

Formale Grundlagen der Informatik 1

Kapitel 1

Endliche Automaten

Frank Heitmann
heitmann@informatik.uni-hamburg.de

1. April 2014

Organisatorisches

Zur **Vorlesung**:

- Mo, 12-14 und Di 8-10 in ErzWiss H
- Die Vorlesung wird aufgezeichnet!
 - Livestream:
<https://lecture2go.uni-hamburg.de/livemin>
 - und kurz nach der Vorlesung bei Lecture2Go
- Die Vorlesung hat zwei Teile:
 - Automatentheorie (Frank Heitmann; bis Mo, 19.5)
 - Logik (Christopher Habel; ab Di, 20.5)

Organisatorisches...

Eine typische Woche:

- Mo+Di: Vorlesung
- Di-Fr: Übungsgruppen
 - Neuer Zettel und Bearbeiten der Präsenzaufgaben
 - Abgabe der Übungsaufgaben
 - Rückgabe der Übungsaufgaben der Vorwoche
- Mi: Neuer Lesestoff
 - Als Vorbereitung für die kommende Woche
- Fr: Neue PDFs:
 - Präsenzlösungen der Aufgaben dieser Woche
 - Musterlösung der gerade abgegebenen Aufgaben
 - Neuer Aufgabenzettel

Organisatorisches

Zum **Lesestoff**:

- Soll informal das Modell der jeweiligen Woche erklären.
- ca. 4 Seiten mit kleinem Selbsttest am Ende
- Auf Seite 5 dann die "mathematische Seite"
- Auf Seite 6 Lösungen zum Selbsttest

Anmerkung

Lesestoff und aktueller Übungszettel "passen" nicht. Der Lesestoff ist als Vorbereitung für die *nächste Woche* gedacht. Der Übungszettel behandelt den Stoff der gerade vergangenen Vorlesungen. Organisiert euch gut! ;-)

Organisatorisches

Zu den **Übungen**:

- In den Übungen wird der neue Aufgabenzettel verteilt (gibt es i.A. ab Freitag schon online)
- In den Übungen werden die *Präsenzaufgaben* behandelt:
 - Ü-LeiterIn teilt euch zu Anfang in 3er-Gruppen ein.
 - In diesen ca. 10 Minuten Aufgaben bearbeiten.
 - Ü-LeiterIn geht rum und hilft.
 - Nach 10 Minuten wird "rotiert". Eine Person pro Gruppe wandert im Uhrzeigersinn, eine Person pro Gruppe entgegen des Uhrzeigersinns.
 - Dies wird wiederholt.
 - Besprechen der Präsenzaufgabe im Plenum
 - Wiederholung mit der zweiten Präsenzaufgabe.
- Am Ende der Ü-Gruppe:
 - Abgabe eurer Lösungen des Aufgabenzettels der Vorwoche.
 - Rückgabe eurer Lösungen der Vorvorwoche.

Organisatorisches

Es wird ein **Tutorium** geben:

- Freitag, 16-18 Uhr, B-201 (ab dem 11.4).
- Hier werden die in der Woche abgegebenen Übungsaufgaben besprochen.
- Gut zur Klausurvorbereitung.

Organisatorisches

Es wird vor den Klausuren ein **Repetitorium** geben.

- Termine stehen noch nicht fest.

Organisatorisches

Die **Kriterien**:

- **Übungen**:
 - 50% der Punkte insgesamt.
 - Jeweils 20% der Punkte auf 10 der 12 Aufgabenzettel.
 - Aktiv mitarbeiten.
 - Nicht mogeln ...
 - Weiteres siehe Webseiten und Modulhandbuch.
- **Klausur**:
 - 50% der Punkte

Organisatorisches

- Vorlesung (und Folien)
- **Übungsgruppe (Präsenzaufgaben und Übungsaufgaben)**
- Lesestoff
- Tutorium
- Repetitorium

Bemerkung

Empfehlung: Lesestoff und Vorlesung und Skript/Buch und Aufgaben, Aufgaben, Aufgaben!

Literaturhinweis

Zwei sehr gute Bücher:

- Hopcroft, John E., Motwani, Rajeev und Ullman, Jeffrey D. (2007) **Introduction to Automata Theory, Languages, and Computation**, 3ed, Pearson/Addison-Wesley (auch auf Deutsch erhältlich).
- Schöning, Uwe (2000). **Logik für Informatiker**. Spektrum, Akademischer Verlag.

Skript...

Wer möchte ein Skript?

Die Boxen...

Wichtige Anmerkung

Wichtige Hinweise, Anmerkungen zur Klausur, ...

Anmerkung

Standard. Definitionen, Sätze, Beweise, "normale" Anmerkungen

Bemerkung

Hinweise, Beispiele, Nebenbemerkungen, ...

