Formale Grundlagen der Informatik 1
Termin 01
Einführung

Daniel Moldt
Moldt+FGI1@Informatik.Uni-Hamburg.de

Folienvorlage: Frank Heitmann
1. April 2019 (pdf vom 01.04.2019)

Organisatorisches...

Sämtliche Informationen von meiner Seite erfolgen über Stine-Emails:

Bitte sorgfältig prüfen,
ob die Email von gestern Abend Sie erreicht hat!
(Eventuell die Email-Weiterleitung korrigieren / aufheben.)

Alle Materialien werden über OpenOlat verteilt.

Der Zugang zu OpenOlat wird per Stine verschickt.

Emails erreichen mich ausschließlich,
wenn sie von uni-hamburg.de aus verschickt werden!

Meine Email: Moldt+fgi1@Informatik.Uni-Hamburg.de

Organisatorisches...

Eine typische Woche:

- Mo+Di: Vorlesung
- Di-Fr: Übungsgruppen
  - Bearbeiten der Aufgaben
  - Bearbeiten der openOLAT Tests (bis zum eingestellten Datum) Für jede Woche gibt einen obligatorischen Vorbereitungs- und einen Nachbereitungstest.
  - Optionale Abgabe der Übungsaufgaben (der Vorwoche)
  - Kreuzen: Erklärung gleich
  - Rückgabe der Übungsaufgaben (der Vorvorwoche)
  - Neuer Aufgabenzettel
- Anmerkungen:
  - Es gibt keinerlei Musterlösungen zu den Aufgaben.
  - Es ist kein Tutorium parallel während der Vorlesungszeit vorgesehen.

Organisatorisches

Zur **Vorlesung**:

- Mo, 12:15-13:45 und Di, 8:15-9:45 in Erzwiss H
- Die Vorlesung wird NICHT aufgezeichnet!
- Die Vorlesung ist inhaltlich zweigeteilt:
  - Automaten, formale Sprachen, Grammatiken und Logik (bis ca. Mitte Mai)
  - Berechenbarkeit und Komplexität (ab ca. Mitte Mai)
- Ab Berechenbarkeit und Komplexität wird Frau Prof. Dr. Petra Berenbrink die Vorlesungen übernehmen.

Organisatorisches

Zu den **Übungen**:

- Aufgabenzettel gibt es online im OpenOlat System
- In den Übungen werden i.A. die Aufgaben von den Studierenden vorgerechnet: Zum Bestehen der Übungsgruppe ist mindestens drei mal erfolgreich eine Aufgabe an der Tafel vorzurechnen. Alle Teilnehmenden sorgen selbst für die notwendige Anzahl und tragen rechtzeitig vor.
- Wer 6 mal erfolgreich vorrechnet, der erhält einen Bonus für die Klausur. (Priorisierung: Pflicht vor Kür)
- Am Ende der Ü-Gruppe:
  Rückgabe eurer Lösungen der Vorvorwoche.

Organisatorisches

Zu den **Abgaben**:

- Es wird voraussichtlich 13 Aufgabenzettel mit Hausaufgaben geben.
- Es ist gibt wöchentlich (meist) vier Aufgaben.
- Hausaufgaben sollten in Arbeitsgruppen bearbeitet werden.
- Eine Arbeitsgruppe soll i.A. aus zwei Studierenden bestehen.
- Arbeitsgruppen geben nur einen Lösungszettel ab, auf dem alle Namen der Teilnehmer(innen) und die Übungsgruppe lesbar notiert sind.
- Abgabe Eurer Lösungen erfolgt in der Regel direkt bei der Übungsgruppenleitung; die Abgabedaten stehen auf den jeweiligen Zetteln

Organisatorisches

Zum **Kreuzen**:

- Zu Beginn jeder Übungsgruppe tragt Ihr auf einem Zettel ein, welche Aufgaben von Euch vorbereitet wurden und vorgerechnet werden können.
- Die Übungsgruppenleitung wählt dann eine Person aus, die eine angekreuzte Aufgabe dann vorrechnen.
- Bei hinreichend guter Präsentation gilt die Aufgabe als vorgerechnet. (Mind. drei sind notwendig)
- Es müssen mind. 50% der Aufgaben angekreuzt sein (über alle Übungstermine hinweg).

OpenOlat

Wir haben für FGI-1 und FGI-2 Fördermittel der Universität zur Verbesserung der Lehre erhalten.
FormAdTe: Formative Adaptive Tests.
Diese Tests haben sich als sehr förderlich für den Lernerfolg gezeigt, da sie die wöchentliche, kontinuierliche Beschäftigung mit dem Stoff erfordern.

- Formative Tests sollen beim eigenständigen Lernen helfen
- Beliebig viele Versuche
- Keine Zeitbeschränkung während der Beantwortung
- Erfolgreich ab 80% richtige Antworten
- Zeitraum der Bearbeitung ca. 5 Tage
  i.A. von Dienstag 10h00 bis Montag 12h00

Organisatorisches

Die **Kriterien (1)**:

- Übungen:
  - Freiwillig anwesend sein.
  - Freiwillig aktiv mitarbeiten.
  - Mindestens drei Aufgaben erfolgreich an der Tafel vorrechnen
  - Erfolgreiches Kreuzen (50% Regel)

Organisatorisches

Die **Kriterien (2)**:

- Bestehen der openOLAT-Tests
  (höchstens zwei Fehlversuche)
- OpenOlat: Für alle Tests müssen innerhalb der in OpenOlat festgelegten Zeiträume mindestens 80% der Fragen richtig beantwortet werden.
- Klausur:
  - 50% der Punkte

  $\Rightarrow$ erste Klausur ist SEHR früh => kontinuierlich mitarbeiten!
  - Bonus bei mind. ausreichend durch zusätzliches Vorrechnen

Organisatorisches

Es wird vor den Klausuren voraussichtlich an drei Terminen ein **Repetitorium** geben.

- Termine stehen noch nicht fest.
- Die Teilnehmerzahlen sind jeweils begrenzt (aufgrund der Raumgrößen).

Organisatorisches

Zur **Literatur**:

- Zwei sehr gute **Bücher** für den ersten Teil:
  - Michael Sipser: **Introduction to the Theory of Computation**, Third Edition, CENGAGE Learning, 2012
  - Schöning, Uwe (2000). **Logik für Informatiker**. Spektrum, Akademischer Verlag.
- Weitere Literaturangaben siehe Stine-Text / kommen später.

Organisatorisches

OPTIONALE ETI-Präsenzübungen:

- Ab SoSe 2020 gibt es ETI und (spätestens ab SoSe 2021) BKA statt FGI-1.
- Modul ETI: Einführung in die Theoretische Informatik (6 Credits)
- Modul BKA: Berechenbarkeit, Komplexität und Approximation (6 Credits)
- Testlauf für einfache Präsenzübungen für ETI als Option für Interessierte (freiwillig, da unabhängig von FGI-1!)
- Dienstag, 16-18 Uhr in D-220 + Freitag, 12-14 in D-115 (**ab dem 02.04**).

