Diskrete Mathematik

2. Übungsblatt

Präsenzübungen

Aufgabe P2.1 (Inklusion–Exklusion)

a) Zeigen Sie, dass für endliche Mengen A, B, C gilt

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|\,.$$

b) Wieviele Zahlen in $\{1, \ldots, 100\}$ sind durch 2, 3 oder 5 teilbar?

Aufgabe P2.2 (Permutationen)

Für eine Bijektion $f\colon M \to M$ schreiben wir $f^k = \overbrace{f \circ \cdots \circ f}^{k \text{ mal}}$. Wir betrachten zwei Elemente von $\mathrm{Sym}(\{1,2,3,4,5\})$:

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix} \quad \text{und} \quad g = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 1 & 2 \end{pmatrix}$$

Die Notation bedeutet, dass die Abbildung ein Element in der ersten Zeile auf das darunterstehende abbildet. Bestimmen Sie die Mengen $\{i, f(i), f^2(i), f^3(i), \ldots\}$ für $i \in \{1,2,3,4,5\}$. Welches ist das kleinste $k \geq 1$ mit $f^k = \mathrm{id}$? Gehen Sie analog für g vor.

Aufgabe P2.3 (k-Partitionen)

Eine *k-Partition* einer Menge M ist eine Partition von M in k Teilmengen. Wir bezeichnen die Menge der k-Partitionen von M mit $\mathrm{Part}(k, M)$. Die Anzahl aller k-Partitionen einer n-elementigen Menge heißt *Stirling-Zahl zweiter Art* (engl. *Stirling number of the second kind*) und wird als $S_{n,k}$ geschrieben.

Zeigen Sie $S_{n,1} = S_{n,n} = 1$ und $S_{n,k} = S_{n-1,k-1} + kS_{n-1,k}$ für $2 \leq k \leq n-1$.

Hinweis: Definieren Sie für $x \in M$ eine Abbildung

$$\mathrm{Part}(k, M) \to \mathrm{Part}(k, M \setminus \{x\}) \cup \mathrm{Part}(k-1, M \setminus \{x\}).$$

Aufgabe P2.4 (Würfelfärbungen)

Ein Farbwürfel ist ein Würfel, dessen Seiten mit den Farben rot, grün, blau, gelb, violett und orange angemalt sind (jede Seite in einer anderen Farbe). Wieviele verschiedene Farbwürfel gibt es?

Hausübungen

Aufgabe H2.1 (Inklusion–Exklusion)

Zeigen Sie per Induktion die Formel "Inklusion–Exklusion": für endliche Mengen $X_1, \ldots, X_n$ gilt

$$\begin{aligned} \left| \bigcup_{1 \leq i \leq n} X_i \right| &= \sum_{j=1}^{n} (-1)^{j+1} \sum_{i_1 < \ldots < i_j} \left| X_{i_1} \cap \cdots \cap X_{i_j} \right| \qquad (1) \\ &= \big(|X_1| + \ldots + |X_n| \big) - \big(|X_1 \cap X_2| + |X_1 \cap X_3| + \ldots \big) + \ldots \pm |X_1 \cap \ldots \cap X_n|\,. \end{aligned}$$

Wieviele Zahlen in $\{1, \ldots, 100\}$ sind durch 2, 3, 5 oder 7 teilbar? Welcher Zusammenhang besteht zur Anzahl der Primzahlen bis 100?

Hinweis: Denken Sie für an das Sieb des Eratosthenes **(2+2 Punkte)**

Aufgabe H2.2 (Zykel)

Sei M eine endliche Menge mit n Elementen.

a) Zeigen Sie: wenn $f \in \mathrm{Sym}(M)$ die Eigenschaft hat, dass $f(A) \neq A$ für alle $\emptyset \neq A \neq M$, dann ist $f^n = \mathrm{id}$.
b) Wieviele solcher Elemente gibt es in $\mathrm{Sym}(M)$? **(2+2 Punkte)**

Aufgabe H2.3 (Surjektive Abbildungen)

a) Geben Sie eine Abbildung $\mathrm{Surj}(M, \{1, \ldots, k\}) \to \mathrm{Part}(k, M)$ von der Menge der surjektiven Abbildungen $\{1, \ldots, k\} \to M$ in die Menge der k-Partitionen von M an.
b) Bestimmen Sie die Anzahl der surjektiven Abbildungen $A \to B$ wenn A und B endliche Mengen sind (der Ausdruck kann Stirling-Zahlen enthalten, die nicht aufgelöst werden brauchen).
c) Wieviele Möglichkeiten gibt es 64 Studentinnen auf drei Übungsgruppen aufzuteilen?

Hinweis: (a) Benutzen Sie Lemma 1.2.3. (c) Suchen Sie online nach einem Rechner für Stirling-Zahlen (vorzugsweise auf englisch). **(2+2+1 Punkte)**

Aufgabe H2.4* (Bänder)

Wir haben in der Vorlesung *Bänder* (auf einem Alphabet A) eingeführt als Äquivalenzklassen $\{x_1 \ldots x_k,\, x_k \ldots x_1\}$ von Worten modulo Spiegelung.

Schreiben Sie eine Funktion `bands`, die ein Alphabet A und eine natürliche Zahl n annimmt und eine Liste von Repräsentanten von allen Bändern der Länge n auf dem Alphabet A zurückgibt.

Für volle Punktzahl sollte Ihre Funktion die folgenden Bedingungen erfüllen:

a) Von den beiden Repräsentanten eines Bandes (falls das Band kein Palindrom ist) wird der lexikographisch kleinere (s.u.) ausgegeben.
b) Die Repräsentanten werden in lexikographisch aufsteigender Reihenfolge aufgelistet.

Das Alphabet $A = \{\ell_1, \ldots \ell_t\}$ wird als Wort $\ell_1 \ldots \ell_t$ (ohne wiederholte Buchstaben) übergeben. Dadurch sind die Buchstaben auf natürliche Weise geordnet: $\ell_i \leq \ell_j \Leftrightarrow i \leq j$. Die *lexikographische Ordnung* $\leq^k$ auf A^k ist nun gegeben durch

$$x_1 \ldots x_k \leq^k y_1 \ldots y_k :\Leftrightarrow x_1 \ldots x_k = y_1 \ldots y_k \text{ oder } x_m < y_m \text{ für } m = \min\{i \mid x_i \neq y_i\}.$$

Das heißt die Worte haben dieselbe Reihenfolge wie in einem Lexikon, daher der Name.

Beispiel Haskell:

```
> bands "01" 3
["000","001","010","011","101","111"]
> bands "ATCG" 2
["AA","AT","AC","AG","TT","TC","TG","CC","CG","GG"]
```

Beispiel Python:

```
> bands('ein',3)
['eee', 'eei', 'een', 'eie', 'eii', 'ein', 'ene', 'eni', 'enn', 'iei', 'ien',
'iii', 'iin', 'ini', 'inn', 'nen', 'nin', 'nnn']
> bands('MA',4)
['MMMM', 'MMMA', 'MMAM', 'MMAA', 'MAMA', 'MAAM', 'MAAA', 'AMMA', 'AMAA', 'AAAA']
```

(3* Punkte)

Die Punkte für Programmieraufgaben (*) werden bei der für die Klausurzulassung zu erreichenden Punktzahl nicht berücksichtigt (bei der erreichten Punktzahl aber schon).

Abgabe bis 6.11.2015.