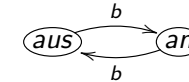
Nebenbemerkung

Literaturhinweis für Interessierte, ...

Motivation ...

Ein (Licht-)Schalter

Ein einfacher (Licht-)Schalter kann *an* oder *aus* sein. Zwischen diesen *Zuständen* kann man wechseln.



Ein (Licht-)Schalter

Ein einfacher (Licht-)Schalter kann *an* oder *aus* sein. Zwischen diesen *Zuständen* kann man wechseln.

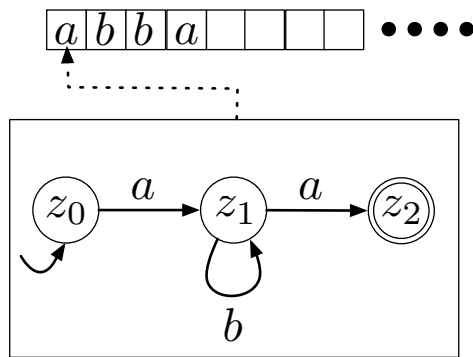


Ein (Licht-)Schalter



- *endlichen Anzahl von Zuständen*;
- davon *einer* als *Startzustand* und
- eine *beliebige Teilmenge* als *Endzustände* hervorgehoben;
- Zustandsübergänge sind möglich sind;
- das System ist stets in *genau einem Zustand*.

Endliche Automaten



Um dieses Modell und Erweiterungen davon wird es gehen...

Und wozu das Ganze?

Und wozu nun das Ganze? Was kann man damit machen?

- Entwicklung von lexikographischen Analysierern, Parsern und Compilern
- Modellierung von Systemen; am Modell können dann Eigenschaften überprüft werden.
- Automaten als grundlegende Datenstruktur (für Mengen und Relationen auf denen bestimmte Operationen nötig sind)
- Grundlage vieler weiterer (Automaten-)Modelle
- Bedeutende Anwendungen
 - bei der Modellierung und Verifikation von Systemen
 - im Compilerbau
 - in der technischen Informatik
 - bei Protokollen
 - ...
- Einüben des abstrakten Denkens! Erkenntnisgewinn!

Mengen

Definition (Mengen)

- Endliche Menge: $M_1 = \{1, 2, 3, 4\}$
- Unendliche Menge: $M_2 = \{1, 2, 3, \dots\}$
- Enthalten sein: $3 \in M_1$, $5 \notin M_1$
- Kardinalität: $|M_1| = 4$ ist die Anzahl der Elemente in M_1 .

Oft werden die Elemente einer Menge durch eine sie *charakterisierende Eigenschaft* beschrieben:

$$M_3 = \{x \in \mathbb{N} \mid \exists i \in \mathbb{N} : x = 2 \cdot i\} = \{0, 2, 4, 6, \dots\}$$

Mengen

Definition (Mengen II)

- ① **Teilmenge:** $A \subseteq B$ gdw. für jedes $a \in A$ auch $a \in B$ gilt
 - $\{1, 4, 5\} \subseteq \{1, 2, 3, \dots\}$
 - $\{1, 4, 5\} \subseteq \{1, 4, 5\}$
 - $\{1, 4, 5\} \not\subseteq \{1, 3, 5, 7, \dots\}$
 - $\{1, 4, 5\} \not\subseteq \{1, 4\}$
- ② **Mengengleichheit:** $A = B$ gdw. $A \subseteq B$ **und** $B \subseteq A$ gilt
- ③ **Echte Teilmenge:** $A \subsetneq B$ gdw. $A \subseteq B$ aber nicht $A = B$

Wichtige Anmerkung

Bei der Mengengleichheit sind **zwei Richtungen** zu zeigen. Einmal $A \subseteq B$ (für jedes Element aus A zeigen, dass es auch in B ist) und einmal $B \subseteq A$ (für jedes Element aus B zeigen, dass es auch in A ist).

Mengen

Definition (Mengen III)

- 1 **Vereinigung:** $A \cup B = \{x \mid x \in A \text{ oder } x \in B\}$
Beispiel: $\{1, 2, 4\} \cup \{1, 2, 3\} = \{1, 2, 4, 1, 2, 3\} = \{1, 1, 2, 2, 3, 4\} = \{1, 2, 3, 4\}$
- 2 **Schnitt:** $A \cap B = \{x \mid x \in A \text{ und } x \in B\}$
Beispiel: $\{1, 2, 4\} \cap \{1, 2, 3\} = \{1, 2\}$
- 3 **Komplement:** $A \setminus B = \{x \mid x \in A \text{ und } x \notin B\}$
Beispiel: $\{1, 2, 4\} \setminus \{1, 2, 3\} = \{4\}$
- 4 **Potenzmenge:** $\mathcal{P}(A) = 2^A = \{B \mid B \subseteq A\}$
Beispiel: $\mathcal{P}(\{1, 2\}) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$
- 5 **Kartesisches Produkt:** $A \times B = \{(a, b) \mid a \in A, b \in B\}$
Beispiel:
 $\{1, 2\} \times \{a, b, c\} = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$

