

Konfigurationsübergang, Rechnung, Erfolgsrechnung, akzeptierte Sprache, reguläre Menge und die Sprachfamilie REG. Alle diese Begriffe sind wichtig und sollten bekannt sein, da wir auf ihrer Grundlage aufbauen und einem sonst im weiteren Verlauf wichtige Vokabeln fehlen. Daher diese Begriffe ruhig noch einmal angucken.

Fragen

1. Wie viele Startzustände kann ein endlicher Automat haben?
 - a) beliebig viele
 - b) zwischen 0 und $|Z|$
 - c) nur genau einen
 - d) hängt von δ ab
2. Wie viele Endzustände kann ein endlicher Automat haben?
 - a) beliebig viele
 - b) zwischen 0 und $|Z|$
 - c) nur genau einen
 - d) hängt von δ ab
3. Wann akzeptiert ein endlicher Automat ein Eingabewort?
 - a) Wenn er ein Wort zu Ende gelesen hat.
 - b) Wenn er einen Endzustand erreicht.
 - c) Wenn er einen Endzustand erreicht und dort blockiert oder das Wort zu Ende gelesen ist.
 - d) Wenn er einen Endzustand erreicht und das Wort zu Ende gelesen ist.
4. Sei $Z = \{z_0, z_1\}$, z_0 der Startzustand, z_1 der einzige Endzustand. Ferner sei $\delta(z_0, 0) = z_0$, $\delta(z_0, 1) = z_1$ und $\delta(z_1, 0) = z_1$. Skizzieren Sie das Zustandsübergangsdiagramm dieses DFA A . Welche Sprache akzeptiert er?
 - a) $L(A) = \{1\}$
 - b) $L(A) = \{0\}^* \cdot \{1\}$
 - c) $L(A) = \{0\}^* \cdot \{1\} \cdot \{0\}^*$
 - d) $L(A) = \{0\}^* \cdot \{1\} \cdot \{0, 1\}^*$

2.1.3 Sprachen akzeptieren mit dem DFA

Üblicherweise wollen wir nun zu einer Sprache M , die uns interessiert, einen DFA A konstruieren, der diese Sprache akzeptiert, d.h. einen mit $L(A) = M$. Eine Sprache, die uns interessieren könnte, könnte z.B. die Menge aller Wörter

sein, die mit a beginnen oder die Menge aller Wörter, die das Wort aa als Teilwort enthalten.

Die Sprachen erscheinen vielleicht etwas künstlich. Die benutzten Symbole wie a , b usw. dienen aber üblicherweise nur als Abstraktion für Aktionen wie “Schalter betätigen” oder “Speicher löschen”.

Außerdem lassen sich kompliziertere Sprachen formulieren. Stellt man sich den Automaten nicht mehr so vor, dass er ein Wort von einem Band liest, sondern dass er Aktionen ausführt (jedes Symbol des Eingabealphabets steht also für eine Aktion), so kann man z.B. versuchen die Anforderung “zwischen zwei Aktionen a soll immer genau einmal die Aktion b passieren” als Sprache zu formulieren und dann versuchen einen Automaten dazu zu konstruieren. Bei solchen Formulierungen ist einfacher einzusehen, dass sie als Anforderungen in Anwendungen auftreten. Sie sind allerdings auch schwieriger und treten daher in einführenden Beispielen noch nicht auf.

Der zur Menge M konstruierte Automat A soll genau die Wörter in M akzeptieren (nicht mehr, nicht weniger), denn sonst gilt nicht $L(A) = M$. Dies ist aber wichtig, denn hat man nur $L(A) \subsetneq M$, dann akzeptiert der Automat nicht jedes Wort aus M und ist daher fehlerhaft. Hat man nur $M \subsetneq L(A)$, dann akzeptiert der Automat weitere Worte, die er eigentlich nicht akzeptieren soll, was ebenfalls fehlerhaft ist.

