Universität
Augsburg
University

Diskrete Strukturen und Logik

Vorlesung 01: Grundlagen

Prof. Dr. Tobias Mömke
Universität Augsburg

2. November 2020

Diskrete Strukturen und Logik

Vorlesung

- Montags 12:15 – 13:45
- Mittwochs 8:15 – 9:45
- Zoom
- Wird aufgenommen, Links auf Digicampus
- Folien, Material und Verweise auf Buchkapitel: Digicampus
- Alle Klausurrelevanten Inhalte in Vorlesung
- Fragen: Direkt (Mikrofon) oder über Tweedback
  Nicht über Zoom Chat

Werkzeuge

- Tweedback
- Pingo:

pingo.coactum.de $\rightarrow$ 724653

- Anonymes Feedback

Globalübung

- Mittwochs 10:15 – 11:45
- Gleicher Zoom Link wie Vorlesung
- Klären von Fragen zur Vorlesung
- Zusätzliche Beispiele und Übungen
- Vertiefung von Themen
  (besserer Überblick, Zusammenhänge verstehen)

Übungen

- Es gibt 10 Übungen

  also 10 verschiedene Übungsgruppen
  Ü01 bis Ü10
- Die ersten vier Wochen:
  Alle Übungen online.
- Übungen 01 - 08: Möglicherweise später Präsenz
  (je nach Corona Situation)
- Studierende ohne Übung:
  Zunächst Übung 09 und 10
- Teilnahmelinks für Online Übungen 09 und 10:
  Zunächst verfügbar für alle, die in der Vorlesung sind.
- Details zu Abgaben:
  Steht spätestens Sonntag fest.

Übungsablauf

- Wöchentliche Übung
- Neue Aufgaben: Montags auf Digicampus
- Abgabe:
  - Bis spätestens **Montags 12:15**
    (also vor der Vorlesung)
  - Über **Digicampus** als **pdf-Datei**
    Vorzugsweise: LaTeX;
    Auf jeden Fall: Gut lesbar.
- Nicht lesbare Abgaben werden als falsch gewertet.
- Übungsgruppen:
  Abgaben in Gruppen der Größe 2 oder 3, vorzugsweise 3.
  Einergruppen: Nur in Ausnahmesituationen,
  bis Anschluss an andere Gruppe gefunden ist.

Übungsinhalt

- Wichtigstes Werkzeug, um Inhalte wirklich zu verstehen
- Übungen sind Training
  Analog zu Sprints und Intervalltraining
  um einen Marathon zu laufen
- Vier Übungen pro Übungsblatt, insgesamt 100 Punkte
- Teilweise Bonuspunkte – mit * gekennzeichnet
- Grundsätzlich Mischung aus leichten und schweren Aufgaben

Klausur

- Schriftliche Klausur
- Mindestens 50% der Übungspunkte **und** Klausur **bestanden**:
  Notenverbesserung um 0.3
  (Man kann **nicht** durch die Übungen bestehen.)
- Erlaubtes Material in Klausur:
  Handschriftlich beschriebenes Din A4 Blatt (beide Seiten)

- Nicht besser als 1.0

Diskrete Strukturen und Logik – Ingenieurinformatik:

- 6 LP anstatt 8 LP
- Nicht prüfungsrelevante Vorlesungen werden angekündigt
- Klausur: Weniger Aufgaben
- Empfehlung: Ganze Vorlesung besuchen

Diskrete Strukturen und Logik

Diskret

Dingen, die sich eindeutig voneinander abgrenzen lassen
Beispiel: Ganze Zahlen
Wörterbuch (Quelle:`https://de.wiktionary.org/wiki/diskret`):

[1] etwas geheim haltend; vertraulich
[2] Mathematik, Physik, Elektrotechnik, ohne Steigerung:
als einzelne Werte oder Komponenten unterscheidbar,
abgrenzbar, voneinander getrennt;
aus einer abzählbaren Menge von Elementen entnommen

Herkunft: von gleichbedeutend französisch discret

→ fr entlehnt, das auf mittellateinisch discretus
→ la „abgesondert", dem Partizip Perfekt von discernere
→ la „unterscheiden", zurückgeht[1]

Wir wollen nichts geheim halten, sondern benutzen die ursprünglichen Bedeutung.

Diskrete Mathematik

Unterscheidung:

Diskrete Mathematik vs Kontinuierliche Mathematik

Diskrete Mathematik

Unterscheidung:

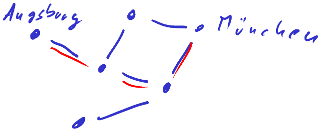
Diskrete Mathematik vs Kontinuierliche Mathematik

Beispiel:

Ganze Zahlen $\mathbb{N}$ vs Reelle Zahlen $\mathbb{R}$
Zählen von Möglichkeiten vs Differenzialgleichungen

Informatik: Zum größten Teil diskrete Objekte, z.B.