Relationen

Definition (Kartesisches Produkt)

Seien A_1, A_2, \dots, A_n Mengen. Dann ist

$$A_1 \times A_2 \times \dots \times A_n := \{(x_1, x_2, \dots, x_n) \mid x_1 \in A_1, \dots, x_n \in A_n\}$$

das **kartesische Produkt** dieser Mengen. (x_1, \dots, x_n) ist ein *n-Tupel*. Eine *Teilmenge* $R \subseteq A_1 \times \dots \times A_n$ heißt **n-stellige Relation**.

Beispiel

Sei $Z = \{1, 2, 3\}$, $A = \{a, b, c\}$, dann ist

$$R = \{(1, a, 1), (1, b, 2), (2, c, 3), (3, c, 3)\}$$

eine 3-stellige Relation über $Z \times A \times Z$.

Funktionen

Definition (Funktion)

Eine **totale Funktion**

$$f : A \rightarrow B$$

weist jedem Element aus ihrem Definitionsbereich A ein Element aus ihrem Bildbereich B zu. Z.B.

- $f : \{0, 1\} \rightarrow \{0, 1\}$ mit $0 \mapsto 1$ und $1 \mapsto 0$
- $f : \mathbb{N} \rightarrow \mathbb{N}$ mit $x \mapsto 2 \cdot x$ (alternativ $f(x) = 2x$)

Funktionen

Anmerkung

$$f : A \rightarrow B$$

Bemerkung

- 1 Gibt es ein $a \in A$ ohne Bild, d.h. ist $f(a)$ nicht definiert, nennt man f eine *partielle* Funktion.
- 2 Jedes $x \in A$ hat höchstens ein Bild. ($f(x) = 5$ und $f(x) = 4$ ist nicht gleichzeitig möglich.)
- 3 Zwei $x_1, x_2 \in A$ können aber das gleiche Bild haben, d.h. $f(x_1) = f(x_2)$ ist möglich.

Alphabete und Wörter

Definition

- 1 Ein **Alphabet** ist eine (total geordnete) endliche Menge von unterschiedlichen **Zeichen** (alternativ: Buchstaben oder Symbole).
- 2 Die **Konkatenation** \cdot ist die Operation zum Hintereinanderschreiben von Buchstaben. So ergeben sich Worte auf die die Operation dann erweitert wird. Z.B. ist $a \cdot b = ab$ und dann $ab \cdot c = abc$.
- 3 Das **leere Wort** (Wort mit 0 Buchstaben) wird mit λ oder ϵ bezeichnet (nie in einem Alphabet enthalten!).
- 4 Wir schreiben $w^1 = w$, $w^2 = w \cdot w$ und $w^k = w^{k-1} \cdot w$ ($k \geq 2$) und als Spezialfall $w^0 = \lambda$.

Alphabete und Wörter

Die Konkatenation wird auf Mengen von Wörtern erweitert:

$$\begin{aligned} \{ab, aba\} \cdot \{ab, b\} &= \{ab \cdot ab, ab \cdot b, aba \cdot ab, aba \cdot b\} \\ &= \{abab, abb, abaab\} \\ \{a, ab, \lambda\} \cdot \{b, \lambda\} &= \{ab, a, abb, b, \lambda\} \end{aligned}$$

Ausserdem definieren wir für eine Menge von Worten R

$$\begin{aligned} R^1 &= R \\ R^2 &= R \cdot R \\ R^3 &= R^2 \cdot R = R \cdot R \cdot R \\ &\dots \\ R^n &= R^{n-1} \cdot R \end{aligned}$$

und als Spezialfall $R^0 = \{\lambda\}$.

Alphabete und Wörter

Definition

Für eine Menge von Worten R definieren wir

$$\begin{aligned} R^+ &:= \bigcup_{i \geq 1} R^i \text{ mit } R^1 := R \text{ und } R^{i+1} = R^i \cdot R \\ R^* &:= R^+ \cup \{\lambda\} \end{aligned}$$

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**.

Formaler...

Formaler kann man $(\Sigma^*, \cdot, \lambda)$ als ein (freies) Monoid auffassen.

