Automaten und Formale Sprachen

SoSe 2007 in Trier

Henning Fernau
Universität Trier
fernau@informatik.uni-trier.de

Automaten und Formale Sprachen

Gesamtübersicht

- Organisatorisches
- Einführung
- Endliche Automaten und reguläre Sprachen
- Kontextfreie Grammatiken und kontextfreie Sprachen
- Chomsky-Hierarchie

Organisatorisches

Vorlesung FR 12-14 im HS 13
Vorschlag: 12.25-13.55 (mensafreundlicher)

Übung vierzehntägig MO 14-16 oder MI 14-16 (Raible)
Beginn in der dritten Vorlesungswoche;
d.h.: Ausgabe des ersten Übungsblattes in der zweiten Vorlesungswoche

Tutorensprechstunde DO 14.15-15.15 im H 407
Assistentensprechstunde DI 9.15-10.15 im H 413
Dozentensprechstunde MO 13-15 in meinem Büro (4. Stock)

Scheinkriterien Es wird ein benoteter Schein vergeben nach folgenden Regeln.

- **Abgabe von Hausaufgaben.** Diese sollte in Gruppen zu 2-3 Personen erfolgen. Abgabe ist in der Regel “kurz vor” Übungsbeginn der Montagsübungen im mit AFS beschrifteten Kasten im 4. Stock vor dem Sekretariat von Prof. Näher.
  Die Lösungen sind handschriftlich anzufertigen; weder Schreibmaschinen- noch Computerausdrucke werden akzeptiert, erst recht keine Kopien.
  Eine Woche später (in der Regel) werden die korrigierten und “bepunkteten” Hausaufgaben wieder zurckgegeben (ebenfalls vor den Übungen).

- **Teilnahme an einer Abschlussklausur.** Deren Termin wird noch bekanntgegeben.

Notenberechnung

Hat ein Kandidat h Prozent der Hausaufgabenpunkt erworben und k Prozent der Klausurpunkte, so ergibt sich die gewichtete Gesamtprozentzahl g gemäß: $\mathrm{g} = 0,4 \cdot \mathrm{h} + 0,6 \cdot \mathrm{k}$.
Gilt $\mathrm{k} < 35$, so erhält der Kandidat keinen Schein (also: nicht bestanden).
Andernfalls erfolgt die Ermittlung der Zensur nach folgender Tabelle:

$\geq 86\%$	$\geq 72\%$	$\geq 58\%$	$\geq 44\%$	$< 44\%$
1	2	3	4	nicht bestanden

Kleiner Werbespot

Hinweis: Im Proseminar: Alles Logo ? (Logik für Informatiker) sind noch Plätze für Kurzentschlossene frei.

Bitte umgehend bei Interesse anmelden !

Veranstaltungsform: Blockform nach Pfingsten

Einführung 1: Einordnung

- AFS und “Berechenbarkeit und Komplexität” (BK) (sowie teilweise Diskrete Strukturen und Logik) werden mancherorts zu einer oder zwei “4+2-Vorlesungen” “Einführung in die Theoretische Informatik” zusammengefasst.

- AFS liefert Grundlagen für bzw. ist verwandt mit:
  Compilerbau
  Textverarbeitungsalgorithmen
  computergesttzte Linguistik / linguistische Datenverarbeitung
  Schaltkreisentwurf / Hardwarebeschreibung
  allgemein: formale Methoden in Spezifikation
  formale Beschreibung von Algorithmen, z.B. VL “Lernalgorithmen”

Einführung 2: Motivation

AFS und BK untersucht die Frage:

Was ist die Natur einer Berechnung / eines Algorithmus ?

oder

Was ist ein “Rechner” (Computer) als mathematisches Objekt ?

Die Vorlesungen beschäftigen sich also mit den **absoluten Grundbegriffen der Informatik** als “Computer Science.”
Hoffnung: Es müsen Fragen der folgenden Art beantwortet werden:

- Was können Rechner prinzipiell (Berechenbarkeitsfragen) ?
- Was können Rechner “effizient” (d.h. schnell oder mit “sparsamem Speicher”) (AFS und Komplexitätstheorie) ?
- Wie können Rechner mathematisch modelliert werden ?
- Wie können Algorithmen in einer Weise notiert werden, dass man über sie mathematisch argumentieren kann ? (Mein Programm ist besser, weil . . .)

