Reguläre Sprachen

Einleitung

In diesen Kapiteln besprechen wir ein mathematisches Standardmodell für Berechnungen. So sollen bei dieser Modellierung einige wichtige Aspekte erfüllt werden. Die Eingabe von Informationen in das Modell und infolgedessen soll unser Modell Befehle ausführen. Die Gesamtheit der Operationen soll endlich bleiben und unser Modell darf "Speicher" besitzen. Diese Anforderungen an das Modell werden von endlichen Automaten und deren korrespondierenden Sprachen erfüllt. Eine Motivation zu solch einem Modell entspringt dem Wortproblem, bzw. allgemeiner dem Untersuchen von Elementen freier Gruppen. So werden wir diesen Zusammenhang im nächsten Kapitel betrachten.

Definition 7.1

Sei $\Sigma = \{x_1, \ldots, x_n\}$ ein Alphabet, jede beliebige Teilmenge $\mathcal{L}$ von Wörtern im freiem Monoid Σ^* auf Σ, heißt Sprache ($\mathcal{L} \subseteq \Sigma^* = \{\prod_{i=1}^{m} y_i^{e_i} \,|\, \forall i \in \{1, \ldots, m\} : y_i \in \Sigma, e_i \in \mathbb{N}_0,\, m \in \mathbb{N}_0\}$).

Bemerkung

Es gibt zahlreiche Beispiele für Sprachen. Ist $\Sigma = \{a\}$, so ist zum Beispiel die Menge aller Ketten mit gerader Länge, $\{a^{2n} | n \in \mathbb{N}\}$, eine Sprache. Ähnlich erhalten wir für $\Sigma = \{a, b, c, ..., A, B, C, ...\}$ und der Menge aller Wörter im Duden eine Sprache. Wir interessieren uns in diesem Kapitel für Sprachen, welche von einem "einfachem" Computer berechnet werden können.

Definition 7.2

Sei $\mathcal{M}$ ein Automat, so definiere diesen wie folgt:

1. $\mathcal{M}$ ist ein gerichteter Graph
2. Zu $\mathcal{M}$ ist Σ das assoziierte Alphabet
3. Es existiert eine Teilmenge von Zuständen, $(\emptyset \neq)\, \mathcal{S}(\mathcal{M}) \subseteq \mathcal{Z}(\mathcal{M})$, die Startzustände (markiert durch ein S im Zustand)
4. Es existiert eine Teilmenge von Zuständen, $(\emptyset \neq)\, \mathcal{A}(\mathcal{M}) \subseteq \mathcal{Z}(\mathcal{M})$, die akzeptierten Zustände (markiert durch einen doppelten Rand im Zustand)
5. Die gerichteten Zustandsübergänge $(\emptyset \neq) \mathcal{U}(\mathcal{M})$ sind mit Elementen von Σ beschriftet

Bemerkung zur Definition 7.2

Die Sprache, welche von einem Automaten $\mathcal{M}$ angenommen wird, ist die Menge aller Wörter $w \in \Sigma^*$, welche zu den gerichteten Pfaden p_w korrespondieren, die an einem Startzustand anfangen und in einem akzeptierten Zustand enden, also genau alle gültigen Wörter die $\mathcal{M}$ erzeugen kann, bzw. die $\mathcal{M}$ annimmt. Wir schreiben $\mathcal{L}(\mathcal{M})$ für diese Sprache. Ist $\mathcal{M}$ endlich ($|\mathcal{Z}(\mathcal{M})|, |\mathcal{U}(\mathcal{M})| < \infty$), so bezeichnen wir den Automaten als endlichen (Zustands-) Automaten, kurz EA (ist $\mathcal{S}(\mathcal{M})$, $\mathcal{A}(\mathcal{M})$ oder $\mathcal{U}(\mathcal{M})$ leer, so sei $\mathcal{L}(\mathcal{M}) = \mathcal{L} = \emptyset$, wir interessieren uns aber insbesondere für nicht-leere Automaten). Bezeichne mit $\mathcal{U}(v)$ alle Übergänge die $v \in \mathcal{Z}(\mathcal{M})$ verlassen. Für den mit $a \in \Sigma$ beschrifteten Übergang $u \in \mathcal{U}(v_1)$ von $v_1 \in \mathcal{Z}(\mathcal{M})$ nach $v_2 \in \mathcal{Z}(\mathcal{M})$ schreibe: $u : \Sigma \times \mathcal{Z}(\mathcal{M}) \to \Sigma \times \mathcal{Z}(\mathcal{M})$, $(\emptyset, v_1) \mapsto (\emptyset a, v_2)$ bzw. $(p_{v_1}, v_1) \mapsto (p_{v_1} a, v_2)$ für das Wort zu $p_{v_1} \in \mathcal{L}(\mathcal{M})$, welches dem Pfad von einem Startzustand bis zu v_1 entspricht. Manchmal wird auch geschrieben $v_1 \to a v_2$, bzw. $p_w v_1 \to p_w a v_2$ oder für die Zustandsübergangsfunktion $\delta : \mathcal{Z}(\mathcal{M}) \times \Sigma \to \mathcal{Z}(\mathcal{M})$, $\delta(v_1, a) = v_2$. So wird $\mathcal{M}$ auch als 5-Tupel aufgefasst: $(\mathcal{Z}(\mathcal{M}),\, \Sigma,\, \mathcal{U}(\mathcal{M})/\delta,\, \mathcal{S}(\mathcal{M}),\, \mathcal{A}(\mathcal{M}))$.

Beispiel 7.3

Die Sprache $\{a^{3n} | n \in \mathbb{N}_0\}$ wird mit dem endlichen Automaten in Fig. 7.1 assoziiert.

Beispiel 7.4

Wir haben $\Sigma = \{a, b\}$ und $\mathcal{L}(\mathcal{M}) = \{w \in \{a, b\}^* |$ gerade viele b in $w\}$ in Fig. 7.2.

Beispiel 7.5

Jedes Element der unendlichen Dieeder-Gruppe D_∞ kann eindeutig als (reduziertes) Wort in den alternierenden Erzeugern $\{a, b\}$ geschrieben werden. Diese Menge an Wörtern ist die Sprache, die von beiden Automaten in Fig. 7.3 angenommen wird. Der linke Automat (Fig. 7.3.1) enthält den rechten Automaten (Fig. 7.3.2) sowie einen zusätzlichen Zustand. Dieser zusätzliche Zustand wird als Fehlzustand bezeichnet. Von diesem aus lässt sich kein akzeptierter Zustand erreichen.

Definition 7.6

Ein nicht-deterministischer Automat, kurz NEA, ist ein Automat, bei welchem Übergänge mit einem neuen Buchstaben, $\varepsilon \notin \Sigma$, beschriftet werden können. Die Sprache eines solchen nicht-deterministischen Automaten ist die Menge aller Wörter, welche zu den Pfaden korrespondieren, die an einem Startzustand anfangen und in einem akzeptierten Zustand enden, wobei am Ende alle ε aus dem Wort entfernt werden.

Bemerkung zur Definition 7.6

Ebenfalls können wir einen Automaten als nicht-deterministisch bezeichnen, falls von einem Zustand aus mehr als ein Übergang mit dem gleichen Buchstaben aus Σ existieren darf. Leicht ist einzusehen, dass diese Definitionen äquivalent sind.

