Formale Grundlagen der Informatik 1
Kapitel 2
Endliche Automaten und reguläre Sprachen

Frank Heitmann
heitmann@informatik.uni-hamburg.de

5. April 2016

Alphabet und Wörter - Zusammengefasst

Die wichtigsten Dinge zu Alphabeten und Wörtern:

- Σ für eine Menge von Symbolen (ein Alphabet), z.B. $\Sigma = \{a, b, c\}$ oder $\Sigma = \{0, 1\}$.
- λ oder ϵ für das leere Wort.
- Das leere Wort ist *nie in einem Alphabet* Σ!
- Konkatenation:
  - Von Symbolen oder Worten: $a \cdot b = ab$, $ab \cdot cd = abcd$
  - Von Wortmengen: $\{a, ac\} \cdot \{\lambda, c\} = \{a, ac, acc\}$
- $|w|$ für die Länge von w, $|w|_x$ für die Anzahl der x in w.
- R^2, R^3 oder auch Σ^2, Σ^3 und w^2, w^3 usw. für mehrfach Hinterinanderausführen von $\cdot$. Sonderfall: $R^0 = \{\lambda\}$ und $w^0 = \lambda$
- R^+ für alle Worte, die sich durch beliebiges Aneinanderreihen von Worten aus R bilden lassen.
- R^* wie R^+, aber λ kommt noch hinzu.

Alphabete, Wörter und Sprachen

Definition

1. Betrachten wir Σ^* für ein Alphabet Σ, so ist Σ^* die **Menge aller endlichen Wörter** (über dem Alphabet Σ).
2. Jede (Teil-)Menge $L \subseteq \Sigma^*$ heißt **formale Sprache**.

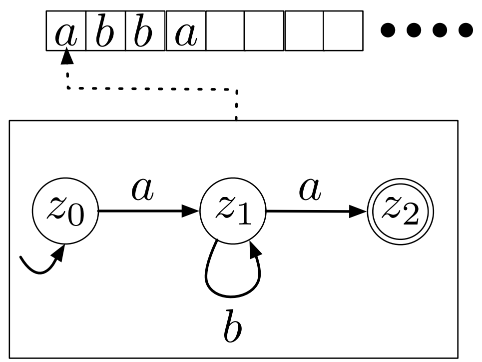
Das Ziel

Ziel ist es nun einen Automaten zu haben, der bestimmte Wörter akzeptiert (und alle anderen nicht). Ein Automat akzeptiert damit eine formale Sprache!

Das Ziel 2

Normalerweise ist eine Sprache M *gegeben* und ein Automat A wird dazu *konstruiert*. Zu zeigen ist dann stets $L(A) = M$.

Endliche Automaten - Beispiel



Der deterministische, endliche Automat

Definition (DFA)

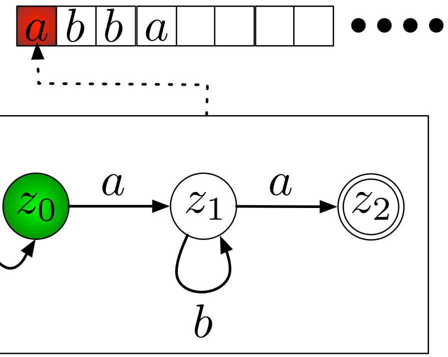
Ein **deterministischer, endlicher Automat** (DFA) ist ein 5-Tupel

$$A = (Z, \Sigma, \delta, z_0, Z_{end})$$

mit:

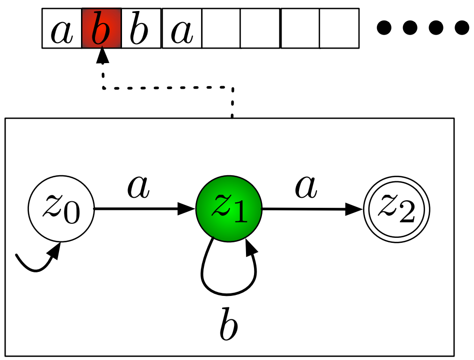
- Der endlichen Menge von *Zuständen* Z.
- Dem endlichen Alphabet Σ von *Eingabesymbolen*.
- Der *Überführungsfunktion* $\delta : Z \times \Sigma \to Z$.
- Dem *Startzustand* $z_0 \in Z$.
- Der Menge der *Endzustände* $Z_{end} \subseteq Z$.