Organisatorisches

- Vorlesung und Folien (erster Teil)
- Übungsgruppe
- OpenOlat Tests
- Bücher
- Repetitorium

Bemerkung 1

Anwesenheit in der Vorlesung ist oft für den Erfolg in der Veranstaltung sehr wichtig!
Eigenständiges Bearbeiten der Aufgaben hilft beim Vertiefen des Stoffes!
Dazu zählt auch aktives Arbeiten in einer kleinen Gruppe!
Ebenso ist es wichtig zeitnah in die Literatur zu schauen!

Organisatorisches

Bemerkung 2

Nehmt den Stift in die Hand!
Notiert euch Dinge selbst (und handschriftlich).
Geht z.B. Folien und Literatur durch und notiert euch, was die wichtigen Dinge der Woche waren.
Rechnet Aufgaben (auch dazu: Stift in die Hand nehmen!).
Redet mit anderen, arbeitet zusammen und investiert Zeit.

Organisatorisches

Bemerkung 3

- Zeitaufwand für dieses Modul laut Modulhandbuch:
- 6 SWS $=$ 9 Credits
- 1 Credit $==$ 30 h
- $\Rightarrow 9 * 30 = 270$ h
- Für 14 Wochen Vorlesungszeit ist das ein sehr hoher Aufwand pro Woche.
- Der vermittelte Stoff erfordert auch tatsächlich diesen Aufwand von durchschnittlichen Studierenden!
- Die freiwillige Teilnahme an den Vorlesungen und Übungen sollte also selbstverantwortlich genutzt werden.

Organisatorisches

Bemerkung 3

- Zeitaufwand für dieses Modul laut Modulhandbuch:
- 6 SWS = 9 Credits
- 1 Credit == 30 h
- $\Rightarrow 9 * 30 = 270$ h
- Für 14 Wochen Vorlesungszeit ist das ein sehr hoher Aufwand pro Woche.
- Der vermittelte Stoff erfordert auch tatsächlich diesen Aufwand von durchschnittlichen Studierenden!
- Die freiwillige Teilnahme an den Vorlesungen und Übungen sollte also selbstverantwortlich genutzt werden.

Die Boxen...

Wichtige Anmerkung

Wichtige Hinweise, Anmerkungen zur Klausur, ...

Anmerkung

Standard. Definitionen, Sätze, Beweise, “normale” Anmerkungen

Bemerkung

Hinweise, Beispiele, Nebenbemerkungen, ...

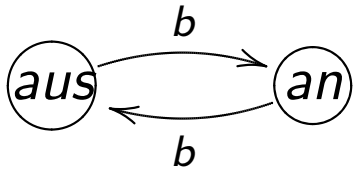
Nebenbemerkung

Literaturhinweis für Interessierte, ...

Eine erste Idee eines Automaten ...

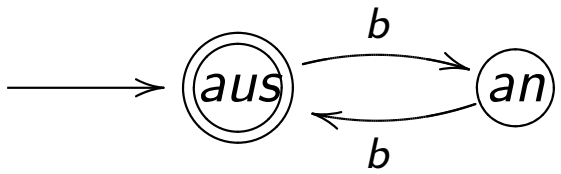
Ein (Licht-)Schalter

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

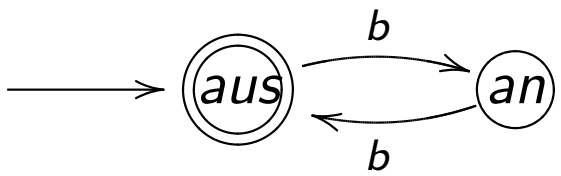


Ein (Licht-)Schalter

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

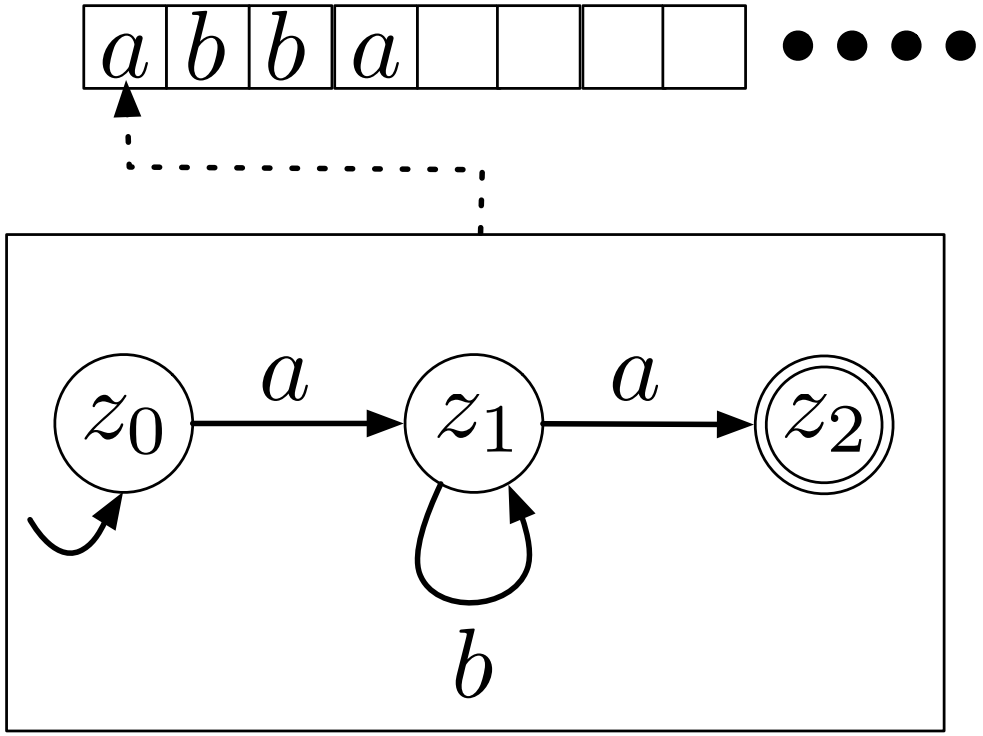


Ein (Licht-)Schalter



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

Endliche Automaten



Um dieses Modell und Erweiterungen davon wird es gehen...
... wobei *a* und *b* hier nur Abstraktionen sind.

Und wozu das Ganze?

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

- Wir beginnen mit einem recht einfachen Modell (DFA). Da denkt man sich "Was soll ich damit?!".
  - Erstmal lernen! Damit gehen dann später nützliche und spannende Dinge! Mit $+$ und $=$ in einer Programmiersprache geht auch nicht viel, trotzdem muss man die erstmal lernen.
- Grundlage vieler weiterer (Automaten-)Modelle

$\Rightarrow$ Schnelle Lösung des (Wort-)Suchproblems

$\Rightarrow$ Entwicklung von lexikographischen Analysierern, Parsern und Compilern

$\Rightarrow$ Modellierung von Systemen;
am Modell können dann Eigenschaften überprüft werden.
  - Kommt insb. auch in FGI2.

- Automaten als grundlegende Datenstruktur (für Mengen und Relationen auf denen bestimmte Operationen nötig sind)

Und wozu das Ganze?

Bedeutende Anwendungen

- bei der Modellierung und Verifikation von Systemen
- Bestandteil softwaretechnischer Verfahren
- in der technischen Informatik
- bei Protokollen
- ...

und Erkenntnisse

- Probleme, die von einem Computer **nicht** gelöst werden können
- Probleme, die von einem Computer **praktisch nicht** gelöst werden können

Und wozu das Ganze?