Alphabet und Wörter - Zusammengefasst

Die wichtigsten Dinge:

- 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: $\{a, ac\} \cdot \{\lambda, c\} = \{a, ac, acc\}$
- R^2, R^3 oder auch Σ^2, Σ^3 und w^2, w^3 usw. für mehrfach Hintereinanderausführen von \cdot .
- Sonderfall: $R^0 = \{\lambda\}$ und auch $w^0 = \lambda$ (w ein Wort)
- R^+ für alle Worte, die sich durch beliebiges Aneinanderreihen von Worten aus R bilden lassen.
- R^* wie R^+ , aber λ kommt noch hinzu.

Fragen...

$$\{a, bc\} \cdot \{de, fg\}$$

- ① $\{abc, defg\}$
- ② $\{abcde, abcfg\}$
- ③ $\{abcde, abcfg, bcade, bcafg\}$
- ④ $\{ade, afg, bcde, bcfg\}$

Fragen...

$$\{abc, a\} \cdot \{bc, \lambda\}$$

- ① $\{abca, bc\}$
- ② $\{abcbc, abc\}$
- ③ $\{abcbc, abc\lambda, abc, a\lambda\}$
- ④ $\{abcbc, abc, a\}$

Fragen...

$$R = \{a, b\} \text{ was ist } R^2?$$

- ① $\{\lambda, a, b, aa, ab, ba, bb\}$
- ② $\{a, b, aa, ab, ba, bb\}$
- ③ $\{aa, ab, ba, bb\}$
- ④ $\{\{aa\}, \{ab\}, \{ba\}, \{bb\}\}$

Fragen...

$$R = \{ab, ba, a, b\} \text{ was ist } R^*?$$

- ① Menge aller endlichen Worte aus a und b
- ② Menge aller *ungeraden* Worte aus a und b
- ③ Menge aller *geraden* Worte aus a und b
- ④ Menge aller unendlichen Worte aus a und b

Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

- 1 4
- 2 3 oder 4, aber 4 ist schöner
- 3 3
- 4 1

Der deterministische, endliche Automat

Definition (DFA)

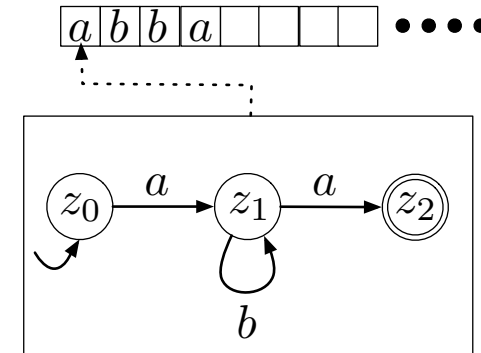
Ein **deterministischer, endlicher Automat** (DFA) ist ein 5-Tupel

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

mit:

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

Endliche Automaten



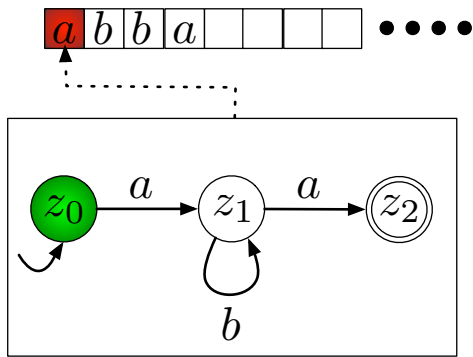
Wie definieren wir das?

Arbeitsweise des DFA (informal)

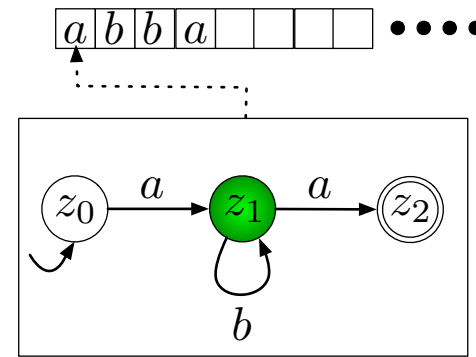
Erhält ein DFA ein Eingabewort $w \in \Sigma^*$, so

- beginnt er im Startzustand z_0 .
- Beginnt w mit dem Symbol $x \in \Sigma$, so
- wird der Nachfolgezustand nun durch $\delta(z_0, x)$ bestimmt.
- Dies wird dann in dem nun aktuellen Zustand und mit dem Restwort fortgeföhrt.
- Das Wort w wird akzeptiert, wenn
 - w bis zum Ende gelesen werden kann **und**
 - der Automat dann in einem Endzustand ist.