Ein Beispiel



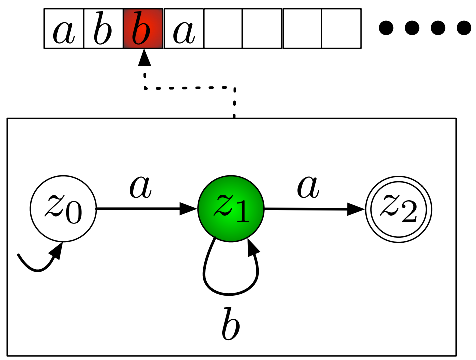
$(z_0, abba)$
oder
$\hat{\delta}(z_0, abba)$

Ein Beispiel



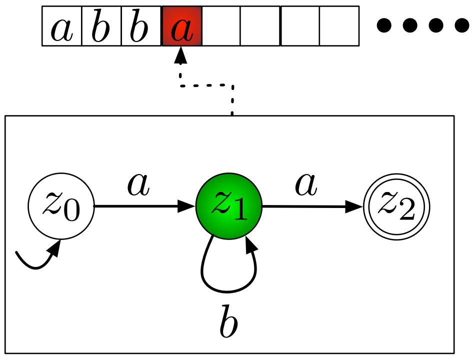
$(z_0, abba) \vdash (z_1, bba)$
oder
$\hat{\delta}(z_0, abba) = \hat{\delta}(z_1, bba)$

Ein Beispiel



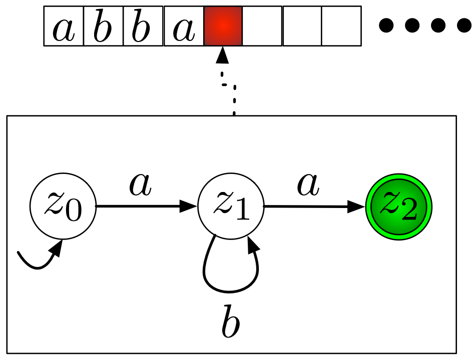
$(z_0, abba) \vdash (z_1, bba) \vdash (z_1, ba)$
oder
$\hat{\delta}(z_0, abba) = \hat{\delta}(z_1, bba) = \hat{\delta}(z_1, ba)$

Ein Beispiel



$$(z_0, abba) \vdash (z_1, bba) \vdash (z_1, ba) \vdash (z_1, a)$$
oder
$$\hat{\delta}(z_0, abba) = \hat{\delta}(z_1, bba) = \hat{\delta}(z_1, ba) = \hat{\delta}(z_1, a)$$

Ein Beispiel



$$(z_0, abba) \vdash (z_1, bba) \vdash (z_1, ba) \vdash (z_1, a) \vdash (z_2, \lambda)$$
oder
$$\hat{\delta}(z_0, abba) = \hat{\delta}(z_1, bba) = \hat{\delta}(z_1, ba) = \hat{\delta}(z_1, a) = z_2$$

Akzeptierte Sprache

Definition (Akzeptierte Sprache)

Die von einem DFA A **akzeptierte Sprache** ist die Menge

$$\begin{aligned} L(A) &:= \{w \in \Sigma^* \mid \hat{\delta}(z_0, w) \in Z_{end}\} \\ &= \{w \in \Sigma^* \mid (z_0, w) \vdash^* (z_e, \lambda), z_e \in Z_{end}\} \end{aligned}$$

Diese Menge wird auch als **reguläre Menge** bezeichnet. Die **Familie aller regulären Mengen** wird mit *REG* bezeichnet.

Wichtige Anmerkung

A akzeptiert ein Wort w genau dann, wenn w bis zum Ende gelesen werden kann **und** er dann in einem Endzustand ist.

Eine Technik

Wichtiges Vorgehen

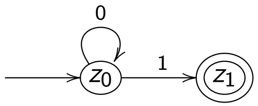
Ermittelt man für einen DFA A seine akzeptierte Sprache M bzw. konstruiert man zu einer Sprache M einen DFA A, so ist $L(A) = M$ zunächst eine Behauptung, die zu zeigen ist!

