

Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2017/18

Weitere Infos unter

- <https://hu.berlin/ethi17>

bzw.

- <https://www.informatik.hu-berlin.de/de/forschung/gebiete/algorithmenII/Lehre/ws17/einftheo>

Übungen

Anmeldung

- bei Agnes (möglichst bald)
- und bei Moodle (wg. Punktevergabe und Zuordnung zu Übungsgruppen)
- Mails von Agnes und von Moodle werden standardmäßig an den CMS-Account gesendet (bitte regelmäßig checken)

Abgabe der Aufgabenblätter

- in der VL sowie bei Moodle und auf der VL-Webseite

Rückgabe

- in den Übungsgruppen

Übungen

Bearbeitung der Aufgaben

- in Gruppen von **zwei bis drei** Teilnehmern
- Teilnehmer müssen **nicht** in der gleichen Übungsgruppe sein
- bitte jede Aufgabe auf einem **separaten** Blatt bearbeiten, da diese getrennt abzugeben sind
- bitte auf **jedem Blatt** Folgendes angeben:
 - die Namen und **CMS-Benutzernamen** der Gruppenteilnehmer
 - den Namen der **Abgabegruppe** aus Moodle (z.B. AG123)
 - den Übungstermin, an dem Sie die korrigierten Blätter zurückerhalten möchten

Schein, Klausur, Skript

Scheinkriterien

- Lösen von $\geq 50\%$ der schriftlichen Aufgaben
- Lösen von $\geq 50\%$ der Multiple-Choice Aufgaben in Moodle

Klausur

- **Termin: wird demnächst bekannt gegeben**
- Zulassung nur mit Übungsschein

Skript

- wird wöchentlich ins Netz gestellt

Gibt es zum organisatorischen Ablauf noch Fragen?

Themen dieser VL:

- Welche Rechenmodelle eignen sich zur Lösung welcher algorithmischen Problemstellungen? **Automatentheorie**
- Welche algorithmischen Probleme sind überhaupt lösbar? **Berechenbarkeitstheorie**
- Welcher Aufwand ist zur Lösung eines geg. algorithmischen Problems nötig? **Komplexitätstheorie**

Themen der VL Algorithmen und Datenstrukturen:

- Wie lassen sich praktisch relevante Problemstellungen möglichst effizient lösen? **Algorithmik**

Themen der VL Logik in der Informatik:

- Mathem. Grundlagen der Informatik, Beweise führen, Modellierung **Aussagenlogik, Prädikatenlogik**

- Überblick über die wichtigsten Rechenmodelle (Automaten) wie z.B.
 - endliche Automaten
 - Kellerautomaten
 - Turingmaschinen
 - Registermaschinen
 - Schaltkreise
- Charakterisierung der Klassen aller mit diesen Rechenmodellen lösbaren Probleme durch
 - unterschiedliche Typen von formalen Grammatiken
 - Abschlusseigenschaften unter geeigneten Sprachoperationen
 - Reduzierbarkeit auf typische Probleme (Vollständigkeit)
- Grenzen der Berechenbarkeit
- Klassifikation wichtiger algorithmischer Probleme nach ihrer Komplexität

- Rechenmaschinen spielen in der Informatik eine zentrale Rolle
- Es gibt viele unterschiedliche math. Modelle für Rechenmaschinen
- Diese können sich in ihrer Berechnungskraft unterscheiden
- Die Turingmaschine (TM) ist ein universales Berechnungsmodell, da sie alle anderen bekannten Rechenmodelle simulieren kann
- Wir betrachten zunächst Einschränkungen des TM-Modells, die vielfältige praktische Anwendungen haben, wie z.B.
 - endliche Automaten (DFA, NFA)
 - Kellerautomaten (PDA, DPDA) etc.