Beispiel 7.7

Sei $\mathcal{L}$ die Sprache in $\Sigma^* = \{a, b, c\}^*$, bestehend aus Wörtern mit anfänglich nicht-leerer Kette von a's, gefolgt von möglicherweise leeren Ketten von b's und anschließend von möglicherweise leeren Ketten von c's. Diese Sprache wird vom NEA in Fig. 7.4.1 beschrieben (die gleiche Sprache wird auch vom DEA in Fig. 7.4.2 beschrieben).

Definition 7.8

Ein deterministischer Automat, kurz DEA, ist ein endlicher Automat mit den Eigenschaften:

1. Es existiert genau ein Startzustand ($|\mathcal{S}(\mathcal{M})| = 1$)
2. Keine 2 Zustandsübergänge an einem Zustand tragen den gleichen Buchstaben, bzw. es existieren keine ε-Übergänge

Ein deterministischer Automat heißt vollständig, wenn für alle Zustände und jeden Buchstaben vom Alphabet genau ein Übergang von jedem Zustand aus existiert, also
$\forall v \in \mathcal{Z}(\mathcal{M})\, \forall a \in \Sigma\, \exists! u \in \mathcal{U}(v) : (p_v, v)u = (p_v a, \tilde{v})$, bzw. $\forall v \in \mathcal{Z}(\mathcal{M})\, \forall a \in \Sigma : |\delta(v, a)| = 1$.

Bemerkung zur Definition 7.6/7.8

Aus den beiden Definitionen geht hervor, dass ein DEA stets ein NEA ist. Die Annahme, dass genau ein Startzustand existiert, ist nicht bei allen Autoren Teil der Definition eines DEA. In Satz 7.12 werden wir sehen, dass beide Konventionen äquivalent bezüglich der Sprachen sind, welche die jeweils so definierten DEAs annehmen.

Lemma 7.9

Sei $\mathcal{L}$ die Sprache, die von einem DEA $\mathcal{M}_1$ angenommen wird, so existiert mindestens ein vollständiger DEA $\mathcal{M}_2$, sodass $\mathcal{L}$ von diesem angenommen wird und wir haben $\mathcal{L}(\mathcal{M}_1) = \mathcal{L}(\mathcal{M}_2)$.

Beweis 7.9

Sei $\mathcal{M}_1$ ein DEA, so ist schon Σ, $\mathcal{Z}(\mathcal{M}_1)$, $\mathcal{U}(\mathcal{M}_1)$ endlich. Ergänze nun einen Fehlzustand v_f zu $\mathcal{M}_1$. Füge dann an jeden Zustand $v \in \mathcal{Z}(\mathcal{M}_1)$, für jeden fehlenden Buchstaben $a \in \Sigma$, bezüglich der Übergänge von v zu anderen Zuständen, einen Übergang mit a zu v_f an. Zusätzlich soll von v_f für jeden Buchstaben aus Σ ein Übergang auf sich selbst hinzugefügt werden. So wird jeder Zustand vervollständigt, womit schlussendlich auch der neue Automat $\mathcal{M}_2$ vollständig ist. Offensichtlich fügen wir auf diese Weise nur endlich viele Zustände und Übergänge zu $\mathcal{M}_2$ hinzu. So ist $\mathcal{M}_2$ ein vollständiger DEA und aufgrund dessen Konstruktion erhalten wir direkt $\mathcal{L}(\mathcal{M}_1) = \mathcal{L}(\mathcal{M}_2)$. $\square$

Bemerkung zum Beweis 7.9

Tatsächlich lassen sich beliebig viele verschiedene vollständige DEA konstruieren, welche die gleiche Sprache $\mathcal{L}$ annehmen.

Definition 7.10

Eine Sprache heißt regulär, wenn diese von einem DEA angenommen wird.

Satz 7.11

Seien $\mathcal{K}$ und $\mathcal{L}$ reguläre Sprachen mit gemeinsamen Alphabet Σ. So sind die folgenden Sprachen ebenfalls reguläre Sprachen:

1. Die komplementäre Sprache, $\Sigma^* \setminus \mathcal{K} = \mathcal{K}^c$
2. Die Vereinigung, $\mathcal{K} \cup \mathcal{L}$
3. Der Schnitt, $\mathcal{K} \cap \mathcal{L}$
4. Die Konkatenation $\mathcal{K}\mathcal{L}$, bestehend aus allen Wörtern der Form $\{w_{\mathcal{K}} w_{\mathcal{L}} \mid w_{\mathcal{K}} \in \mathcal{K}, w_{\mathcal{L}} \in \mathcal{L}\}$
5. $\mathcal{K}^* = \mathcal{K} \cup \mathcal{K}\mathcal{K} \cup \mathcal{K}\mathcal{K}\mathcal{K} \cup \dots$

Beweis von 7.11 folgt nach 7.12

Satz 7.12

Die Menge der Sprachen welche von nicht-deterministischen endlichen Automaten angenommen wird, ist die gleiche, wie die Menge der regulären Sprachen.

Bemerkung (liegt w in der Sprache eines NEA?)

Wir betrachten kurz, wie man testen könnte, ob ein gegebenes Wort w in der Sprache eines NEA liegt. Für DEA ist dies trivial, da jedes Wort im Automaten einem eindeutigen Pfad zugeordnet werden kann, welcher in einem Startzustand beginnt und in einem akzeptierten Zustand endet. Für NEA hingegen haben wir nicht immer eindeutige Pfade. Angenommen man habe genügend Papier, so könnte man w testen in dem man für jeden (verschiedenen) Startzustand im Automaten $\mathcal{M}$ einen Stapel anfertigt. Als nächstes betrachte den ersten Buchstaben von w und notiere für jeden Startzustand alle möglichen Übergänge in die Zustände, welche zum ersten Buchstaben passen (inklusive gegebener ε-Übergänge). Erstelle danach neue Stapel, für jeden Zustand einen, indem man sich nach dem ersten Buchstaben befinden könnte. Fahre mit diesem Vorgehen fort, bis das ganze Wort gelesen wurde. Wenn sich in den letzten Stapeln wenigstens ein akzeptierter Zustand befindet, so existiert mindestens ein Pfad in $\mathcal{M}$, welcher, mit gegebenenfalls zusätzlichen ε, das Wort w beschreibt.

