Grundlagen Theoretischer Informatik I

SoSe 2011 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Organisatorisches

Vorlesungen FR 10.10-11.50 im HS 13

Übungsbetrieb ab dem nächsten Mal wieder drei Gruppen (BEd)

Dozentensprechstunde DO 13-14 in meinem Büro H 410 (4. Stock)

Mitarbeitersprechstunde (Daniel Meister) DO 13-14 H 413

Tutorensprechstunde MO 13-14 H 407; alternativ gti1@informatik.uni-trier.de

Zwischenklausur in der VL-Zeit am 10.6.

Sternchenaufgaben: ab dem nächsten Übungsblatt; diese bringen für Nicht-Kernfach-Informatiker evtl. Bonuspunkte

Grundlagen Theoretischer Informatik I Gesamtübersicht

- Organisatorisches; Einführung
- Logik & Beweisverfahren
- Mengenlehre
- reguläre Sprachen

Folgen sind spezielle Funktionen

Erinnerung: $\mathbb{N}$: Menge der natürlichen Zahlen; $[n] := \{m \in \mathbb{N} \mid m < n\}$.

Eine *unendliche Folge* f mit *Gliedern* aus einer Menge M ist eine Abbildung $f : \mathbb{N} \to M$.

Eine *endliche Folge* f mit *Gliedern* aus einer Menge M ist eine Abbildung $f : [n] \to M$ für ein $n \in \mathbb{N}$.

Folgen dienen zum *Auflisten*, *Abzählen* oder *Nummerieren* von (einigen) Elementen einer Menge. Eine surjektive Folge liefert also eine *vollständige Auflistung* des Wertebereichs.

Folgen: Beispiele

Beispiel: *Listenschreibweise*:

$$\left(1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots, \frac{1}{2^m}, \ldots\right)$$

beschreibt die Folge $f : \mathbb{N} \to \mathbb{Q},\ i \mapsto 2^{-i}$.

Beispiel: Die ganzen Zahlen lassen sich vollständig auflisten. Betrachte

$$f : \mathbb{N} \to \mathbb{Z},\ n \mapsto \begin{cases} -\frac{n}{2}, & \text{falls } n \text{ gerade} \\ \frac{n+1}{2}, & \text{sonst} \end{cases}$$

Das Cantorsche Abzählungsschema I

Satz: Es gibt eine vollständige Auflistung von $\mathbb{N} \times \mathbb{N}$.

Beweis: Betrachte das folgende Schema:

```
/ 0  1  2  3  4  5  ...
0 0  2  5  9  14 20 ...
1 1  4  8  13 19 26 ...
2 3  7  12 18 25 33 ...
3 6  11 17 24 32 41 ...
4 10 16 23 31 40 50 ...
5 15 22 30 39 49 60 ...
| |  |  |  |  |  |  ...
```

Das Cantorsche Abzählungsschema I

Das Schema lässt sich auch formal notieren als *Cantorsche Paarfunktion*:

$$\langle i, j \rangle = (i + j)(i + j + 1)/2 + j$$

Also:
$\langle 0, 0 \rangle = 0$, $\langle 1, 0 \rangle = 1$, $\langle 0, 1 \rangle = 2$, $\langle 2, 0 \rangle = 3$, ...

Folgerung: Für jedes n ist $\mathbb{N}^n = \underbrace{\mathbb{N} \times \cdots \times \mathbb{N}}_{n \text{ mal}}$ vollständig auflistbar.

Beispiel: $n = 3$ (Tripel): $\langle i, j, k \rangle = \langle \langle i, j \rangle , k \rangle$.

Das Cantorsche Abzählungsschema II

Satz: Die rationalen Zahlen lassen sich vollständig auflisten.

Beweis: Dies lässt sich zeigen, indem man die Brüche folgendermaßen in einem zweidimensionalen Schema anordnet und dann den vorigen Satz anwendet:

```
1/1 1/2 1/3 1/4 1/5 ...
2/1 2/2 2/3 2/4 2/5 ...
3/1 3/2 3/3 3/4 3/5 ...
4/1 4/2 4/3 ...
5/1 5/2 ...
...
```

Erinnerung: Äquivalenzklassendefinition der rationalen Zahlen

Rekursiv definierte Folgen

Beispiel: Die Folge $(a_0, a_1, a_2, \dots) = (a_n)_{n\in\mathbb{N}}$ ist gegeben durch:

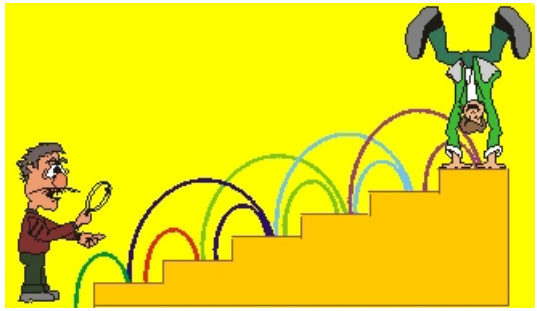
$$\begin{aligned} a_0 &= 1, \\ a_n &= a_{n-1} \cdot n, \text{ für } n > 0. \end{aligned}$$

Schreibweise: $\prod_{j\in[n]} b_j$ bezeichnet das Produkt aller Zahlen der endlichen Folge $(b_0, \dots, b_{n-1})$. Das leere Produkt wird als Eins interpretiert.

Satz: Für die oben definierte Folge a_n gilt: $a_n = \prod_{j\in[n]\setminus\{0\}} j$.
$f(n) = a_n$ heißt auch *Fakultätsfunktion*; Schreibweise: $n!$

Beweis: eine leichte Induktion

Rekursiv definierte Folgen: Treppensteigen



Bei jeder Stufe kann man sich die Frage stellen:
Nehme ich eine Stufe oder überspringe ich eine Stufe?
Die erste Stufe muß auf jeden Fall betreten werden.
Frage: Auf wieviel verschiedene Arten f_n kann man nun eine n-stufige Treppe heraufgehen?

Versuchen wir (an der Tafel), eine Tabelle dafür aufzustellen.

Rekursiv definierte Folgen: Treppensteigen

Finden wir ein *Bildungsgesetz* ?

Für $n \geq 2$ gibt es zwei Möglichkeiten, eine n-stufige Treppe zu erklimmen:
– entweder hatten wir einen Schritt von einer $(n-1)$-stufigen Treppe aus gemacht
– oder zwei Stufen auf einmal von einer $(n-2)$-stufigen Treppe aus genommen.

$\rightsquigarrow f_n = f_{n-1} + f_{n-2}$; Sonderfälle: $f(0) = 0$ und $f(1) = 1$.

Diese Folge kommt sehr häufig in der Natur und Kultur vor und wird gemeinhin die Folge der *Fibonacci-Zahlen* genannt !
Mehr über Leonardo Fibonacci bei einem virtuellen Museumsbesuch.

Mächtigkeit von Mengen

Was bedeutet der Prozess des Zählens (Nummerierens) allgemein ?!
Für eine endliche Menge M könnte er in der (evtl sukzessiven) Konstruktion einer Bijektion von $[\mathrm{n}]$ auf M bestehen. Wir sagen dann auch, M habe n Elemente.

Allgemeiner heißen zwei Mengen A und B *gleichmächtig* gdw es gibt eine Bijektion f von A nach B.
Speziell heißt eine Menge A *abzählbar unendlich*, wenn sie mit $\mathbb{N}$ gleichmächtig ist. Eine Menge heißt *abzählbar*, wenn sie entweder endlich oder abzählbar unendlich ist.

Satz: Sämtliche Teilmengen von $\mathbb{N}$ sind abzählbar.

Abzählbare Mengen

Satz: Sei $(M_n)_{n\in\mathbb{N}}$ eine Folge paarweise disjunkter, endlicher, nichtleerer Mengen. Dann ist $\bigcup_{n\in\mathbb{N}} M_n$ abzählbar unendlich.

Beweis: Gesucht: Bijektion ϕ von $\mathbb{N}$ auf $M^* = \bigcup_{n\in\mathbb{N}} M_n$.

Es bezeichne m_n die Zahl der Elemente von M_n und $m_n^+ = \sum_{0\le j<n} m_j$.

Es gibt also Bijektionen $\phi_n : [m_n] \to M_n$.

Da für alle n gilt: $m_n > 0$, gibt es zu jedem $k \in \mathbb{N}$ eine eindeutig bestimmte Zahl $n(k) \in \mathbb{N}$ mit $m^+_{n(k)} \le k \le m^+_{n(k)+1}$.

Da die M_n paarweise disjunkt sind, ist $k \mapsto \phi_{n(k)}(k - m^+_{n(k)})$ die gesuchte Bijektion ϕ.

Folgerung: Sei $M \neq \emptyset$ eine endliche Menge. Die Menge aller endlichen Folgen, gebildet von Elementen aus M, ist abzählbar unendlich.

Überabzählbare Mengen sind Mengen, die nicht abzählbar sind.