Einführung 3: Literatur

E. Kinber / C. Smith: Theory of Computing. A Gentle Introduction. Prentice Hall.

U. Schöning: Theoretische Informatik kurz gefasst. BI / Spektrum.

Es gibt zahlreiche gute Skripten im Internet, z.B.: Skripten zur Vorlesung Informatik-B2 an der Universität Duisburg-Essen (Prof. Luther, Prof. Hertling)

Wie Sie auch bei anderen Veranstaltungen lernen werden, unterscheiden sich die Formalisierungen oft (leider) in manchen Einzelheiten von Buch zu Buch; am besten suchen Sie nach dem Buch, das Ihrem Geschmack am nächsten kommt. Ich werde mich vornehmlich am Kinber / Smith orientieren.

Einführung 4: Was ist eine Sprache ?

Eine Menge Σ heißt *Alphabet*, falls Σ eine endliche, nicht-leere Menge ist, i.Z.: $|\Sigma| < \infty$ und $\Sigma \neq \emptyset$.

Kurz gesagt: Eine Sprache L (über Σ) ist eine Teilmenge des von der Menge Σ frei erzeugten Monoids, i.Z.: $L \subseteq \Sigma^*$.

Alles klar ?!

Was ist ein Monoid ?

Eine Struktur $(M, \circ, e)$ heißt *Monoid* gdw.:

- M ist eine Menge und $\circ$ ist eine zweistellige Operation (*Verknüpfung*) auf M, d.h., $\circ$ kann aufgefasst werden als Abbildung $\circ : M \times M \to M$ (*Abgeschlossenheit* von M unter $\circ$).

- $\circ$ ist *assoziativ*, d.h.: für alle x, y, z aus M gilt: $(x \circ y) \circ z = x \circ (y \circ z)$, i.Z.:

$$\forall x, y, z \in M : (x \circ y) \circ z = x \circ (y \circ z).$$

- $e \in M$ ist *neutrales Element* von $\circ$, d.h.:

$$\forall x \in M : e \circ x = x \circ e = x.$$

Eine *Halbgruppe* ist eine Struktur $(H, \circ)$, wobei $\circ$ eine assoziative Verknüpfung auf H ist.

Satz: Eine Struktur $(M, \circ, e)$ ist ein Monoid gdw. $(M, \circ)$ eine Halbgruppe ist und e ein neutrales Element von $\circ$ ist.

Lemma: In einer Halbgruppe gibt es höchstens ein neutrales Element.

Beweis: Wenn die Aussage falsch wäre, so gäbe es zwei Elemente e_1 und e_2, die die Eigenschaft eines neutralen Elements erfüllen. Daher gilt:

$$e_1 = e_1 \circ e_2 = e_2 \circ e_1 = e_2.$$

Dies widerspricht der Annahme.

Dies ist ein Beispiel für einen *Widerspruchsbeweis*.

Beispiele

Es sei $\mathbb{N}$ die Menge der natürlichen Zahlen, d.h.:

$$\mathbb{N} = \{0, 1, 2, 3, 4, 5, \ldots\}.$$

Hinweis: $\mathbb{N}$ umfasst "im Wesentlichen" alle auf Rechnern tatsächlich behandelbaren Objekte; insbesondere "reelle Zahlen" sind Fiktion.

Beispiel: $(\mathbb{N}, \max, 0)$ ist ein Monoid.

Beispiel: $(\mathbb{N}, +, 0)$ ist ein Monoid.

Beispiel: $(\mathbb{N}, \min)$ ist eine Halbgruppe ohne neutrales Element. Man könnte allerdings ein neutrales Element zu $\mathbb{N}$ künstlich hinzufügen. Nennen wir es ∞, so bildet $(\mathbb{N} \cup \{\infty\}, \min, \infty)$ ein Monoid.

Endliche Monoide

Ein Monoid $(M, \circ, e)$ heißt endlich gdw. $|M| < \infty$.

Endliche Monoide kann man gut mit einer *Verknüpfungstafel* angeben.

Beispiel: Betrachte $M = \mathbb{Z}_m = \{0, 1 \ldots, m-1\}$ mit der Addition $+$ modulo m als Verknüpfung. Für $m = 3$ ergibt sich folgende Tafel:

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Konvention: “Zeile mal Spalte” (egal bei Kommutativität)

Ablesen von Eigenschaften anhand einer Verknüpfungstafel

Da die mit der 0 indizierte Zeile bzw. Spalte die Grundelemente in der “richtigen Reihenfolge” aufzählt, ist 0 das neutrale Element von $+$.

Die Abgeschlossenheit ist offenbar (warum?),

und die Assoziativität kann man durch erschöpfende Analyse aller Fälle nachrechnen, z.B.:

$$(1+2)+2 = 0+2 = 2 = 1+1 = 1+(2+2).$$

Mengenoperationen als Halbgruppenoperationen

Es sei X eine Menge. Dann bezeichnet 2^X *Potenzmenge* von X, d.i. die Menge der Teilmengen von X.

