2.3 Abschlusseigenschaften

In diesem Abschnitt wollen wir uns mit *Abschlusseigenschaften* der regulären Sprachen, d.h. mit der Frage, ob, gegeben eine Operation $\circ$ und zwei reguläre Sprachen L_1, L_2, auch $L_1 \circ L_2$ regulär ist. $\circ$ könnte z.B. die Vereinigung sein oder als Operation auf nur eine Menge, die Operation, die an jedes Wort aus L_1 ein a anfügt.

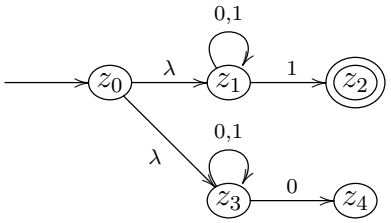
Abschlusseigenschaften erlauben oft Einblicke in Sprachfamilien und helfen auch oft beim Konstruieren von z.B. speziellen Automaten oder beim Beweis, dass es keinen Automat für eine Sprache geben kann.

2.3.1 Weitere Formalismen für REG

Bevor wir zu den Abschlusseigenschaften kommen, wollen wir vorher noch zwei weitere Formalismen einführen, die wie DFAs und NFAs gerade die Sprachfamilie der regulären Sprachen erfassen. Wir wollen diese Formalismen nur einführen und benutzen. Die Beweise, dass sie tatsächlich äquivalent zu DFAs sind werden wir nicht führen. Man findet diese Beweise bspw. in dem Buch von Hopcroft, Motwani und Ullman aus der Literaturliste.

NFAs mit λ-Kanten

Man kann einem NFA zusätzlich zu dem bisherigen noch erlauben, λ-Kanten (auch ϵ-Kanten genannt) zu benutzen. Diese Kanten sind also statt mit einem Symbol mit dem leeren Wort beschriftet. Die Bedeutung dahinter ist dann auch, dass man nichts vom Eingabeband liest (also quasi das leere Wort vom Eingabeband liest), der Lesekopf auf dem Eingabeband also nicht weiter bewegt wird. Ein Zustandswechsel findet allerdings statt.



In dem abgebildeten Beispiel kann der NFA zuerst ohne ein Symbol zu lesen mittels der λ-Kanten nach z_1 oder z_3 wechseln. Dort arbeitet er dann weiter. Auf dem Wort 11 gibt es dann beispielsweise vier verschiedene Rechnungen (in z_1 bleiben, in z_3 bleiben, einmal in z_1 eine 1 lesen und dann nach z_2 wechseln und mit der ersten 1 nach z_2 wechseln und dort blockieren). Da eine dieser Rechnungen $(z_0, 11) \vdash (z_1, 11) \vdash (z_1, 1) \vdash (z_2, \lambda)$ eine Erfolgsrechnung ist, akzeptiert der NFA. Man beachte, dass bei dem ersten Konfigurationsübergang $(z_0, 11) \vdash (z_1, 11)$ kein Symbol gelesen wird, aber ein Zustandsübergang stattfindet.

Das Beispiel ist zwar recht einfach gewählt, dennoch sollte man sich nicht täuschen lassen. Diese Kanten können äußerst nützlich sein. Beim Entwurf von Automaten, erleichtern sie einem wieder an vielen Stellen die Arbeit und bei Beweisen, helfen sie oft Argumente zu vereinfachen.

Man müsste nun noch Begriffe wie Überführungsfunktion, Rechnung, Konfigurationsübergang usw. anpassen. Wir wollen dies hier nicht tun und das Modell einfach informal benutzen. Wir nennen diesen Automaten dann einen NFA mit λ-Kanten oder auch kürzer einen λ-NFA.

Wir betonen noch den folgenden wichtigen Satz:

Satz 2.3.1. *Zu jedem λ-NFA A kann ein DFA B konstruiert werden mit $L(B) = L(A)$ und umgekehrt.*

Der Satz besagt, dass DFAs, NFAs und λ-NFAs äquivalent sind. Die Familie der regulären Sprachen wird also von jedem dieser drei Modelle erfasst. Auf einen Beweis dieses Satzes wollen wir hier verzichten. Man findet ihn z.B. in dem Buch von Hopcroft, Motwani und Ullman aus der Literaturliste.