Wichtiges Vorgehen

Hierzu sind dann **zwei Richtungen** zu zeigen:

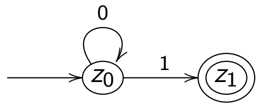
- $L(A) \subseteq M$. Jedes Wort, dass der Automat akzeptiert ist tatsächlich in M. Bei der Argumentation geht man von einem Wort w aus, das der Automat akzeptiert ($w \in L(A)$) und zeigt, dass dann auch $w \in M$ gilt.
- $M \subseteq L(A)$. Jedes Wort aus M wird auch von dem Automaten akzeptiert. Man 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.

Ein Beispiel



Welche Sprache akzeptiert dieser DFA?

Ein Beispiel



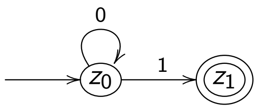
Behauptung

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

Wichtige Anmerkung

Dies ist zunächst eine **Behauptung**! Die Richtigkeit ist zu zeigen. Üblicherweise indem man $L(A) \subseteq M$ **und** $M \subseteq L(A)$ zeigt.

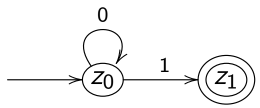
Ein Beispiel



Beweis

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 (einzige Kante in z_1 hinein). Vorher war der Automat in z_0 hier können nur 0en gelesen werden. D.h. w hat die Form $w = u1$ mit $u \in \{0\}^*$. Damit gilt auch $w \in M$.

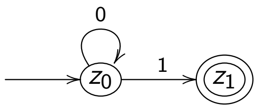
Ein Beispiel



Beweis

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 $(z_0, 1 \cdot x_3') \vdash (z_1, x_3')$. ...
UPS! Da es in z_1 keine Kante gibt, blockieren wir hier! Das Wort wird gar nicht akzeptiert!

Ein Beispiel



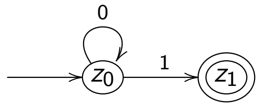
Gegenbeispiel und von vorne...

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

Aus dem Beweis von eben ergibt sich auch gleich ein **Gegenbeispiel**: 010 kann z.B. von dem Automaten nicht akzeptiert werden, ist aber in M. Ein neuer Versuch...

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

Ein Beispiel



Beweis

1. $L(A) \subseteq \{0\}^* \cdot \{1\} = M$. 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 $(z_0, u1) \vdash^* (z_0, 1) \vdash (z_1, \lambda)$. Da z_1 ein Endzustand ist, gilt $w \in L(A)$, was zu zeigen war.

Anmerkung

- Wenn ihr auf eine Menge M kommt, von der ihr meint, dass $L(A) = M$ gilt, dann ist das zunächst eine Behauptung!
- Diese ist zu beweisen/zu begründen!
- Hierzu genügt es nicht nur eine der Teilmengenbeziehungen zu zeigen!
  - $L(A) \subseteq M$. Zeigt nur, dass jedes Wort, das der Automat akzeptiert auch in M ist! Aber in M kann mehr sein! (Würde das genügen, kann man $M = \Sigma^*$ wählen und ist immer gleich fertig!)
  - $M \subseteq L(A)$. Zeigt nur, dass jedes Wort in M vom Automaten akzeptiert wird, aber der Automat kann evtl. mehr! (Insb. vielleicht auch Sachen die er nicht können soll! Und würde dies genügen, kann man stets mit $M = \emptyset$ alles zeigen!
- Also: Sauber die Behauptung aufstellen und sauber die zwei Richtungen trennen und zeigen!

Zusammenfassung - Begriffe

Begriffe bisher (endliche Automaten):

- DFA
- Zustände, Startzustand, Endzustände
- Überführungsfunktion $\delta : Z \times \Sigma \to Z$
- erweitere Überführungsfunktion $\hat{\delta} : Z \times \Sigma^* \to Z$
- vollständig, initial zusammenhängend
- Konfiguration, Konfigurationsübergang
- Rechnung, Erfolgsrechnung
- akzeptierte Sprache
- reguläre Menge, REG

Wichtige Anmerkung

Alle diese Begriffe sind wichtig (auf dieser Grundlage bauen wir auf). Daher die fleissig lernen!

Fragen...

