Prof. Javier Esparza
Philipp Czerner, Martin Helfrich

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2021 – Übungsblatt 1

- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.
- Das ist nicht das Hausaufgabenblatt! Die Hausaufgaben finden Sie auf einem separaten Blatt.

Vorbereitung (→ vor der Übung selbständig zu bearbeiten)

Individualaufgabe Ü1.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

- Alphabet Σ; Σ^*
- Wort w, $|w|$, ε, ww, w^n
- formale Sprache A; AA, A^*, A^+
- reflexive transitive Hülle
- Grammatik G; $L(G)$, AA
- Ableitungsrelation
- Chomsky-Hierarchie
- Wortproblem
- DFA, NFA
- Akzeptanzbedingung von DFAs/NFAs

Individualaufgabe Ü1.2. (*Automata Tutor: DFA / NFA / Grammatiken*)

Automata Tutor (AT) ist ein Online-Tool, mit dem Sie zu vielen Themen aus dem THEO-Kurs Aufgaben bearbeiten können. Dabei bekommen Sie sogar automatisch individuelles Feedback! Da wir AT auch für Hausaufgaben verwenden werden, ist es wichtig, dass Sie sich auf AT ein Konto anlegen, dass mit Ihrem TUM-Login verknüpft ist. Dafür gehen Sie wie folgt vor:

(a) Besuchen die die Webseite https://automata-tutor.model.in.tum.de.

(b) Klicken Sie auf „TUM Login“ (**nicht auf „Register“**) und geben Sie ihre TUM-Kennung (z.B. ab12cde) mit Passwort ein, um ein Konto zu erstellen.

(c) Schreiben sich sich unter „Enroll in course“ mit der Kurs-ID „TUM Theo SS21“ und Passwort „BKIFJCTE“ ein.

(d) Klicken Sie auf „View“, um die aktuellen Aufgaben einzusehen.

Dann können Sie die Aufgaben Ü1.2 (a–l) lösen.
Achtung: Für die Übungsaufgaben haben Sie beliebig viele Versuche. Für die Hausaufgaben H1.1 (a–c) nicht!

Individualaufgabe Ü1.3. (*NFA / DFA Konstruktionsideen*)

Beschreiben Sie in eigenen Worten, wie man im Allgemeinen einen Automaten konstruiert, der eine Sprache erkennt, die

(a) nur Wörter enthält, die eine bestimmte Sequenz von Buchstaben enthalten,

(b) am Anfang/Ende jedes Wortes eine bestimmte Sequenz von Buchstaben fordert,

(c) nur Worte gerader/ungerader Länge enthält,

(d) von jedem Wort verlangt, eine bestimmte Anzahl an Buchstaben zu enthalten,

(e) deren Worte an einer fixierten Position einen bestimmten Buchstaben haben müssen.

Lösungsskizze. Zu dieser Aufgabe gibt es auch eine Video-Lösung.

(a) Sei $u = u_1 \dots u_n$ ein Wort. Wir betrachten dann den folgenden NFA:

$$\longrightarrow \boxed{q_0} \xrightarrow{u_1} \boxed{q_1} \xrightarrow{u_2} \cdots \xrightarrow{u_n} \boxed{\boxed{q_n}}$$

Dieser NFA akzeptiert die Sprache $\{u\}$. Wenn wir bei q_0 und q_n noch eine Schleife mit Σ hinzufügen, akzeptiert der NFA genau die Sprache aller Wörter, die u enthalten. Um daraus einen DFA zu bekommen muss man bei q_0 die Schleife mit dem Zeichen u_1 entfernen, und von den Zuständen q_i Rückwärtskanten für die Zeichen $\Sigma \setminus u_{i+1}$ einfügen. Das Ziel dieser Kanten muss immer das längste Präfix von u sein, das ein Suffix der bisher gelesenen Zeichenkette ist. Alternativ kann man den NFA natürlich mit der Potenzmengenkonstruktion determinisieren.

(b) Ähnlich wie (a). Wenn u am Ende des Wortes vorkommen soll, dann macht man eine Schleife mit Σ an q_0, soll es am Anfang vorkommen gehört die Schleife an q_n.

Im letzteren Fall ist die Determinisierung leicht, man fügt von allen Zuständen q_i (außer q_n) eine Transition mit $\Sigma \setminus u_{i+1}$ in einen Fangzustand ein.