Reguläre Ausdrücke

Wir wollen nun noch einen vierten Formalismus einführen, mit dem wir ebenfalls die Familie der regulären Sprachen erfassen. Dies sind die sogenannten regulären Ausdrücke.

Definition 2.3.2 (Reguläre Ausdrücke)**.** *Sei Σ ein Alphabet. Die **regulären Ausdrücke über** Σ sind induktiv definiert durch:*

1. *$\emptyset$ ist ein regulärer Ausdruck, der die Menge $M_\emptyset = \emptyset$ beschreibt.*
2. *Für jedes $a \in \Sigma$ ist a ein regulärer Ausdruck, der die Menge $M_a = \{a\}$ beschreibt.*
3. *Sind X und Y reguläre Ausdrücke, die die Mengen M_X und M_Y beschreiben, dann beschreibt*
   - *$(X + Y)$ die Menge $M_X \cup M_Y$*
   - *$(X \cdot Y)$ die Menge $M_X \cdot M_Y$*
   - *X^* die Menge M_X^* und*
   - *X^+ die Menge M_X^+.*
4. *Nur die so erzeugten (endlichen) Ausdrücke sind reguläre Ausdrücke.*

Ein paar Beispiele für reguläre Ausdrücke, wobei $\equiv$ hier für "beschreibt die Menge" steht:

- $(a + b)^* \equiv (\{a\} \cup \{b\})^* = \{a, b\}^*$

- $(a \cdot b) \equiv \{a\} \cdot \{b\} = \{ab\}$
- $a \cdot b^* + b \cdot a^* \equiv \{a\} \cdot \{b\}^* \cup \{b\} \cdot \{a\}^*$
- $\emptyset^* \equiv \emptyset^* = \{\lambda\}$
- Worte, die auf 1 enden: $(0 + 1)^* \cdot 1$
- Worte, die 00 oder 11 enthalten: $(0 + 1)^* \cdot (00 + 11) \cdot (0 + 1)^*$

Reguläre Ausdrücke und die Mengenoperationen sind also recht ähnlich. Mit regulären Ausdrücken spart man sich aber oft eine Menge Klammern. Daneben wollen wir auch die üblichen Regeln zur Klammerersparnis nutzen wie $\cdot$ vor $+$ und die hochgestellten $+$ und $*$ vor $\cdot$.

Aufgrund der Ähnlichkeit, wollen wir dann reguläre Ausdrücke gleich wie Mengen nutzen und z.B. $aaa \in a^*$ erlauben. Gemeint ist dann mit dem regulären Ausdruck in diesem Kontext gerade die Menge, die er beschreibt.

Man kann nun wieder zeigen, dass es zu jedem regulären Ausdruck R einen DFA A gibt, der gerade die von R beschriebene Menge akzeptiert und dass es zu jedem DFA A eine regulären Ausdruck gibt, der gerade $L(A)$ beschreibt. Auch hier wollen wir reguläre Ausdrücke der Definition entsprechend benutzen, den eben angesprochenen Beweis jedoch nicht führen. Er findet sich wieder in der Literatur.

Insgesamt haben wir damit folgenden Satz:

Satz 2.3.3. *Zu jeder regulären Sprache L gibt es*

- *einen DFA A mit $L(A) = L$*
- *einen NFA B mit $L(B) = L$*
- *einen NFA mit λ-Kanten C mit $L(C) = L$*
- *einen regulären Ausdruck D, der L beschreibt ($M_D = L$)*

DFAs, NFAs, NFAs mit λ-Kanten und reguläre Ausdrücke sind also äquivalent.

2.3.2 Abschlusseigenschaften

Wir wollen uns nun damit beschäftigen, ob die regulären Sprachen gegenüber bestimmten Operationen abgeschlossen sind. Betrachten wir als Operation z.B. die Vereinigung, dann interessiert uns die Frage, ob $L_1 \cup L_2 \in REG$ ist, wenn $L_1, L_2 \in REG$ gilt. Gilt dies stets, so sagt man die Sprachfamilie der regulären Sprachen ist gegenüber Vereinigung *abgeschlossen*. Die Bezeichnung rührt daher, dass man mittels dieser Operation dann quasi nicht aus der Familie der regulären Sprachen herauskommt.