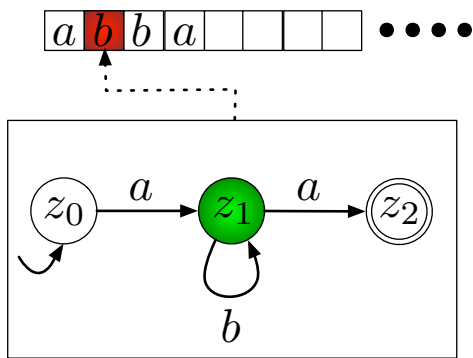
Ein Beispiel



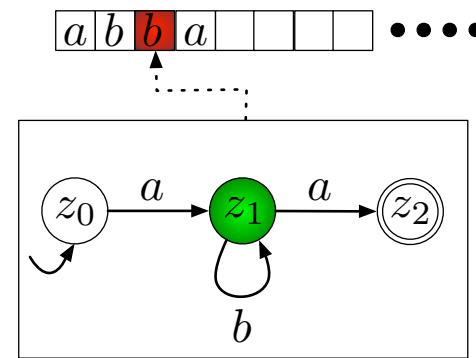
Ein Beispiel



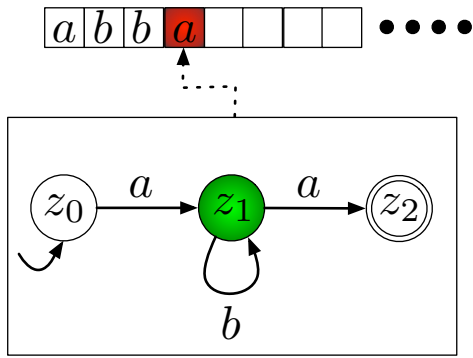
Ein Beispiel



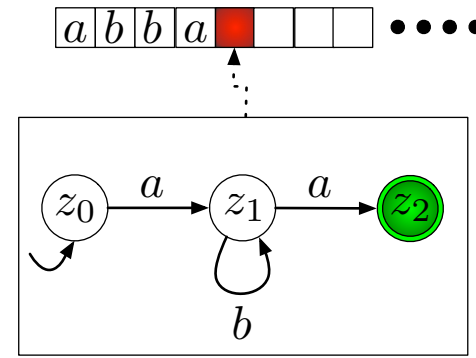
Ein Beispiel



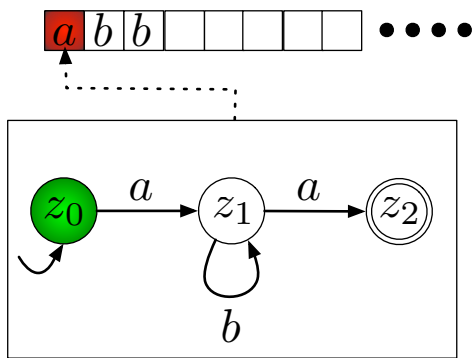
Ein Beispiel



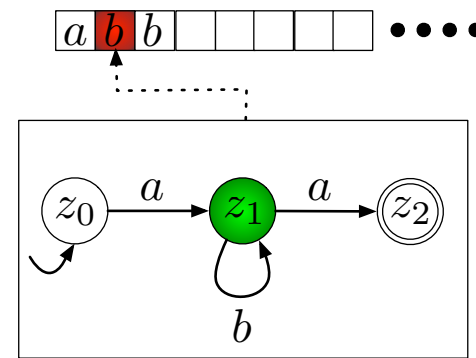
Ein Beispiel



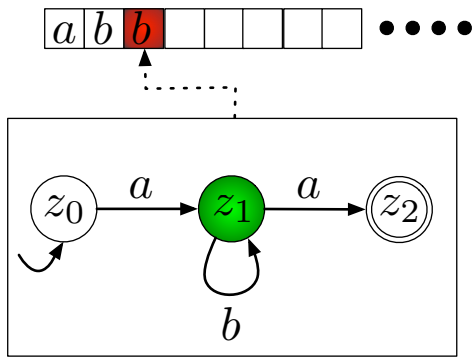
Ein Beispiel



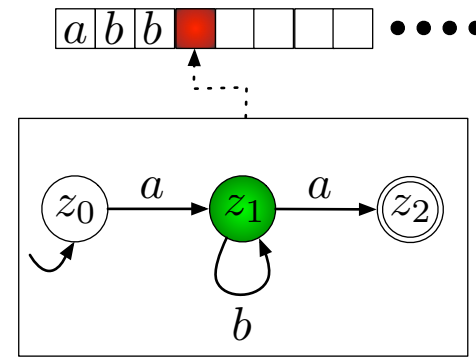
Ein Beispiel



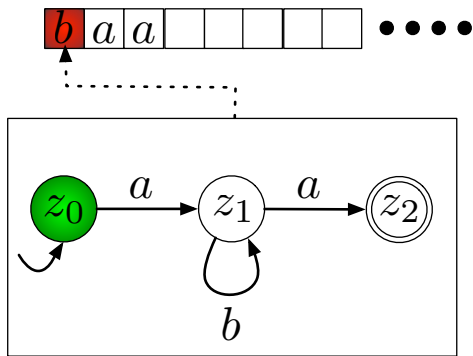
Ein Beispiel



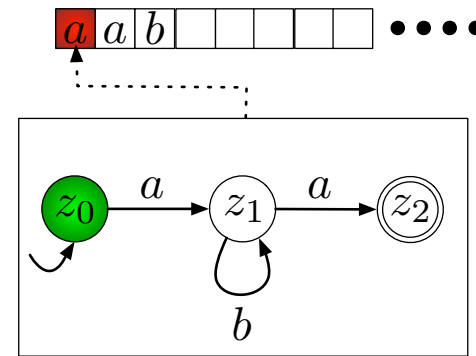
Ein Beispiel



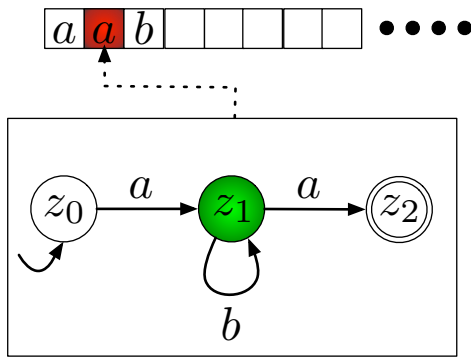
Ein Beispiel



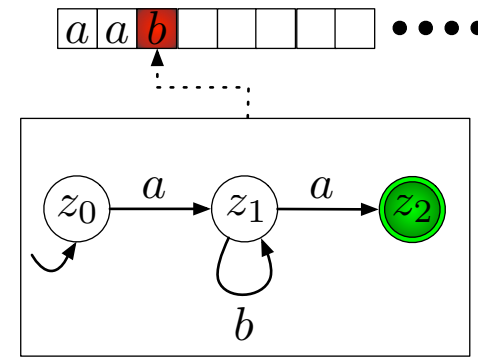
Ein Beispiel



Ein Beispiel



Ein Beispiel



Zusammenfassung

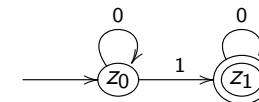
- Das Wort *abba* wird akzeptiert!
(Zu Ende gelesen **und** im Endzustand!)
- Das Wort *abb* wird *nicht* akzeptiert!