Satz: $2^{\mathbb{N}}$ ist überabzählbar.

Beweis: (Cantors *Diagonalisierungsargument*; ein spezieller Widerspruchsbeweis)
Andernfalls gäbe es eine Bijektion $f : \mathbb{N} \to 2^{\mathbb{N}}$.
Betrachte $S := \{n \in \mathbb{N} \mid n \notin f(n)\}$.
Da $S \in 2^{\mathbb{N}}$, gibt es ein $s \in \mathbb{N}$ mit $f(s) = S$.
Wäre $s \in S$, so $s \notin S$ nach Def. von S.
Wäre $s \notin S$, so folgt nach Def. von S: $s \in S$.
Also folgt aus der Annahme der Existenz von f die Kontradiktion:
$s \in S \iff s \notin S$.
Daher ist die Annahme falsch.
Anschaulicher an der Tafel: Relationenmatrix für $\{(s,t) \in \mathbb{N} \times \mathbb{N} \mid t \in f(s)\}$.

Allgemeiner gilt: A und 2^A sind nicht gleichmächtig.

Überabzählbare Mengen: ein weiteres Beispiel

Es bezeichne $M^N := \{f \mid f : N \to M\}$.

Satz: $2^{\mathbb{N}}$ und $\{0,1\}^{\mathbb{N}}$ sind gleichmächtig.

Beweis: Zu jeder Menge $M \subseteq \mathbb{N}$ kann man seine *Indikatorfunktion* oder *charakteristische Funktion* $\chi_M : \mathbb{N} \to \{0,1\}$ definieren durch $\chi_M(x) = 1$ gdw. $x \in M$. Umgekehrt lässt sich jede Funktion $f : \mathbb{N} \to \{0,1\}$ als charakteristische Funktion der Menge $M_f = \{x \in \mathbb{N} \mid f(x) = 1\}$ auffassen.

Eine überraschende Folgerung für die Informatik

Programmtexte (in einer fixierten Programmiersprache) lassen sich (rein syntaktisch) als endliche Folgen über einer endlichen Menge (dem *Alphabet*) begreifen.

Folgerung: Die Menge aller Programmtexte ist abzählbar.

Programmtexte können zur Beschreibung von Funktionen $f : \mathbb{N} \to \{0, 1\}$ verwendet werden.

Folgerung: Nicht jede Funktion $f : \mathbb{N} \to \{0, 1\}$ kann durch einen Programmtext beschrieben werden.

$\leadsto$ “Computer können nicht alles.”

Die **Mächtigkeit einer Menge** $\mathcal{M}$ wird auch mit $|\mathcal{M}|$ bezeichnet.
Ist $\mathcal{M}$ endlich, so ist dies gerade die Anzahl der Elemente; dann schreibt man auch $\#\mathcal{M}$.

Wir können somit auch Mengen bzgl. ihrer Mächtigkeit vergleichen.
Lemma: Für ein gegebenes Universum $\mathcal{U}$ ist dieses eine Quasiordnung auf den Teilmengen von $\mathcal{U}$.

Formale Sprachen—eine Annäherung

Was ist eine Sprache ?

Eine *Sprache* ist eine Menge von Wörtern.

Das verschiebt unser Problem nur... Was ist ein Wort ?

Ein *Wort* ist eine Aneinanderreihung von Buchstaben.

Das verschiebt unser Problem nur noch weiter... Was ist ein Buchstabe ? Was ist eine Aneinanderreihung ?

Ein *Buchstabe* ist ein Element einer endlichen, nicht leeren Grundmenge, genannt *Alphabet*.

Eine *Aneinanderreihung* ist eine endliche Folge von Elementen einer Grundmenge.

Mengenpotenzen und Wörter

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 (bekanntes) 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 *Aneinanderreihung*, *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.

Wir lieben jedoch Operationen, bezüglich derer unsere Grundmenge abgeschlossen ist.

Verkettung: eine Verknüpfung auf X^+ !

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

Satz: Die Menge X^+ der Wörter beliebiger positiver Länge über dem Alphabet X ist bezüglich der Verkettung abgeschlossen. Überdies handelt es sich um eine assoziative Operation.

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 $(a,(a,b))$.

Verkettung: eine Verknüpfung auf X^+ !

Vornehmere Sprechweise: Eine Menge M zusammen mit einer Verknüpfung $\circ$ auf M, d.h., einer Abbildung $\circ : M \times M \to M$, heißt *Halbgruppe*, falls $\circ$ assoziativ ist.