Beweis 7.12

Wir wollen zeigen $\{\mathcal{L}(\mathcal{M}) \,|\, \mathcal{M} \text{ ist NEA}\} = \{\mathcal{L}(\mathcal{M}) \,|\, \mathcal{M} \text{ ist DEA}\}$
"$\Leftarrow$"
Nach Definition ist schon jeder DEA ein NEA, womit die Menge der von DEA angenommenen Sprachen in der Menge der von NEA angenommenen Sprachen enthalten ist, also gilt offensichtlich $\{\mathcal{L}(\mathcal{M}) \,|\, \mathcal{M} \text{ ist NEA}\} \supseteq \{\mathcal{L}(\mathcal{M}) \,|\, \mathcal{M} \text{ ist DEA}\}$.
"$\Rightarrow$"
Sei $\mathcal{M} = (\mathcal{Z}(\mathcal{M}), \Sigma, \delta, \mathcal{S}(\mathcal{M}), \mathcal{A}(\mathcal{M}))$ ein NED. Als Erstes konstruieren wir einen Automaten, welcher die gleiche Sprache wie $\mathcal{M}$ annimmt und welcher keine mit ε beschrifteten Übergänge besitzt. Sei $S \subseteq \mathcal{Z}(\mathcal{M})$. Der ε-Abschluss von S, $\varepsilon(S) = \{\delta(s,\varepsilon) \,|\, s \in S\} \dot\cup S$, ist die Menge aller Zustände von $\mathcal{M}$, welche durch einen ε-Übergang, beginnend in S, erreicht werden können. Insbesondere haben wir $S \subseteq \varepsilon(S)$, da wir von jedem Zustand aus S mit dem leeren Pfad $p_\emptyset$ weiterhin in diesem verbleiben. Eine Teilmenge $S \subseteq \mathcal{Z}(\mathcal{M})$ ist ε-Abgeschlossen wenn $S = \varepsilon(S)$, also von S aus keine ε-Übergänge existieren. Sei $\mathcal{M}_\varepsilon$ der Automat, wessen Zustände wir mit den ε-Abgeschlossene Teilmengen von $\mathcal{Z}(\mathcal{M})$ identifizieren, $\mathcal{Z}(\mathcal{M}_\varepsilon) = \{S \subseteq \mathcal{Z}(\mathcal{M}) \,|\, S = \varepsilon(S)\}$. Auf diese Weise eliminieren wir alle ε-Übergänge. Die Startzustände von $\mathcal{M}_\varepsilon$ sind alle ε-Abgeschlossenen Teilmengen von $\mathcal{Z}(\mathcal{M})$, welche mindestens einen Startzustand enthalten. Die akzeptierten Zustände von $\mathcal{M}_\varepsilon$ sind alle ε-Abgeschlossenen Teilmengen von $\mathcal{Z}(\mathcal{M})$, welche mindestens einen akzeptierten Zustand enthalten. Wenn $S \subseteq \mathcal{Z}(\mathcal{M})$ und $a \in \Sigma$ gilt, definiere Sa als Menge aller Zustände v, wobei der Zustandsübergang zu v hin mit a beschriftet ist und v in S beginnt, also $Sa = \{\delta(s,a) \,|\, s \in S\}$. Existiert für $a \in \Sigma$ ein Übergang, der mit a beschriftet ist, von S nach $\varepsilon(Sa)$, so ist $\varepsilon(Sa) = \delta_\varepsilon(S,a)$. Nach Konstruktion enthält $\mathcal{M}_\varepsilon$ keine Übergänge die ε tragen, so gilt dann $\mathcal{M}_\varepsilon = (\{Z \subseteq \mathcal{Z}(\mathcal{M}) | Z = \varepsilon(Z)\}, \Sigma, \varepsilon(Va), \{S \in \mathcal{Z}(\mathcal{M}_\varepsilon) | S \cap \mathcal{S}(\mathcal{M}) \neq \emptyset\}, \{A \in \mathcal{Z}(\mathcal{M}_\varepsilon) | A \cap \mathcal{A}(\mathcal{M}) \neq \emptyset\})$. Induktiv über die "Länge" von $w \in \Sigma^*$ erhalten wir $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{M}_\varepsilon)$. Nun wollen wir $\mathcal{M}_\varepsilon$ zu einem deterministischen Automaten $\mathcal{D}$ modifizieren. Die Menge der Zustände von $\mathcal{D}$ identifizieren wir mit allen Teilmengen der Zustände von $\mathcal{M}_\varepsilon$, also $\mathcal{Z}(\mathcal{D}) = \mathcal{P}(\mathcal{Z}(\mathcal{M}_\varepsilon))$. Der einzige Startzustand von $\mathcal{D}$ ist dann gerade die Teilmenge von $\mathcal{Z}(\mathcal{M}_\varepsilon)$, welche alle Startzustände von $\mathcal{M}_\varepsilon$ enthält, also $|\mathcal{S}(\mathcal{D})| = 1$ und $\mathcal{S}(\mathcal{D}) = \bigcup \mathcal{S}(\mathcal{M}_\varepsilon)$. Die akzeptierten Zustände von $\mathcal{D}$ sind dann die Teilmengen von $\mathcal{Z}(\mathcal{D})$, welche mindestens einen akzeptierten Zustand von $\mathcal{M}_\varepsilon$ enthalten, also $\mathcal{A}(\mathcal{D}) = \{A \in \mathcal{Z}(\mathcal{D}) |\, A \cap \mathcal{A}(\mathcal{M}_\varepsilon) \neq \emptyset\}$. In $\mathcal{D}$ existiert so ein Übergang mit x von V nach V' $(V, V' \in \mathcal{Z}(\mathcal{D}))$, wenn für jedes $v \in V$ ein Übergang mit x von v zu einem beliebigen $v' \in V'$ existiert und V' völlig aus solchen Zuständen v' besteht, $V' = \{\delta_\varepsilon(v,x) \,|\, v \in V\}$ und $\delta_\mathcal{D}(V,x) = \bigcup_{v \in V} \delta_\varepsilon(v,x)$. Nach Konstruktion ist $\mathcal{D} = (\mathcal{P}(\mathcal{Z}(\mathcal{M}_\varepsilon)), \Sigma, \varepsilon(\bigcup_{v \in V} \delta(v,a)), \varepsilon(\mathcal{S}(\mathcal{M}_\varepsilon)), \{V \in \mathcal{Z}(\mathcal{D}) | V \cap \mathcal{A}(\mathcal{M}) \neq \emptyset\})$ ein deterministischer Automat. Ist $w \in \mathcal{L}(\mathcal{M}_\varepsilon)$, so beschreibt w einen gerichteten Pfad über die Zustände in $\mathcal{D}$, vom Startzustand ausgehend. Der Endzustand zu einem assoziierten Pfad p_w korresspondiert zur Menge der Zustände von $\mathcal{M}_\varepsilon$, in welchen mindestens ein akzeptierter Zustand liegt, da $w \in \mathcal{L}(\mathcal{M}_\varepsilon)$. Somit ist $\mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{M}_\varepsilon) = \mathcal{L}(\mathcal{M})$, also $\{\mathcal{L}(\mathcal{M}) \,|\, \mathcal{M} \text{ ist NEA}\} \subseteq \{\mathcal{L}(\mathcal{M}) \,|\, \mathcal{M} \text{ ist DEA}\}$ $\square$

Bemerkung zum Beweis 7.12

Anhand dieser Konstruktion sehen wir, dass die nicht-deterministischen endlichen Automaten und die deterministischen endlichen Automaten die gleiche "Mächtigkeit" bezüglich der von endlichen Automaten berechenbaren Sprachen haben. So ist es jedoch möglich, dass zu einer gegebenen Sprache der NEA "kleiner" als der DEA ist, also weniger Zustände oder Zustandsübergänge besitzt.