- Gespeicherte Bits
- Codes (Fehlererkennung, Kryptographie...)
- Graphen
- Laufzeit von Algorithmen



Informatik 3

Mathematische Logik

A	B	$A \Rightarrow B$
w	w	w
w	f	f
f	w	w
f	f	w

Formale Beschreibung von Zusammenhängen und Folgerungen.

Beispiele:

Wenn es regnet, dann ist die Straße nass.
Wenn $n > 10$ oder $m < 4$, dann breche die While-Schleife ab.

Warum wichtig?

- Fundament der Mathematik
- Schaltkreise (Hardware)
- Model Checking (Verifikation)
- Algorithmische Anwendungen
- Bedeutung in der Programmierung

Teilgebiete der Logik

- **Mengenlehre**
  Zermelo-Fraenkel-Mengenlehre (ZFC)
- **Beweistheorie**
  Formale Beweise
- **Modelltheorie**
  Aussagen über mit gewissen Struktur versehenen Mengen
- **Rekursionstheorie**
  Berechenbarkeit, Definierbarkeit

Diskrete Strukturen und Logik

Vorkurs Mathematik

Viele Grundlagen: Im Vorkurs Mathematik behandelt.

- Wichtig: Notation aus Vorkurs wird als bekannt vorausgesetzt
- Wir wiederholen Sätze, aber nicht Beweise aus dem Vorkurs
- Wir setzen voraus, dass grundlegende Prinzipien bekannt sind, beispielsweise Begriffe wie
  - Menge
  - Relation
  - Abbildung
  - Injektiv, Surjektiv, Bijektiv

Hier: Nur kurze Wiederholung, nicht so ausführlich wie im Vorkurs.

Vorkurs Skript ist in Digicampus vorhanden.

Diskrete Strukturen und Logik

Summation

Beispiele:

Harmonische Zahlen:

$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$

$$H_n = \sum_{i=1}^{n} \frac{1}{i}, \quad \ln n + 1/n \leq H_n \leq \ln n + 1$$

Euler Zahl

$e^x = n$

$x = \ln n$

Geometrische Reihe:

$$\sum_{i=1}^{\infty} \frac{1}{2^i} = \frac{1}{1 - 1/2} = 2$$

Gaußsche Summenformel:

$$\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$$

Rekursion

Beispiel: Fibonacci Zahlen

F_n: n-te Fibonacci Zahl für $n \in \mathbb{N}$ (Erinnerung: $\mathbb{N} = \{0, 1, 2, \ldots\}$ und $\mathbb{N}^* = \{1, 2, 3, \ldots\}$)

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \quad \text{falls } n \geq 2 \end{aligned}$$

Beispiel:

$F_0 = 0,\ F_1 = 1,\ F_2 = 1$

$F_3 = 2,\ F_4 = 3,\ F_5 = 5$

Wir werden später sehen:

$$F_n = \frac{1}{\sqrt{5}} \cdot \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right)$$

$n = 2$

$$\frac{1}{\sqrt{5}} \cdot \left(\left(\frac{1+\sqrt{5}}{2} \right)^2 - \left(\frac{1-\sqrt{5}}{2} \right)^2 \right)$$

$$= \frac{1}{\sqrt{5}} \cdot \left(\frac{\cancel{1} + 2\sqrt{5} \cancel{+5}}{4} - \frac{\cancel{1} - 2\sqrt{5} \cancel{+5}}{4} \right)$$

$$= \frac{1}{\sqrt{5}} \cdot \frac{\cancel{4}\sqrt{5}}{\cancel{4}} = 1$$

Erzeugende Funktion

Methode, um Werte einer Zählfunktion $f(n)$ als Koeffizienten einer Potenzreihe aufzufassen.

$$F(z) = \sum_{n=0}^{\infty} f(n) z^n$$

Hier: $F(z)$ ist die Erzeugende Funktion der Zählfunktion f.

Wir werden sehen: Wichtiges Hilfsmittel zum Lösen von Rekursionen.

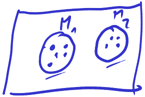
Asymptotische Analyse

Ausblick auf Informatik 3:

- Genaue Bestimmung von $f(n)$ oft sehr schwierig
- Näherung $a(n) \leq f(n) \leq b(n)$ oft deutlich leichter
- Asymptotische Analyse: Systematische Bestimmung von **Größenordnungen**
- Wichtig in der Analyse von Algorithmen

Diskrete Strukturen und Logik

Summenregel



$\{1,2,3\} = \{2,1,3\} = \{1,1,2,3\}$

M_1 und M_2 disjunkt: $M_1 \cap M_2 = \{\} = \emptyset$

Satz

Gegeben: *Menge S, disjunkte Teilmengen $S_1, S_2, \ldots, S_t$ so dass $\bigcup_{i=1}^{t} S_i = S$*
(Wir haben also eine Partition von S)
Es gilt:

$$|S| = \sum_{i=1}^{t} |S_i|$$

$S_i \cap S_j = \emptyset$

Beweis: Vorkurs 17h

Grundlage für viele Rekursionen.

