Diskrete Mathematik – Unterlagen

Fabian Hahn, Dino Wernli, Stefan Götschi
Mai 2008

1. Relationen

Allgemeines

Def: binäre **Relation** R zwischen A und B: $R \subseteq A \times B$
wir schreiben $a\ R\ b$ für $(a,b) \in R$

Relationen können als binäre $|A| \times |B|$ Matrix oder als gerichteter Graph dargestellt werden.

Mögliche Eigenschaften von Relationen auf A

- **reflexiv:** $\forall a \in A: \ (a,a) \in R$
- **antireflexiv:** $\forall a \in A: \ (a,a) \notin R$
- **symmetrisch:** $\forall a,b \in A: \ (a,b) \in R \rightarrow (b,a) \in R$
- **antisymmetrisch:** $\forall a,b \in A: \ \big((a,b) \in R \wedge (b,a) \in R\big) \rightarrow a = b$
- **transitiv:** $\forall a,b,c \in A: \ \big((a,b) \in R \wedge (b,c) \in R\big) \rightarrow \big((a,c) \in R\big)$

Operationen auf Relationen

Übliche Mengenoperationen sind zulässig.

Komposition:

$R \subseteq A \times B\,; Q \subseteq B \times C \rightarrow R \circ Q \subseteq A \times C\,;$
$(a,c) \in R \circ Q \leftrightarrow \exists b \in B: (a,b) \in R \wedge (b,c) \in Q$

Transitiver Abschluss: $\bigcup_{n=1}^{\infty}(\underbrace{R \circ R \circ \ldots \circ R}_{n})$

Arten von Relationen

Äquivalenzrelationen:
Eine Relation $\sim$ auf A heisst Äquivalenzrelation, falls sie symmetrisch, transitiv und reflexiv ist.

Eine Partition einer Menge A ist eine disjunkte Unterteilung in Untermengen $(A_i)_{i \in I}$ mit: $\bigcup A_i = A$

Eine Äquivalenzrelation partitioniert eine Menge in Äquivalenzklassen.
Die Äquivalenzklasse/Partition von a: $[a] = \{b \mid b \sim a\}$

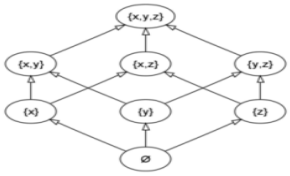
Ordnungsrelationen:
Eine Relation $\leq$ heisst Partialordnung auf A, falls sie reflexiv, antisymmetrisch und transitiv ist. Dabei ist es möglich, dass manche Paare nicht vergleichbar sind.

Eine Partialordnung, wo alle Paare paarweise vergleichbar sind, heisst Totalordnung/Kette/lineare Ordnung auf A

Begriffe:

- x maximales Element, falls: $\nexists y \neq x: \ y \geq x$
- x grösstes Element, falls: $\forall y: \ x \geq y$
- $(A, \leq)$ heisst wohlgeordnet, falls jede nicht-leere Teilmenge von A ein kleinstes Element besitzt

Endliche Ordnungsrelationen können als Hasse-Diagramme dargestellt werden. Dabei werden immer nur direkte Nachbarn verbunden.
Bsp: $A = \{a \mid a \text{ Teiler von } x \cdot y \cdot z\}$



Relationen als Funktionen

Relation $f \subseteq A \times B$ heisst funktional $(f: A \rightarrow B)$, falls:

- $\forall a \exists b: (a,b) \in f$
- $\big((a,b) \in f \wedge (a,b') \in f\big) \rightarrow b = b'$

Wir schreiben für $(a,b) \in f$: $f(a) = b$ oder $f: a \mapsto b$

Zusätzliche Eigenschaften:

- $f: A \rightarrow B$ **injektiv**, falls: $a \neq a' \rightarrow f(a) \neq f(a')$
  Folgerung: $|A| \leq |B|$
- $f: A \rightarrow B$ **surjektiv**, falls: $\forall b \exists a: \ f(a) = b$
  Folgerung: $|A| \geq |B|$
- $f: A \rightarrow B$ **bijektiv**, falls surjektiv und injektiv
  Folgerung: $|A| = |B|$, bzw. gleichmächtig

2. Kombinatorik

Allgemeines

Urnenmodell: ziehe k aus n	geordnet	ungeordnet
mit Zurücklegen	n^k	$\binom{n+k-1}{k}$
ohne Zurücklegen	$\frac{n!}{(n-k)!}$	$\binom{n}{k}$

Der **Binomialkoeffizient** $\binom{n}{k}$ ist das Bildungsgesetz für das Pascal-Dreieck.

