists w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$		
$\exists w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$		
$\exists w \notin L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$		
$\exists w \notin L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \notin F$		
$\forall w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$		
$\forall w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \notin F$		

Lösung:

(a) $\forall q \in Q\colon (\delta(q, \varepsilon) = \emptyset \wedge \forall x \in \Sigma\colon |\delta(q, x)| = 1)$

(b)

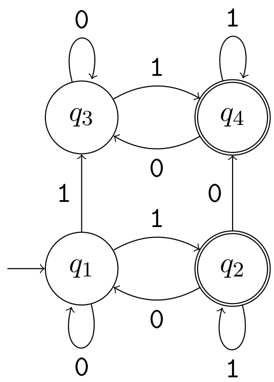
Aussage	NEA	DEA
$\exists w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$		×
$\exists w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$		
$\exists w \notin L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$		
$\exists w \notin L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \notin F$	×	×
$\forall w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$	×	×
$\forall w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \notin F$	(*)	×

(*) Auf den ersten Blick mag es so scheinen, als ob dieses Feld angekreuzt werden muss: Für Wörter $w \in L(\mathcal{A})$ sind *alle* möglichen Abarbeitungen nicht akzeptierend, also erscheint es logisch, dass auch *eine* nicht akzeptierende Abarbeitung existieren muss. Das ist nicht der Fall, weil es bei einem NEA vorkommen kann, dass es für ein Wort *gar keine* Abarbeitung gibt, also $\bar{\delta}(s, w) = \emptyset$. Das liegt daran, dass die NEA-Überführungsfunktion auf eine Menge von möglichen Folgezuständen abbildet. Diese Menge darf auch leer sein. In diesem Fall wird das gelesene Wort w nicht akzeptiert, weil die Akzeptanzbedingung $\bar{\delta}(s, w) \cap F \neq \emptyset$ nicht erfüllt ist.

Aufgabe 5

(3 + 2 = 5 Punkte)

Gegeben ist folgender nichtdeterministischer endlicher Automat:



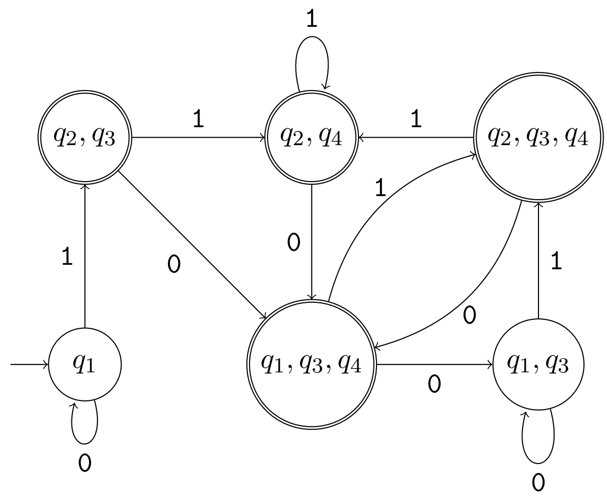
(a) Nutzen Sie die Potenzmengenkonstruktion aus der Vorlesung, um den Automaten deterministisch zu machen. Unerreichbare Zustände dürfen Sie dabei direkt entfernen.

(b) Der durch die Potenzmengenkonstruktion entstandene DEA ist nicht minimal. Geben Sie einen äquivalenten DEA mit höchstens 4 Zuständen an. Sie müssen dafür nicht die Äquivalenzklassenkonstruktion aus der Vorlesung anwenden.

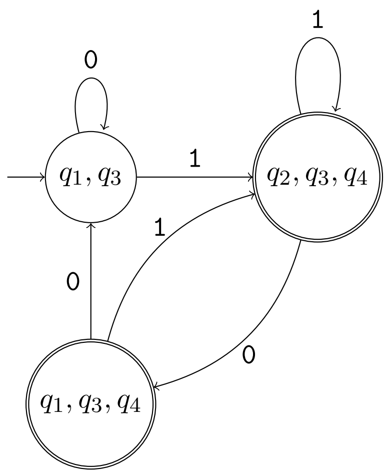
Hinweis: Die ursprüngliche Aufgabenstellung verlangte einen Automaten mit genau 4 Zuständen. Tatsächlich hat der minimale Automat nur 3 Zustände.