Ein verbreitetes Problem zu Anfang ist, dass man das Gefühl hat, eine der Richtungen ($L(A) \subseteq M$ oder $M \subseteq L(A)$) würde genügen. Meist stört man sich dann nicht daran, dass der Automat zu viel oder zu wenig kann. Nehmen wir einmal das Beispiel der Menge M aller Worte, die mit a beginnen. Wir wollen also einen Automaten A , der ein Wort w genau dann akzeptiert, wenn es mit a beginnt. Ansonsten soll es abgelehnt werden. Hat man jetzt nur $L(A) \subseteq M$ und gilt tatsächlich nicht $L(A) = M$ (d.h. es ist $L(A) \subsetneq M$), dann gibt es Wörter, die mit a beginnen, die der Automat nicht akzeptiert. Wir wollten ja aber gerade, dass er alle Worte akzeptiert, die mit a beginnen. Mit einem solchen Automaten können wir also nicht zufrieden sein. Haben wir andersherum nur $M \subsetneq L(A)$ erreicht, so akzeptiert der Automat neben den Worten, die mit a beginnen, noch weitere, womit wir ebenfalls nicht zufrieden sein können.

Man stelle sich vor, in der Sprache M sind die Möglichkeit Münzen korrekt in einen Automaten zu werfen (z.B. für eine Fahrkarte für den Bus; nehmen wir an die Karte kostet 50 Cent und es gibt 10 und 20 Cent Münzen (symbolisiert durch a und b). Dann ist z.B. die Folge abb (eine 10er, zwei 20er Münzen) ein korrekter Münzeinwurf, die Folge bbb hingegen nicht, wobei wir die Möglichkeit, Wechselgeld zurückzugeben, nicht beachten wollen). $L(A) \subsetneq M$ bedeutet dann, dass es Möglichkeiten gibt, Münzen korrekt einzuwerfen, die der Automat nicht akzeptiert (das Geld behält er aber vielleicht trotzdem ein, in jedem Fall wird der Kunde genervt sein). $M \subsetneq L(A)$ bedeutet, dass der Automat noch Folgen akzeptiert, die nicht vorgesehen sind. Dies könnte für den Anbie-

ter schlecht sein, wenn der Kunde die Fahrkarte z.B. auch kriegt, wenn er noch gar nicht genug Geld eingeworfen hat.

In diesem Abschnitt wollen wir uns zwei Konstruktionsmethoden für DFAs ansehen und uns mit den Beweisen von $L(A) = M$ beschäftigen, denn auch wenn man einen Automaten A konstruiert hat, dann ist $L(A) = M$ zunächst nur eine Behauptung oder Vermutung. Diese ist dann noch zu beweisen.

Der hier aus didaktischen Gründen auftretende Fall, zu einem gegebenen Automaten A die Menge M mit $L(A) = M$ zu ermitteln (und dies zu beweisen), tritt in Anwendungen kaum auf. Dies wäre ungefähr so als müsste man zu unkommentierten Code, dessen Bedeutung ermitteln. Zum Erlernen der Funktionsweise von Automaten ist dies aber ein ganz praktisches Vorgehen, daher tritt dies hier und in der Literatur meist ein paar Mal auf.

In den Beispielen dieses Abschnitts wird man an mehreren Stellen vielleicht denken, dass man die Lösung doch ohnehin schon sieht oder dass bei beiden Richtungen des Beweises von $L(A) = M$ das gleiche passiert.

Dass man die Lösung schon sieht, mag an einigen Stellen stimmen. Die einfachen Automaten und Sprachen dienen ja aber gerade zur Übung, sie sollen das Vorgehen eintrainieren, dass man dann bei komplizierteren Automaten oder Sprachen oder auch bei fortgeschritteneren Automatenmodellen anwenden kann.