- Der Begriff **Algorithmus** geht auf den persischen Gelehrten **Muhammed Al Chwarizmi** (8./9. Jhd.) zurück
- Ältester bekannter nicht-trivialer Algorithmus: **Euklidischer Algorithmus** zur Berechnung des größten gemeinsamen Teilers zweier natürlicher Zahlen (300 v. Chr.)
- Von einem Algorithmus wird erwartet, dass er bei jeder zulässigen **Problemeingabe** nach endlich vielen Rechenschritten eine korrekte **Ausgabe** liefert
- Die (maximale) Anzahl der Rechenschritte bei allen möglichen Eingaben ist nicht beschränkt, d.h. mit wachsender Eingabelänge kann auch die Rechenzeit beliebig anwachsen
- Die Beschreibung eines Algorithmus muss jedoch endlich sein
- Problemeingaben können Zahlen, Formeln, Graphen etc. sein
- Diese werden über einem Eingabealphabet Σ kodiert

Definition

- Ein **Alphabet** ist eine geordnete endliche Menge

$$\Sigma = \{a_1, \dots, a_m\}, \quad m \geq 1$$

von **Zeichen** a_i

- Eine Folge $x = x_1 \dots x_n \in \Sigma^n$ von Zeichen heißt **Wort**
- Die **Länge** von $x = x_1 \dots x_n \in \Sigma^n$ ist n und wird mit $|x|$ bezeichnet
- Die Menge aller Wörter über Σ ist

$$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n$$

- Das (einzige) Wort der Länge $n = 0$ ist das **leere Wort**, welches wir mit ε bezeichnen, d.h. $\Sigma^0 = \{\varepsilon\}$
- Jede Teilmenge $L \subseteq \Sigma^*$ heißt **Sprache** über dem Alphabet Σ

Beispiel

- Sprachen über Σ sind beispielsweise \emptyset , Σ^* , Σ und $\{\varepsilon\}$
- \emptyset enthält keine Wörter und heißt **leere Sprache**
- Σ^* enthält dagegen alle Wörter über Σ
- Σ enthält alle Wörter über Σ der Länge 1
- $\{\varepsilon\}$ enthält nur das leere Wort, ist also einelementig
- Sprachen, die genau ein Wort enthalten, werden auch als **Singletonsprachen** bezeichnet

- Da Sprachen Mengen sind, können wir sie bzgl. Inklusion vergleichen
- Zum Beispiel gilt $\emptyset \subseteq \{\varepsilon\} \subseteq \Sigma^*$
- Wir können Sprachen auch vereinigen, schneiden und komplementieren
- Seien A und B Sprachen über Σ . Dann ist
 - $A \cap B = \{x \in \Sigma^* \mid x \in A \wedge x \in B\}$ der **Schnitt** von A und B ,
 - $A \cup B = \{x \in \Sigma^* \mid x \in A \vee x \in B\}$ die **Vereinigung** von A und B , und
 - $\overline{A} = \{x \in \Sigma^* \mid x \notin A\}$ das **Komplement** von A

Konkatenation von Wörtern

Definition

Seien $x = x_1 \dots x_n$ und $y = y_1 \dots y_m$ Wörter. Dann wird das Wort

$$x \circ y = x_1 \dots x_n y_1 \dots y_m$$

als **Konkatenation** von x und y bezeichnet. Für $x \circ y$ schreiben wir auch einfach xy .

Beispiel

- Für $x = aba$ und $y = abab$ erhalten wir $xy = abaabab$ und $yx = abababa$
- Die Konkatenation ist also nicht kommutativ
- Allerdings ist \circ assoziativ, d.h. es gilt $x(yz) = (xy)z$
Daher können wir hierfür auch einfach xyz schreiben
- Es gibt auch ein neutrales Element, da $x\varepsilon = \varepsilon x = x$ ist
- Eine algebraische Struktur (M, \square, e) mit einer assoziativen Operation $\square : M \times M \rightarrow M$ und einem neutralen Element e heißt **Monoid**
- $(\Sigma^*, \circ, \varepsilon)$ ist also ein Monoid

Neben den Mengenoperationen Schnitt, Vereinigung und Komplement gibt es auch spezielle Sprachoperationen