Lösung:

(a)



(b)



Aufgabe 6

(3 + 3 + 2 (+ 4) = 8 (+ 4) Punkte)

Sei Σ ein beliebiges Alphabet und L eine beliebige reguläre Sprache über Σ. Zeigen Sie, dass folgende Sprachen regulär sind:

(a) $L_1 := \{v \in \Sigma^* \mid \exists w \in \Sigma^* \colon vw \in L\}$, also die Menge der Präfixe von Wörtern aus L.

(b) $L_2 := \{v \in \Sigma^* \mid \exists u \in \Sigma^* \colon uv \in L\}$, also die Menge der Suffixe von Wörtern aus L.

(c) $L_3 := \{v \in \Sigma^* \mid \exists u, w \in \Sigma^* \colon uvw \in L\}$, also die Menge der Teilwörter von Wörtern aus L.

(d) **Bonusaufgabe:** $L_4 := \{v = v_1 \dots v_n \text{ mit } v_1, \dots, v_n \in \Sigma \mid \exists u_0, \dots, u_n \in \Sigma^* \colon u = u_0 v_1 u_1 v_2 \dots u_{n-1} v_n u_n \in L\}$, also die Menge der Teil*folgen* von Wörtern aus L.

Um zu zeigen, dass L_i regulär ist, gehen Sie wie folgt vor: Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ der deterministische endliche Automat, der L erkennt. Bauen Sie $\mathcal{A}$ zu einem endlichen Automaten $\mathcal{A}_i$ um, der L_i erkennt. Beweisen Sie, dass $\mathcal{A}_i$ genau die Wörter aus L_i akzeptiert.

Lösung:

(a) Betrachte den DEA $\mathcal{A}_1 = (Q, \Sigma, \delta, s, F_1)$ mit $F_1 := \{q \in Q \mid \exists w \in \Sigma^* \colon \delta(q, w) \in F\}$. Es werden also alle Zustände zu Endzuständen gemacht, von denen aus ein ursprünglicher Endzustand über einen Pfad in $\mathcal{A}$ erreichbar ist. Ansonsten bleibt der Automat unverändert.

Betrachte ein beliebiges Wort $v \in \Sigma^*$. Wir zeigen, dass v genau dann von $\mathcal{A}_1$ akzeptiert wird, wenn v in L_1 liegt, d.h. $\delta(s, v) \in F_1 \Leftrightarrow v \in L_1$:

$$\begin{aligned}
\delta(s, v) \in F_1 &\Leftrightarrow \exists w \in \Sigma^* \colon \delta(\delta(s, v), w) \in F \\
&\Leftrightarrow \exists w \in \Sigma^* \colon \delta(s, vw) \in F \\
&\Leftrightarrow \exists w \in \Sigma^* \colon vw \in L \\
&\Leftrightarrow v \in L_1
\end{aligned}$$

(b) Betrachte den NEA $\mathcal{A}_2 = (Q \cup \{s'\}, \Sigma, \delta_2, s', F)$ mit

$$\begin{aligned}
\delta_2(q, a) &:= \{\delta(q, a)\} && \text{für } q \in Q, a \in \Sigma \\
\delta_2(s', a) &:= \emptyset && \text{für } a \in \Sigma \\
\delta_2(q, \varepsilon) &:= \emptyset && \text{für } q \in Q \\
\delta_2(s', \varepsilon) &:= \{q \in Q \mid \exists u \in \Sigma^* : \delta(s, u) = q\}.
\end{aligned}$$

Es wird also ein neuer Startzustand s' eingeführt, von dem aus ε-Übergänge zu allen Zuständen führen, die über einen Pfad in $\mathcal{A}$ von s aus erreichbar sind. Ansonsten bleibt der Automat unverändert. Also wählt $\mathcal{A}_2$ am Anfang einen beliebigen von s aus erreichbaren Zustand als „Startzustand" und arbeitet dann von dort deterministisch wie $\mathcal{A}$ weiter.