Der Eindruck, dass bei beiden Richtungen das gleiche passiert, entsteht zu Anfang schnell, ist aber falsch. Bei der einen Richtung zeigen wir $L(A) \subseteq M$, also dass jedes Wort, das der Automat A akzeptiert, auch in M ist. Um dies zu zeigen gehen wir von einem Wort $w \in L(A)$ aus und zeigen dann, dass auch $w \in M$ gilt. Wir nutzen also nur die Eigenschaften, die ein Wort hat, das von dem Automaten A akzeptiert wird und versuchen daraus abzuleiten, dass dieses Wort auch in M ist.

Bei der anderen Richtung zeigen wir $M \subseteq L(A)$, also dass jedes Wort, das in M ist, auch von A akzeptiert wird. Um dies zu zeigen gehen wir nun von einem Wort $w \in M$ aus und zeigen dann, dass auch $w \in L(A)$ gilt. Wir nutzen also nur die Eigenschaften, die ein Wort hat, das in M ist und versuchen daraus abzuleiten, dass dieses Wort auch in $L(A)$ ist.

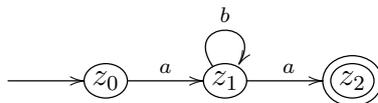
Der fälschliche Eindruck, dass hier das gleiche passieren würde entsteht bei den Beispielen zu Anfang dadurch, dass die Eigenschaften, die ein Wort hat, das von A akzeptiert wird und die Eigenschaft die ein Wort hat, das in M ist, meist sehr ähnlich sind und daher die Argumentation auch sehr ähnlich aussieht. Man mache sich aber in den folgenden Beispielen und auch bei der eigenen Arbeit aber klar, dass tatsächlich beim Beweis von z.B. $L(A) \subseteq M$ von einem Wort $w \in L(A)$ ausgegangen wird und dann mittels Eigenschaften die ein solches Wort hat (wobei sich diese Eigenschaften aus dem speziellen Automaten A und der Art wie Automaten akzeptieren ergeben) argumentiert wird, um letztendlich zu zeigen, dass das Wort auch in M ist. Bei späteren,

komplizierteren Sprachen und Automaten wird es dann wichtig sein, diese klare Trennung erlernt zu haben.

(In einigen Fällen kann man auch gut mit “genau dann, wenn” Beziehungen zeigen, dass $w \in L(A)$ gilt genau dann, wenn $w \in M$ gilt und ist dann sofort fertig. Wir machen dies hier aber höchstens in Ausnahmefällen, um die obige klare Trennung zu betonen.)

Ein erstes Beispiel

Betrachten wir als erstes Beispiel den abgebildeten Automaten.



Nachdem man den Automaten etwas untersucht hat, kommt man nach einigem Nachdenken und Ausprobieren einzelner Worte zu der Vermutung, dass die akzeptierte Sprache gerade $\{a\} \cdot \{b\}^* \cdot \{a\}$ sein könnte. Zur Erleichterung der Notation setzen wir $M := \{a\} \cdot \{b\}^* \cdot \{a\}$ und nennen den Automaten A . Unsere Vermutung oder Behauptung ist nun also, dass $L(A) = M$ gilt. Auch wenn man sich bei diesem kleinen Beispiel vielleicht sehr sicher ist, dass diese Vermutung stimmt, so ist sie dennoch zu beweisen (und insb. für spätere, kompliziertere Fälle wird es wichtig sein, das Vorgehen zu beherrschen). Wir teilen den Beweis von $L(A) = M$ in zwei Teile auf. Die beiden nachfolgenden Teilbeweise sind absichtlich sehr ausführlich. Im Anschluss finden sich beide Richtungen mit kürzeren Argumentationen bewiesen. Wir wollen das hier aber zunächst ganz ausführlich machen, damit man sehen kann, was im einzelnen argumentiert werden muss.