Und ganz wichtig:

- Einüben des abstrakten Denkens und des exakten Argumentierens!
- ... sonst klappt es mit obigem nicht
- ... und dann kann man drüber nachdenken, wie man die Grenzen doch umgeht... ;-)

Und wir beginnen jetzt ganz grundlegend ...

Und wozu das Ganze?

Wir schauen jetzt kurz auf Grundlagen, die im Folgenden durchgängig benötigt werden.

- Mengen
- Sequenzen / Folgen und Tupel
- Funktionen und Relationen
- Graphen
- Alphabete und Wörter
- Endliche Automaten (erst einmal nur die Idee)

Mengen

Definition (Mengen)

- Endliche Menge: $M_1 = \{1, 2, 3, 4\}$
- Unendliche Menge: $M_2 = \{1, 2, 3, \ldots\}$
- 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:

$$M_3 = \{x \in \mathbb{N} \mid \exists i \in \mathbb{N} : x = 2 \cdot i\} = \{2, 4, 6, 8 \ldots\}$$

Mengen

Definition (Mengen II)

1. **Teilmenge**: $A \subseteq B$ gdw. für jedes $a \in A$ auch $a \in B$ gilt
   - $\{1, 4, 5\} \subseteq \{1, 2, 3, \ldots\}$
   - $\{1, 4, 5\} \subseteq \{1, 4, 5\}$
   - $\{1, 4, 5\} \not\subseteq \{1, 3, 5, 7, \ldots\}$
   - $\{1, 4, 5\} \not\subseteq \{1, 4\}$
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. **Multimenge**: In einer Multimenge können Elemente mehrfach auftreten. Die beiden Multimengen $\{4\} \textit{und} \{4, 4\}$ unterscheiden sich, während sie identisch sind bei einer Betrachtung als Menge.
2. Eine unendliche Menge enthält unendlich viele Elemente. Die natürlichen Zahlen $\mathbb{N}$ werden als $\{1, 2, 3, ...\}$ und die ganzen Zahlen $\mathbb{Z}$ als $\{..., -2, -1, 0, 1, 2, 3, ...\}$ dargestellt.
3. Die leere Menge wird als $\emptyset$ (selten auch als $\{\}$) dargestellt.

Mengen

Definition (Mengen IV)

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. **Mengendifferenz**: $A \setminus B = \{x \mid x \in A \text{ und } x \notin B\}$
   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. **Komplement**: $\overline{A} = \{x \mid x \notin A\}$
   Das Komplement wird auf einer Grundmenge von berücksichtigten Elementen definiert

Mengen

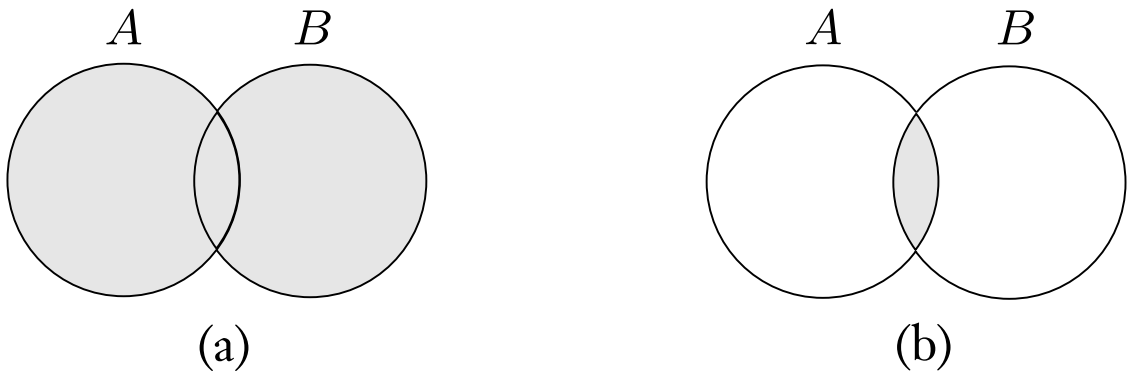


Abbildung: Venn Diagramm für (a) $A \cup B$ und (b) $A \cap B$

Folgen und Tupel

Definition (Folgen)

- Folgen sind Listen von Elementen in einer festgelegten Reihenfolge.
- Für Folgen gilt:

$$(1,2,3) \neq (3,2,1), \text{ sowie } (1,1,2) \neq (1,2)$$

- Folgen können ebenfalls endlich oder unendlich sein.
- Endliche Folgen (der Länge k) werden auch **(k-)Tupel** genannt
- Ein 2-Tupel wird auch **geordnetes Paar** als bezeichnet

Folgen und Tupel

Definition (Kartesisches Produkt (Kreuzprodukt))

Seien $A_1, A_2, \ldots, A_n$ Mengen. Dann ist

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

das **kartesische Produkt** dieser Mengen. $(x_1, \ldots, x_n)$ ist ein *n-Tupel*.
Kreuzprodukte einer Menge mit sich selbst wird verkürzt notiert:

$$\overbrace{A \times A \times \ldots \times A}^{k} = A^k$$

Relationen und Funktionen

Definition (Relation I)

Ein Eigenschaft, deren Definitionsbereich eine Menge von n-Tupeln $R \subseteq A_1 \times \ldots \times A_n$ ist, wird **Relation** genannt.
Eine *Teilmenge* $R \subseteq A_1 \times \ldots \times A_n$ heißt ***n*-stellige Relation (auf A)**.

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 $Z \times A \times Z$.

Relationen und Funktionen

Definition (Relation II)

Eine zweistellige Relation heißt **binäre Relation**. Für Ausdrücke zu einer binären Relation wird typischerweise die Infix-Notation genutzt.

Beispiel

Ist R eine binäre Relation, dann meint aRb, dass $aRb = TRUE$ gilt. Entsprechend $R(a_1, \ldots, a_n)$ und $R(b_1, \ldots, b_n) = TRUE$.

Relationen und Funktionen

Definition (Äquivalenzrelation)

Eine spezielle binäre Relation $R \subseteq A \times A$ ist die **Äquivalenzrelation**, die folgende drei Eigenschaften erfüllt:

1. R ist **reflexiv** $\iff \forall x \in A : xRx$
2. R ist **symmetrisch** $\iff \forall x, y : xRy \implies yRx$
3. R ist **transitiv** $\iff \forall x, y, z : xRy, yRz \implies xRz$

Relationen und Funktionen

Definition (Funktion)

Eine **Funktion** ist ein Objekt, welches eine Eingabe-Ausgabe Beziehung aufstellt.

Definition (Totale Funktion)

Eine **totale Funktion**

$$f : A \to B$$

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

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

Relationen und Funktionen

Anmerkung

$$f : A \to B$$

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.

Relationen und Funktionen

Begriffe

Mit dem Definitionsbereich

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

ist die **Eingabe für f** ein n-Tupel. x_i ist ein **Argument** von f.
Eine Funktion mit k Argumenten ist eine k-**stellige Funktion**, entsprechend einstellige / binäre Funktion bei 1 / 2 Argumenten.
Infixnotation entspricht $a + b$.
Präfixnotation entspricht $add(a, b)$ oder auch $+a\ b$.