Um den Abschluss genauer zu formulieren, definieren wir:

Definition 2.3.4. *Sei f_1 eine einstellige Operation auf Mengen und f_2 eine zweistellige Operationen. D.h. wenn M_1, M_2 zwei Mengen sind, dann sind auch $f_1(M_1)$ und $f_2(M_1, M_2)$ Mengen.*

Eine Sprachfamilie $\mathcal{C}$ ist ***abgeschlossen*** *gegenüber der Operation f_1 bzw. f_2, wenn für jedes $R \in \mathcal{C}$ auch $f_1(R) \in \mathcal{C}$ gilt bzw. wenn für $R_1, R_2 \in \mathcal{C}$ auch $f_2(R_1, R_2) \in \mathcal{C}$ gilt.*

Man kann dies auch allgemeiner für Mengen und Operationen auf diesen Mengen einführen. Die Menge der natürlichen Zahlen ist dann z.B. gegenüber der Addition abgeschlossen, da die Summe zweier natürlicher Zahlen wieder eine natürliche Zahl ist. Die natürlichen Zahlen sind hingegen nicht gegenüber der Subtraktion abgeschlossen, da es zwar natürliche Zahlen gibt, die subtrahiert voneinander wieder eine natürliche Zahl ergeben, aber eben auch welche, bei denen das nicht passiert (wobei es hier primär auf die Reihenfolge ankommt, aber die dürfte man sich ja auch nicht aussuchen, sonst hätte man eine andere Operation).

Einfache Abschlusseigenschaften

Wir wollen nun für einige Operationen zeigen, dass die regulären Sprachen gegenüber ihnen abgeschlossen sind. Die ersten vier Operationen kriegen wir fast geschenkt.

Satz 2.3.5. *Die regulären Sprachen sind gegenüber $\cup, \cdot, +, *$ abgeschlossen.*

Beweis. Seien $L_1, L_2 \in \text{REG}$. Dann gibt es reguläre Ausdrücke A_1, A_2, die L_1 bzw. L_2 beschreiben. Nach Definition der regulären Ausdrücke sind nun auch $A_1 + A_2$, $A_1 \cdot A_2$, A_1^+, A_1^* reguläre Ausdrücke. Diese beschreiben gerade die Mengen $L_1 \cup L_2$, $L_1 \cdot L_2$, L_1^+ und L_1^*, die aufgrund des Satzes zur Äquivalenz regulärer Ausdrücke und DFAs wieder regulär sind. □

Wir haben für diese vier Operationen also gar nicht viel zu tun gehabt. Wir müssen hier lediglich beachten, dass jede durch einen regulären Ausdruck beschriebene Menge eine reguläre Sprache ist. Dann benutzen wir noch die Operationen, die wir bei regulären Ausdrücken syntaktisch haben und sind schon fertig.

Komplementbildung

Schwieriger ist da schon zu zeigen, dass die regulären Sprachen gegenüber Komplementbildung abgeschlossen sind.

Definition 2.3.6. *Sei $L \subseteq \Sigma^*$. Dann ist*

$$\overline{L} = \{w \in \Sigma^* \mid w \notin L\}$$

das Komplement von L.

In $\overline{L}$ sind also gerade all jene Worte aus Σ^*, die *nicht* in L sind. Die Worte, die in L sind, sind gerade nicht in $\overline{L}$.

Sind die regulären Sprachen nun gegenüber Komplementbildung abgeschlossen? Wir können zumindest nicht so vorgehen wie bei dem vorherigen Satz. Dort nutzten wir reguläre Ausdrücke. Das ist hier nicht so leicht möglich, da sich in der Definition der regulären Ausdrücke keine Operation findet, mit der wir Komplement ausdrücken könnten.

Pause to Ponder: Geht das also? Sind die regulären Sprachen gegenüber Komplementbildung abgeschlossen? Man versuche das statt mit regulären Ausdrücken mit den Automaten zu zeigen, d.h. man geht von einem Automaten A für eine reguläre Sprache L aus und konstruiert daraus einen Automaten A' für $\overline{L}$. Aber wie?