1. Zuerst wollen wir $L(A) \subseteq M$ beweisen. Wir wollen also zeigen, dass jedes Wort, das der Automat A akzeptiert, auch in M enthalten ist. Betrachten wir also ein beliebiges Wort w , das von A akzeptiert wird, also ein $w \in L(A)$. Wir wissen, dass der Automat A in seinem Startzustand z_0 beginnen muss. Wir wissen ferner, dass w akzeptiert wird, also wissen wir, dass A das Wort w zu Ende lesen kann und dann in einem Endzustand ist (nur dann akzeptiert A ein Wort). Da es nur einen Endzustand gibt, wissen wir, dass A das Wort w lesen kann und er dann in z_2 ist. Damit wissen wir schon, dass wir von der Konfiguration (z_0, w) irgendwie (d.h. mittels möglicher Übergänge in A) in die Konfiguration (z_2, λ) gelangen. Betrachten wir den Automaten weiter, so sehen wir, dass aus z_0 nur eine a -Kante heraus führt und nach z_2 nur eine a -Kante hinein und dass in z_2 nicht mehr gelesen werden kann. Die a -Kante aus z_0 heraus führt zudem nach z_1 und die a -Kante nach z_2 kommt von z_1 . Insgesamt wissen wir damit, dass w mit einem a beginnen und mit einem a enden muss und wir wissen dass dies nicht das gleiche Symbol in w sein kann

(w also aus mindestens zwei a besteht). Damit ist w von der Form $a \cdot v \cdot a$ mit $v \in \{a, b\}^*$. Wir wissen damit bisher, dass beim Lesen von $w = ava$ folgende Konfigurationsübergänge passieren:

$$(z_0, ava) \vdash (z_1, va) \vdash \cdots \vdash (z_1, a) \vdash (z_2, \lambda).$$

Die Stelle mit den Punkten ist nun noch unbekannt. Wir wissen nun, dass wir von z_1 ausgehend mit erlaubten Übergängen im Automaten nach z_1 gelangen und dabei das Teilwort v komplett lesen. Ein weiterer Blick auf den Automaten zeigt uns, dass es in z_1 zwei Kanten gibt, eine a - und eine b -Kante. Die a -Kante führt aber zu z_2 und von dort kommen wir nicht mehr nach z_1 . Diese Kante kann uns beim Lesen von v also nicht helfen. Die b -Kante führt uns von z_1 nach z_1 und dies kann uns beim Lesen von v helfen. Da dies außerdem die einzige Kante ist (und sie nicht von z_1 wegführt; dies würde eine weitere Analyse erforderlich machen) können wir schließen, dass wir von der Konfiguration (z_1, va) zur Konfiguration (z_1, a) nur über b -Kanten gelangen, d.h. $v = b^n$ für ein $n \geq 0$ (der Sonderfall $v = 0$ ist auch möglich, wovon man sich schnell durch einen Blick auf den Automaten überzeugt). Damit haben wir argumentiert, dass, wenn w von A akzeptiert wird, w von der Form $a \cdot b^n \cdot a$ für ein $n \geq 0$ sein muss. Zur Erinnerung: Unser ursprüngliches Ziel ist es $L(A) \subseteq M$ zu zeigen. Wir gehen also von einem $w \in L(A)$ aus und wollen zeigen, dass $w \in M$ gilt. Sei also $w \in L(A)$, dann wissen wir nun, dass es ein $n \geq 0$ gibt mit $w = a \cdot b^n \cdot a$. Wegen $a \in \{a\}$ und $b^n \in \{b\}^*$ ist nun $a \cdot b^n \cdot a \in \{a\} \cdot \{b\}^* \cdot \{a\}$, also $w \in M$ und wir sind fertig.