Relationen und Funktionen

Begriffe

Ein **Prädikat** (auch **Eigenschaft** genannt) ist eine Funktion, deren Bildbereich $\{TRUE, FALSE\}$ ist.

Beispiel

$even(4) = TRUE$ und $even(5) = FALSE$.

Graphen

Begriffe

Ein **ungerichteter Graph** (auch **Graph** genannt) ist eine Menge von Punkten und Linien, die einige der Punkte verbinden. Die Punkte werden **Knoten** (oder auch **Ecken**) und die Linien **Kanten** genannt.
Die Anzahl der Kanten an einem Knoten ist der **Grad** des Knoten.
Eine **Schleife** ist eine Kante, die einen Knoten mit sich selbst verbindet.
Beschriftete Graphen führen Beschriftungen an den Kanten.
Ein **Teilgraph** H hat eine Teilmenge der Knoten des Graph G und alle entsprechenden Kanten aus G für seine Knoten.

Graphen

Graphenbeispiel

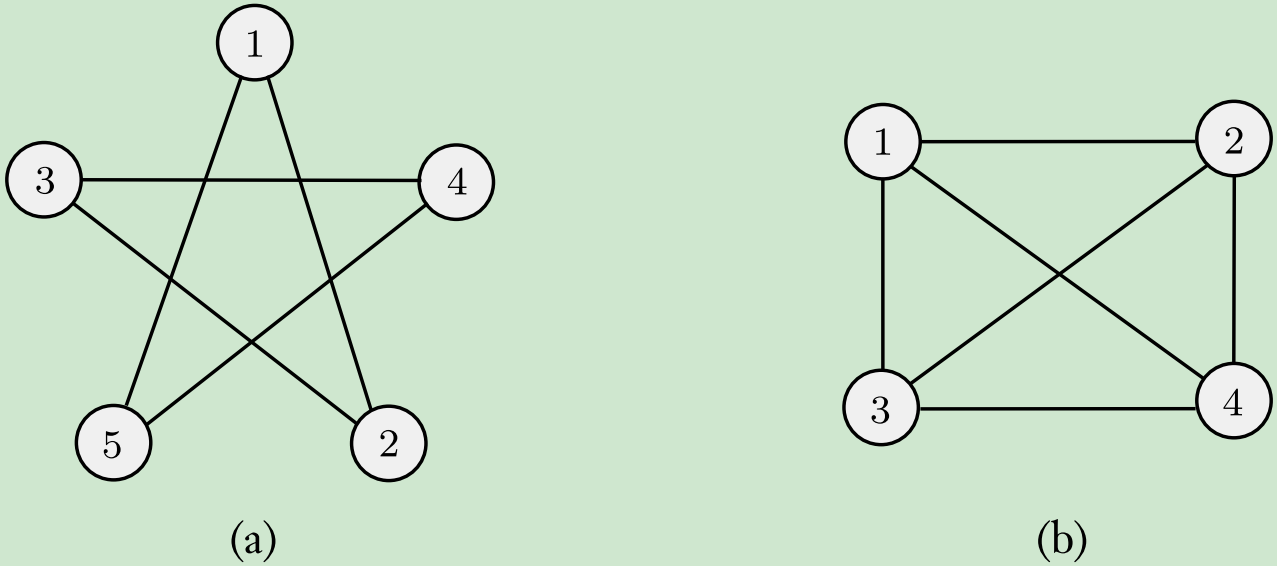


Abbildung: Beispielgraphen mit (a) Grad 2 und (b) Grad 3

Graphen

Graphenbeispiel

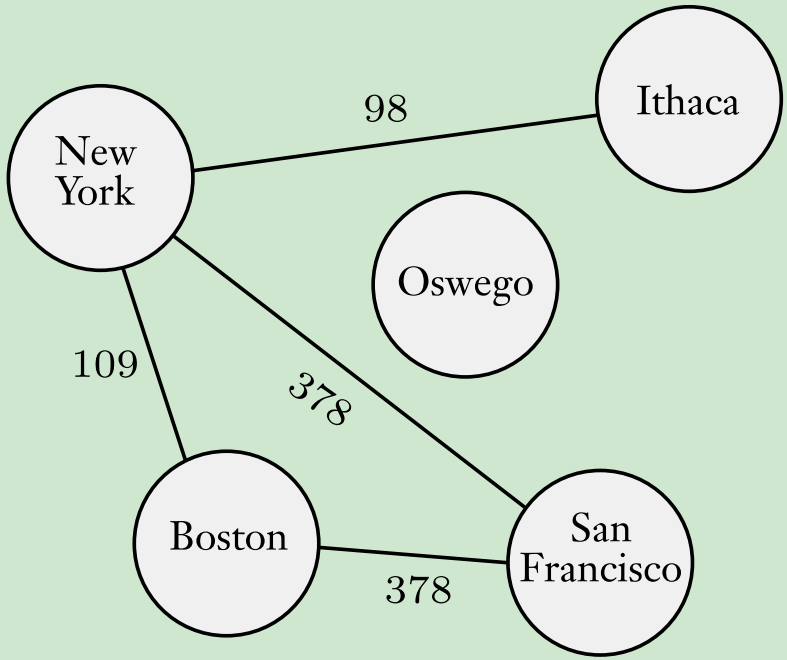


Abbildung: Beschrifteter Graph

Graphen

Graphenbeispiel

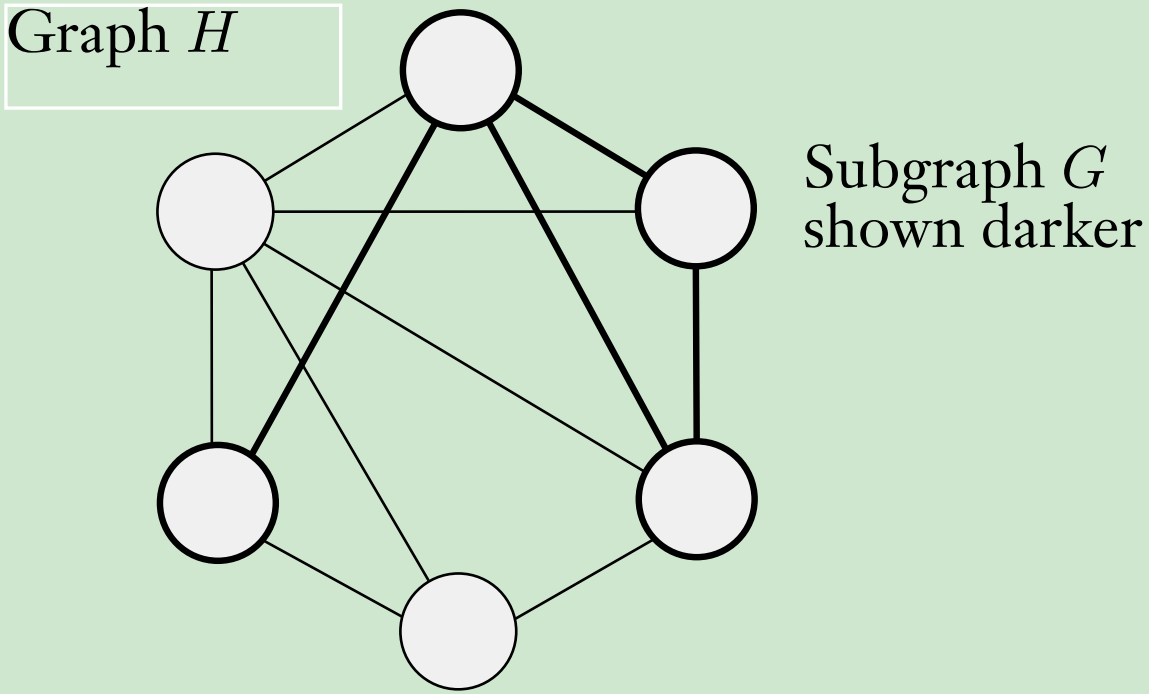


Abbildung: Teilgraph H: dunklere Kanten und Knoten aus G

Graphen

Begriffe

Ein **Pfad** in einem Graphen ist eine Folge von Knoten, die durch Kanten verbunden sind. Ein **einfacher Pfad** enthält keinen Knoten mehrfach.
Ein Graph ist **zusammenhängend** (verbunden), wenn jedes Knotenpaar durch einen Pfad verbunden ist.
Ein Pfad ist ein **Kreis**, falls er im selben Knoten beginnt und endet.
Ein Kreis ist ein **einfacher Kreis**, falls er mindestens drei Knoten hat und nur den ersten / letzten Knoten wiederholt.
Ein Graph ist ein **Baum**, wenn er zusammenhängend ist und keine einfachen Kreise enthält.
In einem Baum kann ein Knoten als **Wurzel** festgelegt werden.
Ein Knoten mit dem Grad eins, der nicht die Wurzel ist, wird als **Blatt** bezeichnet.

Graphen

Graphenbeispiel

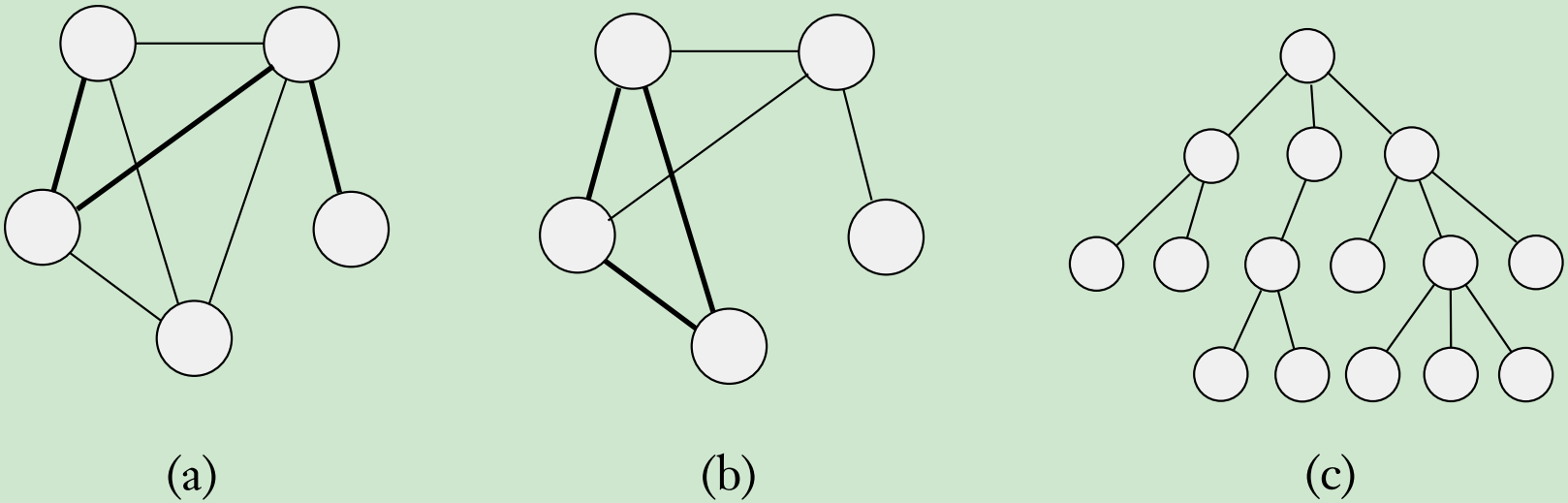


Abbildung: (a) Pfad im Graphen: fette Kanten, (b) Kreis: fette Kanten und (c) Baum

Graphen

Begriffe

Ein **gerichteter Graph** hat **Pfeile** (gerichtete Kanten).
Ein gerichteter Graph $G = (V, E)$ besteht aus der Knotenmenge V und der Menge an Pfeilen E.
Ein Pfeil von i nach j wird als geordnetes Paar (i, j) notiert.
Bei einem **gerichteten Pfad** zeigen alle Pfeile in dieselbe Richtung.
Ein gerichteter Graph ist **streng zusammenhängend**, wenn ein gerichteter Pfad alle Knotenpaare verbindet.

Alphabete und Wörter

Definition