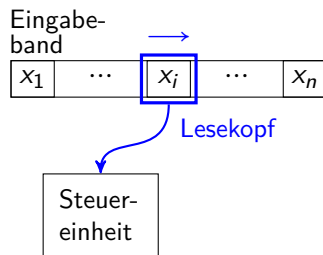
Definition

- Das **Produkt** (**Verkettung**, **Konkatenation**) der Sprachen A und B ist
$$AB = \{xy \mid x \in A, y \in B\}$$
- Ist $A = \{x\}$ eine Singletonsprache, so schreiben wir für $\{x\}B$ auch einfach xB
- Die **n -fache Potenz** A^n einer Sprache A ist induktiv definiert durch

$$A^n = \begin{cases} \{\varepsilon\}, & n = 0, \\ A^{n-1}A, & n > 0 \end{cases}$$

- Die **Sternhülle** von A ist $A^* = \bigcup_{n \geq 0} A^n$
- Die **Plushülle** von A ist $A^+ = \bigcup_{n \geq 1} A^n = AA^*$

- Ein einfaches Rechenmodell zum Erkennen von Sprachen ist der endliche Automat:



- Ein endlicher Automat
 - nimmt zu jedem Zeitpunkt genau einen von endlich vielen Zuständen an
 - macht bei Eingaben der Länge n genau n Rechenschritte und
 - liest in jedem Schritt genau ein Eingabezeichen

Definition

- Ein **endlicher Automat** (kurz: **DFA**; *Deterministic Finite Automaton*) wird durch ein 5-Tupel $M = (Z, \Sigma, \delta, q_0, E)$ beschrieben, wobei
 - $Z \neq \emptyset$ eine **endliche** Menge von **Zuständen**
 - Σ das **Eingabealphabet**,
 - $\delta: Z \times \Sigma \rightarrow Z$ die **Überföhrungsfunktion**
 - $q_0 \in Z$ der **Startzustand** und
 - $E \subseteq Z$ die Menge der **Endzustände** ist
- Die von M **akzeptierte** oder **erkannte Sprache** ist

$$L(M) = \left\{ x_1 \dots x_n \in \Sigma^* \mid \begin{array}{l} \text{es gibt } q_1, \dots, q_{n-1} \in Z, q_n \in E \text{ mit} \\ \delta(q_i, x_{i+1}) = q_{i+1} \text{ f\u00fcr } i = 0, \dots, n-1 \end{array} \right\}$$

- Eine Zustandsfolge q_0, q_1, \dots, q_n hei\u00dft **Rechnung** von $M(x_1 \dots x_n)$, falls $\delta(q_i, x_{i+1}) = q_{i+1}$ f\u00fcr $i = 0, \dots, n-1$ gilt
- Sie hei\u00dft **akzeptierend**, falls $q_n \in E$ ist, und andernfalls **verwerfend**

Frage

Welche Sprachen lassen sich durch endliche Automaten erkennen und welche nicht?

Definition

Eine von einem DFA akzeptierte Sprache wird als **regulär** bezeichnet. Die zugehörige Sprachklasse ist

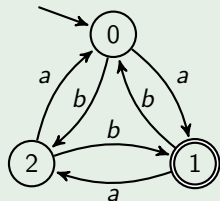
$$\text{REG} = \{L(M) \mid M \text{ ist ein DFA}\}$$

Beispiel

Sei $M_3 = (Z, \Sigma, \delta, 0, E)$ ein DFA mit $Z = \{0, 1, 2\}$, $\Sigma = \{a, b\}$, $E = \{1\}$ und der Überföhrungsfunktion

δ	0	1	2
a	1	2	0
b	2	0	1

Graphische Darstellung:



Endzustände werden durch einen doppelten Kreis und der Startzustand wird durch einen Pfeil gekennzeichnet

Frage: Welche Wörter akzeptiert M_3 ?

- $w_1 = aba$? Ja (akzeptierende Rechnung: 0, 1, 0, 1)
- $w_2 = abba$? Nein (verwerfende Rechnung: 0, 1, 0, 2, 0)

Behauptung

Die von M_3 erkannte Sprache ist

$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}, \text{ wobei}$$

- $\#_a(x)$ die Anzahl der Vorkommen von a in x bezeichnet und
- $i \equiv_m j$ (in Worten: i ist kongruent zu j modulo m) bedeutet, dass $i - j$ durch m teilbar ist

Beweis der Behauptung durch Induktion über die Länge von x

Wir betrachten zunächst das Erreichbarkeitsproblem für DFAs