Sei $R = \{a, b\}$ und $S = R^* \cdot \{a\} \cdot R^*$. Was ist $R^* \setminus S$?

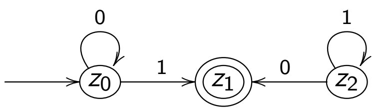
1. R^*
2. $\emptyset$
3. $\{b\}^*$
4. $\{w \in \{a, b\}^* \mid |w|_a \geq 2\} \cup \{\lambda\}$

Fragen...

Sei A ein Automat und M eine Wortmenge. Wenn man $L(A) \subseteq M$ zeigt, dann ...

1. ist man fertig, wenn man M nicht zu groß gewählt hat.
2. weiß man, dass jedes Wort, das in M ist, vom Automaten akzeptiert wird.
3. weiß man, dass der Automat teilweise stimmt.
4. ...

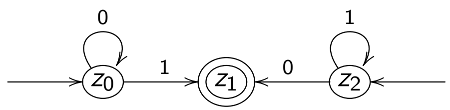
Fragen...



Ist dieser DFA initial zusammenhängend?

1. Ja!
2. Nein!

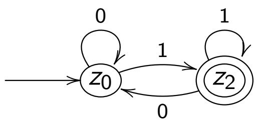
Fragen...



Ist dieser DFA initial zusammenhängend?

1. Ja!
2. Nein!

Fragen...



Welche Sprache akzeptiert dieser DFA A?

1. $L(A) = \{0,1\}^*$
2. $L(A) = \{0\}^* \cdot \{1\} \cdot \{1\}^*$
3. $L(A) = \{0,1\}^* \cdot \{1\}$
4. $L(A) = (\{0\}^* \cdot \{1\} \cdot \{1\}^* \cdot \{0\})^*$

Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

1. 3
2. 4 (jedes Wort, das vom Automaten akzeptiert wird, ist in M)
3. 2
4. 2
5. 3

Die Aufgabe...

Es ist nun also üblicherweise

- eine **Sprache gegeben** (bzw. wir wissen, was wir brauchen; z.B. wollen wir die Wörter erkennen, die auf 0 enden)

Wir wollen dazu

- einen **Automaten konstruieren**

Wie kann man das machen?

Es gibt keine Vorschrift/kein Rezept dafür - aber Techniken.

Konstruktion von DFAs 1

Technik

Techniken zum Konstruieren von DFAs

Methode 1: Was will ich speichern?

(Und wie sehen dann die Zustände aus?)

Konstruktion von DFAs 1 - Beispiel

Beispiel

$$M := \{w \in \{0,1\}^* \mid |w|_0 \text{ ist gerade und } |w|_1 \text{ ist ungerade}\}$$

Wie konstruieren wir hierfür einen DFA?

Tipp: Was wollen wir uns merken?

Definition/Notation (zur Wiederholung)

Sei Σ ein Alphabet, $x \in \Sigma$ ein Symbol und $w \in \Sigma^*$ ein Wort. Mit $|w|_x$ ist dann die Anzahl der x in w gemeint. Z.B. ist mit $\Sigma = \{0,1,2\}$

$$|2202|_2 = 3, \quad |2202|_1 = 0 \quad \text{und } |2202|_0 = 1.$$

Konstruktion von DFAs 1 - Beispiel

Das Problem

$$M := \{w \in \{0,1\}^* \mid |w|_0 \text{ ist gerade und } |w|_1 \text{ ist ungerade}\}$$

Die Idee

Wir merken uns im Zustand, ob die Anzahl der bisher gelesenen 0en gerade (g) oder ungerade (u) ist. Ebenso für die 1en. Das sind **zwei** Informationen, die **jeweils genau einen von zwei Werten annehmen** können (führt zu vier Zuständen) und die **wir leicht mit jedem gelesenen Symbol aktualisieren können**.

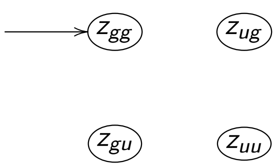
Die Durchführung

Wir notieren gg, gu, ug, uu für die Zustände.
$gu \hat{=}$ gerade Anzahl 0en, ungerade Anzahl 1en
$ug \hat{=}$ ungerade Anzahl 0en, gerade Anzahl 1en ...