Soll u am Ende des Wortes vorkommen, müssen wir wie bei (a) Rückwärtskanten einfügen.

(c)

$$\begin{tikzcd} \arrow[r] & q_0 \arrow[r, bend left, "\Sigma"] & q_1 \arrow[l, bend left, "\Sigma"] \end{tikzcd}$$

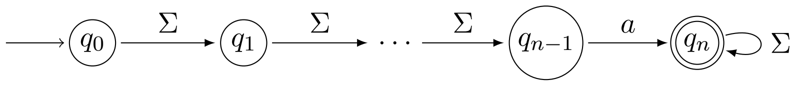
Wenn q_0 zu einem Endzustand gemacht wird, akzeptiert dieser DFA genau alle Wörter gerader Länge. Wird hingegen q_1 zum Endzustand gemacht, so akzeptiert er genau alle Wörter ungerader Länge.

(d) Ähnlich zu (a). Der folgende NFA akzeptiert genau alle Wörter der Länge n:

$$\longrightarrow \boxed{q_0} \xrightarrow{\Sigma} \boxed{q_1} \xrightarrow{\Sigma} \cdots \xrightarrow{\Sigma} \boxed{\boxed{q_n}}$$

Fügt man noch einen Fangzustand $\emptyset \circlearrowleft \Sigma$ hinzu und verbindet $q_n \xrightarrow{\Sigma} \emptyset$, so erhalten wir sogar einen DFA.

(e) Es geht um alle Wörter w, die an der Stelle n den Buchstaben a haben. Man kann den NFA aus dem ersten Teil von (a) leicht anpassen:



Auch hier erreicht man durch Hinzufügen eines Fangzustands wieder einen DFA.

Übung und Nachbereitung

Übungsaufgabe Ü1.4. (*Kahoot*)

Spielen Sie in der Übungsgruppe das Kahoot zum Stoff der aktuellen Woche. Gehen Sie dafür auf www.kahoot.it und geben Sie den Game PIN auf dem Bildschirm Ihres Tutors an.

Lösungsskizze.

- Auf Kahoot Übungsblatt 1.4 klicken,
- „Play as guest“ $\rightarrow$ „Continue as a guest“,
- unter „Game options“ die Fragen und Antworten auf den Geräten der Spieler anzeigen lassen,
- auf „Classic“ klicken,
- oben links die Sprache auf Deutsch ändern, und
- nachdem die Studenten mit dem Game-Pin beigetreten sind, auf „Start“ drücken.

Fokusaufgabe Ü1.5. (*kleine DFAs vs kleine NFAs*)

Sei $\Sigma := \{a, b\}$ und $B_n := \{w \in \Sigma^* \mid \exists i : w_i = w_{i+n}\}$ die Sprache aller Wörter über Σ, in denen an irgendeiner Stelle der gleiche Buchstabe im Abstand n vorkommt. Insbesondere ist B_0 die Menge aller nichtleeren Wörter, und B_1 die Menge aller Wörter, in denen ein Buchstabe zweimal hintereinander vorkommt. *Versuchen Sie in allen Aufgabenteilen NFAs und DFAs mit möglichst wenigen Zuständen anzugeben.*

(a) Geben Sie jeweils einen NFA für B_0, B_1 und B_2 an.

(b) Geben Sie jeweils einen DFA für B_0, B_1 und B_2 an.

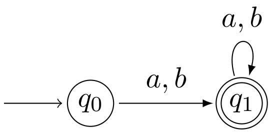
(c) Beschreiben Sie kurz, wie der DFA B_n und der NFA B_n für beliebige $n \in \mathbb{N}$ aussehen.

(d) Beurteilen Sie die folgende Aussage: *Es gibt einen NFA für B_n mit $\mathcal{O}(n)$-vielen Zuständen, aber jeder DFA zu B_n hat mindestens $\Omega(2^n)$-viele Zustände.*

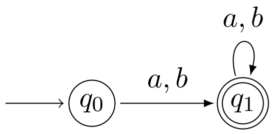
Lösungsskizze. **Hinweis:** Wir haben uns dazu entschlossen, in der ersten Woche des Übungsbetriebs die Lösung zur Fokusaufgabe zu veröffentlichen. Beachten Sie bitte, dass wir nicht die Absicht haben, zu allen Fokusaufgaben Lösungen zu veröffentlichen.

Zu dieser Aufgabe gibt es eine Video-Lösung.