Beweis 7.11

1. Mit Lemma 7.9 erhalten wir eine Reguläre Sprache $\mathcal{K} = \mathcal{L}(\mathcal{M}_\mathcal{K})$ für den vollständigen DEA $\mathcal{M}_\mathcal{K}$. Konstruiere dann einen neuen Automaten $\overline{\mathcal{M}_\mathcal{K}}$ ausgehend von $\mathcal{M}_\mathcal{K}$, indem man die akzeptierten mit den nicht-akzeptierten Zuständen vertauscht, also $\mathcal{Z}(\overline{\mathcal{M}_\mathcal{K}}) = \mathcal{Z}(\mathcal{M}_\mathcal{K})$, $\mathcal{S}(\overline{\mathcal{M}_\mathcal{K}}) = \mathcal{S}(\mathcal{M}_\mathcal{K})$, $\mathcal{A}(\overline{\mathcal{M}_\mathcal{K}}) = \mathcal{Z}(\mathcal{M}_\mathcal{K}) \setminus \mathcal{A}(\mathcal{M}_\mathcal{K})$, $\mathcal{U}(\overline{\mathcal{M}_\mathcal{K}}) = \mathcal{U}(\mathcal{M}_\mathcal{K})$. Somit ist nach Konstruktion $\overline{\mathcal{M}_\mathcal{K}}$

ebenfalls ein vollständiger DEA, da die Zustandübergänge invariant unter dem Vertauschen der akzeptierten mit den nicht-akzeptierten Zuständen bleiben. Sei dann v_w der Endzustand in $\mathcal{M}_{\mathcal{K}}$, bzw. in $\overline{\mathcal{M}_{\mathcal{K}}}$ der zu einem Pfad p_w korrespondiert für ein beliebiges Wort in Σ^*. Insbesondere ist p_w für w in beiden Automaten eindeutig bestimmt. So ist dann genau $w \in \mathcal{L}(\mathcal{M}_{\mathcal{K}}) \wedge w \notin \mathcal{L}(\overline{\mathcal{M}_{\mathcal{K}}})$ oder $w \notin \mathcal{L}(\mathcal{M}_{\mathcal{K}}) \wedge w \in \mathcal{L}(\overline{\mathcal{M}_{\mathcal{K}}})$, also $\mathcal{L}(\overline{\mathcal{M}_{\mathcal{K}}}) = \{w \in \Sigma^* \mid w \notin \mathcal{L}(\mathcal{M}_{\mathcal{K}})\} = \Sigma^* \setminus \mathcal{L}(\mathcal{M}_{\mathcal{K}}) = \mathcal{K}^c$. Dann ist $\mathcal{K}^c$ ebenfalls eine reguläre Sprache, da $\mathcal{K}^c$ von einem vollständigem DEA angenommen wird.

2. Seien $\mathcal{K} = \mathcal{L}(\mathcal{M}_{\mathcal{K}})$ und $\mathcal{L} = \mathcal{L}(\mathcal{M}_{\mathcal{L}})$ reguläre Sprachen mit jeweils DEA. Betrachte dann den disjunkten NEA $\mathcal{M}_{\mathcal{K}} \,\dot{\cup}\, \mathcal{M}_{\mathcal{L}}$ mit 2 Startzuständen. Dieser NEA nimmt dann offensichtlich alle Wörter aus $\mathcal{K}$ und $\mathcal{L}$ an, also $\mathcal{L}(\mathcal{M}_{\mathcal{K}} \,\dot{\cup}\, \mathcal{M}_{\mathcal{L}}) = \mathcal{K} \cup \mathcal{L}$. Mit Satz 7.12 ist dann $\mathcal{K} \cup \mathcal{L}$ regulär.

3. $\mathcal{K} \cap \mathcal{L} = \Sigma^* \setminus ((\Sigma^* \setminus \mathcal{K}) \cup (\Sigma^* \setminus \mathcal{L})) = (\mathcal{K}^c \cup \mathcal{L}^c)^c$. Mit 7.11.1 und 7.11.2 ist dann $\mathcal{K} \cap \mathcal{L}$ regulär.

4. Konstruiere einen neuen NED aus $\mathcal{M}_{\mathcal{K}} \,\dot{\cup}\, \mathcal{M}_{\mathcal{L}}$ mit genau einem Startzustand $\mathcal{S}(\mathcal{M}_{\mathcal{K}} \,\dot{\cup}\, \mathcal{M}_{\mathcal{L}}) = \mathcal{S}(\mathcal{M}_{\mathcal{K}})$. Für jeden akzeptierten Zustand von $\mathcal{M}_{\mathcal{K}}$ fügen wir einen mit ε beschrifteten Übergang zum Startzustand von $\mathcal{M}_{\mathcal{L}}$ hinzu, also $\forall v \in \mathcal{A}(\mathcal{M}_{\mathcal{K}}) \subseteq \mathcal{Z}(\mathcal{M}_{\mathcal{K}} \,\dot{\cup}\, \mathcal{M}_{\mathcal{L}}) \exists u \in \mathcal{U}(v) : (p_v, v)u = (p_v\varepsilon, s)$ für $s \in \mathcal{S}(\mathcal{M}_{\mathcal{L}})$, bzw. $\forall v \in \mathcal{A}(\mathcal{M}_{\mathcal{K}}) \subseteq \mathcal{Z}(\mathcal{M}_{\mathcal{K}} \,\dot{\cup}\, \mathcal{M}_{\mathcal{L}}) : \delta(v, \varepsilon) = s \in \mathcal{S}(\mathcal{M}_{\mathcal{L}})$. So nimmt dieser NEA $\mathcal{KL}$ an und $\mathcal{KL}$ ist nach 7.12 regulär.

5. Konstruiere einen NEA, ähnlich wie für 7.11.4. Für jeden akzeptierten Zustand von $\mathcal{M}_{\mathcal{K}}$ fügen wir einen mit ε beschrifteten Übergang zum Startzustand von $\mathcal{M}_{\mathcal{K}}$ hinzu, also $\forall v \in \mathcal{A}(\mathcal{M}_{\mathcal{K}}) \exists u \in \mathcal{U}(v) : (p_v, v)u = (p_v\varepsilon, s)$ für $s \in \mathcal{S}(\mathcal{M}_{\mathcal{K}})$, bzw. $\forall v \in \mathcal{A}(\mathcal{M}_{\mathcal{K}}) : \delta(v, \varepsilon) = s \in \mathcal{S}(\mathcal{M}_{\mathcal{K}})$. So nimmt dieser NEA $\mathcal{K}^*$ an und $\mathcal{K}^*$ ist nach 7.12 regulär. □

Beispiel 7.13

Die Menge der reduzierten Wörter (Normalform) in $\Sigma^* = \{x, x^{-1}, y, y^{-1}\}^*$ formt eine reguläre Sprache. Ein Automat $\mathcal{M}$, der diese Sprache annimmt, ist in Fig. 7.5.1 zu sehen. $\mathcal{M}$ besitzt 5 Zustände, wobei jeder dieser bereits ein akzeptierter Zustand ist (somit beschreibt $\mathcal{M}$ auch das leere Wort). Zusätzlich kann sich $\mathcal{M}$ "merken", für ein gegebenes Wort $w \in \Sigma^*$, welcher Buchstabe als letztes gelesen wurde. So existiert in jedem Zustand (ausgenommen dem Start) genau ein Übergang mit jedem Buchstaben von Σ, bis auf das Inverse des zuletzt gelesenen Buchstabens. Insbesondere ist $\mathcal{M}$ damit nicht vollständig. Die Elemente der freien Gruppe $\mathbb{F}_2$ korrespondieren zu den reduzierten Wörtern in Σ^*. Somit korrespondieren die Elemente der Gruppe zu Wörtern in einer regulären Sprache. Die gerade Untergruppe von $\mathbb{F}_2$ (vgl. Prop. 3.12) ist die Menge in Σ^* reduzierter Wörter mit gerader Länge (vgl. Fig. 7.5.2). So ist nach 7.11.3 die Menge dieser Wörter ebenso eine reguläre Sprache.