Konstruktion von DFAs 1 - Beispiel

Das Problem

$$M := \{w \in \{0,1\}^* \mid |w|_0 \text{ ist gerade und } |w|_1 \text{ ist ungerade}\}$$



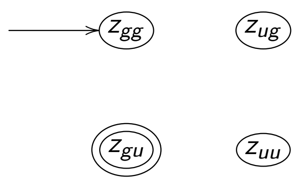
Die Durchführung

Wir notieren gg, gu, ug, uu für die Zustände.
$gu \hat{=}$ gerade Anzahl 0en, ungerade Anzahl 1en
$ug \hat{=}$ ungerade Anzahl 0en, gerade Anzahl 1en ...

Konstruktion von DFAs 1 - Beispiel

Das Problem

$$M := \{w \in \{0,1\}^* \mid |w|_0 \text{ ist gerade und } |w|_1 \text{ ist ungerade}\}$$



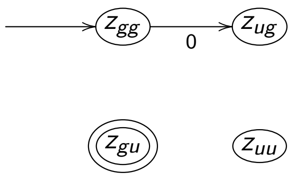
Die Durchführung

Wir notieren gg, gu, ug, uu für die Zustände.
$gu \hat{=}$ gerade Anzahl 0en, ungerade Anzahl 1en
$ug \hat{=}$ ungerade Anzahl 0en, gerade Anzahl 1en ...

Konstruktion von DFAs 1 - Beispiel

Das Problem

$$M := \{w \in \{0,1\}^* \mid |w|_0 \text{ ist gerade und } |w|_1 \text{ ist ungerade}\}$$



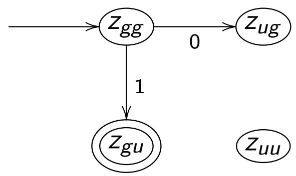
Die Durchführung

Wir notieren gg, gu, ug, uu für die Zustände.
$gu \,\hat{=}$ gerade Anzahl 0en, ungerade Anzahl 1en
$ug \,\hat{=}$ ungerade Anzahl 0en, gerade Anzahl 1en . . .

Konstruktion von DFAs 1 - Beispiel

Das Problem

$$M := \{w \in \{0,1\}^* \mid |w|_0 \text{ ist gerade und } |w|_1 \text{ ist ungerade}\}$$



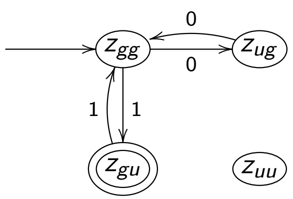
Die Durchführung

Wir notieren gg, gu, ug, uu für die Zustände.
$gu \,\hat{=}$ gerade Anzahl 0en, ungerade Anzahl 1en
$ug \,\hat{=}$ ungerade Anzahl 0en, gerade Anzahl 1en . . .

Konstruktion von DFAs 1 - Beispiel

Das Problem

$$M := \{w \in \{0,1\}^* \mid |w|_0 \text{ ist gerade und } |w|_1 \text{ ist ungerade}\}$$



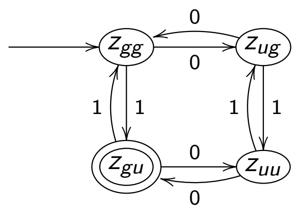
Die Durchführung

Wir notieren gg, gu, ug, uu für die Zustände.
$gu \,\hat{=}$ gerade Anzahl 0en, ungerade Anzahl 1en
$ug \,\hat{=}$ ungerade Anzahl 0en, gerade Anzahl 1en . . .

Konstruktion von DFAs 1 - Beispiel

Das Problem

$$M := \{w \in \{0,1\}^* \mid |w|_0 \text{ ist gerade und } |w|_1 \text{ ist ungerade}\}$$



Die Durchführung

Wir notieren gg, gu, ug, uu für die Zustände.
$gu \,\hat{=}$ gerade Anzahl 0en, ungerade Anzahl 1en
$ug \,\hat{=}$ ungerade Anzahl 0en, gerade Anzahl 1en . . .