2. Nun wollen wir noch $M \subseteq L(A)$ beweisen. Wir wollen nun also zeigen, dass jedes Wort der Menge M auch vom Automaten A akzeptiert werden kann. Diese Richtung ist meist einfacher zu zeigen. Wir wählen ein beliebiges Wort aus M und zeigen durch Angabe von Konfigurationsübergängen, dass A das Wort zu Ende lesen kann und vom Start- in einen Endzustand gelangt (das Wort also akzeptiert). Da das Wort beliebig gewählt wurde, gilt es dann für alle Worte in M und wir sind fertig. Sei also $w \in M$ beliebig gewählt, d.h. w ist aus $\{a\} \cdot \{b\}^* \cdot \{a\}$ und aufgrund der Definition der Konkatenation und der Sternbildung wissen wir also, dass es ein $n \geq 0$ gibt derart, dass $w = ab^n a$ ist. Man beachte an dieser Stelle, dass wir w weiterhin beliebig gewählt haben. Die verschiedenen n führen dann zu verschiedenen Worten in M . Hat man kompliziertere Sprachen, so sind oft Fallunterscheidungen nötig. Hat man z.B. als Sprache $N := \{a\} \cdot (\{b\}^* \cup \{c\}^*) \cdot \{d\}$ und wählt wieder $w \in N$ beliebig, so weiß man, dass w mit a beginnt und dass dann beliebig viele b oder beliebig viele c folgen. Daran schließt sich dann in beiden Fällen noch ein d an. Man würde dann meist beide Fälle getrennt behandeln und für beide Fälle zeigen, dass der Automat die Worte akzeptiert. Man hat dann also

die beliebig gewählten Worte in zwei Klassen zerlegt und für beide Klassen gezeigt, dass der Automat diese Worte akzeptiert. Da beide Klassen zusammen wieder die ganze Menge N ergeben, ist man dann fertig. Nun legen wir $w = ab^n a$ dem Automaten A vor. Als Übergänge ergibt sich dann aufgrund der Konstruktion von A

$$(z_0, ab^n a) \vdash (z_1, b^n a) \vdash^* (z_1, a) \vdash (z_2, \lambda)$$

wobei \vdash^* für endliche viele (auch 0) Übergänge steht. Damit haben wir eine Erfolgsrechnung auf A und akzeptieren w , also gilt $w \in L(A)$.

Einigen ist der Beweis oben bestimmt zu länglich. Wir waren dort absichtlich ganz besonders ausführlich. Der Beweis geht aber auch viel kürzer.

Wir zeigen zuerst $L(A) \subseteq M$. Sei also $w \in L(A)$. Nach Konstruktion muss w mit a beginnen und mit a enden, denn die einzige Kante aus dem Startzustand z_0 heraus ist mit einem a beschriftet, ebenso wie die einzige Kante in den einzigen Endzustand z_2 hinein. Da diese Kanten zudem nach z_1 gehen bzw. von z_1 kommen, muss ein Wort, das A zwischen diesen beiden a liest, den Automaten von z_1 nach z_1 überführen. Da es an z_1 nur eine b -Schleife gibt, folgt $w \in \{a\}\{b\}^*\{a\}$ und damit $w \in M$.

Sei nun $w \in M$, dann gibt es ein $n \geq 0$ mit $w = a \cdot b^n \cdot a$. Damit ergibt sich

$$(z_0, ab^n a) \vdash (z_1, b^n a) \vdash^* (z_1, a) \vdash (z_2, \lambda)$$

woraus $w \in L(A)$ folgt.

Nachfolgend werden die Beweise in ähnlichen Beispielen an vielen Stellen deutlich kürzer ausfallen. Es ist aber eine gute Übung sich zu Anfang jedes Argument sehr kleinschrittig zu überlegen und aufzuschreiben und sich auch später bei schnelleren Argumentationen, genau zu überlegen, ob die Schritte nachvollziehbar sind (sowohl wenn man selbst schneller vorgeht als auch wenn man schneller voranschreitende Beweise liest).

Da es nun also scheinbar unterschiedlich lange Beweise für die gleiche Aussage gibt, kann man sich an dieser Stelle fragen, ob dieser Unterschied überhaupt erlaubt ist. Ist der kurze Beweis erlaubt? Sind beide Beweise richtig? Und falls ja, sollte man selbst lieber lange oder lieber kurze Beweise führen?