(Zu Ende gelesen, aber nicht im Endzustand!)
- Das Wort *baa* wird *nicht* akzeptiert!
(Kann nicht zu Ende gelesen werden!)
- Das Wort *aab* wird *nicht* akzeptiert!
(Kann nicht zu Ende gelesen werden! Wir sind zwar in einem Endzustand, aber wir können das Wort nicht zu Ende lesen!)

Überföhrungsfunktion und Rechnung

Definition (Erweiterte Überföhrungsfunktion)

Sei $A = (Z, \Sigma, \delta, z_0, Z_{end})$ ein DFA. Die **erweiterte Überföhrungsfunktion** $\hat{\delta} : Z \times \Sigma^* \rightarrow Z$ wird für alle $z \in Z, x \in \Sigma$ und $w \in \Sigma^*$ rekursiv definiert durch

$$\begin{aligned} \hat{\delta}(z, \lambda) &:= z \\ \hat{\delta}(z, xw) &:= \hat{\delta}(\delta(z, x), w) \end{aligned}$$



$$\hat{\delta}(z_0, 001) = \hat{\delta}(z_0, 01) = \hat{\delta}(z_0, 1) = z_1$$

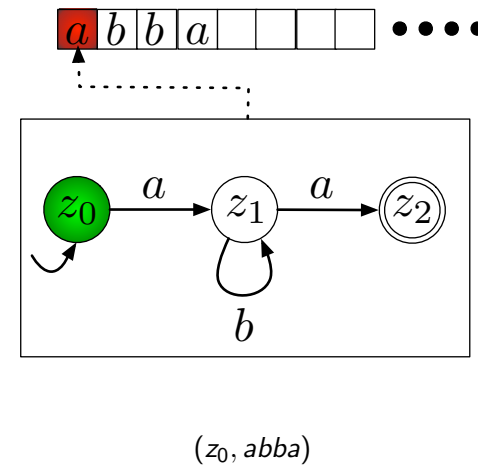
Überföhrungsfunktion und Rechnung

Definition (Konfiguration und Rechnung)