1. Ein **Alphabet** ist eine nichtleere, (total geordnete) endliche Menge von unterschiedlichen **Symbolen** (alternativ: Buchstaben oder Zeichen).
2. Die **Konkatenation** $\cdot$ ist die Operation zum Hintereinanderschreiben von Symbolen.
   So ergeben sich **Wörter** (Strings, Zeichenketten) 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 Symbolen) wird mit ε (oder auch mit λ) bezeichnet.
   (ε ist 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 $w^0 = \varepsilon$.

Alphabete und Wörter

Die Konkatenation wird auf Mengen von Wörtern erweitert:

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

Außerdem definieren wir für eine Menge von Wörtern 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 $R^0 = \{\varepsilon\}$.

Alphabete und Wörter

Definition

Für eine Menge von Wörtern R definieren wir

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

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

Alphabete und Wörter - Noch zwei Notationen

Noch zwei Notationen:

Definition/Notation

Sei Σ ein Alphabet, $x \in \Sigma$ ein Symbol und $w \in \Sigma^*$ ein Wort.

1. $|w|$ ist die Länge von w, z.B. ist $|001| = 3$ und $|00111| = 5$. Außerdem ist $|\varepsilon| = 0$.
2. $|w|_x$ ist die Anzahl der x in w. Z.B. ist mit $\Sigma = \{0, 1, 2\}$

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

Zusammenfassung

Mathematische Terme

SUMMARY OF MATHEMATICAL TERMS

Alphabet	A finite, nonempty set of objects called symbols
Argument	An input to a function
Binary relation	A relation whose domain is a set of pairs
Boolean operation	An operation on Boolean values
Boolean value	The values TRUE or FALSE, often represented by 1 or 0
Cartesian product	An operation on sets forming a set of all tuples of elements from respective sets
Complement	An operation on a set, forming the set of all elements not present
Concatenation	An operation that joins strings together
Conjunction	Boolean AND operation
Connected graph	A graph with paths connecting every two nodes
Cycle	A path that starts and ends in the same node
Directed graph	A collection of points and arrows connecting some pairs of points
Disjunction	Boolean OR operation
Domain	The set of possible inputs to a function
Edge	A line in a graph
Element	An object in a set
Empty set	The set with no members
Empty string	The string of length zero
Equivalence relation	A binary relation that is reflexive, symmetric, and transitive
Function	An operation that translates inputs into outputs