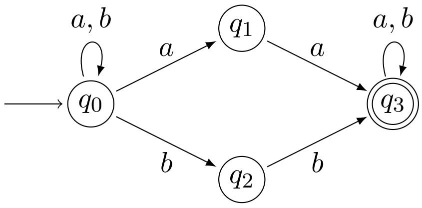
(a), (b):



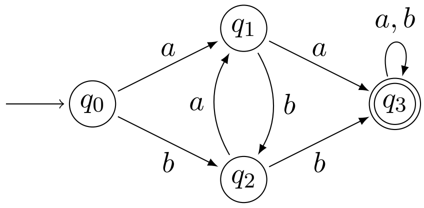
NFA für B_0



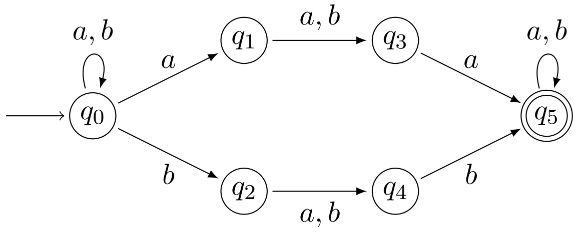
DFA für B_0



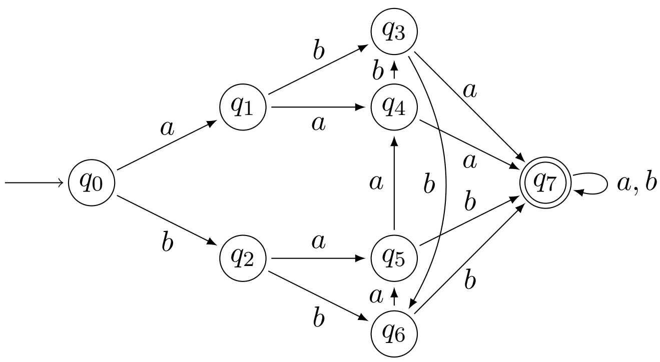
NFA für B_1



DFA für B_1



NFA für B_2



DFA für B_2

(c) Der NFA rät, ob jetzt das i-te Zeichen gelesen wird, welches die Bedingung erfüllt. Der DFA muss sich hingegen immer die letzten n Zeichen die er gelesen hat merken, um zu überprüfen, ob die Bedingung $w_i = w_{i+n}$ erfüllt ist.

(d) Die Aussage ist korrekt. Der Beweis ist ähnlich zu dem Beweis in den Vorlesungsfolien zur Sprache L_k. (Lemma 3.12)

Übungsaufgabe Ü1.6. (*Grammatiken entwerfen*)

Bestimmen Sie für jede der folgenden Sprachen eine passende Grammatik G, so dass $L(G)$ genau die Sprache ist.

(a) $A = \{w \in \{0,1\}^* : |w| \text{ gerade}\}$

(b) $B = \{w \in \{0,1\}^+ : (w)_2 \text{ gerade}\}$

(c) $C = \{ww^R : w \in \{a,b\}^*\}$

(d) $D = \{ww : w \in \{a,b\}^*\}$ *

(e) $E = \{a^{n^2} : n \geq 0\}$ ***

Hinweise:

- $(w)_2$ ist der Wert von $w \in \{\mathtt{0},\mathtt{1}\}^*$ zur Basis 2, also z.B. $(\mathtt{101010})_2 = 42$.
- Wir bezeichnen mit w^R die Spiegelung von w, d.h. $(abb)^R = bba$, $\varepsilon^R = \varepsilon$.
- Eine mögliche Lösung von Aufgabenteil (d) erweitert die Grammatik von Aufgabenteil (c) passend.
- Für Aufgabe (e) ist diese Gleichung hilfreich.

Lösungsskizze.

(a) $G = (\{S\}, \{0,1\}, P, S)$ mit $P\colon S \to 01S \mid 10S \mid 11S \mid 00S \mid \varepsilon$

(b) $G = (\{S\}, \{0,1\}, P, S)$ mit $P\colon S \to 0S \mid 1S \mid 0$. Lösung ohne führende Nullen:
$G = (\{S,X\}, \{0,1\}, P, S)$ mit $P\colon S \to 1X \mid 0 \quad X \to 0X \mid 1X \mid 0$

(c) $G = (\{S\}, \{a,b\}, P, S)$ mit $P\colon S \to aSa \mid bSb \mid \varepsilon$

(d) Es gibt hier drei Ansätze zur Lösung:

1. $G = (\{S,X,O,A,B\}, \{a,b\}, P, S)$ mit Produktionen P wie folgt

$$\begin{array}{lll} S \to XO & O \to \varepsilon & X \to XaA \mid XbB \mid \varepsilon \\ Aa \to aA & Ba \to aB & AO \to Oa \\ Ab \to bA & Bb \to bB & BO \to Ob \end{array}$$

Wir schreiben alle Buchstaben das Wortes doppelt und verschieben A und B bis zum Ende (markiert mit O) und wandeln dort das Zeichen wieder um. Vielen Dank an David Schneller für diese Lösung.