Methode 1 - Anmerkungen

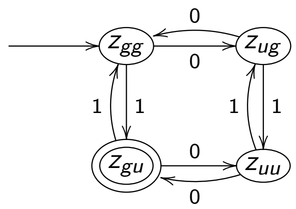
Bemerkung

Bisweilen (nicht hier, aber manchmal) wird mit dieser Methode ein nicht erreichbarer Teil mitkonstruiert. Dieser kann dann weggelassen werden. Es genügt dann sich auf die initiale Zusammenhangskomponente einzuschränken.

Wichtige Anmerkung

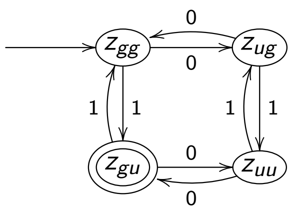
Egal, wie man den Automaten konstruiert, danach ist stets ein Beweis von $L(A) = M$ nötig! Ist die Konstruktion "gut" (d.h. liegt ihr eine eingängige Idee zugrunde), so ist der Beweis aber oft sehr viel einfacher.

Beweis der Korrektheit



- Sei $w \in M$, dann kann w nach Konstruktion (der Automat ist vollständig) zu Ende gelesen werden. Nach Konstruktion der Zustände und der Zustandsübergänge enden wir dann in z_{gu} (da w eine gerade Anzahl von 0en und eine ungerade Anzahl von 1en enthält) und akzeptieren. Also ist auch $w \in L(A)$.

Beweis der Korrektheit



- Sei $w \in L(A)$. Da w akzeptiert wird, muss A in z_{gu} enden. Dies ist aber nach Konstruktion gleichbedeutend damit, dass w gerade viele 0en und ungerade viele 1en enthält. Damit gilt auch $w \in M$.

Beweis der Korrektheit

Anmerkung

Man könnte noch die Zustände und Zustandsübergänge durchgehen und argumentieren, dass sie wirklich das gewünschte leisten (z.B. das der Wechsel von z_{gg} nach z_{gu} eine 1 erfordert und das dann ja wirklich ungerade viele 1en gelesen wurden, wenn vorher gerade viele gelesen wurden). Bei kleineren Beispielen ist das aber meist klar. **Es ist dann aber nötig vorher gut die Konstruktion zu erklären! Und aus dieser Konstruktion müssen sich Dinge ableiten lassen, die gebraucht werden, sonst macht die Formulierung "nach Konstruktion" keinen Sinn!**

Konstruktion von DFAs 2

Technik

Techniken zum Konstruieren von DFAs

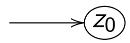
Methode 2: On-the-fly-Konstruktion ...

Konstruktion von DFAs 2 - Beispiel

Beispiel

$$M := \{0,1\}^*\{1\} = \{w \in \Sigma^* \mid \exists v \in \Sigma^* : w = v1\}$$

Wir starten mit dem Startzustand:



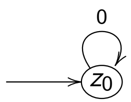
Was passiert bei einer 0 ?

Konstruktion von DFAs 2 - Beispiel

Beispiel

$$M := \{0,1\}^*\{1\} = \{w \in \Sigma^* \mid \exists v \in \Sigma^* : w = v1\}$$

Bei 0: Keine neuen Informationen. Wir bleiben in z_0:



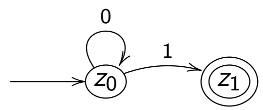
Was passiert bei einer 1 ?

Konstruktion von DFAs 2 - Beispiel

Beispiel

$$M := \{0,1\}^*\{1\} = \{w \in \Sigma^* \mid \exists v \in \Sigma^* : w = v1\}$$

Bei 1: Vielleicht endet das Wort hier. Dann sind wir fertig! Also brauchen wir einen neuen Zustand, in dem wir akzeptieren!



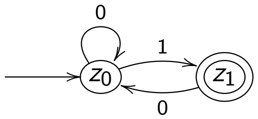
Mit z_0 sind wir fertig (keine weiteren Symbole in Σ).
Aber wir haben eine neuen Zustand! Was passiert hier bei 0?

Konstruktion von DFAs 2 - Beispiel

Beispiel

$$M := \{0,1\}^*\{1\} = \{w \in \Sigma^* \mid \exists v \in \Sigma^* : w = v1\}$$

Bei 0: Wir wollen nicht mehr akzeptieren und könnten einen neuen Zustand einführen. **Aber** wir haben wieder genau so viele Infos wie in z_0, also gehen wir einfach dahin!



