Diskrete Mathematik

LVA 405.020

C. Fuchs
Inhaltsübersicht **31.01.2017**

Inhaltsübersicht

Die diskrete Mathematik beschäftigt sich, im Gegensatz zur Analysis, mit diskreten anstatt kontinuierlichen (stetigen) Abläufen. Man kann die diskrete Mathematik auch als die Theorie der endlichen Mengen bezeichnen. Eine Grundfrage ist dabei immer: Wie groß ist die Menge? Diese Frage wird typischerweise in der Kombinatorik beantwortet. Oft sind auf der Menge auch Beziehungen zwischen den Elementen gegeben. Ein eindrücklicher Spezialfall bilden die sogenanten Graphen, die in der Graphentheorie behandelt werden. Bevor wir uns Kombinatorik und Graphentheorie im Detail ansehen, müssen wir uns aber über die Grundlagen unterhalten. Daher beginnen wir mit den natürlichen Zahlen und betrachten unendliche Mengen. Anschließend behandelt wir die elementare Kombinatorik. Die grundlegende Theorie der Graphen und Netzwerke schließt die Vorlesung ab.

Die Vorlesung behandelt (voraussichtlich) die folgenden Themen:

§1. Die natürlichen Zahlen
vollständige Induktion, Arithmetik, Teilbarkeitslehre

§2. Unendliche Mengen
endliche vs. unendliche Mengen, abzählbare vs. überabzählbare Mengen

§3. Elementare Kombinatorik
elementares Zählen, Schubachschlussprinzip, Inklusions-/Exklusionsprinzip, Kombinationen, Repetitionen, Permutationen, Variationen, Partitionen

§4. Graphentheorie
Grundbegriffe, Wege, Kreise, Zusammenhang, planare Graphen

§5. Netzwerke
kürzeste Wege, minimale Bäume, maximale Flüsse, Anwendungen

Bei Fragen oder Bemerkungen (speziell Hinweise auf Fehler aller Art sind willkommen) schicken Sie ein Email an clemens.fuchs@sbg.ac.at.

§1. Die natürlichen Zahlen

1.1 Vollständige Induktion

1.1.1 Motivierendes Beispiel: $1 + 2 + \ldots + n = n(n+1)/2$

1.1.2 Peano Axiomensystem

1.1.3 Prinzip der vollständigen Induktion
Beispiel: $1 + 3 + \ldots + (2n - 1) = n^2$ (F. Maurolicus, 1575)

1.1.4 Varianten und Bemerkungen
Weiter Beispiele: 2.1.13 (es gibt 2^n Teilmengen einer n-Menge), 2.3.4 (n-Tupel), 3.2.15

1.2 Arithmetik

1.2.1 Darstellung der natürlichen Zahlen

1.2.2 Addition und Multiplikation

1.2.3 **Satz** Summe und Produkt sind wohldefiniert.

1.2.4 **Satz** (Eigenschaften)

1.2.5 Potenzen einer Relation

1.2.6 **Satz** Potenzen und gerichtete Wege
Beispiel: $R = \{(a, b), (b, c), (c, d), (d, b)\}$

§2. Unendliche Mengen

2.1 Endliche vs. unendliche Mengen

2.1.1 Endlich, unendlich

2.1.2 **Satz** $\mathbb{N}$ ist unendlich.

2.1.3 **Satz** $|A| = \infty \Rightarrow \exists f : \mathbb{N} \to A$ mit f injektiv.

2.1.4 **Satz** A ist unendlich $\Leftrightarrow \exists f : \mathbb{N} \to A$ mit f injektiv und $f(\mathbb{N}) \subset A$.

2.1.5 gleichmächtig

2.2 Abzählbare vs. überabzählbare Mengen

2.2.1 Begriffe
Beispiele: $\mathbb{N}$, $\mathbb{N}\backslash\{0\}$ sind abzählbar

2.2.2 **Satz** $\mathbb{Z}$ sind abzählbar

2.2.3 **Satz** $\mathbb{Q}$ ist abzählbar

2.2.4 **Satz** Die Menge der Wörter über einem endlichen Alphabet ist abzählbar.