2. $G = (\{S,X,O,A,B,\overrightarrow{A},\overrightarrow{B}\}, \{a,b\}, P, S)$ mit Produktionen P wie folgt

$$\begin{array}{lll} S \to XO & O \to \varepsilon & X \to aXA \mid bXB \mid O \\ OA \to O\overrightarrow{A} & \overrightarrow{A}A \to A\overrightarrow{A} & \overrightarrow{A}B \to B\overrightarrow{A} \\ OB \to O\overrightarrow{B} & \overrightarrow{B}A \to A\overrightarrow{B} & \overrightarrow{B}B \to B\overrightarrow{B} \\ \overrightarrow{A}O \to a & \overrightarrow{A}a \to aa & \overrightarrow{A}b \to ab \\ \overrightarrow{B}O \to b & \overrightarrow{B}a \to ba & \overrightarrow{B}b \to bb \end{array}$$

Wir erzeugen erst wOw^RO und verschieben dann alle Buchstaben von der Mitte beginnend an das Ende (O). Vielen Dank an Jan Schuchardt für diese Lösung.

3. $G = (\{S, S', A, \overleftarrow{A}, B, \overleftarrow{B}, \overrightarrow{X}, \bullet, \circ\}, \{a, b\}, P, S)$ mit Produktionen P wie folgt

$$
\begin{array}{llll}
S \to S'\bullet & S' \to AS'A \mid BS'B \mid \circ & & \\
A \to a & A\bullet \to \overleftarrow{A}\circ & A\overleftarrow{A} \to \overleftarrow{A}A & B\overleftarrow{A} \to \overleftarrow{A}B \\
B \to b & B\bullet \to \overleftarrow{B}\circ & A\overleftarrow{B} \to \overleftarrow{B}A & B\overleftarrow{B} \to \overleftarrow{B}B \\
\circ\overleftarrow{A} \to A\bullet & \circ\overleftarrow{B} \to B\bullet & \bullet \to \circ\overrightarrow{X} & \\
\overrightarrow{X}A \to A\overrightarrow{X} & \overrightarrow{X}B \to B\overrightarrow{X} & \overrightarrow{X}\circ \to \bullet & \\
\circ\bullet \to \varepsilon & \bullet\circ \to \varepsilon & &
\end{array}
$$

Ein beliebiges Wort w zu erzeugen ist leicht. Im Aufgabenteil (c) sieht man, wie man ww^R bilden kann. Die Idee hinter dieser Aufgabe ist, zuerst ww^R zu bilden, aber alle Elemente von w^R voerst als Nichtterminale darzustellen. Im Anschluss verschieben wir die Nichtterminale in w^R an die richtige Position und ersetzen Sie schlussendlich durch Terminale. Die Intuition der Lösung ist, dass $\bullet, \overrightarrow{X}, \overleftarrow{X}$ den aktiven Bereich der Berechnung darstellen, wobei $X \in \{A, B\}$. $\circ$ ist der deaktivierte Begrenzer, der markiert, bis zu welcher Position wir noch Buchstaben verschieben müssen. $\circ$ trennt zunächst die w und w^R voneinander. Mit jedem Buchstaben, den wir vom Ende von w^R nach vorne verschieben, wandert $\circ$ weiter nach rechts.