- $\binom{n}{k} = \frac{n!}{(n-k)!\cdot k!} = \frac{n(n-1)(n-2)\ldots(n-k+1)}{k!}$
- $\binom{n}{k} = \binom{n}{n-k} = \binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n+1}{k} - \binom{n}{k-1}$
- Binomischer Satz: $(x+y)^n = \sum_{k=0}^{n}\binom{n}{k}x^k y^{n-k}$
- $\sum_{k=0}^{n}\binom{n}{k} = 2^n$
- $\binom{n}{k} \geq \left(\frac{n}{k}\right)^k$; $\frac{\binom{n}{k}}{\left(\frac{n}{k}\right)^k} \leq e^k$; $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{n\cdot e}{k}\right)^k$
- Vandermonde: $\binom{n}{k} = \sum_{t=0}^{k}\binom{r}{t}\binom{n-r}{k-t}$

Datenkompression

Binäre Entropiefunktion:

$$h(x) = -x\log_2 x - (1-x)\log_2(1-x)$$

Approximation für Binomialkoeffizient: $\binom{n}{k} \approx 2^{n\cdot h\left(\frac{k}{n}\right)}$

Ein String der Länge n aus einer Stringmenge mit Nullwahrscheinlichkeit x ist auf die Länge $\log_2\binom{n}{xn} \approx nh(x)$ komprimierbar.

Kombinatorische Regeln

- für A_i Partitionen von A: $|A| = \sum|A_i|$
- für $(A_i)_{i=1,\ldots,n}$ beliebig: $\left|\mathrm{X}A_i\right| = \prod_{i=1}^{n}|A_i|$
- Inklusion/Exklusion:
  $|\cup_{i=1}^{n} A_i| = \sum_{i=1}^{r}(-1)^{r-1}\sum_{1\leq i_1<i_2<\ldots<i_r\leq n}\left|\cap_{j=1}^{n} A_{i_j}\right|$

Euler'sche φ-Funktion

$\varphi(n) = |\{k \in \{0,\ldots,n-1\} \mid ggT(k,n) = 1\}|$.
Zum Beispiel: $\varphi(6) = 2$; p prim $\varphi(p) = p-1$
Allgemein: $n = \prod_i p_i^{l_i} \rightarrow \varphi(n) = \prod_i p_i^{l_i-1}(p_i - 1)$

Schubfachprinzip

Wenn n Objekte auf $k(<n)$ Schubfächer verteilt werden, enthält mindestens 1 Schubfach mindestens 2 Objekte.
Anwendung: für jede Folge der Länge n ist die längste monotone Teilfolge höchstens $\sim\sqrt{n}$ lang.

Anzahl Äquivalenzrelationen auf eine n-Menge

Es gibt $S_{n,k} = S_{n-1,k-1} + k \cdot S_{n-1,k}$ Möglichkeiten, eine n-Menge in genau k Partitionen zu zerlegen.

Nun ist die #Äquivalenzrelationen: $B_n = \sum_{k=1}^{n} S_{n,k}$
Die Rekursion $S_{n,k}$ bildet das **Stirling-Dreieck 2.Art:**

n \ k	0	1	2	3	4	5	6	7	8	9
0	1									
1	0	1								
2	0	1	1							
3	0	1	3	1						
4	0	1	7	6	1					
5	0	1	15	25	10	1				
6	0	1	31	90	65	15	1			

Permutationen
Eine Permutation π ist eine bijektive Abbildung einer Menge auf sich selbst und kann dargestellt werden:

- als Matrix: $\pi = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ \pi(1) & \pi(2) & \pi(3) & \dots & \pi(n) \end{pmatrix}$
- Menge eindeutiger, disjunkter Zyklen.
  (Zyklen selbst nicht eindeutig $(1,3,8) = (3,8,1)$)

1-elementige Zyklen heissen Fixpunkte der Permutation.
fixpunktfreier Permutationen einer n-Menge → $n!\, e^{-1}$

Alle Permutationen einer Menge bilden eine nicht-kommutative (nicht-Abelsche) Gruppe.