Mit regulären Ausdrücken kommen wir tatsächlich nicht gut weiter. Wohl aber mit Automaten! Haben wir einen Automaten A für eine Sprache L, dann wollen wir jetzt also erreichen, dass ein Wort, das A akzeptiert hat, nicht mehr akzeptiert wird und ein Wort, das A nicht akzeptiert hat, nun akzeptiert wird. Dies erreichen wir gerade, indem wir Zustände, die keine Endzustände sind, zu solchen machen und umgekehrt. Endeten wir dann vorher in einem normalen Zustand (und akzeptierten also nicht), so enden wir jetzt in einem Endzustand (und akzeptieren also), endeten wir vorher in einem Endzustand (und akzeptierten also), so enden wir nun in einem normalen Zustand (und akzeptieren also nicht). Auf eine Kleinigkeit müssen wir noch achten. Damit dies funktioniert, muss A auch tatsächlich jedes Wort lesen können. Kann er ein Wort nämlich nicht zu Ende lesen, so wird das Wort nicht akzeptiert. Unser neuer Automat kann das Wort dann aber weiterhin nicht zu Ende lesen und akzeptiert also auch nicht. Wir müssen also dafür sorgen, dass A jedes Wort lesen kann. Dies erreichen wir ganz einfach dadurch, dass wir gleich zu Beginn von einem *vollständigen* DFA A für L ausgehen, der ja ebenfalls stets existiert. Wir fassen den Beweis noch einmal zusammen.

Satz 2.3.7. *Die regulären Sprachen sind gegenüber Komplementbildung abgeschlossen, d.h. ist $L \in REG$, dann auch $\overline{L} \in REG$.*

Beweis. Sei $L \in$ REG. Dann gibt es einen vollständigen DFA A mit $L(A) = L$. Wir konstruieren nun einen vollständigen DFA A' aus A wie folgt:

- Zunächst übernehmen wir alles von A.
- Ist dann $z \in Z$ in A Endzustand, dann ist z in A' kein Endzustand.
- Ist $z \in Z$ in A kein Endzustand, dann ist z in A' ein Endzustand.

Wir tauschen also lediglich End- und Nicht-Endzustände.

Wurde ein Wort w nun von A akzeptiert, dann gab es eine Rechnung $(z_0, w) \vdash (z_e, \lambda)$, wobei z_e ein Endzustand von A ist. Diese Rechnung passiert nun genauso

auch in A' nur ist dort z_e kein Endzustand und folglich wird w von A' nicht akzeptiert.

Wurde andererseits ein Wort von A nicht akzeptiert, dann gab es eine Rechnung $(z_0, w) \vdash (z, \lambda)$, wobei z kein Endzustand von A ist. Man beachte an dieser Stelle, dass A vollständig ist. Der Fall, dass ein Wort nicht akzeptiert wird, weil es nicht zu Ende gelesen wird, tritt also nicht auf. Die obige Rechnung passiert nun wieder genauso in A', nur ist dort z ein Endzustand und folglich wird w von A' akzeptiert.

Damit gilt gerade $L(A') = \overline{L}$. □

Man beachte, dass es uns völlig frei steht, mit welcher Darstellung für die reguläre Sprache L wir starten. Wichtig ist nur, dass diese Darstellung tatsächlich L beschreiben kann, also entweder ein DFA ist oder ein NFA, ein λ-NFA, ein regulärer Ausdruck oder eben ein DFA, der zudem noch vollständig ist. Da all diese Formalismen die regulären Sprachen erfassen, gibt es zu jeder regulären L Sprache auch z.B. einen DFA mit $L(A) = L$ oder einen regulären Ausdruck R mit $M_R = L$.

Wir können hier also frei wählen, womit wir starten wollen und können dann diese Darstellung manipulieren, also z.B. bei einem Automaten Zustände oder Kanten hinzufügen, Kanten umbiegen oder eben Endzustände manipulieren. Wichtig ist dann, dass bei diesen Manipulationen etwas entsteht, von dem wir dann zeigen können, dass wir damit unser Ziel erreichen. Oben musste also ein Automat entstehen, der gerade $\overline{L}$ akzeptiert.

Durchschnittsbildung

Als nächstes wenden wir uns dem Durchschnitt oder dem Produkt zweier Sprachen zu, d.h. der Operation $L_1 \cap L_2$.

Pause to Ponder: Sind L_1 und L_2 regulär, ist dann auch $L_1 \cap L_2$ regulär?

Dies lässt sich mit unserem bisherigen Wissen sehr schnell zeigen. Wenn man bedenkt, dass $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ nach dem Gesetz von de Morgan gilt, dann sind wir sofort fertig, da wir ja schon wissen, dass die regulären Sprachen gegenüber Komplement und Vereinigung abgeschlossen sind. (Sind also L_1 und L_2 regulär, dann ist auch $\overline{L_1}$ und $\overline{L_2}$ und damit auch $\overline{L_1} \cup \overline{L_2}$ und damit letztendlich $\overline{\overline{L_1} \cup \overline{L_2}}$ regulär, was gerade $L_1 \cap L_2$ ist.)