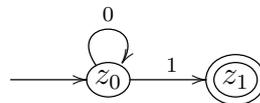
Wichtig beim Beweisen ist, dass die einzelnen Argumentationsschritte für den Leser nachvollziehbar sind. Das eigentliche Ziel bei einem Beweis ist ja, dass der Beweis den Leser von der Richtigkeit der Aussage überzeugt und das durch logisch nachvollziehbare Argumente. Ein kürzerer Beweis erfordert daher im Allgemeinen mehr Kenntnisse (oder mehr Mitdenken) beim Leser. Oben könnte man z.B. sagen, dass jedes Wort, das der Automat liest, mit a beginnen muss oder mit a enden muss. Will man dem Leser mehr Informationen mitgeben, schreibt man vielleicht noch dazu, dass dies nach Konstruktion gilt. Noch mehr Hilfe gibt man, wenn man noch notiert, dass dies an dem Start- und dem Endzustand liegt bzw. an den Kanten die von ihnen kommen bzw. zu

ihnen hinführen. Welche Detailtiefe man wählt hängt sehr von der anvisierten Leserschaft ab und von der eigenen Übung.

Sowohl der lange als auch der kurze Beweis oben sind also richtig. Die kurzen Beweise erwarten aber mehr vom Leser. Ist man in einem Themengebiet neu, so lohnt es sich stets in den Beweisen sehr ausführlich vorzugehen und jeden Argumentationsschritt möglichst ausführlich zu begründen. Dies ist eine gute Übung für einen selbst und es ermöglicht dem Leser gut zu folgen. Für Korrekturen ist insb. von Vorteil, dass man einfacher sagen kann, wo ggf. ein Denkfehler besteht. Zu Anfang gilt daher: bei jedem Beweise an jeder Stelle einmal die ‘Warum?’-Frage stellen. Warum gilt dieses Argument? Wenn die Antwort darauf zu lang ist, dann sollte man hier noch ausführlicher sein.

Noch ein Beispiel

Wir wollen als weiteres Beispiel den unten stehenden Automaten betrachten und diesmal mit einer Annahme beginnen, die sich während des Beweises als falsch herausstellt. Das Beispiel ist zwar wieder so klein, dass man die richtige Sprache gleich erkennen kann, es geht hier aber darum, die Vorgehensweise einzuüben und dafür bieten sich kleinere Beispiele an. Außerdem werden wir hier eine Beweisidee kennenlernen, die an verschiedenen Stellen nützlich sein kann. Später folgt dann auch noch ein größeres und komplizierteres Beispiel.



Angenommen wir vermuten nun, dass die von diesem Automaten akzeptierte Sprache die Menge $M := \{0, 1\}^* \cdot \{1\} \cdot \{0, 1\}^*$ ist. Wir vermuten also, dass

$$L(A) = \{0, 1\}^* \cdot \{1\} \cdot \{0, 1\}^* = M$$

gilt.

Tatsächlich ist die akzeptierte Sprache natürlich $\{0\}^* \cdot \{1\}$ und wir haben beim Aufstellen von M einen Fehler gemacht. Dieses negative Beispiel soll noch einmal illustrieren, wie wir beim Beweis von $L(A) = M$ vorgehen und das Fehler beim Aufstellen von M passieren können, die dann beim Beweisen (hoffentlich) auffallen und korrigiert werden können.

Wir gehen so vor wie eben und versuchen, die zwei Teilmengenbeziehungen zu zeigen.

1. $L(A) \subseteq \{0, 1\}^* \cdot \{1\} \cdot \{0, 1\}^* = M$. Sei $w \in L(A)$, dann wird w von A akzeptiert. Da z_1 der einzige Endzustand ist, muss w auf 1 enden, da die einzige Kante in z_1 hinein mit 1 beschriftet ist. Vorher war der Automat in z_0 , wo nur 0en gelesen werden können. D.h. w hat die Form $w = u1$ mit $u \in \{0\}^*$. Damit gilt auch $w \in M$. (Man kann schon hier merken, dass M

wohl zu “groß” ist. Allerdings muss man es hier noch nicht merken. Die Teilmengenbeziehung $L(A) \subseteq M$ gilt nämlich!