Sei $S_{n,k}$ die Anzahl Permutationen einer n-Menge mit genau k Zyklen → $S_{n,k} = S_{n-1,k-1} + (n-1) \cdot S_{n-1,k}$
Durch diese Rekursion entsteht das **Stirling-Dreieck 1.Art:**

n \ k	0	1	2	3	4	5	6
0	1						
1	0	1					
2	0	−1	1				
3	0	2	−3	1			
4	0	−6	11	−6	1		
5	0	24	−50	35	−10	1	
6	0	−120	274	−225	85	−15	1

3. Lösen von Rekursionsgleichungen

Fibonacci-Zahlen
$f_0 = 0;\ \ f_1 = 1;\ \ f_n = f_{n-1} + f_{n-2}$

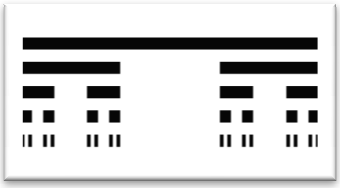
Man nehme den Ansatz: $f_n = \lambda^n$ und setze ein:

$\lambda^n = \lambda^{n-1} + \lambda^{n-2} \rightarrow\ \lambda^2 - \lambda - 1 = 0$

Die Nullstellen des Polynoms $\lambda_{1,2} = \frac{1 \pm \sqrt{5}}{2}$ liefern eine Lösung der Form: $f_n = a\left(\frac{1+\sqrt{5}}{2}\right)^n + b\left(\frac{1-\sqrt{5}}{2}\right)^n$

Anfangswerte einsetzen ergibt: $a = \frac{1}{\sqrt{5}};\ \ b = -\frac{1}{\sqrt{5}}$

Cantormenge
„Faktor f in jede Richtung“ erzeugt k Kopien:
Es gilt: $k = f^d \rightarrow d = \frac{\log k}{\log f}$
für CM: $f = 3, k = 2 \rightarrow d = \log_3 2$



Master-Theorem

$T(n) = a \cdot T\left({}^{n}/_{b}\right) + f(n) \rightarrow$ Lösung :

$$T(n) = \begin{cases} \theta\big(f(n)\big), & f(n) > n^{\log_b a} \\ \theta\big(n^{\log_b a}\big), & f(n) < n^{\log_b a} \\ \theta\big(\log n \cdot n^{\log_b a}\big), & f(n) = n^{\log_b a} \end{cases}$$

4. Graphentheorie

Definition
Graph $G(V,E) \rightarrow V$ Menge(Knoten), E Relation(Kanten)

- $\deg_+ v$ = Anzahl Kanten, die von v weggehen
- $\deg_- v$ = Anzahl Kanten, die in v ankommen

Graphen Allgemein
Mögliche Eigenschaften von Graphen $G = (V,E)$:

- **ungerichtet:** E symmetrisch
- **isomorpher** Graph $G' = (V', E')$ zu G:
  $\exists f: V \rightarrow V'$ bijektiv: $\forall v, w \in V$:
  $(v,w) \in E \quad \leftrightarrow \quad \big(f(v), f(w)\big) \in E'$

Es gilt immer: $\sum_{v \in V} \deg_- v = \sum_{v \in V} \deg_+ v = |E|$

Nachbarschaftsfunktion: $\Gamma(v) = \{w \in V \mid v, w \in E\}$

Für ungerichtete Graphen gilt
- $\forall v$: $\deg_+ v = \deg_- v = \deg v$
- $\sum_{v \in V} \deg v = 2 \cdot |E|$

Einfacher Graph: keine Loops und Mehrfachkanten

Bipartiter Graph: Zweifärbbar, keine Kante ist Verbindung zwischen zwei gleichfarbigen Knoten.
Es gilt: G bipartit ↔ G enthält keine ungeraden Kreise

Weg: $W = (v_0, \dots, v_r);\ \ (v_i, v_{i+1} \in E)$

Pfad: Weg, bei dem alle Knoten verschieden sind

Zusammenhängender Graph:
$\forall u, v \in V, \exists$Pfad $(v_0, \dots, v_r)$: $v_0 = u\ \wedge v_r = v$

Kreis $(v_0, \dots, v_r)$: Pfad mit $(v_r, v_0) \in E$

Teilgraph (V', E') von G: $\quad V' \subseteq V; E' \subseteq E; E' \subseteq V' \times V'$

Induzierter Teilgraph H' von $V' \subseteq V$, falls: $\forall u, v \in V'$: $(u,v) \in E \rightarrow (u,v) \in E'$ Notation: $H' = G[V']$
Zwei verbundene Knoten in G sind auch in H' verbunden.