(e) $G = (\{S, A, \overleftarrow{A}, \overrightarrow{A}, \bullet, \circ, \#, \$, X_\#\}, \{a, b\}, P, S)$ mit Produktionen P wie folgt

$$
\begin{array}{lll}
S \to \bullet AX_\#\$ & X_\# \to \#X_\# \mid \varepsilon & A \to a \\
\bullet A \to A \circ \overrightarrow{A} & \overrightarrow{A}A \to A\overrightarrow{A} & A\overleftarrow{A} \to \overleftarrow{A}A \\
\circ\overrightarrow{A}\# \to \bullet AAA & A\overrightarrow{A}\# \to \overleftarrow{A}\#A & \\
\overrightarrow{A}\$ \to \varepsilon & \circ \to \varepsilon & \circ\overleftarrow{A} \to \bullet A
\end{array}
$$

Die Grammatik rät zuerst die natürliche Zahl n und erzeugt dann alle ungerade Zahlen $1, 3, ..., 2n$ (1 wird mit einem Nichtterminal dargestellt, welches am Ende durch ein Terminal ersetzt wird, 3 mit drei solchen Nichtterminalen, etc.). Die Summe all dieser ungeraden Zahlen ergibt gerade n^2, damit erhalten wir gerade eine Zeichensequenz mit n^2 Nichtterminalen, die im letzten Schritt dann mit Terminalen ersetzt werden.

Übungsaufgabe Ü1.7. (*baaaaaaaaa...*)

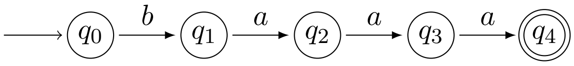
Sei $\Sigma = \{a, b\}$ und $L = \{ba^n\}$ für ein beliebiges aber festes $n \in N_0$.

(a) Sei $n = 3$. Zeichnen Sie einen NFA, der L erkennt.

(b) Zeigen sie, dass jeder NFA, der die Sprache L erkennt, mindestens $n + 2$ Zustände hat.

Lösungsskizze.

(a) Skizze:



(b) *Beweis.* Annahme zum Widerspruch: Es gibt einen Automaten $M = (Q, \Sigma, \delta, q_0, \mathsf{F})$ mit weniger als $n+2$ Zuständen, der L akzeptiert. Da M das Wort $w = ba^n$ akzeptiert und $|ba^n| = n+1$, muss es einen akzeptierenden Lauf $L = s_0 \xrightarrow{b} s_1 \xrightarrow{a} \ldots \xrightarrow{a} s_{n+1} \in \mathsf{F}$ geben wobei $s_0 = q_0$. Es gibt aber höchstens $n+1$ Zustände, daher folgt nach Schubfachprinzip, dass mindestens ein Zustand s_i durch den Lauf zweimal besucht wird. Dadurch gibt es einen Kreis $s_i \xrightarrow{w_i} s_{i+1} \xrightarrow{w_{i+1}}_D \cdots \xrightarrow{w_{j-1}} s_j$ in L, wobei $s_i = s_j$. Wenn wir den Kreis abkürzen, d.h. den Lauf $s_0 \xrightarrow{w_0} \ldots \xrightarrow{w_{i-1}} s_i \xrightarrow{w_j} s_{j+1} \xrightarrow{w_{j+1}} \ldots \xrightarrow{w_n} q_{n+1} \in F$ betrachten, dann ist dies auch ein akzeptierender Lauf. Deshalb akzeptiert M ein Wort w' mit $|w'| < n+1$, was ein Widerspruch zur Annahme ist, dass M die Sprache W akzeptiert. □

Übungsaufgabe Ü1.8. (*endlich Beweise*)

Sei Σ ein Alphabet und $A, B, C \subseteq \Sigma^*$ beliebig. Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Geben Sie für korrekte Aussagen einen Beweis an und widerlegen Sie falsche mithilfe eines geeigneten Gegenbeispiels.

(a) $A^* = A^+$ genau dann wenn (gdw.) $\varepsilon \in A$

(b) $A(B \cap C) = AB \cap AC$

(c) Falls $A \subseteq B$, dann $A^n \subseteq B^n$ für jedes $n \in N_0$.

(d) Unter der Annahme $A \neq \emptyset$ gilt: $A = AA$ gdw. $A = A^*$.

Lösungsskizze. Zu dieser Aufgabe gibt es auch Videolösungen: (a), (b), (c), (d)

(a) Die Aussage ist wahr. Wir zeigen beide Richtungen der Aussage getrennt.