2. $M = \{0, 1\}^* \cdot \{1\} \cdot \{0, 1\}^* \subseteq L(A)$. Sei $w \in M$, dann lässt sich w zerlegen in $w = x_1 \cdot x_2 \cdot x_3$ mit $x_1, x_3 \in \{0, 1\}^*$ und $x_2 = 1$. Da in w eine 1 auftritt, muss es auch eine Stelle geben, an der die 1 das erste Mal auftritt, d.h. w lässt sich sogar zerlegen in $w = x'_1 \cdot x'_2 \cdot x'_3$ mit $x'_1 \in \{0\}^*$, $x'_2 = 1$ und $x'_3 \in \{0, 1\}^*$. Legen wir w nun A als Eingabe vor, so gilt zunächst $(z_0, x'_1 \cdot 1 \cdot x'_3) \vdash^* (z_0, 1 \cdot x'_3)$ und dann $(z_0, 1 \cdot x'_3) \vdash (z_1, x'_3)$. – An dieser Stelle bemerken wir nun, dass wir einen Fehler gemacht haben. Da es in z_1 keine Kante gibt, blockieren wir nun nämlich. Das Wort wird also gar nicht akzeptiert!

Aus dem Beweis(versuch) von eben ergibt sich auch gleich ein *Gegenbeispiel*. Das Wort 010 kann z.B. von dem Automaten nicht akzeptiert werden, ist aber in M .

Im besten Fall kann man nach solch einem Beweisversuch auf eine bessere Idee für M kommen und einen neuen Beweisversuch starten. Wir vermuten nun, dass $M' := \{0\}^* \cdot \{1\}$ die akzeptierte Sprache ist, unsere Vermutung ist nun also

$$L(A) = \{0\}^* \cdot \{1\} = M'$$

1. $L(A) \subseteq \{0\}^* \cdot \{1\} = M'$. Dies zeigt man genau wie eben.
2. $M' = \{0\}^* \cdot \{1\} \subseteq L(A)$. Sei $w \in M'$, dann lässt sich w zerlegen in $w = u1$ mit $u \in \{0\}^*$. Es gilt nun $(z_0, u1) \vdash^* (z_0, 1) \vdash (z_1, \lambda)$ und da z_1 ein Endzustand ist, gilt $w \in L(A)$, was zu zeigen war.

Nun gelingt der Beweis und wir haben $L(A) = M'$ gezeigt.

Obwohl der erste Beweisversuch nicht gelang, konnte man dort eine wichtige Beweisidee sehen. Wissen wir, dass in einem Wort w ein Symbol oder eine Zeichenkette auftritt, wissen wir also z.B. das in w eine 1 auftritt oder die Zeichenkette 000 als Teilwort, dann wissen wir auch, *dass dies an einer Stelle in w das erste Mal passiert*. Dies hilft oft bei der Zerlegung von w in Teilworte und bei der Betrachtung der einzelnen Teilworte. Zerlegt man w z.B. in $v \cdot 000 \cdot u$ und hat dies so gemacht, dass die 000 dort das erste Mal auftritt, dann weiß man, dass in v nicht die Zeichenkette 000 auftritt und sogar, dass v nicht auf 0 endet (sonst wäre das erste Mal, dass 000 auftritt, nämlich ein Symbol vorher gewesen).

Dies schließt unsere Betrachtungen zu Automaten und ihren akzeptierten Sprachen ab. Im nächsten Abschnitt geht es dann um zwei Konstruktionsmethoden, um einen Automaten zu einer Sprache M zu konstruieren.

Wir betonen zur Wiederholung das wichtige Vorgehen in diesem Abschnitt. Hat man für eine Sprache M einen DFA A konstruiert oder, was außer in der

Lehre recht selten vorkommt, zu einem DFA A eine Idee, was die akzeptierte Sprache M ist, so ist $L(A) = M$ zunächst nur eine Behauptung, die dann noch zu zeigen ist.