Bemerkung (Nicht alle Sprache sind regulär)

Eine einfache Intuition, um zu entscheiden, ob eine gegebene Sprache regulär ist, ist den benötigten "Speicher" zu betrachten, den man benötigen würde, um zu bestimmen, ob ein beliebiges Wort tatsächlich in der gegebenen Sprache liegt. Der Automat einer regulären Sprache kommt schon mit endlichem Speicher aus (vgl. 7.15).

Satz 7.14 (Pumping Lemma (Schleifensatz), 1959)

Sei $\mathcal{L}$ eine reguläre Sprache. Dann existiert ein $n \in \mathbb{N}$, sodass jedes beliebige Wort $x \in \mathcal{L}$ mit $|x| > n$, geschrieben werden kann als $x = uvw$, wobei:

1. $v \neq \emptyset$
2. $|u| < n$ (bzw. $|uv| \leq n$)
3. $\forall i \in \mathbb{N}_0 : uv^iw \in \mathcal{L}$ $(v^0 = \emptyset)$

Beweis 7.14 (Beweisskizze: Fig. 7.6.1)

Sei $\mathcal{M}$ ein vollständiger DEA mit $\mathcal{L}(\mathcal{M}) = \mathcal{L}$. Sei $n = |\mathcal{Z}(\mathcal{M})|$ und sei x ein Wort mit $|x| > n$. Da $\mathcal{M}$ ein vollständiger DEA ist, existiert ein eindeutiger gerichteter Pfad von Zustandübergängen, ausgehend vom Startzustand, sodass dieser Pfad x beschreibt. Da x länger als n ist, muss dieser Pfad mindestens einen Zustand $z \in \mathcal{Z}(\mathcal{M})$ zweimal durchschreiten. Somit enthält der Pfad assoziiert zu x eine Schleife. Sei dann u der möglicherweise leere Prefix von x assoziiert zum anfänglichen Teilpfad (bevor der Schleife), ausgehend vom Startzustand, bis hin zum Zustand z. Sei dann ebenso w der möglicherweise leere Suffix von x assoziiert zu dem Teilpfad von z bis hin zum akzeptierten Endzustand vom ursprünglichem Pfad. So folgt dann, dass alle die Wörter $uv^iw\,(\forall i \in \mathbb{N}_0)$ Pfade beschreiben, welche in einem Startzustand beginnen und in einem akzeptierten Zustand enden, womit diese Wörter also vom Automaten angenommen werden, also zur Sprache gehören. $|u| < n$ folgt aus $|x| > n$, da $v \neq \emptyset \Rightarrow |v| \geq 1$, also muss nach Vorraussetzung die Schleife vor dem $(n+1)$-ten Zustand beginnen und da v mindestens einen Buchstaben trägt muss schon $|u| \leq (n-1)$ sein, also $|u| < n$, bzw. $|uv| \leq n$. □

Bemerkung zum Pumping Lemma

Das kleinste n, das diese Eigenschaft erfüllt wird die Pumping-Zahl von $\mathcal{L}$ genannt. Beachte, dass es wichtig war, dass wir an w keine weiteren Bedingungen gestellt haben, also darf gelten $|w| \geq n$, womit w selbst Schleifen enthalten darf. Desweiteren sind diese Zerlegungen von x nicht zwingend eindeutig (vgl. Fig. 7.6.2).

Beispiel 7.15

Das Pumping Lemma zeigt, dass die Sprache $\mathcal{L} = \{a^n b^n \,|\, n \geq 1\}$ nicht regulär ist (für $\{a^n b^n \,|\, n\}$, $\{a^m b^m \,|\, m \leq n\}$ mit n fest, erhalten wir jedoch reguläre Sprachen, vgl. Fig. 7.7 mit n=3). Nehme das Gegenteil an, also dass $\mathcal{L}$ regulär sei und sei n die Anzahl der Zustände eines (endlichen) Automaten $\mathcal{M}$ mit korrespondierender Sprache $\mathcal{L}$. Dann enthält der Pfand zu a^{n+1} einen geschlossenen Teilpfad der Länge $1 \leq k \leq n$. So folgt, dass $a^{n+1+k}b^{n+1}$ ebenfalls vom Automaten angenommen werden müsste, was jedoch ein Widerspruch zu $\mathcal{L} = \mathcal{L}(\mathcal{M})$ ist.

Definition 7.16

Sei $\mathcal{L}$ eine Sprache in Σ^* und sei w ein Wort in Σ^*. Der Konustyp von w ist die Menge aller Wörter w' aus Σ^* sodass ww' in $\mathcal{L}$ liegen. Schreibe dazu $Cone(w) = \{w' \in \Sigma^* \,|\, ww' \in \mathcal{L}\}$ und weiter $Cone(\mathcal{L}) = \{Cone(w) \,|\, w \in \Sigma^*\}$ ist der Konustyp der Sprache $\mathcal{L}$.

Beispiel 7.17

Sei $\Sigma = \{a\}$ und sei $\mathcal{L}$ die Sprache aller Wörter mit gerader Länge. So ist $Cone(a) = Cone(a^3) = \cdots = Cone(a^{2n+1}) = \{a^{2n+1} \,|\, n \in \mathbb{N}_0\}$ und $Cone(\emptyset) = Cone(a^2) = \cdots = Cone(a^{2n}) = \{a^{2n} \,|\, n \in \mathbb{N}_0\}$. Somit besteht $Cone(\mathcal{L})$ aus genau 2 Mengen, also jeweils die Menge aller ungeraden, bzw. geraden Wörter in Σ^*.

Satz 7.18 (Myhill-Nerode Theorem, 1957/1958)

Eine Sprache $\mathcal{L}$ ist regulär genau dann, wenn die Menge der Konustypen endlich ist, also
$\mathcal{L}$ regulär $\Leftrightarrow$ $|Cone(\mathcal{L})| < \infty$.

Beweis 7.18 (Beweisskizze: Fig. 7.8)

"$\Rightarrow$"
Sei $\mathcal{L}$ eine reguläre Sprache, so wird diese von einem vollständigen DEA $\mathcal{M}$ angenommen. Für jeden Zustand $v \in \mathcal{Z}(\mathcal{M})$ sei $\mathcal{M}_v$ der modifizierte Automat $\mathcal{M}$, mit dem einzigen Startzustand v, also