2.2.5 **Satz** Die Menge der endlichen Folgen über einer abzählbaren Menge ist abzählbar.

2.2.6 **Satz** $[0, 1]$ ist überabzählbar.
Beispiel

2.2.7 **Satz** $(0, 1)$ ist überabzählbar.

2.2.8 **Satz** $\mathbb{R}$ ist überabzählbar.

2.2.9 **Satz** Die Potenzmenge von $\mathbb{N}$ ist überabzählbar.

2.2.10 $\leq$, $<$ für beliebige Mengen

2.2.11 **Satz** $|A| < |\text{Potenzmenge von } A|$.

2.2.12 Transfinites Zählen, Kontinuumshypothese

§3. Elementare Kombinatorik

3.1 Elementares Zählen

3.1.1 Gleichheitsprinzip

3.1.2 Additionsprinzip

3.1.3 Prinzip der doppelten Abzählung
Beispiel: 32 Studierende, 32 Studenten mit je 5 befreundeten Studentinnen, n Studentinnen mit je 8 befreundeten Studenten

3.1.4 Multiplikationsprinzip
Beispiel: A-NN-1234

3.1.5 (Dirichletsches) Schubfachprinzip
Beispiel: Turnier mit 10 Mannschaften, jeder gegen jeden. Behauptung: Nach dem ersten Tag haben mind. 2 Mannschaften diesselbe Anzahl von absolvierten Spielen.

3.1.6 Inklusions-Exklusions-Prinzip

3.1.7 Siebprinzip
Beispiel: $\{1 \leq n \leq 1000; 2 \nmid n, 3 \nmid n, 5 \nmid n\}$

3.1.8 Grundprobleme der klassischen Kombinatorik: $f : N \to R$, $|N| = k, |R| = n$, Kugel-Fächer- und Wortinterpretation Die Abbildung f kann interpretiert werden als

- Wort: die Elemente von N sind die Positionen des Wortes, $f(i)$ gibt den Buchstaben an Position i an,
- Zuordnung: N ist die Menge der Kugeln und R die Menge der Fächer, $f(i)$ gibt an, in welches Fach die Kugel mit Nummer i kommt,
- Auswahl: $f(i)$ gibt an, welches Element aus R im Schritt i ausgewählt wird.

Sind die Kugeln einfärbig, so werden die Abbildungen f bis auf Äquivalenz gezählt, wobei zwei Abbildung f_1, f_2 äquivalent heißen, wenn eine Permutation $\pi \in \mathfrak{S}(N)$ vom Grad k existiert mit $f_2 = f_1 \circ \pi$.
Sind die Fächer einfärbig, so werden entsprechend die Abbildungen f bis auf Äquivalenz gezählt, wobei f_1, f_2 nun äquivalent heißen, wenn eine Permutation $\rho \in \mathfrak{S}(R)$ vom Grad n exisitert mit $f_2 = \rho \circ f_1$.
Bezüglich dieser Äquivalenzrelationen zerfallen die Abbildungen von N nach R in Klassen, die in den Grundproblemen gezählt werden sollen.
Zusammenfassend erhält man:

	f beliebig	f injektiv	f surjektiv	f bijektiv
Kugeln gefärbt Fächer gefärbt				
Kugeln gefärbt Fächer einfärbig				
Kugeln einfärbig Fächer gefärbt				
Kugeln einfärbig Fächer einfärbig				

3.2 Kombinationen

3.2.1 k-Kombinationen und Wörter
Beispiel: 2-Kombinationen von 4

3.2.2 **Satz** Eigenschaften von Binomialkoeffizienten
Beispiel zur Illustration der Rekursionsformel

3.2.3 Pascalsches Dreieck

3.2.4 **Satz** Identitäten mit Binomialkoeffizienten inklusive Binomischer Lehrsatz

3.2.5 Interpretation
Beispiel: Lotto 6 aus 45

3.2.6 k-Kombinationen und charakteristische Wörter
Beispiel: k-Kombinationen von 4

3.3 Repetitionen

3.3.1 Begriff
Beispiel: 2-Repetitionen von 4

3.3.2 **Satz**
Beispiel: 3-Repetitionen von 2

3.3.3 Interpretation
Beispiel: Wurf mit 5 Würfeln

3.4 Permutationen

3.4.1 Begriff
Beispiel: 2-Permutationen von 4

3.4.2 **Satz** inklusive $(n)_k = n!/(n-k)!$.