Wir halten zunächst fest, dass für jeden Zustand $q \in Q$ und jedes Wort $v \in \Sigma^*$ gilt: $\bar{\delta}_2(q, v) = \{\delta(q, v)\}$. In Worten: Sobald $\mathcal{A}_2$ sich nicht mehr im Startzustand befindet, arbeitet er deterministisch wie $\mathcal{A}$.

Nun schauen wir uns für ein beliebiges Wort $v \in \Sigma^*$ an, woraus $\bar{\delta}_2(s', v)$ besteht:

$$\begin{aligned}
\bar{\delta}_2(s', v) &= \bigcup_{q \in \delta_2(s', \varepsilon)} \bar{\delta}_2(q, v) \\
&= \{\delta(q, v) \mid q \in \delta_2(s', \varepsilon)\} \\
&= \{\delta(q, v) \mid \exists u \in \Sigma^* \colon \delta(s, u) = q\} \\
&= \{\delta(\delta(s, u), v) \mid u \in \Sigma^*\} \\
&= \{\delta(s, uv) \mid u \in \Sigma^*\}
\end{aligned}$$

Damit können wir nun für jedes Wort $v \in \Sigma^*$ zeigen, dass v genau dann von $\mathcal{A}_2$ akzeptiert wird, wenn v in L_2 liegt, d.h. $\bar{\delta}_2(s', v) \cap F \neq \emptyset \Leftrightarrow v \in L_2$:

$$\begin{aligned}
\bar{\delta}_2(s', v) \cap F \neq \emptyset &\Leftrightarrow \exists q \in \bar{\delta}_2(s', v) \colon q \in F \\
&\Leftrightarrow \exists u \in \Sigma^* \colon \delta(s, uv) \in F \\
&\Leftrightarrow \exists u \in \Sigma^* \colon uv \in L \\
&\Leftrightarrow v \in L_2
\end{aligned}$$

(c) L_3 wird von $\mathcal{A}_3 = (Q, \Sigma, \delta_2, s, F_1)$ erkannt. Es werden also die Präfixerkennung von $\mathcal{A}_1$ und die Suffixerkennung von $\mathcal{A}_2$ in einem Automaten kombiniert.

Wir greifen auf die Beweise aus den vorigen beiden Aufgabenteilen zurück, um zu zeigen, dass ein Wort $v \in \Sigma^*$ genau dann von $\mathcal{A}_3$ akzeptiert wird, wenn v in L_3 liegt, d.h. $\bar{\delta}_2(s', v) \cap F_1 \neq \emptyset \Leftrightarrow v \in L_3$:

$$\begin{aligned}
\bar{\delta}_2(s', v) \cap F_1 \neq \emptyset &\Leftrightarrow \exists u \in \Sigma^* \colon \delta(s, uv) \in F_1 \\
&\Leftrightarrow \exists u, w \in \Sigma^* \colon \delta(s, uvw) \in F \\
&\Leftrightarrow \exists u, w \in \Sigma^* \colon uvw \in L \\
&\Leftrightarrow v \in L_3
\end{aligned}$$

(d) Betrachte den Automaten $\mathcal{A}_4 = (Q, \Sigma, \delta_4, s, F)$ mit

$$\begin{aligned}
\delta_4(q,a) &:= \{\delta(q,a)\} && \text{für } q \in Q, a \in \Sigma \\
\delta_4(q,\varepsilon) &:= \{p \in Q \mid \exists w \in \Sigma^* : \delta(q,w) = p\} && \text{für } q \in Q
\end{aligned}$$

Es werden also für jeden Zustand ε-Übergänge zu allen Zuständen eingefügt, die über einen Pfad in $\mathcal{A}$ erreichbar sind. Dadurch kann $\mathcal{A}_4$ zu jedem Zeitpunkt ohne Lesen von Zeichen in Zustände springen, in die $\mathcal{A}$ durch das Lesen eines Teilwortes aus L gekommen wäre.

Die Relation der ε-Übergänge in $\mathcal{A}_4$ ist reflexiv und transitiv abgeschlossen, d.h. wenn ein Zustand q von einem anderem Zustand q' über eine Folge von ε-Übergängen erreichbar ist, dann gibt es auch einen direkten ε-Übergang. Also ist der ε-Abschluss $E(q)$ eines Zustands $q \in Q$ gerade $\delta_4(q, \varepsilon)$.