Beispiel: Ist $X = \{0, 1, 2\}$, so ist $2^X = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, X\}$.

Lemma: Gilt $|X| < \infty$, so ist $|2^X| = 2^{|X|}$.

Satz: Für jede Menge X sind

$$(2^X, \cup, \emptyset)$$

und

$$(2^X, \cap, X)$$

Monoide.

Abbildungen

Es seien X, Y nicht-leere Mengen.
Eine *Abbildung* (oder *Funktion*) $f : X \to Y$ ist eine Vorschrift, die jedem Element aus X höchstens ein Element aus Y zuordnet.
In dieser Vorlesung werden Abbildungen stets *total* sein, d.h., jedem Element aus X wird genau ein Element aus Y zugeordnet.
In diesem Sinne ist dann Y^X die Menge aller Abbildungen von X nach Y.

Lemma: Gilt $|X|, |Y| < \infty$, so ist $|Y^X| = |Y|^{|X|}$.

Hinweis: Es macht oft Sinn, n mit $\mathbb{Z}_n$ zu identifizieren...

Hintereinanderausführung von Abbildungen

Sind $f : X \to Y$ und $g : Y \to Z$ Abbildungen, so ordnet
ihre *Hintereinanderausführung* (oder *Komposition*) $g \circ f : X \to Z$
(beachte das "Vertauschen" von g und f in der Schreibweise)
einem $x \in X$ dasjenige Element $z \in Z$ zu, das sich durch $z = g(f(x))$ ergibt, i.Z.:
$g \circ f : X \to Z, x \mapsto g(f(x))$.

Satz: $(X^X, \circ, id_X)$ ist ein Monoid, wobei $id_X(x) = x$ für alle $x \in X$ gilt (*Identität*).

Kartesische Mengenprodukte

Es seien X, Y Mengen.
Das *kartesische Produkt* oder *Mengenprodukt* von X und Y beinhaltet diejenigen *geordneten Paare* (x, y) mit $x \in X$ und $y \in Y$, i.Z.:

$$X \times Y = \{(x, y) \mid x \in X \wedge y \in Y\}.$$

Lemma: Gilt $|X|, |Y| < \infty$, so gilt $|X \times Y| = |X| \cdot |Y|$.

Spezialfall: $X^2 = X \times X$.

Beispiel: $X = \mathbb{R}$, d.h.: $\mathbb{R}^2$ beschreibt die kartesische *Ebene*.

Mengenpotenzen allgemein

Ist X eine Menge und $n \in \mathbb{N}$, $n > 1$, so definiere: $X^1 := X$ und $X^n = X \times X^{n-1}$.
Dies ist ein Beispiel für eine *rekursive Definition*.
Ein Element aus X^n heißt auch *Folge der Länge n über X*.
Ist X ein Alphabet, so nennen wir eine Folge auch ein *Wort* (der Länge n).

Lemma: Gilt $|X| < \infty$, so ist $|X^n| = |X|^n$ für beliebige $n \in \mathbb{N}, n \geq 1$.

Verkettung: eine Verknüpfung auf X^n ?

Beispiel: Nach der rekursiven Definition z.B. für $\{a, b, c\}^3$ gilt:

$$(a, (a, b)) \in \{a, b, c\}^3.$$

Die Folge $(a, (a, b))$ der Länge drei entsteht durch *Verkettung* (oder *Hintereinanderschreiben*, *(Kon-)Katenation*) des Wortes a der Länge eins mit dem Wort (a, b) der Länge zwei, und letzteres wieder durch Verkettung der Wörter a und b der Länge eins.

Problem: X^n ist bezüglich der $\cdot$ geschriebenen Operation “Verkettung” nicht abgeschlossen.

Verkettung: eine Verknüpfung auf X^+ !

Lösung: Betrachte $X^+ := \bigcup_{n \geq 1} X^n$.

Satz: $(X^+, \cdot)$ ist eine Halbgruppe, die so genannte *frei erzeugte Halbgruppe (über X)*.

Im Beweis benötigt man: X^n und $X^{n-\ell} \times X^\ell$ bezeichnen für jedes $1 \leq \ell < n$ "dasselbe."

(Dies wäre evtl. leichter ersichtlich, hätten wir X^n als $X^{\mathbb{Z}_n}$ definiert...)

Deshalb (Assoziativität) kann man auch die vielen Klammern bei der Notation von Wörtern fortlassen:

Wir schreiben also aab statt $(a, (a, b))$.

Beispiel: $LASS \in \{A, D, S, L\}^4$ und $DAS \in \{A, D, S, L\}^3$, also gilt für die Konkatenation $LASSDAS \in \{A, D, S, L\}^7$.