3.4.3 Interpretation
Beispiel: Wörter der Länge 3 mit Buchstaben $2, 3, 4, 5, 6$

3.4.4 Faktorielle, Anzahl der n-Permutationen

3.4.5 Stirling-Zahlen der 1. Art: $s_0(n, k), s(n, k)$

3.4.6 **Satz** Eigenschaften der Stirling-Zahlen

3.4.7 Zykeltypen von Permutationen
Beispiel

3.4.8 **Satz** Anzahl der Permutationen vom Grad n mit vorgegebenem Zykeltyp

3.4.9 **Satz** Anzahl der fixpunktfreien Permutationen vom Grad n

3.5 Variationen

3.5.1 Begriff: k-Variation von n
Beispiel: 2-Variationen von 4

3.5.2 **Satz**: n^k

3.5.3 Interpretation

3.5.4 Typ einer k-Variation vom Typ n, Multinomialkoeffizient

3.5.5 **Satz** Multinomialkoeffizient
Beispiel: $k = 2$

3.5.6 Interpretationen
Beispiel: a) MISSISSIPPI, b) 4-Variationen vom Typ $1^2 2^1 3^1$.

3.6 Partitionen

3.6.1 Wachstumsbeschränkte Wörter, k-Partitionen von n
Beispiele: $n = 3$

3.6.2 **Satz**
Beispiel: Partitionen von 3

3.6.3 Stirling-Zahlen der 2. Art

3.6.4 **Satz**

3.6.5 Interpretation

3.6.6 Typ einer k-Partition von n
Beispiel: Typ $1^1 2^1 3^0$

3.6.7 **Satz** Anzahl

3.6.8 Interpretation

3.6.9 **Satz** Anzahl der surjektiven Abbildungen ist $n!S(k, n)$

3.6.10 Geordnete k-Zahlpartitionen von n
Beispiel: $n = 5$

3.6.11 **Satz**
Beispiel: Veranschaulichung mit den 3-Zahlpartitionen von 5

3.6.12 Ungeordnete k-Zahlpartitionen von n
Beispiel: $n = 5$

3.6.13 **Satz**

3.6.14 Interpretationen

§4. Graphentheorie

4.1 Grundbegriffe

4.1.1 Definition von Graph, adjazent, inzident, endlich, Ordnung, Größe, Diagramm, Schlingen, Mehrfachkanten
Beispiel: $V = \{v_1, \dots, v_4\}, E = \{v_1v_3, v_2v_3, v_2v_4, v_3v_4\}$

4.1.2 Grad, isoliert, k-regulär, Gradfolge

4.1.3 **Satz** Handschlaglemma und Folgerung
Beispiel: Können 333 Telefone so zusammengeschaltet werden, dass jedes mit 3 direkt verbunden ist? Nein.

4.1.4 Teilgraph, induzierte Teilgraphen
Beispiel

4.1.5 Isomorphismus (solche Graphen haben dieselbe Ordnung, Größe und Gradfolge; sie sind aber nicht dadurch bestimmt); Automorphismen
Beispiel

4.2 Wege, Kreise, Zusammenhang

4.2.1 Weg (Start-, Endknoten, Länge, einfach) und Kreis (einfach)
Beispiele

4.2.2 Zusammenhang, Verbindbarkeit ist eine Äquivalenzrelation
Beispiel

4.2.3 **Satz** Streichen von Kanten auf einem einfachen Kreis erhält Zusammenhang

4.2.4 Abstand (definiert eine Metrik)

4.2.5 kreisfrei, Wald, Baum, Spannbaum (Gerüst), Brücke (Isthmus)
Beispiele

4.2.6 **Satz** Jeder Baum a) enthält mindestens 2 Knoten vom Grad 1; b) erfüllt $|E| = |V| - 1$.