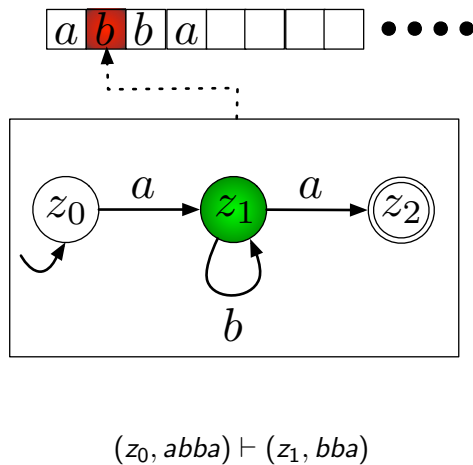
- 1 Eine **Konfiguration** eines DFA A ist ein Tupel $(z, w) \in Z \times \Sigma^*$ mit der Bedeutung, dass A im Zustand z ist und noch das Wort w zu lesen ist.
- 2 Ein **Konfigurationsübergang** ist dann

$$(z, w) \vdash (z', v)$$
 gdw. $w = xv$, $x \in \Sigma$ und $\delta(z, x) = z'$ ist.
- 3 Eine **Rechnung** auf dem Wort $w \in \Sigma^*$ ist eine Folge von Konfigurationsübergängen, die in (z_0, w) beginnt.
- 4 Endet die Rechnung in (z', λ) und ist $z' \in Z_{end}$, so ist dies eine **Erfolgsrechnung** und w wird akzeptiert.

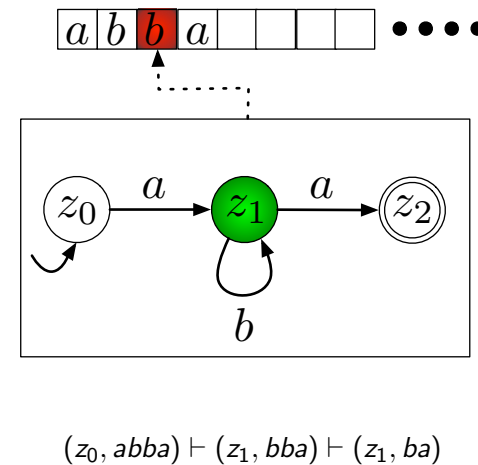
Ein Beispiel



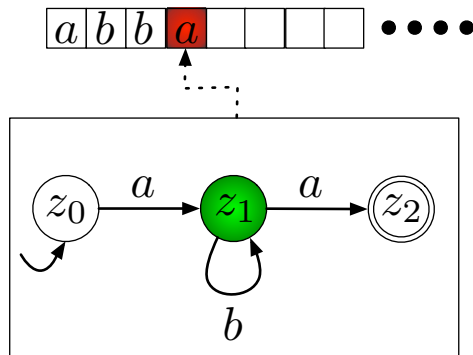
Ein Beispiel



Ein Beispiel



Ein Beispiel



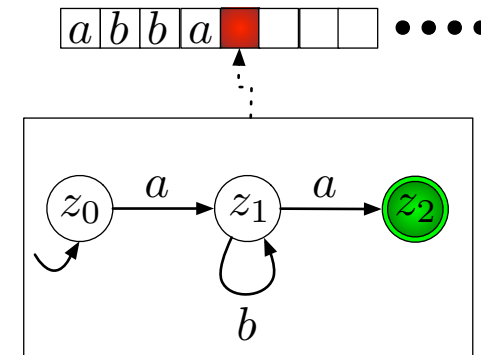
$$(z_0, abba) \vdash (z_1, bba) \vdash (z_1, ba) \vdash (z_1, a)$$

Hinweis

Hinweis

Der Begriff der Rechnung wird oft nicht für DFAs eingeführt (sondern erst später). Auch im Skript ist er nicht zu finden! Druckt euch die eine Folie am besten aus und klebt sie euch ins Skript ;-)

Ein Beispiel



$$(z_0, abba) \vdash (z_1, bba) \vdash (z_1, ba) \vdash (z_1, a) \vdash (z_2, \lambda)$$

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.

Hinweis

Ist A ein DFA, so ist also $L(A)$ eine Menge von Worten (die von A akzeptierte Sprache) und außerdem ist $L(A) \in REG$ (REG enthält also Mengen!).

Noch zwei Begriffe

Definition

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

- ① **vollständig**, wenn zu jedem $(z, x) \in Z \times \Sigma$ ein z' existiert mit $\delta(z, x) = z'$.
- ② **initial zusammenhängend**, wenn es zu jedem $z \in Z$ ein Wort w gibt mit $\hat{\delta}(z_0, w) = z$.

Hinweis

vollständig = jede Eingabe kann gelesen werden

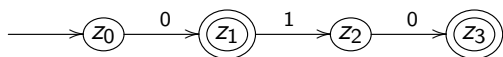
initial zusammenhängend = jeder Zustand erreichbar

Zusammenfassung

Begriffe bisher:

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

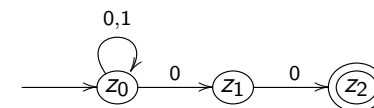
Fragen...