Es gibt aber eine sehr schöne Methode, wie man dies direkt mit Automaten machen kann. Wir wollen dies hier einmal tun und dabei den Produktautomaten zu zwei Automaten A_1 und A_2 kennenlernen. Dieser akzeptiert gerade dann ein Wort, wenn A_1 und A_2 dieses Wort beide akzeptieren. Damit akzeptiert der Produktautomat also gerade den Schnitt von $L(A_1)$ und $L(A_2)$.

Pause to Ponder: Wie lässt sich aus einem endlichen Automaten A_1 für L_1 und einem endlichen Automaten A_2 für L_2 ein Automat C für $L_1 \cap L_2$ konstruieren?

Erinnern wir uns an die Potenzautomatenkonstruktion. Hier haben wir uns im Zustand des konstruierten DFAs gemerkt, in welchen Zuständen der ursprüngliche NFA nichtdeterministisch sein kann. Hier können wir ähnlich vorgehen. Allerdings wollen wir uns genau merken, welcher Zustand zum ersten und welcher Zustand zum zweiten Automaten gehört. Wir arbeiten daher mit Tupeln (z_1, z_2) als neue Zustände von C, wobei im ersten Tupelelement ein Zustand von A_1 und im zweiten Tupelelement ein Zustand von A_2 notiert wird. Ein Übergang mit Symbol a geschieht dann genau dann, wenn im ersten Tupelelement (also im Automaten A_1) ein Übergang mit a möglich ist und ebenso im zweiten Tupelelement (also im Automaten A_2). Wenn C dann ein Wort w liest, dann macht er im ersten Tupelelement seiner Zustände quasi eine Rechnung von A_1 und im zweiten eine von A_2. Akzeptieren soll er daher genau dann, wenn im ersten Tupelelement ein Endzustand von A_1 und im zweiten Tupelelement ein Endzustand von A_2 ist. So ist sichergestellt, dass beide Automaten gleichzeitig in einem Endzustand sind, also beide das Wort w akzeptieren. Dies ist gerade, was wir für C erreichen wollen. C soll akzeptieren, wenn A_1 *und* A_2 akzeptieren und sonst ablehnen. Wir formalisieren die Konstruktion jetzt genauer.

Satz 2.3.8. *Seien $L_1, L_2 \in REG$, dann ist auch $L_1 \cap L_2 \in REG$.*

Beweis. Seien $A_1 = (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,end})$ und $A_2 = (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,end})$ vollständige DFAs mit $L(A_1) = L_1$ und $L(A_2) = L_2$. Wir konstruieren $C = (Z_3, \Sigma_3, \delta_3, z_{3,0}, Z_{3,end})$ mit

$$\begin{aligned} Z_3 &:= Z_1 \times Z_2 \\ \Sigma_3 &:= \Sigma_1 \cap \Sigma_2 \\ z_{3,0} &:= (z_{1,0}, z_{2,0}) \\ Z_{3,end} &:= Z_{1,end} \times Z_{2,end} \\ \delta_3((z_1, z_2), x) &:= (\delta_1(z_1, x), \delta(z_2, x)) \end{aligned}$$

Da A_1 und A_2 vollständig sind, ist jeder Zustandsübergang in C definiert.

Hat man nun eine Rechnung

$$(z_{1,0}, z_{2,0}, w_1w_2w_3\ldots) \vdash ((z_1, z_2), w_2w_3\ldots) \vdash ((z_1', z_2'), w_3\ldots) \vdash \ldots$$

von C, so ist $(z_{1,0}, w_1w_2w_3\ldots) \vdash (z_1, w_2w_3\ldots) \vdash (z_1', w_3\ldots) \vdash \ldots$ eine Rechnung von A_1 und ebenso $(z_{2,0}, w_1w_2w_3\ldots) \vdash (z_2, w_2w_3\ldots) \vdash (z_2', w_3\ldots) \vdash \ldots$ eine von A_2. Zumindest wollen wir dies mit der Konstruktion erreichen. Wir formulieren dies daher als eine Behauptung. Man beachte, dass mit A_1 und A_2 auch C vollständig ist, daher kann jedes Wort gelesen werden. Wir behaupten nun, dass für jedes Wort $w \in \Sigma_3^*$ gilt:

$$\hat{\delta_3}(z_{3,0}, w) = (z_1, z_2) \Leftrightarrow \hat{\delta_1}(z_{1,0}, w) = z_1 \wedge \hat{\delta_2}(z_{2,0}, w) = z_2$$