4.2.7 **Satz** Jeder zusammenhängende Graph a) hat einen Spannbaum; b) ist ein Baum genau dann, wenn = gilt.
Beispiel für Spannbaum

4.2.8 Bipartite Graphen
Beispiel

4.2.9 **Satz** Charakterisierung von zusammenhängenden bipartiten Graphen

4.3 Planare Graphen

4.3.1 planar, eben $\rightarrow$ Flächen, Gebiete
Beispiel: Hexaeder

4.3.2 **Satz** (Eulersche Polyederformel): $n - m + f = 2$

4.3.3 **Satz** Ein planarer zshg Graph mit $n \geq 3$ Knoten hat $\leq 3n - 6$ Kanten.

4.3.4 vollständig, K_n
Beispiel: $K_1, \ldots, K_5$

4.3.5 Vollständigkeit bei bipartiten Graphen, $K_{m,n}$
Beispiel: $K_{3,3}$

4.4 Die Adjazenz- und Inzidenzmatrix

4.4.1 Adjazenzmatrix
Beispiel

4.4.2 **Satz** (i, j)-Eintrag der k-ten Potenz = Anzahl der Wege der Länge k von v_i nach v_j
Zur Vorbereitung: Matrixprodukt und Potenzen einer Matrix

4.4.3 Inzidenzmatrix
Beispiel

4.4.4 Algorithmus *Breitensuche* (Breadth-First):
Eingabe: Graph $G = (V, E)$
Ausgabe: Der Algorithmus findet einen Spannbaum (einer Komponente) und beantwortet dabei die Frage, ob G zusammenhängend ist.

(1) Starte mit einem Knoten und gib ihm die Nummer 1, 1 ist der aktuelle Knoten.
(2) Der aktuelle Knoten habe die Nummer i und es seien bereits die Nummern $1, \ldots, r$ vergeben. Falls $r = n$, stop. Andernfalls betrachte die noch nicht nummerierten Nachbarn von i und gibt ihnen der Reihe nach die Nummern $r + 1, r + 2, \ldots$ und füge die Kanten $i(r + 1), i(r + 2), \ldots$ hinzu. Falls nun die Nummer $i + 1$ nicht existiert, stop (G ist nicht zusammenhängend), andernfalls gehe zum Knoten mit Nummer $i + 1$, dies ist die neue aktuelle Nummer, und iteriere (2).

Korrektheitsbeweis und Beispiel

4.4.5 Algorithmus *Tiefensuche* (Depth-First):
Eingabe: Graph $G = (V, E)$
Ausgabe: Wie bei 4.4.4.

(1) Starte mit einem Knoten und gib ihm die Nummer 1, 1 ist der aktuelle Knoten. Wähle einen Nachbarn von 1 und gib ihm die Nummer 2. Füge die Kante 12 ein. 2 ist nun der aktuelle Knoten und 1 ist der Vorgängerknoten.

(2) Der aktuelle Knoten habe Nummer i und es seien die Nummern $1, \ldots, r$ bereits vergeben. Falls $r = n$, stop. Andernfalls wähle einen noch nicht nummerierten Nachbarn von i, gib ihm die Nummer $r+1$, und füge die Kante $i(r+1)$ ein. Der aktuelle Knoten ist nun $r+1$ und i der Vorgängerknoten. Falls keine nichtnummerierten Nachbarn von i existieren, gehe zum Vorgängerknoten von i, falls $i > 1$. Dies ist nun der aktuelle Knoten. Iteriere (2). Wenn $i = 1$ ist, und keine nicht nummerierten Nachbarn existieren, so ist G nicht zusammenhängend, stop.

Korrektheitsbeweis und Beispiel

§5. Netzwerke

5.1 Kürzeste Wege

5.1.1 Wegenetz=Netzwerk, Länge, kürzester Weg, Abstand (definiert Metrik bei ungerichteten Netzen)
Beispiel

5.1.2 Algorithmus *Floyd-Warshall*:
Eingabe: gerichtetes Wegenetz $D = ((V, E), \omega)$ mit $V = \{v_1, \ldots, v_n\}$, in dem es keine Kreise negativer Länge gibt
Ausgabe: Der Algorithmus berechnet die Länge von kürzesten Wegen zwischen allen Knoten des Wegenetzes.