Hierzu zeigt man dann oft die zwei Teilmengenbeziehungen $L(A) \subseteq M$ und $M \subseteq L(A)$.

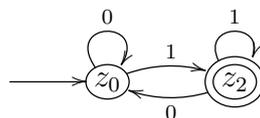
- Mit $L(A) \subseteq M$ zeigt man, dass jedes Wort, das der Automat akzeptiert, in M ist. Bei der Argumentation geht man von einem Wort w aus, das der Automat akzeptiert ($w \in L(A)$) und zeigt, dass dann auch $w \in M$ gilt.
- Mit $M \subseteq L(A)$ zeigt man, dass jedes Wort aus M von dem Automaten akzeptiert wird. Man geht von einem (beliebigen!) Wort aus M aus (“Sei $w \in M$, dann ...”) und zeigt, dass dieses auch von A akzeptiert wird, also in $L(A)$ ist.

Es sei noch mal betont, dass es nicht genügt nur eine der Teilmengenbeziehungen zu zeigen. $L(A) \subseteq M$ zeigt nur, dass jedes Wort, das der Automat akzeptiert, auch in M ist. In M können aber noch Worte sein, die nicht in $L(A)$ sind! Andersherum zeigt $M \subseteq L(A)$ nur, dass jedes Wort in M vom Automaten akzeptiert wird, aber der Automat akzeptiert eventuell weitere Worte. Beides ist nicht wünschenswert. Entweder fehlen dem Automaten Fähigkeiten oder er kann zu wenig.

Man überlege sich zudem, dass es stets sehr leicht wäre, zu einer gegebenen Menge M einen Automaten zu bauen, der nur eine der Teilmengenbeziehungen erfüllt. Will man $L(A) \subseteq M$ sicherstellen, so nimmt man einen Automaten der keine Endzustände hat. Für diesen gilt $L(A) = \emptyset$ und damit sofort $L(A) \subseteq M$. Will man $M \subseteq L(A)$ sicherstellen, so nimmt man als Automat A einen Automaten mit einem einzelnen Zustand z , der sowohl Start- als auch Endzustand ist und für jedes Symbol $x \in \Sigma$ eine Kante von z nach z besitzt. Dann gilt $L(A) = \Sigma^*$ und damit sofort $M \subseteq L(A)$. Nur eine der Teilmengenbeziehungen zu zeigen, genügt also nicht. Es muss $L(A) = M$ gezeigt werden.

Fragen

1. Welche Sprache akzeptiert der folgenden DFA A ?



- a) $L(A) = \{0, 1\}^*$
- b) $L(A) = \{0\}^* \cdot \{1\} \cdot \{1\}^*$
- c) $L(A) = \{0, 1\}^* \cdot \{1\}$

d) $L(A) = (\{0\}^* \cdot \{1\} \cdot \{1\}^* \cdot \{0\})^*$

2. Wie kann man $L(A) = M$ beweisen?

a) Für jedes $w \in L(A)$ zeigen, dass $w \in M$ gilt und andersherum.

b) Für ein $w \in L(A)$ zeigen, dass $w \in M$ gilt und andersherum.

c) Für ein beliebiges $w \in M$ zeigen, dass $w \in L(A)$ gilt und andersherum.

d) Für ein festes $w \in M$ zeigen, dass $w \in L(A)$ gilt und andersherum.

3. Wie kann $L(A) = M$ auch bewiesen werden?

a) $L(A) \setminus M = \emptyset$

b) $L(A) \setminus M = L(A)$

c) $L(A) \cap \overline{M} = \emptyset$

d) $\overline{L(A)} \cup M = \Sigma^*$

4. Wie kann $L(A) \subseteq M$ auch bewiesen werden?

a) $L(A) \setminus M = \emptyset$

b) $L(A) \setminus M = L(A)$

c) $L(A) \cap \overline{M} = \emptyset$

d) $\overline{L(A)} \cup M = \Sigma^*$