Zusammenhangskomponente (ZHK) $G[V_i]$ von G: (V_i)
Partition von V, so dass: $\exists$Pfad $(u,v)\ \leftrightarrow\ \exists i$: $u, v \in V_i$

Es gilt immer:
- Ein Graph $G(V,E)$ besitzt mind. $|V| - |E|$ ZHK
- G zusammenhängend → $|E| \geq |V| - 1$

Brücke $e \in E$ von $G \leftrightarrow G'(V, E \setminus \{e\})$ eine ZHK mehr als G

Bäume
Definitionen und Eigenschaften:
- Wald: einfacher, ungerichteter Graph ohne Kreise
- Baum: zusammenhängender Wald
- Blatt: $v \in V$ mit $\deg(v) = 1$
- Jeder Baum (ausser dem Punkt) hat ≥ 2 Blätter

Äquivalente Aussagen:
1. G Baum: kreislos, zusammenhängend
2. G zusammenhängend, $|V| = |E| + 1$
3. G kreislos, $|V| = |E| + 1$
4. G zusammenhängend, alle e sind Brücken
5. $\forall u, v \in G,\ \exists$ ein eindeutiger Pfad(u,v)

Spannbaum $H = (V, E')$ eines zusammenhängenden Graphen G, falls H ein Baum ist und $E' \subseteq E$

Einige spezielle ungerichtete Graphen
$\boldsymbol{K_n}$: vollständiger Graph, n Knoten, hat n^{n-2} Spannbäume

$\boldsymbol{C_n}$: Kreisgraph mit n Knoten

$\boldsymbol{M_{m,n}}$: Gittergraph, m waagrechte, n senkrechte Linien: $V = \{(i,j)\},\ i = 1 \dots m,\ j = 1 \dots n : \big((i,j),(i',j')\big) \in E \leftrightarrow [(i = i' \wedge |j - j'| = 1) \vee (j = j' \wedge |i - i'| = 1)]$

$\boldsymbol{K_{m,n}}$: bipartiter Graph, wobei jeder Knoten einer Farbe mit jedem Knoten der anderen Farbe verbunden ist.

$\boldsymbol{Q_d}$: d-dimensionaler Hyperkubus.Nehme 2 Kopien von Q_{d-1}, füge zusätzliches Bit links hinzu:

- $(u,v) \in E \leftrightarrow d_H(u,v) = 1$
- # Kanten: $|E| = d \cdot 2^{d-1}$

Eulertouren und Hamiltonkreise

Eulertour: geschlossene Tour in zusammenhängendem G, bei der jede Kante genau einmal besucht wird.

$\exists$ Eulertour in $G \leftrightarrow \forall v \in V: \deg(v)$ gerade

$\exists$ ungeschlossene Eulertour: in $G \leftrightarrow$ alle Kantengrade ausser genau zwei sind gerade

Hamiltonkreis: geschlossener Kreis in zusammenhängendem G, jeder Knoten wird genau einmal besucht. Der Graph G heisst dann Hamiltonsch.

Für $m,n \geq 2$: m,n gerade $\leftrightarrow M_{m,n}$ Hamiltonsch

Q_d ist Hamiltonsch für alle d ausser 1

$G(V,E)$ mit $\forall v \deg(v) \geq \frac{|V|}{2} \rightarrow G$ Hamiltonsch

Planare Graphen

Ein Graph ist genau dann planar, wenn er so gezeichnet werden kann, dass sich keine Kanten überkreuzen. Die Kanten müssen aber nicht zwingend gerade sein.

Eigenschaften planarer Darstellungen von Graphen:
Für einen zusammenhängenden planaren Graphen G, der die Ebene in f Gebiete unterteilt, gilt die Euler-Formel:

$$|V| + f - |E| = 2 \rightarrow |E| \leq 3 \cdot |V| - 6$$

G planar, dreiecksfrei $\rightarrow |E| \leq 2 \cdot |V| - 4$

Ein zusammenhängender Graph ist genau dann NICHT planar, wenn er sich auf $K_{3,3}$ oder K_5 reduzieren lässt, durch:

- Eine beliebige Kante streichen
- 2 Knoten verschmelzen, Verbindungen behalten

Färbung von Graphen

Ziel: Die Knoten von G mit k Farben färben, so dass keine Kante 2 Knoten gleicher Farbe verbindet.