$\mathcal{S}(\mathcal{M}_v) = \{v\}$. Der Konustyp eines gegebenen Wortes w wird durch den Endzustand $v_w \in \mathcal{Z}(\mathcal{M})$ (also der Zustand, in dem w endet bzw. "fertig geschrieben wurde") zu einem assoziierten Pfad p_w bestimmt. So ist $Cone(w) = \{w' \in \Sigma^* \,|\, ww' \in \mathcal{L}(\mathcal{M})\} = \mathcal{L}(\mathcal{M}_{v_w}), \mathcal{A}(\mathcal{M}_{v_w}) \subseteq \mathcal{A}(\mathcal{M})$. Betrachte dazu die Wörter $w_1 \neq w_2$ mit $v_{w_1} = v_{w_2} = v \in \mathcal{Z}(\mathcal{M})$, so folgt schon: $Cone(w_1) = Cone(w_2) = \mathcal{L}(\mathcal{M}_v)$. Somit ist die Anzahl der Konustypen von $\mathcal{L}$ höchstens gleich der Anzahl der Zustände in $\mathcal{M}$ ($|Cone(\mathcal{L})| \leq |\mathcal{Z}(\mathcal{M})| < \infty$).
"$\Leftarrow$"
Sei nun $Cone(\mathcal{L})$ endlich. Wir konstruieren den Automaten $\mathcal{M}$, indem wir zuerst die Zustände von $\mathcal{M}$ mit den Konustypen von $\mathcal{L}$ identifizieren. Weiter sei $v_\emptyset$, bzw. $Cone(\emptyset)$ der eindeutige Startzustand, wobei $\emptyset$ für das leere Wort steht. Die akzeptierten Zustände von $\mathcal{M}$ sind dann die Konustypen, welche $\emptyset$ enthalten ($\emptyset \in Cone(w) \Rightarrow w\emptyset = w \in \mathcal{L}$, $v_w \in \mathcal{A}(\mathcal{M})$, bzw. $Cone(w) \in \mathcal{A}(\mathcal{M})$). Für alle Konustypen und alle $x \in \Sigma$ fügen wir einen mit x beschrifteten Übergang, vom Zustand, gegeben durch $Cone(w)$, zum Zustand, gegeben durch $Cone(wx)$, hinzu. Nun ist zu prüfen, ob die Möglichkeit besteht, dass diese Definition vom Wort w abhängt und nicht vom Konustyp von w. Also ist möglicherweise $Cone(w) = Cone(\hat{w})$, aber $Cone(wx) \neq Cone(\hat{w}x)$? Das kann jedoch nicht passieren, denn $(x \in \Sigma \wedge w \in \Sigma^*) \Rightarrow Cone(wx) = \{w' \in \Sigma^* \,|\, wx\,w' \in \mathcal{L}\} = \{xw' \in \Sigma^* \,|\, w\,xw' \in \mathcal{L}\} = \{w' \in \Sigma^* \,|\, xw' \in Cone(w)\}$. Somit folgt direkt $(Cone(w) = Cone(\hat{w})) \Rightarrow (\forall x \in \Sigma : Cone(wx) = Cone(\hat{w}x))$, da $Cone(wx) = \{w' \in \Sigma^* \,|\, xw' \in Cone(w)\} = \{w' \in \Sigma^* \,|\, xw' \in Cone(\hat{w})\} = Cone(\hat{w}x)$. So ist nach Konstruktion $\mathcal{M}$ ein vollständiger DEA. Nun bleibt zu überprüfen, dass $\mathcal{L} = \mathcal{L}(\mathcal{M})$ gilt. Als erstes fällt auf, dass gilt: $w \in \mathcal{L} \Leftrightarrow \emptyset \in Cone(w)$. Sei $w \in \Sigma^*$ beliebig. Da $\mathcal{M}$ ein vollständiger DEA ist, existiert ein eindeutiger Pfad in $\mathcal{M}$, von einem Startzustand ausgehend, über die entsprechenden Zustände, gemäß w, bis zu einem Endzustand v hin. Nach Konstruktion von $\mathcal{M}$, wird dieser Endzustand mit $Cone(w)$ assoziiert. Womit $w \in \mathcal{L}(\mathcal{M})$ genau dann, wenn $v \in \mathcal{A}(\mathcal{M})$, also wenn $Cone(w)$ ein akzeptierter Zustand ist, was aber genau nur dann gilt, wenn $\emptyset \in Cone(w)$. Somit ist dann nach unserer Konstruktion $\mathcal{L} = \mathcal{L}(\mathcal{M})$. $\square$

Bemerkung zur Notation

Ähnlich wie der Konustyp, bzw. $Cone(w)$, wird anstelle dessen auch die Nerode-Relation $\sim_\mathcal{L}$ verwendet: $(x \sim_\mathcal{L} y) \Leftrightarrow (\forall z \in \Sigma^* : xz \in \mathcal{L} \Leftrightarrow yz \in \mathcal{L})$

Beispiel 7.19

Sei $\Sigma = \{a, b\}$ und sei $\mathcal{L} = \{ww \,|\, w \in \Sigma^*\}$ die Sprache der "formalen" Quadrate in Σ^*. So ist $ba^ib \in Cone(a^i) \wedge ba^ib \notin Cone(a^j)\, \forall i \neq j$. Somit hat die Sprache $\mathcal{L}$ unendlich viele verschiedene Konustypen, womit nach Satz 7.18 $\mathcal{L}$ eine nicht-reguläre Sprache ist.

Weiter Bemerkungen

Einige äquivalente Definitionen regulärer Sprachen

1. $\mathcal{L}$ wird von einer regulären Grammatik erzeugt
2. $\mathcal{L}$ wird von einem EA angenommen
3. $\mathcal{L}$ wird von einem NEA angenommen
4. $\mathcal{L}$ wird von einem DEA angenommen
5. $\mathcal{L}$ kann durch einen regulären Ausdruck dargestellt werden
6. Die auf Σ^* definierte Nerode-Relation $\sim_{\mathcal{L}}$ hat endlichen Index

Lösbarkeit von Problemen und Chomsky-Hierarchie

Für reguläre Sprachen $\mathcal{L}_1$ und $\mathcal{L}_2$ sind folgende Probleme entscheidbar:

1. Wortproblem: $\Sigma^* \ni w \stackrel{?}{\in} \mathcal{L}_1$
2. Leerheitsproblem: $\mathcal{L}_1 \stackrel{?}{=} \emptyset$
3. Endlichkeitsproblem: $|\mathcal{L}_1| \stackrel{?}{<} \infty$
4. Äquivalenzproblem: $\mathcal{L}_1 \stackrel{?}{=} \mathcal{L}_2$ (Inklusionsproblem: $\mathcal{L}_1 \stackrel{?}{\subseteq} \mathcal{L}_2$)
5. Schnittproblem: $\mathcal{L}_1 \cap \mathcal{L}_2 \stackrel{?}{=} \emptyset$

Grammatik	Entscheidbarkei	Automaten	Zeitabschätzung
Typ-0 beliebige formale Grammatik	-	Turingmaschine	unbeschränkt
Typ-1 Kontextsensitive Grammatik	Wortproblem	linear platzbeschränkte ND-Turingmaschine	$\mathcal{O}(2^n)$
Typ-2 Kontextfreie Grammatik	Wortproblem, Leerheitsproblem, Endlichkeitsproblem	ND-Kellerautomat	$\mathcal{O}(n^3)$
Typ-3 reguläre Grammatik	Wortproblem, Leerheitsproblem, Endlichkeitsproblem, Äquivalenzproblem, Inklusionsproblem, Schnittproblem	EA	$\mathcal{O}(n)$

Abbildungen

Fig. 7.1

$\mathcal{L}(\mathcal{M}) = \{a^{3n} \mid n \in \mathbb{N}_0\}$

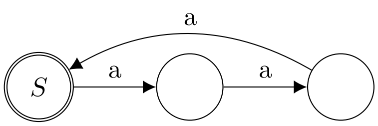


Fig. 7.2

$\mathcal{L}(\mathcal{M}) = \{w \in \{a, b\}^* \mid \#b \in w \text{ gerade } \}$

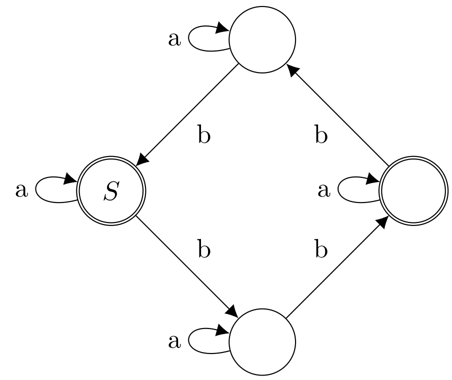


Fig. 7.3 (7.3.1 links, 7.3.2 rechts)