Wir schauen uns nun mithilfe der Definition aus der Vorlesung für einen Zustand $q \in Q$ und ein Zeichen $a \in \Sigma$ an, woraus $\bar{\delta}_4(q,a)$ besteht:

$$\begin{aligned}
\bar{\delta}_4(q,a) &= \bigcup_{p \in E(q)} \bigcup_{r \in \delta_4(p,a)} E(r) \\
&= \bigcup_{p \in \delta_4(q,\varepsilon)} \bigcup_{r \in \delta_4(p,a)} \delta_4(r,\varepsilon) \\
&= \bigcup_{p \in \delta_4(q,\varepsilon)} \delta_4(\delta(p,a),\varepsilon) \\
&= \bigcup_{p \in \delta_4(q,\varepsilon)} \{\delta(\delta(p,a),y) \mid y \in \Sigma^*\} \\
&= \bigcup_{p \in \delta_4(q,\varepsilon)} \{\delta(p,ay) \mid y \in \Sigma^*\} \\
&= \{\delta(\delta(q,x),ay) \mid x,y \in \Sigma^*\} \\
&= \{\delta(q,xay) \mid x,y \in \Sigma^*\}
\end{aligned}$$

Durch das Lesen von a kann $\mathcal{A}_4$ also genau die Zustände erreichen, die in $\mathcal{A}$ durch das Lesen eines Wortes, das a enthält, erreichbar waren.

Wir betrachten nun ein beliebiges Wort $v = v_1 \dots v_n \in \Sigma^*$ und zeigen durch Induktion über n: $\bar{\delta}_4(s,v) = \{\delta(s, u_0 v_1 u_1 \dots v_n u_n) \mid u_0, \dots, u_n \in \Sigma^*\}$.

- **Induktionsanfang:** $\bar{\delta}_4(s,\varepsilon) = \{\delta(s,u_0) \mid u_0 \in \Sigma^*\}$
- **Induktionsvoraussetzung:** Wir nehmen an, dass gilt: $\bar{\delta}_4(s, v_1 \dots v_{n-1}) = \{\delta(s, u_0 v_1 u_1 \dots v_{n-1} u_{n-1}) \mid u_0, \dots, u_{n-1} \in \Sigma^*\}$.
- **Induktionsschritt:**

$$\begin{aligned}
\bar{\delta}_4(s, v_1 \dots v_n) &= \bigcup_{q \in \bar{\delta}_4(s, v_1 \dots v_{n-1})} \delta_4(q, v_n) \\
&= \bigcup_{q \in \bar{\delta}_4(s, v_1 \dots v_{n-1})} \{\delta(q, x v_n y) \mid x, y \in \Sigma^*\} \\
&= \{\delta(\delta(s, u_0 v_1 u_1 \dots v_{n-1} u_{n-1}), x v_n u_n) \mid u_0, \dots, u_n, x \in \Sigma^*\} \\
&= \{\delta(s, u_0 v_1 u_1 \dots v_{n-1} u_{n-1} x v_n u_n) \mid u_0, \dots, u_n, x \in \Sigma^*\} \\
&= \{\delta(s, u_0 v_1 u_1 \dots v_n u_n) \mid u_0, \dots, u_n \in \Sigma^*\}
\end{aligned}$$

Dabei wurde im letzten Schritt x mit u_{n-1} verschmolzen.

Damit zeigen wir nun, dass v genau dann von $\mathcal{A}_4$ akzeptiert wird, wenn v in L_4 liegt, d.h. $\bar{\delta}_4(s,v) \cap F \neq \emptyset \Leftrightarrow v \in L_4$:

$$\begin{aligned}
\bar{\delta}_4(s,v) \cap F \neq \emptyset &\Leftrightarrow \exists q \in \bar{\delta}_4(s,v)\colon q \in F \\
&\Leftrightarrow \exists u_0, \ldots, u_n \in \Sigma^*\colon \delta(s, u_0 v_1 u_1 \ldots v_n u_n) \in F \\
&\Leftrightarrow \exists u_0, \ldots, u_n \in \Sigma^*\colon u_0 v_1 u_1 \ldots v_n u_n \in L \\
&\Leftrightarrow v \in L_4
\end{aligned}$$