$\chi(G)$: minimale Anzahl nötiger Farben zur Färbung von G

Es gilt: $\chi(G) = 2 \leftrightarrow G$ ist bipartit

Färbbarkeit von speziellen Graphen:

$\chi(K_n) = n$
$\chi(K_{m,n}) = 2$
$\chi(M_{m,n}) = 2$
$\chi(\text{Baum}) = 2$
$\chi(C_n) = \begin{cases} 2, & n \text{ gerade} \\ 3, & n \text{ ungerade} \end{cases}$
$\chi(planar) \leq 4$

Matchings

M heisst Matching von $G(V,E)$, falls:

- $M \subseteq E$
- Kein Knoten von G ist in > 1 Kante von M

Ein Matching ist perfekt, falls jeder Knoten eine Kante hat.

Perfektes Matching im Graph G mit $V = A \cup B$ bipartit: Annahme oBdA $|A| \leq |B|$. Das Matching heisst perfekt, genau dann, wenn gilt: $|M| = |A|$.

5. Zahlentheorie

Teilbarkeit

Teilbarkeitsoperator: $a \mid b \leftrightarrow a$ teilt $b \leftrightarrow \exists n: a \cdot n = b$

Regeln zum Teilbarkeitsoperator:

- $a|b \wedge b|c \rightarrow a|c$
- $a|b \wedge b|a \rightarrow a = b \vee a = -b$
- $a|b \wedge a|c \rightarrow a|(ub + vc)$
- $a|b \vee a|c \rightarrow a|bc$

$\forall a,d \in \mathbb{Z},\ d \neq 0\ \exists q,\ r \in \mathbb{Z}$ eindeutig mit: $a = q \cdot d + r$
Es gilt: $0 \leq r < d$ und wir schreiben $R_d(a) = r$

Theoreme zu ggT und kgV

Grösster gemeinsamer Teiler von 2 oder mehr Zahlen: kleinste positive Zahl, die sich als Linearkombination eben dieser Zahlen darstellen lässt: $d = ggT(a,b) = ua + vb$

Formell: $d|a \ \wedge\ d|b \ \wedge (c|a \wedge c|b \rightarrow c|d)$

Zahlen a,b heissen teilerfremd, falls: $ggT(a,b) = 1$

Erweiterter Euklid-Algorithmus (EEA):

$ggT(24,9) = 3 = (-1) \cdot 24 + 3 \cdot 9$	(24)	(9)
24	1	0
9	0	1
6	1	-2
3	-1	3
0		

Kleinstes gemeinsames Vielfaches von 2 oder mehr Zahlen: kleinste Zahl, die sich als positive Linearkombination der Zahlen darstellen lässt: $l = kgV(a,b)$
Formell: $a|l \ \wedge\ b|l \ \wedge\ (a|m \wedge b|m \rightarrow l|m)$

Zusammenhang zwischen ggT und kgV:

Sei $a = \prod_i p_i^{l_i}$ und $b = \prod_i p_i^{f_i}$ $(l_i, f_i \geq 0)$, dann gilt:

- $ggT(a,b) = \prod_i p_i^{\min(l_i,f_i)}$
- $kgV(a,b) = \prod_i p_i^{\max(l_i,f_i)}$

Ausserdem gilt: $kgV(a,b) \cdot ggT(a,b) = a \cdot b$

Primzahlen und Primfaktorzerlegung (PFZ)

Definition: p prim $\leftrightarrow \forall a > 0\ (a \mid p \rightarrow (a = 1 \vee a = p))$
Es gilt: $p|a \cdot b \rightarrow p|a \vee p|b$

Es gibt unendlich viele PZ. Es gibt beliebig grosse Lücken zwischen 2 auf einander folgende PZ: $[n! + 2, n! + n]$

Primzahldichte: $\pi(n) = |\{1 < k \leq n \mid k\ prim\}|$
Approximation: $\pi(n) \cdot \ln n \approx n \rightarrow \frac{\pi(n)}{n} \approx \frac{1}{\ln n} \rightarrow$ etwa jede 231. hundertstellige Zahl ist eine Primzahl

Jede Zahl besitzt eine eindeutige PFZ: $a = \prod_i p_i^{l_i}$

Modulare Arithmetik

$a \equiv_m b \leftrightarrow a = tm + b \leftrightarrow m|(a - b) \leftrightarrow R_m(a) = R_m(b)$

Rechenregeln: (p prim)

- $R_m(a+b) = R_m\big(R_m(a) + R_m(b)\big)$
- $R_m(a \cdot b) = R_m\big(R_m(a) \cdot R_m(b)\big)$
- **Satz von Fermat(1):** $a^p \equiv_p a \quad \leftrightarrow \quad a^{p-1} \equiv_p 1$
- **Satz von Fermat(2):** $R_p\big(a^b\big) = R_p\big(R_p(a)^{R_{p-1}(b)}\big)$
- **Euler:** $\forall n \in \mathbb{N},\ a \in \mathbb{Z}_n^* \ \rightarrow \ a^{\varphi(n)} \equiv_n 1$