Zusammenfassung

Mathematische Terme

Graph	A collection of points and lines connecting some pairs of points
Intersection	An operation on sets forming the set of common elements
k-tuple	A list of k objects
Language	A set of strings
Member	An object in a set
Node	A point in a graph
Ordered pair	A list of two elements
Path	A sequence of nodes in a graph connected by edges
Predicate	A function whose range is $\{\text{TRUE}, \text{FALSE}\}$
Property	A predicate
Range	The set from which outputs of a function are drawn
Relation	A predicate, most typically when the domain is a set of k-tuples
Sequence	A list of objects
Set	A group of objects
Simple path	A path without repetition
Singleton set	A set with one member
String	A finite list of symbols from an alphabet
Symbol	A member of an alphabet
Tree	A connected graph without simple cycles
Union	An operation on sets combining all elements into a single set
Unordered pair	A set with two members
Vertex	A point in a graph

Zusammenfassung - Übersetzung

Alphabet	Endliche, nicht-leere Menge an Objekten, genannt Symbole
Argument	Eine Eingabe einer Funktion
Binäre Relation	Eine Relation dessen Urbildmenge eine Menge aus Paaren ist
Bool'sche Operation	Eine Operation auf bool'schen Werten
Bool'scher Wert	Die Werte `TRUE` und `FALSE`, häufig auch durch 0 und 1 repräsentiert
Kartesisches Produkt	Eine Operation auf Mengen die eine Menge aus allen Tupeln der Elemente der Mengen erzeugt
Komplement	Eine Operation auf eine Menge die eine Menge aus allen nicht vorhandenen Elementen erzeugt
Konkatenation	Eine Operation die Wörter aneinanderreiht
Konjunktion	Bool'sche `UND` Operation
Zusammenhängender Graph	Ein Graph mit Pfaden die alle Knoten paarweise verbinden
Zyklus	Ein Pfad der im selben Knoten startet und endet
Gerichteter Graph	Eine Sammlung von Punkten und Pfeilen die Paare von Punkten verbinden
Disjunktion	Bool'sche `ODER` Operation
Urbildmenge	Die Menge aller möglichen Eingaben einer Funktion. Auch: Quellmenge
Kante	Eine Linie in einem Graphen
Element	Ein Objekt in einer Menge
Leere Menge	Eine Menge ohne Elemente
Leeres Wort	Ein Wort mit der Länge null
Äquivalenzrelation	Eine binäre Relation die reflexiv, symmetrisch und transitiv ist
Funktion	Eine Operation die Eingaben in Ausgaben übersetzt
Graph	Eine Sammlung von Punkten und Linien die Paare von Punkten verbinden
Schnitt	Eine Operation auf Mengen die eine Menge aus allen gemeinsamen Elementen erzeugt
k-Tupel	Eine Liste von k Objekten
Sprache	Eine Menge an Wörtern

Zusammenfassung - Übersetzung

Element	Ein Objekt in einer Menge (Duplikat)
Knoten	Ein Punkt in einem Graphen
Geordnetes Paar	Eine Liste aus zwei Elementen
Pfad	Eine Sequenz von Knoten in einem Graphen die durch Kanten verbunden sind
Prädikat	Eine Funktion mit dem Bildbereich {`TRUE`, `FALSE`}
Eigenschaft	Ein Prädikat
Bildbereich	Die Menge aus der die Ausgaben einer Funktion gewählt werden
Relation	Ein Prädikat, typischerweise wenn die Urbildmenge eine Menge an k-Tupeln ist
Sequenz	Eine Liste von Objekten
Menge	Eine Gruppe von Objekten
Einfacher Pfad	Ein Pfad ohne Wiederholung
Einelementige Menge	Menge mit einem Element
Wort	Eine endliche Liste an Symbolen aus einem Alphabet
Symbol	Ein Element eines Alphabets
Baum	Ein zusammenhängender Graph ohne einfache Zyklen
Vereinigung	Eine Operation auf Mengen die alle Elemente in eine Menge kombiniert
Ungeordnetes Paar	Eine Menge mit zwei Elementen
Ecke	Ein Punkt in einem Graphen

Fragen...

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

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

Fragen...

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

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

Fragen...

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

1. $\{abca, bc\}$
2. $\{abcbc, abc\}$
3. $\{abcbc, abc\varepsilon, abc, a\varepsilon\}$
4. $\{abcbc, abc, a\}$

Fragen...

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

1. $\{abca, bc\}$
2. $\{abcbc, abc\}$
3. $\{abcbc, abc\varepsilon, abc, a\varepsilon\}$
4. $\{abcbc, abc, a\}$ (3. ginge auch)

Fragen...

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

1. $\{\varepsilon, a, b, aa, ab, ba, bb\}$
2. $\{a, b, aa, ab, ba, bb\}$
3. $\{aa, ab, ba, bb\}$
4. $\{\{aa\}, \{ab\}, \{ba\}, \{bb\}\}$

Fragen...

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

1. $\{\varepsilon, a, b, aa, ab, ba, bb\}$
2. $\{a, b, aa, ab, ba, bb\}$
3. $\{aa, ab, ba, bb\}$
4. $\{\{aa\}, \{ab\}, \{ba\}, \{bb\}\}$

Fragen...

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

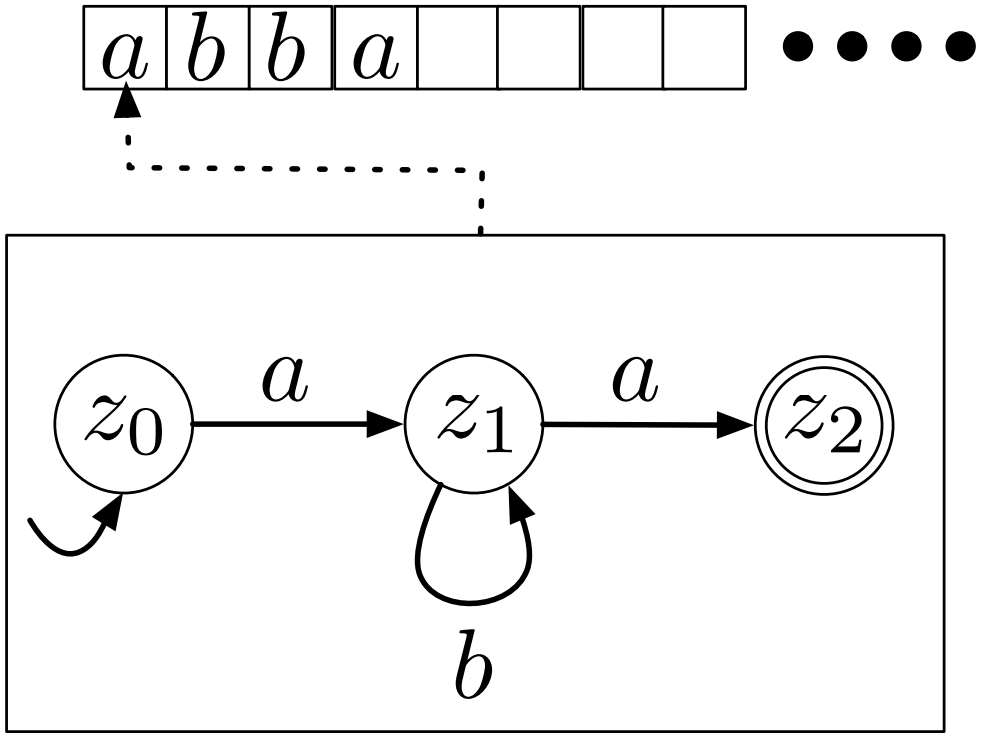
1. Menge aller endlichen Wörter aus a und b
2. Menge aller *ungeraden* Wörter aus a und b
3. Menge aller *geraden* Wörter aus a und b
4. Menge aller unendlichen Wörter aus a und b

Fragen...