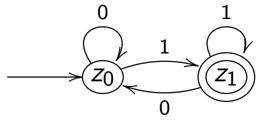
Und was passiert bei 1?

Konstruktion von DFAs 2 - Beispiel

Beispiel

$$M := \{0,1\}^*\{1\} = \{w \in \Sigma^* \mid \exists v \in \Sigma^* : w = v1\}$$

Bei 1 wollen wir weiterhin akzeptieren und bleiben einfach in z_1.



Und sind fertig, da wir in allen Zuständen alle Symbole lesen können!

Methode 2 - Anmerkungen

Bemerkung

Da man sich für jeden Zustand zu jeder Eingabe überlegt, was passiert, wird der konstruierte Automat i.A. vollständig und initial zusammenhängend sein.

Wichtige Anmerkung

Kritisch bei dieser Konstruktionsmethode ist irgendwann zu merken, wann bereits vorhandene Zustände benutzt werden können – und im besten Fall auch, wofür sie stehen; bspw. kann man oben z_x mit "letztes gelesenes Symbol ist x" identifizieren ($x \in \{0,1\}$).

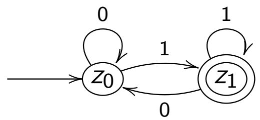
Wichtige Anmerkung

Nochmal: Egal, wie man den Automaten konstruiert, danach ist stets ein Beweis von $L(A) = M$ nötig! Ist die Konstruktion "gut" (d.h. liegt ihr eine eingängige Idee zugrunde), so ist der Beweis aber oft sehr viel einfacher.

Beweis der Korrektheit

Beispiel

$$M := \{0,1\}^*\{1\} = \{w \in \Sigma^* \mid \exists v \in \Sigma^* : w = v1\}$$

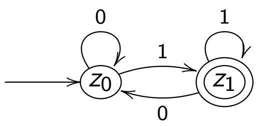


- Sei $w = v1 \in M$. Da der Automat vollständig ist, kann w auf jeden Fall ganz gelesen werden. Da w auf 1 endet, muss der Automat mit dieser letzten 1 nach Konstruktion in den Zustand z_1 wechseln (da jeder Zustand eine 1-Kante zu z_1 hat). Mit $z_1 \in Z_{end}$ folgt $w \in L(A)$.

Beweis der Korrektheit

Beispiel

$$M := \{0,1\}^*\{1\} = \{w \in \Sigma^* \mid \exists v \in \Sigma^* : w = v1\}$$



- Sei $w \in L(A)$. Dann muss nach Konstruktion das Wort auf 1 enden, da z_1 der einzige Endzustand ist und alle Kanten nach z_1 mit 1 beschriftet sind. Dann ist aber sofort auch $w \in M$.

Zusammenfassung

Wir haben heute:

- Zwei **Konstruktionsmethoden für DFAs** kennengelernt.
- Uns mit Beweisen von $L(A) = M$ beschäftigt.

Zusammenfassung - Begriffe

Begriffe bisher (endliche Automaten):

- DFA
- Zustände, Startzustand, Endzustände
- Überführungsfunktion
- erweitere Überführungsfunktion
- vollständig, initial zusammenhängend
- Konfiguration, Konfigurationsübergang
- Rechnung, Erfolgsrechnung
- akzeptierte Sprache
- reguläre Menge, REG

Wichtige Anmerkung

Alle diese Begriffe sind wichtig (auf dieser Grundlage bauen wir auf). Daher die fleissig lernen!

Zusammenfassung - Technik

Wichtiges Vorgehen

Ermittelt man für einen DFA A seine akzeptierte Sprache oder konstruiert zu einer Menge M einen Automaten A, der M akzeptieren soll, so ist $L(A) = M$ zunächst eine Behauptung, die zu zeigen ist!

Wichtiges Vorgehen

Hierzu sind dann **zwei Richtungen** zu zeigen:

- $L(A) \subseteq M$. Jedes Wort, dass der Automat akzeptiert ist tatsächlich in M. 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.
- $M \subseteq L(A)$. Jedes Wort aus M wird auch von dem Automaten akzeptiert. 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.