Siehe z.B. Beweis im Vorkurs von

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

für $k \geq 1$.

Produktregel

$$\prod_{i=1}^{5} (i+1) = (1+1)\cdot(2+1)\cdot(3+1)\cdot(4+1)\cdot(5+1)$$

Satz

Sei $S = S_1 \times S_2 \times \cdots \times S_t$. Dann gilt

$a_1 \quad a_2 \quad a_t \qquad a_i \in S_i$

$$|S| = \prod_{i=1}^{t} |S_i|.$$

$(a_1, a_2, a_3, \ldots, a_t)$

Beweis: Vorkurs 23e

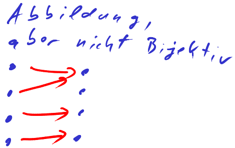
Beispiel: Wir fahren von Augsburg über Ulm und Frankfurt nach Aachen.

- 2 Wege von Augsburg nach Ulm
- 1 Weg von Ulm nach Frankfurt
- 3 Wege von Frankfurt nach Aachen

$\underbrace{\{A_1, A_2\}}_{2}, \underbrace{\{U_1\}}_{1}, \underbrace{\{F_1, F_2, F_3\}}_{3}$

Insgesamt: $2 \cdot 1 \cdot 3 = 6$ Mögliche Wege.

Gleichheitsregel



Satz

*Zwei Mengen S, T sind **gleichmächtig** ($|S| = |T|$) genau dann wenn es eine Bijektion zwischen S und T gibt.*

Beweis: Vorkurs, 23c

Beispiel: Es werden 10 verschiedene Vorlesungen angeboten.
Wie viele Möglichkeiten gibt es, davon Vorlesungen auszuwählen?

Gleichheitsregel

Satz

Zwei Mengen S, T sind ***gleichmächtig*** *($|S| = |T|$) genau dann wenn es eine Bijektion zwischen S und T gibt.*

Beweis: Vorkurs, 23c

Beispiel: Es werden 10 verschiedene Vorlesungen angeboten.
Wie viele Möglichkeiten gibt es, davon Vorlesungen auszuwählen?

Abstahiert und verallgemeinert:

Wie viele Teilmengen $T \subseteq S$ besitzt eine endliche Menge S?

Gesehen im Vorkurs: $|\mathcal{P}(S)| = 2^{|S|}$ (also $2^{10} = 1024$ im Beispiel).

Beweis: Bijektion zwischen $\mathcal{P}(S)$ und Bitfolgen der Länge $|S|$.

Inzindenzsystem

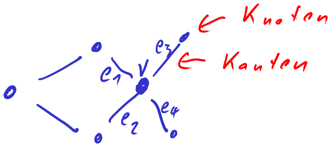
Seien S, T Mengen und $I \subseteq S \times T$ eine Relation zwischen S und T ("Inzidenz").

Dann ist (S, T, I) ein **Inzi~~n~~denzsystem**.

Seien $a \in S$ und $b \in T$:

- a und b sind inzident, genau dann wenn $(a, b) \in I$
- Sonst: a und b sind nicht-inzident

Beispiel: Graph



$V = \{v \ldots\}$
$E = \{e_1, e_2, \ldots\}$

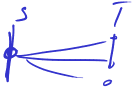
v, e_1 sind inzident

$\delta \subseteq V \times E$

$(v,e) \in \delta$, wenn v an e angrenzt

z.B. $(v,e_1),(v,e_2),(v,e_3),(v,e_4) \in \delta$

Zweifaches Abzählen



Satz

Sei (S, T, I) ein Inzidenzsystem. Für $a \in S$ und $b \in T$,

$r(a)$: *Anzahl zu a inzidenter Elemente aus T*

$r(b)$: *Anzahl zu b inzidenter Elemente aus S*

Dann gilt

$$\sum_{a \in S} r(a) = \sum_{b \in T} r(b)$$

Beweis. Sei
$S = \{a_1, a_2, \ldots, a_m\}$ für ein $m \in \mathbb{N}^*$ und
$T = \{a_1, a_2, \ldots, a_n\}$ für ein $n \in \mathbb{N}^*$.

$m \times n$ Matrix $M = (m_{ij})$ (Inzidenzmatrix):

$$m_{ij} = \begin{cases} 1 & \text{falls } (a_i, b_j) \in I \\ 0 & \text{sonst} \end{cases}$$