Verkettung: eine Verknüpfung auf X^* !

Problem: X^+ ist kein Monoid ?!

Lösung: Betrachte $X^* := \bigcup_{n \geq 0} X^n = X^+ \cup \{\lambda\}$. λ (andere Notationen: ϵ oder e) ist *das leere Wort*, formal ein künstlich hinzugefügtes neutrales Element.

Satz: $(X^*, \cdot, \lambda)$ ist ein Monoid, das so genannte *frei erzeugte Monoid (über X)*.

Morphismen I

Allgemein bezeichnet ein *(Homo-)Morphismus* eine *strukturerhaltende Abbildung*.

Für Halbgruppen gilt daher: Sind $(H, \circ)$ und $(G, \square)$ Halbgruppen, so ist eine Abbildung $h : H \to G$ ein *(Halbgruppen-)Morphismus* gdw.

$$\forall x, y \in H : h(x \circ y) = h(x) \square h(y).$$

Satz: Sind $(H, \circ)$ und $(G, \square)$ Halbgruppen und $h : H \to G$ ein Morphismus, so ist $(\{h(x) \mid x \in H\}, \square)$ eine Halbgruppe. Besitzt $(H, \circ)$ darüber hinaus ein neutrales Element $e \in H$, so ist $h(e)$ neutrales Element von $(\{h(x) \mid x \in H\}, \square)$.

Morphismen II

Für Monoide gilt: Sind $(H, \circ, e)$ und $(G, \Box, 1)$ Monoide, so ist eine Abbildung $h : H \to G$ ein *(Monoid-)Morphismus* gdw.

$$\forall x, y \in H : h(x \circ y) = h(x) \Box h(y)$$

sowie $h(e) = 1$.

Es ist notwendig, die zweite Bedingung auch zu prüfen, wie folgendes Beispiel lehrt: Beispiel: Betrachte die Monoide $(\mathbb{Z}_k, \min, k-1)$ für $k > 0$; die identischen Abbildungen $h_k : x \mapsto x$ können als Halbgruppenmorphismen $(\mathbb{Z}_k, \min) \to (\mathbb{Z}_{k+1}, \min)$ aufgefasst werden. Das neutrale Element $k-1$ von $(\mathbb{Z}_k, \min)$ wird aber nicht auf das neutrale Element k von $(\mathbb{Z}_{k+1}, \min)$ abgebildet.

Satz: Sind $(H, \circ, e)$ und $(G, \Box, 1)$ Monoide und $h : H \to G$ ein Morphismus, so ist $(\{h(x) \mid x \in H\}, \Box, 1)$ ein Monoid.

Morphismen III

Beispiel: ℓ ordne einem Wort $w \in \Sigma^*$ dasjenige n zu, für welches $w \in \Sigma^n$ gilt. Die so definierte Abbildung $\ell : \Sigma^* \to \mathbb{N}$ ist ein Monoidmorphismus von $(\Sigma^*, \cdot, \lambda)$ nach $(\mathbb{N}, +, 0)$.

Beispiel: $(\mathrm{mod}\ m)$ ordnet einer natürlichen Zahl n diejenige Zahl aus $\mathbb{Z}_m$ zu, die sich als Rest von n beim Teilen durch m ergibt. Für jedes $m \in \mathbb{N}, m > 0$, ist $(\mathrm{mod}\ m) : \mathbb{N} \to \mathbb{Z}_m$ ein Monoidmorphismus von $(\mathbb{N}, +, 0)$ nach $(\mathbb{Z}_m, +, 0)$.

Satz: Die Komposition von zwei Morphismen liefert einen Morphismus.

Beispiel: Für jedes $m \in \mathbb{N}, m > 0$, und jedes Alphabet Σ ist $\ell_m := (\ (\mathrm{mod}\ m) \circ \ell)$ ein Monoidmorphismus.

Reguläre Sprachen

Eine Sprache $L \subseteq \Sigma^*$ heißt *regulär* gdw. es ein endliches Monoid $(M, \circ, e)$, einen Monoidmorphismus $h : (\Sigma^*, \cdot, \lambda) \to (M, \circ, e)$ sowie eine endliche Menge $F \subseteq M$ gibt mit

$$L = \{w \in \Sigma^* \mid h(w) \in F\}.$$

Beispiel: Die Sprache $L = \{w \in \{a, b\}^* \mid \ell(w) \text{ ist gerade }\}$ ist regulär.

Beweis: Betrachte den Morphismus ℓ_2 und die endliche Menge $\{0\} \subset \mathbb{Z}_2$.