Welche Worte akzeptiert dieser Automat?

- ① Nur 0
- ② 0 und 010
- ③ Nur 010
- ④ 0, 01 und 010
- ⑤ Gar keine Worte (zwei Endzustände sind nicht möglich!)

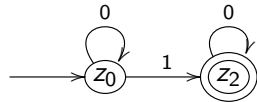
Fragen...



Ist dies ein DFA?

- ① Ja!
- ② Nein!
- ③ Weiß ich nicht!
- ④ Sag ich nicht! ;)

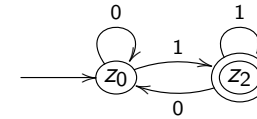
Fragen...



Welche Sprache akzeptiert dieser DFA A?

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

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

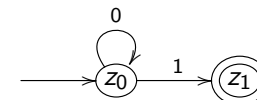
Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

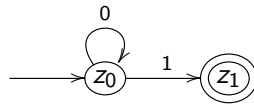
- 1 2
- 2 2
- 3 3
- 4 3

Ein Beispiel



Welche Sprache akzeptiert dieser DFA?

Ein Beispiel



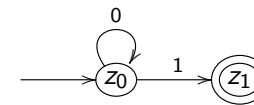
Behauptung

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

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.

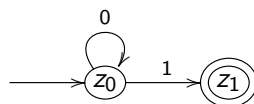
Ein Beispiel



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 $w \in M$.

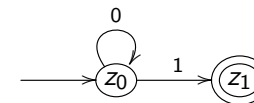
Ein Beispiel



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



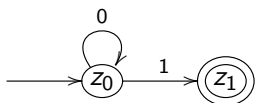
Gegenbeispiel und von vorne...

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

Aus dem Beweis von oben 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

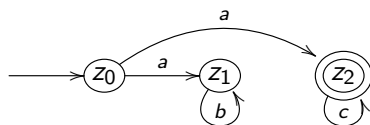
1. $L(A) \subseteq \{0\}^* \cdot \{1\} = M$. Genau wie eben.
2. $M = \{0\}^* \cdot \{1\} \subseteq L(A)$. Sei $w \in M$, dann lässt sich w zerlegen in $w = u1$ mit $u \in \{0\}^*$. Es gilt nun $(z_0, u1) \vdash^* (z_0, 1) \vdash (z_1, \lambda)$. Da z_1 ein Endzustand ist, gilt $w \in L(A)$, was zu zeigen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

Anmerkung

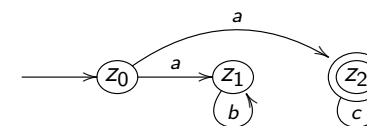
- Wenn ihr auf eine Menge M kommt, von der ihr meint, dass $L(A) = M$ gilt, dann ist das zunächst eine Behauptung!
- Diese ist zu beweisen/zu begründen!
- Hierzu genügt es nicht nur eine der Teilmengenbeziehungen zu zeigen!
 - $L(A) \subseteq M$. Zeigt nur, dass jedes Wort, das der Automat akzeptiert 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!

Ein NFA...

Beim DFA gibt es zu einem $z \in Z$ und einem $x \in \Sigma$ einen Nachfolgezustand $\delta(z, x)$. Der Nachfolgezustand ist also *determiniert*. Dies kann man aufweichen...



Ein NFA...



Wie würdet ihr die Überföhrungsfunktion definieren?

- 1 $\delta : Z \times \Sigma \rightarrow Z$
- 2 $\delta : Z \times \Sigma \rightarrow Z^*$
- 3 $\delta : Z \times \Sigma \rightarrow 2^Z$
- 4 $\delta : 2^{Z \times \Sigma} \rightarrow Z$
- 5 ganz anders ...

Zusammenfassung - Begriffe

Begriffe bisher (mathematische Grundlagen):

- Mengen, (echte) Teilmenge, Mengengleichheit
- Vereinigung, Schnitt, Komplement, Potenzmenge
- kartesisches Produkt, Relation
- Funktion

Begriffe bisher (Buchstaben, Worte):

- Alphabet, Konkatenation, Wort, leeres Wort
- R^+ , R^* , Σ^+ , Σ^* , R^0 , R^1 , w^0 , w^1 , ...

Zusammenfassung - Begriffe

Begriffe bisher (endliche Automaten):

- DFA
- Zustände, Startzustand, Endzustände
- Überföhrungsfunktion
- erweiterbare Überföhrungsfunktion
- vollständig, initial zusammenhängend
- Konfiguration, Konfigurationsübergang
- Rechnung, Erfolgsrechnung
- akzeptierte Sprache
- reguläre Menge, REG
- (Nichtdeterminismus)

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, 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.

Ausblick

Ausblick ...
... und Ende! :)