Wir behaupten also, wenn C nach Lesen von w in (z_1, z_2) ist, dann ist A_1 nach Lesen von w in z_1 und A_2 nach Lesen von w in z_2 und wenn umgekehrt A_1 nach Lesen von w in z_1 und A_2 nach Lesen von w in z_2 ist, dann ist C nach Lesen von w in (z_1, z_2). Dies ist genau das, was wir uns als Ziel unserer Konstruktion überlegt hatten und dies kann man recht schnell mittels Induktion über die Wortlänge beweisen.

Im Anschluss folgt dann schnell die Korrektheit der Konstruktion, weil nach Definition der Endzustandsmenge von C dann aus obigem

$$\hat{\delta_3}(z_{3,0}, w) \in Z_{3,end} \Leftrightarrow \hat{\delta_1}(z_{1,0}, w) \in Z_{1,end} \wedge \hat{\delta_2}(z_{2,0}, w) \in Z_{2,end}$$

folgt, womit gezeigt ist, dass C ein Wort w genau dann akzeptiert, wenn w von A_1 und von A_2 akzeptiert wird, d.h. es ist $L(C) = L(A_1) \cap L(A_2)$.

Den Induktionsbeweis oben wollen wir zur Übung überlassen. Man kann sich hierfür gut an den Induktionsbeweisen, die wir bisher hatten, orientieren. □

Noch eine Anmerkung zur Konstruktion: Wenn man den Produktautomaten konstruieren will (oder einen Algorithmus dafür implementieren will), dann verfährt man wieder so wie beim Potenzautomaten und beginnt nicht sofort mit $Z_3 = Z_1 \times Z_2$, sondern beginnt mit dem Startzustand $z_{3,0} = (z_{1,0}, z_{2,0})$ und konstruiert dann sukzessive die initiale Zusammenhangskomponente.

Spiegelworte

Zuletzt wollen wir einen Blick auf Spiegelworte werfen. Spiegelworte sind Worte, die sich durch rückwärts Lesen eines ursprünglichen Wortes ergeben. Ist z.B. $w = 011$, dann ist das Spiegelwort $w^{rev} = 110$. Wir definieren dies formal mittels einer induktiven Definition.

Definition 2.3.9 (w^{rev})**.** *Das* ***Spiegelwort*** w^{rev} *zu einem Wort* $w \in \Sigma^*$ *ist definiert durch*

1. $\lambda^{rev} = \lambda$ *und*
2. $(ux)^{rev} = x \cdot u^{rev}$ *mit* $x \in \Sigma$ *und* $u \in \Sigma^*$

Bspw. ist für $w = 1011$ *dann* $w^{rev} = 1101$.
Für Mengen notieren wir $L^{rev} = \{w^{rev} \mid w \in L\}$.

Intuitiv ist die Bedeutung von w^{rev} vermutlich schnell klar. Die Definition steht hier insb. aufgrund ihres Charakters: Zuerst wird die Bedeutung von *rev* für das leere Wort definiert. Im Anschluss wird die Bedeutung für längere Worte mittels der Bedeutung von *rev* für kürzere Worte definiert. Genau auf diese Art und Weise gehen wir auch immer bei induktiven Beweisen über die Wortlänge vor. Zunächst haben wir den Fall des leeren Wortes, dann den eines längeren Wortes, wobei wir die Induktionsannahme für kürzere Worte benutzen.

Die Induktionsannahme entspricht hier der Definition für kürzerer Worte (also oben beim zweiten Punkte dem u).

Wir können nun zeigen, dass die regulären Sprachen gegenüber der *rev*-Operation abgeschlossen sind.