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

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

Eine Struktur $(M, \circ, e)$ ist ein *Monoid* gdw. $(M, \circ)$ eine Halbgruppe ist und e ein neutrales Element von $\circ$ ist, also $\forall x \in M : x = x \circ e = e \circ x$.

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)*.

Formale Sprachen—eine Annäherung

Was ist eine Sprache ?
Eine *Sprache* ist eine Menge von Wörtern.

Das sollte jetzt auch mathematisch klar sein:

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.: $L \subseteq \Sigma^*$.

Ein **deterministischer endlicher Automat** oder DEA wird beschrieben durch ein Quintupel

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

wobei gilt:

Q: endliche Menge von *Zuständen* (Zustandsalphabet)

Σ: endliche Menge von *Eingabezeichen* (Eingabealphabet)

$\delta : Q \times \Sigma \to Q$: *Überführungsfunktion*

$q_0 \in Q$: *Anfangszustand*

$F \subseteq Q$: *Endzustände*

Überführungstafel

Ein endlicher Automat kann vollständig durch seine *Überführungstafel* beschrieben werden.

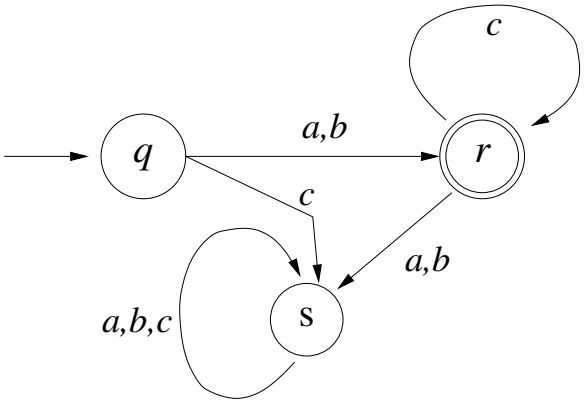
Beispiel: Betrachte:

δ	a	b	c
$\rightarrow$ q	r	r	s
r $\rightarrow$	s	s	r
s	s	s	s

In der ersten Zeile ist das Eingabealphabet beschrieben, in der ersten Spalte das Zustandsalphabet; der eingehende Pfeil kennzeichnet den Anfangszustand und der ausgehende den Endzustand (es könnten auch mehrere sein).

Automatengraph

Ein gerichteter, Σ-kantenbeschrifteter Graph mit Knotenmenge Q:



“Eingangspfeile” kennzeichnen den Startzustand und doppelte Umkreisungen die Endzustände.

Wie arbeitet ein DEA ?

Es sei $w = a_1 \dots a_n \in \Sigma^n$ das *Eingabewort* von A.
Die Arbeit von A auf w kann wie folgt beschrieben werden:

1. Setze $q = q_0$.

2. Für $x = a_1$ bis a_n tue:
   Setze $q := \delta(q, x)$

3. Akzeptiere w gdw. $q \in F$ gilt.

$L(A)$ bezeichnet die von A akzeptierte Sprache.

Die **von einem DEA $A = (Q, \Sigma, \delta, q_0, F)$ akzeptierte Sprache** kann man formal wie folgt beschreiben.

Ein Element aus $C = Q \times \Sigma^*$ heißt *Konfiguration* von A.
Definiere eine binäre Relation $\vdash_A$ auf C durch $(q, w) \vdash_A (q', w')$ gdw.
$\exists a \in \Sigma : w = aw'$ und $q' = \delta(q, a)$.
Die zweite Komponente einer Konfiguration ist die "übrige Eingabe".
$\vdash_A$ beschreibt den Konfigurationsübergang in einem Schritt.
Entsprechend beschreibt $\vdash_A^n$ n Schritte von A.
Daher können wir definieren:

$$L(A) = \{w \in \Sigma^* \mid \exists q \in F : (q_0, w) \vdash_A^* (q, \lambda)\}.$$

Was tut also “unser” Automat ?

$$L(A) = \{w \in \Sigma^* \mid (q, w) \vdash_A^* (r, \lambda)\}.$$

$$\{w \in \Sigma^* \mid (q, w) \vdash_A^* (r, \lambda)\} = \{u \in \Sigma^* \mid (q, u) \vdash_A (r, \lambda)\} \cdot \{v \in \Sigma^* \mid (r, v) \vdash_A^* (r, \lambda)\}$$

$$\Rightarrow$$

$$\{w \in \Sigma^* \mid (q, w) \vdash_A^* (r, \lambda)\} = \{a, b\}\{c^n \mid n \in \mathbb{N}\}$$