(1) Setze $d^{(0)}(v_i, v_j)$ gleich $\omega(v_i, v_j)$ falls $v_i v_j \in E$, 0 falls $v_i = v_j$ und ∞ sonst.
(2) Für alle $k \in \{1, \ldots, n\}$ berechne für alle $(i, j) \in \{1, \ldots, n\}^2$ die Zahlen $d^{(k)}(v_i, v_j) = \min\{d^{(k-1)}(v_i, v_j), d^{(k-1)}(v_i, v_k) + d^{(k-1)}(v_k, v_j)\}$.
(3) Der kürzeste Weg von v_i nach v_j hat Länge $d^{(n)}(v_i, v_j)$.

Satz (Floyd-Warshall)
Beispiel

5.1.3 Algorithmus *Dijkstra*:
Eingabe: gerichtetes Wegenetz $D = ((V, E), \omega)$ und Startknoten $v_0 \in V$ mit $\omega(e) \geq 0$ für alle $e \in E$
Ausgabe: Der Algorithmus berechnet die Länge von kürzesten Wegen von v_0 zu allen anderen Knoten.

(1) Setze $S = \emptyset, d(v_0) = 0$ und $d(v) = \infty$ für alle $v \in V \backslash \{v_0\}$.
(2) Falls $S = V$, gehe zu (3). Andernfalls wähle ein $v \in V \backslash S$ mit $d(v) = \min\{d(u); u \in V \backslash S\}$ und füge v zu S hinzu. Für alle anderen Elemente $w \in V \backslash S$ setze $d(w) = \min\{d(w), d(v) + \omega(vw)\}$. Iteriere (2).
(3) Der kürzeste Weg von v_0 nach v_i hat Länge $d(v_i)$.

Satz (Dijkstra)
Beispiel

5.2 Minimale Bäume

5.2.1 Kosten, minimaler Spannbaum

5.2.2 Algorithmus *Kruskal*:
Eingabe: ungerichtetes Wegenetz $D = (G, \omega)$ mit $G = (V, E)$ zusammenhängend
Ausgabe: Der Algorithmus liefert die Kantenmenge F eines minimalen Spannbaumes $B = (V, F)$.

(1) Setze $F = \emptyset$.
(2) Falls $E = \emptyset$, stop. Andernfalls entferne eine Kante e mit $\omega(e) = \min\{\omega(f); f \in E\}$ von E. Falls der Graph $(V, F \cup \{e\})$ kreisfrei ist, füge e zu F hinzu. Iteriere (2).

5.2.3 Sehne, Fundamentalkreis
Beispiel

5.2.4 **Satz** (Kruskal)
Beispiel

5.3 Maximale Flüsse

5.3.1 Flussnetz, Kapazität, innere Knoten
Beispiel

5.3.2 Flüsse (kirchhoffsche Bedingung)
Beispiel

5.3.3 Hilfsaussage, Kapazität eines Flusses, maximale Flüsse

5.3.4 Schnitte, Kapazität, Schnitte mit minimaler Kapazität in einem Flussnetz
Beispiel

5.3.5* Algorithmus *Ford-Fulkerson* (MaxFlow-MinCut):
Eingabe: Flussnetz (D, κ, q, s)
Ausgabe: Der Algorithmus berechnet einen maximalen Fluss $f : E \to \mathbb{R}_{\geq 0}$ im Flussnetz.