$\mathcal{L}(\mathcal{M}) = \{w \in \{a, b\}^* \mid w$ bestehend aus altanierenden a, b mit beliebiger Länge und beliebigem ersten Buchstaben $\}$

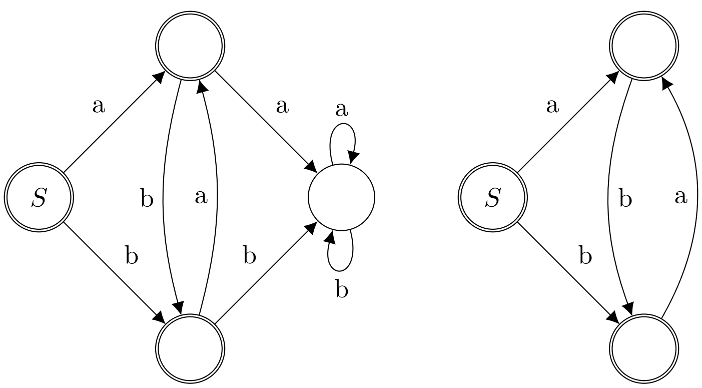


Fig. 7.4 (7.4.1 links, 7.4.2 rechts)

$\mathcal{L}(\mathcal{M}) = \{a^n b^m c^k \mid n \in \mathbb{N},\, m, k \in \mathbb{N}_0\}$

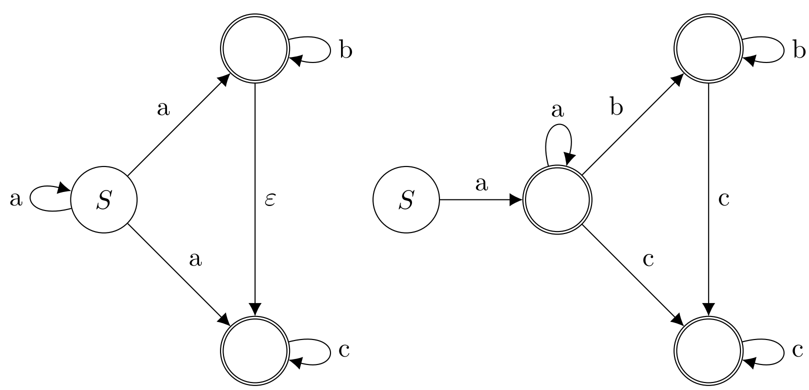
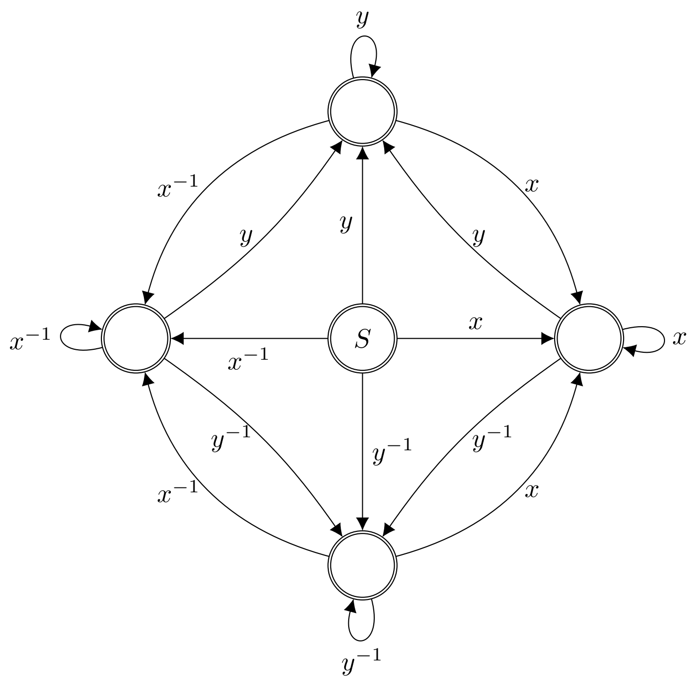


Fig. 7.5 (7.5.1 oben, 7.5.2 unten)

$\mathcal{L}(\mathcal{M}) = \{w \in \{x, x^{-1}, y, y^{-1}\}^* \mid w \text{ in reduzierter Form}\}$



$\mathcal{L}(\mathcal{M}) = \{w \in \{x, x^{-1}, y, y^{-1}\}^* \mid w \text{ in reduzierter Form und gerade}\}$

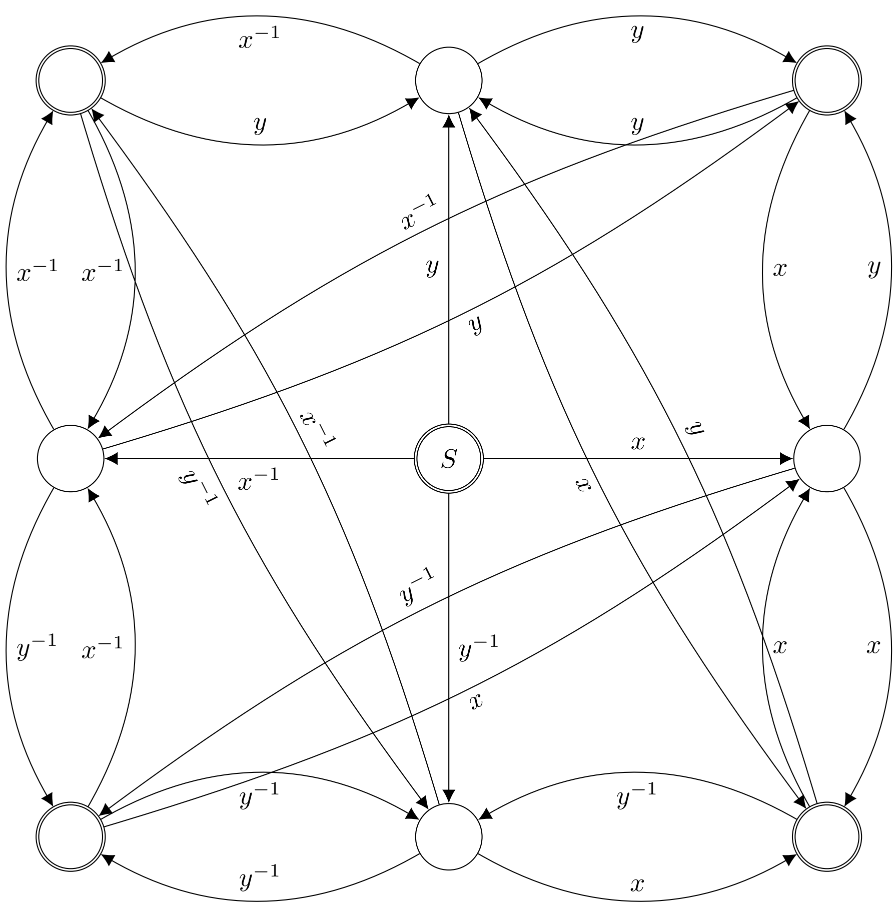


Fig. 7.6 (7.6.1 links, 7.6.2 rechts)