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

1. Menge aller endlichen Wörter aus a und b
2. Menge aller *ungeraden* Wörter aus a und b
3. Menge aller *geraden* Wörter aus a und b
4. Menge aller unendlichen Wörter aus a und b

Endliche Automaten



Wie definieren wir das?

Der deterministische, endliche Automat

Definition (DFA)

Ein **deterministischer, endlicher Automat** (DFA)
(engl.: deterministic finite automata) (DFA)
ist ein 5-Tupel

$$A = (Q, \Sigma, \delta, q_0, F)$$

mit:

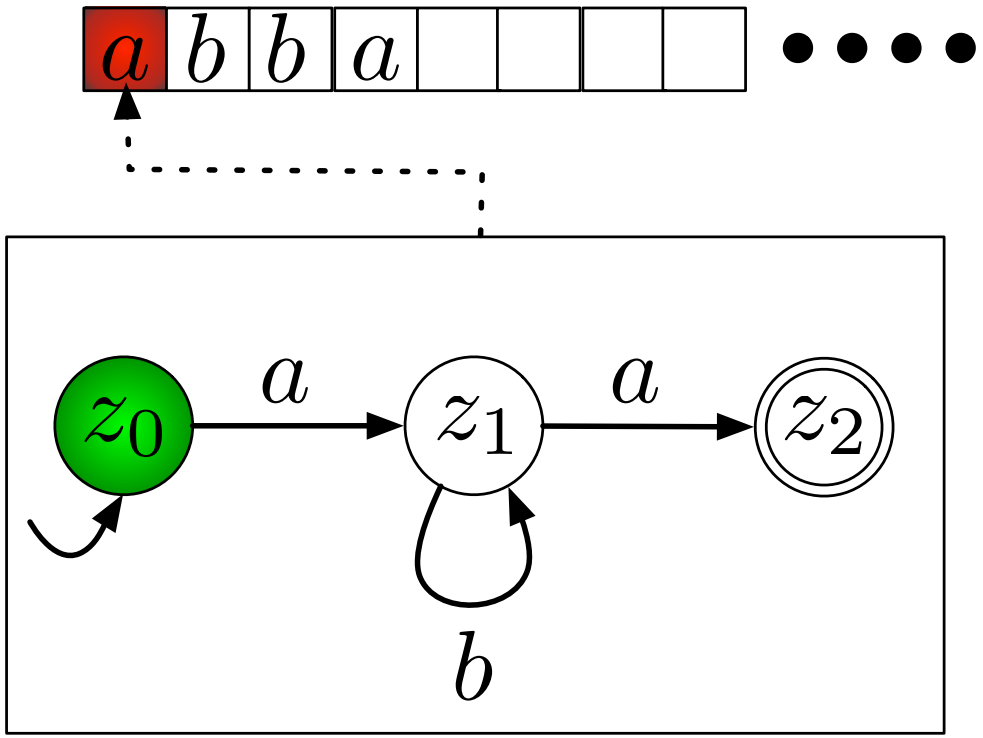
- Der endlichen Menge von *Zuständen* Q.
- Dem endlichen Alphabet Σ von *Eingabesymbolen*.
- Der *Überführungsfunktion* $\delta : Q \times \Sigma \to Q$.
- Dem *Startzustand* $q_0 \in Q$.
- Der Menge der *Endzustände* $F \subseteq Q$.

Arbeitsweise des DFA (informal)

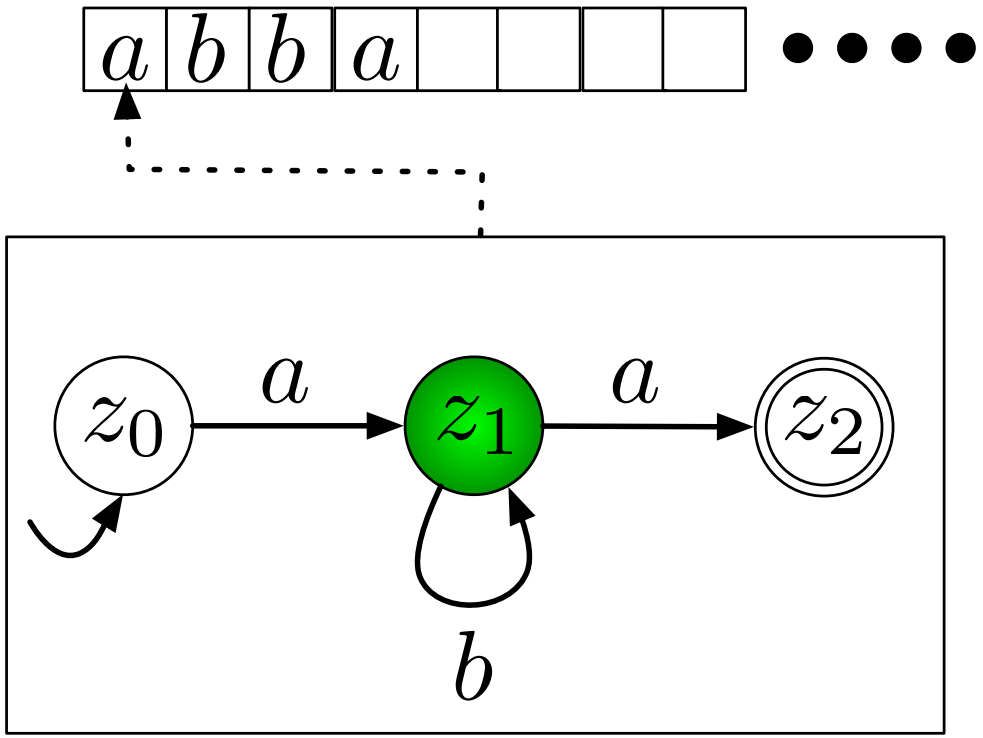
Erhält ein DFA ein Eingabewort $w \in \Sigma^*$, so

- beginnt er im Startzustand q_0.
- Beginnt w mit dem Symbol $x \in \Sigma$, so
- wird der Nachfolgezustand nun durch $\delta(q_0, x)$ bestimmt.
- Dies wird dann in dem nun aktuellen Zustand und mit dem Restwort fortgeführt.
- Das Wort w wird akzeptiert, wenn
  - w bis zum Ende gelesen werden kann **und**
  - der Automat dann in einem Endzustand ist.
- Für einen DFA soll die Überführungsfunktion (erstmal) vollständig sein, die erste Bedingung ist also immer erfüllt.