(1) Setze $f = 0$.
(2) Setze $S = M = \{q\}$ und $\epsilon(q) = \infty$.
(3) Falls $M = \emptyset$, stop (f ist der maximale Fluss). Sonst entferne einen Knoten v aus M und führe die folgenden Schritte durch: für jeden Knoten $w \in V \backslash S$ mit $vw \in E$ und $f(vw) < \kappa(vw)$ setze $\mathrm{pred}(w) = v, R(w) =$“$\rightarrow$”,$\epsilon(w) = \min\{\kappa(vw) - f(vw), \epsilon(v)\}$ und füge w zu S und M hinzu; für jeden Knoten $u \in V \backslash S$ mit $uv \in E$ und $f(uv) > 0$ setze $\mathrm{pred}(u) = v, R(u) =$“$\leftarrow$”,$\epsilon(u) = \min\{f(uv), \epsilon(v)\}$ und füge u in S und M hinzu.
(4) Setze $\epsilon = \epsilon(s)$ und $v = s$.
(5) Setze $f(\mathrm{pred}(v), v) = f(\mathrm{pred}(v), v) + \epsilon$ falls $R(v) =$“$\rightarrow$” und $f(v, \mathrm{pred}(v)) = f(v, \mathrm{pred}(v)) - \epsilon$ sonst. Falls $v = q$, gehe zu (2). Sonst ersetze v durch $\mathrm{pred}(v)$ und iteriere (5).

5.3.6 **Satz** (Ford-Fulkerson) MaxFlow=MinCut
Beweisskizze

§A. Zusammenfassung

A.1 Die natürlichen Zahlen:
Vollständige Induktion, Peano-Axiome, Arithmetik der natürlichen Zahlen
Wichtige Sätze: 1.2.3+4

A.1 Unendliche Mengen:
Endlich vs. unendlich + Beispiele, Abzählbar vs. überabzählbar + Beispiele, Kardinalzahlen, Kontinuumshypothese
Wichtige Sätze: 2.1.3+4, 2.2.3, 2.2.6, 2.2.11

A.3 Elementare Kombinatorik:
Gleichheitsprinzip, Additionsprinzip, Prinzip des doppelten Abzählens, Multiplikationsprinzip, Schubfachschlussprinzip, Inkl./Exklusionsprinzip, Siebprinzip, Grundprobleme der klassischen Kombinatorik, Kombinationen, Repetitionen, Permutationen - Zykeltyp, Variationen - Typ, Partitionen - Typ, Zahlpartitionen - geordnet/ungeordnet
Wichtige Sätze: 3.1.6, 3.2.2, 3.2.4, 3.3.2, 3.4.2, 3.4.6, 3.5.2, 3.5.5, 3.6.2, 3.6.4, 3.6.9, 3.6.11, 3.6.13

A.4 Graphentheorie:
Graphen - Ordnung, Größe, Grad, Handschlaglemma, Teilgraphen, Isomorphie, Zusammenhang, Distanz, Wälder und Bäume + Charakterisierung von Bäumen, Bipartite Graphen + Charakterisierung von bipartiten Graphen, Planare Graphen - Eulersche Polyederformel, Adjazenz- und Inzidenzmatrix, Breiten- und Tiefensuche
Wichtige Sätze: 4.1.3, 4.2.3, 4.2.6, 4.2.7, 4.2.9, 4.3.2, 4.3.3, 4.4.2, 4.4.4

A.5 Netzwerke:
Wegenetze (Länge), Flussnetze (Flüsse, kirchhoffsche Bedingung, Kapazität), Kürzeste Wege: Floyd-Warshall, Dijkstra, Minimale Bäume: Kruskal (Sehne, Fundamentalkreis), Maximale Flüsse: Ford-Fulkerson (Schnitte, Kapazität)*
Wichtige Sätze: 5.1.2, 5.1.3, 5.2.4, 5.3.5+6*

§B. Literatur

1. K.-H. Zimmermann, Diskrete Mathematik, Books on Demand, 2006, ISBN978-3-8334-5529-2
2. M. Aigner, Diskrete Mathematik, Vieweg+Teubner, 2009, ISBN978-3-8348-0084-8
3. A. Beutelspacher und M.-A. Zschiegner, Diskrete Mathematik für Einsteiger, Vieweg+ Teubner, 2011, ISBN978-3-8348-1248-3
4. G.+S. Teschl, Mathematik für Informatiker (Band 1, Diskrete Mathematik und Lineare Algebra), Springer, 2008, ISBN978-3540-77431-0