Satz 2.3.10. *Ist L regulär, dann ist auch L^{rev} regulär.*

Beweisidee Wir wollen den Beweis hier nur skizzieren. Wir gehen wieder von einem vollständigen DFA A für L aus. Die Worte aus L^{rev} sind gerade die Worte, die in einem Endzustand beginnen und in einem Startzustand enden, wenn man die Kanten rückwärts entlang geht. Wir konstruieren daher aus A einen NFA B, indem wir alle Kanten in A umdrehen, den Start- zu einem Endzustand machen und alle Endzustände zu Startzuständen.

Um noch zu zeigen, dass $L(B) = L(A)^{rev}$ gilt, kann man einfach akzeptierende Rechnungen des einen Automaten nehmen und zeigen, dass der jeweils andere Automat eine Erfolgsrechnung mit dem gleichen, aber umgedrehten, Weg im Zustandsdiagramm hat. □

Mit diesem Resultat können wir dann endlich auch schlussfolgern, dass die komplizierte Sprache

$$Sum = \{w \in \Sigma^* \mid \text{ die unterste Zeile ist Summe der oberen Zeilen}\}$$

tatsächlich regulär ist (zur Erinnerung: die Symbole in Σ waren im Grunde Vektoren der Länge 3 also z.B. $(0, 0, 1)$ als Spalte geschrieben). Bisher hatten wir ja nur gezeigt, dass Sum^{rev} regulär ist.

Wir haben in diesem Abschnitt λ-NFAs kennengelernt, also NFAs, die zusätzlich λ-Kanten benutzen können. Bei λ-Kanten wird kein Buchstabe gelesen. Es findet aber ein Zustandswechsel statt.

Ferner haben wir reguläre Ausdrücke kennengelernt, also Ausdrücke der Art $(a + b)^* \cdot a^+$.

Von beiden Formalismen nehmen wir mit, dass sie äquivalent zu DFAs sind. Wir kennen damit nun vier Formalismen, die die Familie der regulären Sprachen erfassen: DFAs, NFAs, λ-NFAs und reguläre Ausdrücke.

Im Anschluss haben wir uns dann mit Abschlusseigenschaften beschäftigt. Hierbei geht es darum für zwei Sprachen L_1, L_2 einer Sprachfamilie und eine Operation $\circ$ zu zeigen, dass auch $L_1 \circ L_2$ in dieser Sprachfamilie ist (oder auch ein Gegenbeispiel anzugeben, um dies zu widerlegen).

Wir haben gesehen, dass die regulären Sprachen gegenüber $\cup, \cdot, +, *$, Komplementbildung, Durchschnitt und *rev* abgeschlossen sind. Bei den ersten vier haben wir in den Beweisen mit regulären Ausdrücken argumentiert, bei den letzten drei mit (vollständigen) DFAs. Hier hat man also stets eine Wahlfreiheit und kann den Formalismus nehmen, der einem am geeignetsten erscheint. Dies ist auch einer der Gründe, warum unterschiedliche Formalismen eingeführt werden und selbst nachdem man gezeigt hat, dass sie äquivalent zu schon

bestehenden sind, weiterhin benutzt werden. Oft beleuchten unterschiedliche Formalismen Dinge nämlich von unterschiedlichen Blickwinkeln.

Fragen

1. Sei $R = 0^* \cdot 1^+ \cdot (0+1)^*$. In welchen Fällen ist $w_1 \in M_R$ und $w_2 \notin M_R$?
   a) $w_1 = 001$ und $w_2 = 100$
   b) $w_1 = 100$ und $w_2 = 0011$
   c) $w_1 = 0101$ und $w_2 = 1010$
   d) $w_1 = 101$ und $w_2 = 000$

2. Welcher reguläre Ausdruck beschreibt die Menge $\overline{\{a\}^*}$?
   a) $(a+b)^* \cdot b \cdot (a+b)^*$
   b) b^*
   c) $(a+b)^* \setminus a^*$
   d) $(a+b)^* \cdot b^* \cdot (a+b)^*$

3. Sind die regulären Sprachen gegenüber Vereinigung abgeschlossen?
   a) Ja
   b) Nein

4. Welcher reguläre Ausdruck beschreibt die Menge jener Worte, bei denen auf eine 0 stets eine 1 folgt?
   a) $(00 + 01 + 11)^*$
   b) $(1 + 01)^*$
   c) $(0+1)^* \setminus 0^*$
   d) $(1 + 01)^* \cdot 0^*$
   e) $(1 + 01)^* \cdot (0 + \emptyset^*)$
   f) $(1 + 01)^* \cdot \emptyset^*$