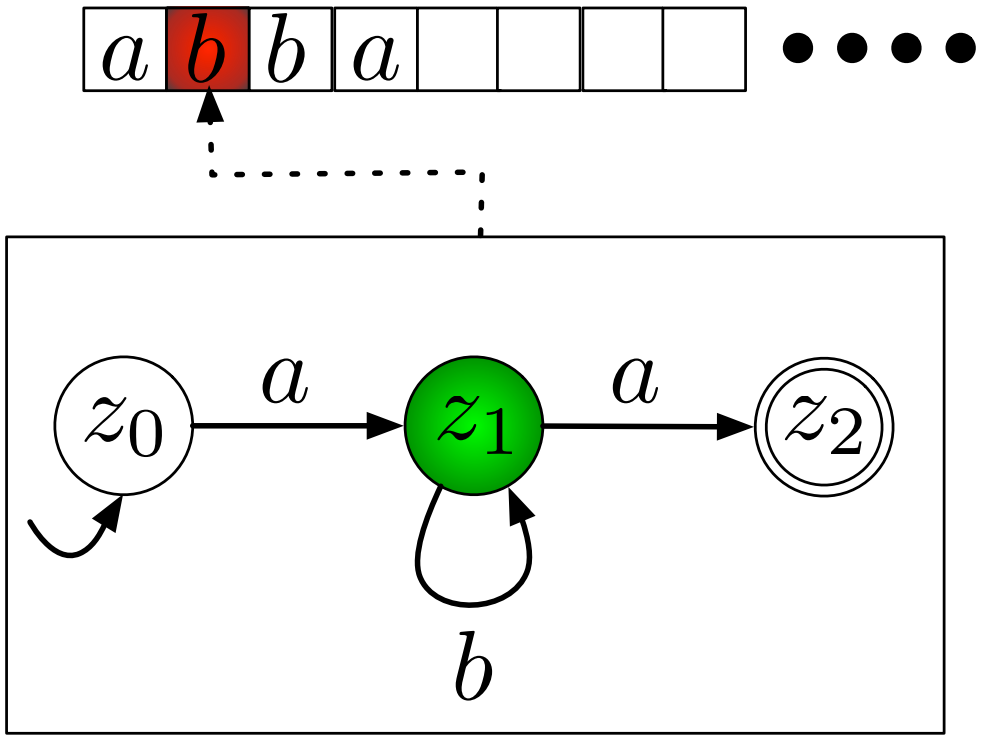
Ein informales Beispiel (abba)



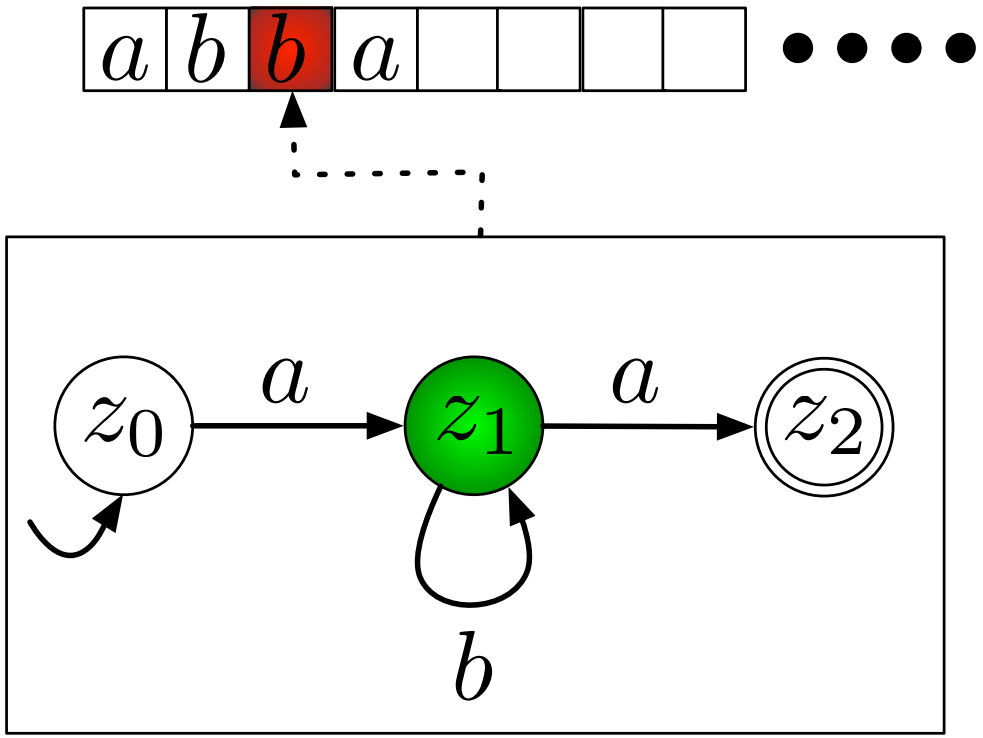
Ein informales Beispiel (abba)



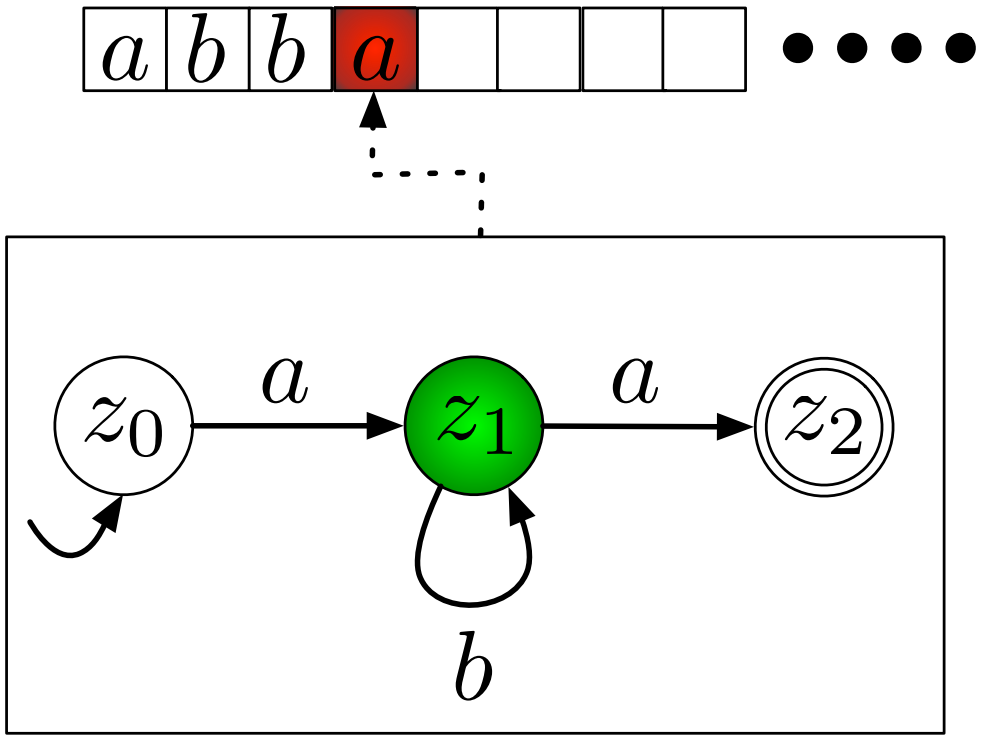
Ein informales Beispiel (abba)



Ein informales Beispiel (abba)



Ein informales Beispiel (abba)



Ein informales Beispiel (abba)