Beweisidee zu 7.14

$\mathcal{L}(\mathcal{M}) = \{ab^nac^mad^ka \mid n, m, k \in \mathbb{N}_0\}$

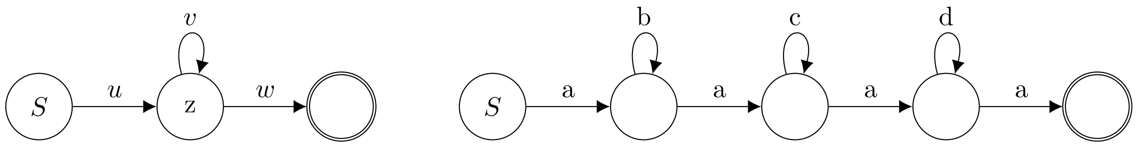
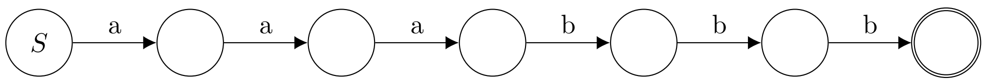


Fig. 7.7 (7.7.1 oben, 7.7.2 unten)

$\mathcal{L}(\mathcal{M}) = \{a^3b^3\}$



$\mathcal{L}(\mathcal{M}) = \{a^nb^n \mid 0 \leq n \leq 3\}$

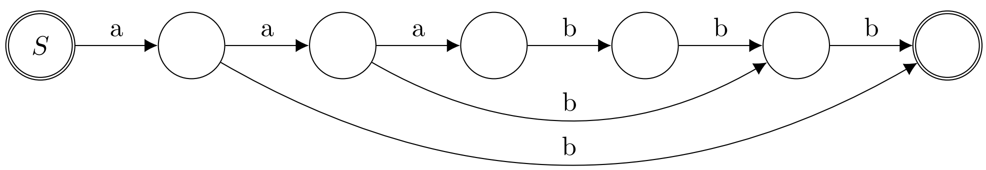
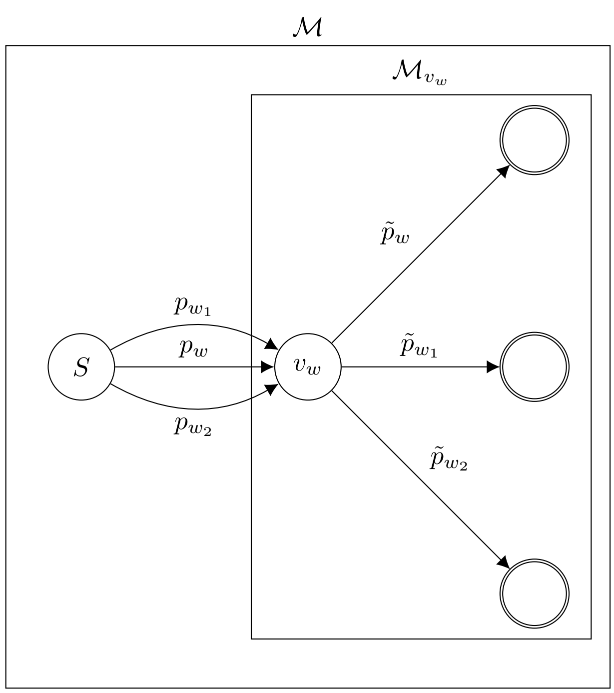
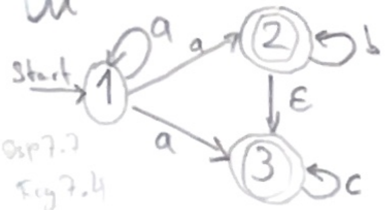


Fig. 7.8

Beweisidee zu 7.18



$\mathcal{M}$



$\mathcal{M} = (\underbrace{\{\{1\},\{2\},\{3\}\}}_{Z(\mathcal{M})}, \underbrace{\{a,b,c\}}_{\Sigma}, \delta, \underbrace{\{1\}}_{S(\mathcal{M})}, \underbrace{\{\{2\},\{3\}\}}_{\mathcal{A}(\mathcal{M})})$

FIN++

$\varepsilon(S) = \{\delta(s,\varepsilon) \mid s \in S\} \cup S$, $S_a = \{\delta(s,a) \mid s \in S\}$, $\delta_\varepsilon(S,a) = \varepsilon(S_a)$

↳ Entferne ε-Übergänge

$\mathcal{M}_\varepsilon = (\underbrace{\{V \subseteq Z(\mathcal{M}) \mid V = \varepsilon(V)\}}_{Z(\mathcal{M}_\varepsilon)}, \Sigma, \overset{\varepsilon(\{\delta(s,a) \mid s \in S\}) = \delta_\varepsilon(S,a)}{\delta_\varepsilon}, \underbrace{\{S \in Z(\mathcal{M}_\varepsilon) \mid S \cap S(\mathcal{M}) \neq \emptyset\}}_{S(\mathcal{M}_\varepsilon)}, \underbrace{\{A \in Z(\mathcal{M}_\varepsilon) \mid A \cap \mathcal{A}(\mathcal{M}) \neq \emptyset\}}_{\mathcal{A}(\mathcal{M}_\varepsilon)})$

Mache Deterministisch

↳ $\mathcal{D} = (\underbrace{\mathcal{P}(Z(\mathcal{M}_\varepsilon))}_{Z(\mathcal{D})}, \Sigma, \underbrace{\varepsilon(\bigcup_{v \in V} \delta(v,a))}_{\delta_\mathcal{D}(V,a)}, \underbrace{\varepsilon(S(\mathcal{M}_\varepsilon))}_{S(\mathcal{D})}, \underbrace{\{V \in Z(\mathcal{D}) \mid V \cap \mathcal{A}(\mathcal{M}) \neq \emptyset\}}_{\mathcal{A}(\mathcal{D})})$

Konkret

↳ $\mathcal{D} = (\{\emptyset, \{1\}, \cancel{\{2\}}, \{3\}, \cancel{\{1,2\}}, \{1,3\}, \{2,3\}, \{1,2,3\}\}, \{a,b,c\}, \delta_\mathcal{D}, \{1\}, \{\cancel{\{2\}}, \{3\}, \cancel{\{1,2\}}, \{1,3\}, \{2,3\}, \{1,2,3\}\})$ und "$|\delta_\mathcal{D}| \leq 2^{|Z(\mathcal{M})|} \cdot |\Sigma| = 2^3 \cdot 3 = 8 \cdot 3 = 24$"

↳ $\delta_\mathcal{D}(V,x)$: $\delta(\{1\},a) = \varepsilon(\{1,2,3\}) = \{1,2,3\}$, $\delta(\{1\},b) = \delta(\{1\},c) = \varepsilon(\emptyset) = \emptyset$

"$\mathcal{D}$ weglassen zum Zeitsparen"

$\delta(\{1,2,3\},a) = \varepsilon(\{1,2,3\} \cup \emptyset \cup \emptyset) = \{1,2,3\}$

$\delta(\{1,2,3\},b) = \varepsilon(\emptyset \cup \{2\} \cup \emptyset) = \underline{\varepsilon(\{2\}) = \{2,3\}}$, $\delta(\{2,3\},b) = \varepsilon(\{2\}) = \{2,3\}$

$\delta(\{1,2,3\},c) = \varepsilon(\emptyset \cup \emptyset \cup \{3\}) = \{3\}$, $\delta(\{3\},c) = \varepsilon(\{3\}) = \{3\}$

...

$\mathcal{L}(\mathcal{D}) = \{a^n b^m c^k \mid n \in \mathbb{N}, m,k \in \mathbb{N}_0\}$
$= \mathcal{L}(\mathcal{M})$