$\Longleftarrow$: Annahme $\varepsilon \in A$. Per Definition gilt

$$A^* := \bigcup_{n \geq 0} A^n = A^0 \cup \bigcup_{n \geq 1} A^n := A^0 \cup A^+ := \{\varepsilon\} \cup A^+.$$

Es reicht also $\varepsilon \in A^+$ zu zeigen. Nach Vorlesung (VL) wissen wir $A^+ \overset{\text{VL}}{=} AA^* := \{uv \mid u \in A \text{ und } v \in A^*\}$. Wir wissen $\varepsilon \in A$ per Annahme und es gilt stets $\varepsilon \in A^*$. Somit $\varepsilon \in AA^* = A^+$. ■

$\Longrightarrow$: Beweis per Kontraposition (Erinnerung: Eine Aussage $A \implies B$ ist äquivalent zu ihrer Kontraposition $\neg B \implies \neg A$). Annahme $\varepsilon \notin A$. Zu zeigen $A^* \neq A^+$. Da $\varepsilon \in A^*$, reicht es $\varepsilon \notin A^+$ zu zeigen. Da $A^+ = \bigcup_{n \geq 1} A^n$ per Definition, reicht es $\varepsilon \notin A^n$ für $n \geq 1$ zu zeigen. Der Fall $n = 1$ gilt per Annahme. Für den Fall $n > 1$ haben wir $A^n := AA^{n-1} := \{uv \mid u \in A \text{ und } v \in A^{n-1}\}$. Sei nun $w \in A^n$. Dann gibt es $u \in A$ und $v \in A^{n-1}$ mit $w = uv$. Da $\varepsilon \notin A$ gilt $|u| > 0$, somit $|w| = |u| + |v| > 0 = |\varepsilon|$ und somit $w \neq \varepsilon$. □

(b) Die Aussage ist falsch. Wähle $A = \{a, aa\}$, $B = \{b\}$ und $C = \{ab\}$. Dann gilt

$$A(B \cap C) = A\emptyset = \emptyset \neq \{aab\} = \{ab, aab\} \cap \{aab, aaab\} = AB \cap AC.$$

(c) Die Aussage ist wahr. Annahme (Ann.) $A \subseteq B$. Wir zeigen die Aussage per Induktion über n. Für den Fall $n = 0$ gilt $A^0 = \{\varepsilon\} = B^0$. Im Fall $n+1$ gilt $A^n \subseteq B^n$ per Induktionshypothese (IH). Wähle nun $w \in A^{n+1} := AA^n := \{uv \mid u \in A \text{ und } v \in A^n\}$. Das heißt es gibt $u \in A \overset{\text{Ann.}}{\subseteq} B$ und $v \in A^n \overset{\text{IH}}{\subseteq} B^n$ mit $w = uv$. Somit $w = uv \in BB^n := B^{n+1}$. □

(d) Die Aussage ist wahr. Annahme $A \neq \emptyset$. Wir zeigen beide Richtungen der Aussage getrennt.

$\Longleftarrow$: Annahme $A = A^*$. Wir zeigen $A^* = A^*A^*$. Da $\varepsilon \in A^*$ gilt $A^* = \{\varepsilon\}A^* \subseteq A^*A^*$. Sei nun $w \in A^*A^*$. Dann existieren nach Definition $u, v \in A^*$ mit $w = uv$. Weiterhin existieren $u_1, \ldots, u_m, v_1, \ldots, v_n \in A$ mit $m, n \in N$ sodass $u = u_1 \cdots u_m$ und $v = v_1 \cdots v_n$. Somit $w = u_1 \cdots u_m v_1 \cdots v_n \in A^{m+n} \subseteq A^*$ und damit $A^*A^* \subseteq A^*$. Insgesamt also $A^* = A^*A^* \overset{\text{Ann.}}{=} AA$. ■

$\Longrightarrow$: Annahme $A = AA$. Nach Definition gilt $A \subseteq A^*$. Bleibt noch $A^* \subseteq A$ zu zeigen. Aufgrund der Definition von A^* reicht es $A^n \subseteq A$ für $n \in N$ zu zeigen. Wir zeigen die Aussage per Induktion über n.

Im Fall $n = 0$ gilt es $A^0 := \{\varepsilon\} \subseteq A$ zu zeigen. Sei hierfür $w \in A = AA$ ein kürzestes Wort. Dann lässt sich w in zwei Wörter $u, v \in A$ mit $|w| = |u| + |v|$ faktorisieren. Da w ein kürzestes Wort ist, gilt $|w| \leq |u|$ und $|w| \leq |v|$. Somit $|w| = |u| = |v| = 0$, also $w = \varepsilon$.

Im Fall $n+1$ folgt $A^{n+1} := AA^n \overset{\text{IH}}{\subseteq} AA \overset{\text{Ann.}}{=} A$. □