Beispiel einer Anwendung der Restsätze:
$R_5(2007^{2007}) = R_5(R_5(2007)^{2007}) = R_5(2^{2007}) = R_5(8 \cdot 2^{2004}) = R_5\left(R_5(16^{501}) \cdot R_5(8)\right) = R_5\left(3 \cdot R_5(1^{501})\right) = 3$

Ansatz für Restberechnung: Umkehrung des CRS !
Bsp: $x \equiv_{22} a \quad \rightarrow \quad x \equiv_2 R_2(a) \ \wedge \ x \equiv_{11} R_{11}(a)$

Iterierte Quersummen:
Die IQS einer Zahl ist die Ziffer, die man erhält wenn man so lange immer wieder die Quersumme berechnet, bis das Resultat ≤ 9 ist. Es gilt für IQS:

- $IQS(a+b) = IQS\big(IQS(a) + IQS(b)\big)$
- $IQS(a \cdot b) = IQS\big(IQS(a) \cdot IQS(b)\big)$
- $IQS(n) = \begin{cases} 9 & n \neq 0 \ \wedge R_9(n) = 0 \\ 0 & n = 0 \\ R_9(n) & sonst \end{cases}$

IQS-Kontrolle von Rechnungen: falsch → sicher falsch

Struktur und Aufbau der Restklassen:
Die Kongruenz modulo m ist eine Äquivalenzrelation.
Die Partitionierung in Restklassen modulo m bildet $\mathbb{Z}_m$.

$$\mathbb{Z}_m = \{\,[0], [1], \dots, [m-1]\,\}$$

Es gilt für eine Partitionierung: (Klassen $[a]$ und $[b]$)

- $a + b \ \in [a+b]$
- $a \cdot b \ \in [a \cdot b]$
- $a + b \equiv_m a' + b'$
- $a \cdot b \equiv_m a' \cdot b'$

Jede Restklasse von $\mathbb{Z}_m$ besitzt eine **additive Inverse:**

$$\forall [a], \exists [b], \forall a' \in [a], \forall b' \in [b]: \quad a' + b' \in [0]$$

Eine **multiplikative Inverse** existiert genau dann wenn:

$$\exists x: \ [a] \cdot [x] = [1] \ \leftrightarrow \ ggT(a,m) = 1$$

Multiplikative Inversenberechnung mit EEA:

$$ggT(a,m) = 1 = \ u \cdot a + v \cdot m \ \rightarrow \ u = a^{-1} (mod\ m)$$

Definiere: $\mathbb{Z}_m^* = \{[a] \mid ggt(a,m) = 1\}$
$\mathbb{Z}_5^* = \{[1],[2],[3],[4]\} = \mathbb{Z}_5 \setminus \{0\}$

Deshalb gilt: $|\mathbb{Z}_m^*| = \varphi(m)$ und für p prim: $\mathbb{Z}_p^* \equiv \mathbb{Z}_{p-1}$

Der Chinesische Restsatz (CRS):
Seien $m_1 \dots m_r$ paarweise teilerfremd $\big(ggT(m_i, m_j) = 1\big)$ und sei ein x gesucht mit: $x \equiv_{m_1} a_1, \dots, x \equiv_{m_r} a_r$
Dann lassen sich die Bedingungen vereinfachen:
$\leftrightarrow \quad x \equiv_M a \quad ; \quad M = \prod_i m_i$

Zur Berechnung von a geht man wie folgt vor:

$$a = R_M\left(\sum_i a_i \cdot \frac{M}{m_i} \cdot \left(\frac{M}{m_i}\right)^{-1} (mod\ m_i)\right)$$

$\boldsymbol{r = 2}$:
$a = R_M\big(a_1 \cdot m_2 \cdot m_2^{-1} (mod\ m_1) + \ a_2 \cdot m_1 \cdot m_1^{-1} (mod\ m_2)\big)$

$\boldsymbol{r = 3}$:
$a = R_M\big(a_1 m_2 m_3 (m_2 m_3)^{-1} (mod\ m_1) + a_2 m_1 m_3 (m_1 m_3)^{-1} (mod\ m_2) + a_3 m_2 m_1 (m_2 m_1)^{-1} (mod\ m_3)\big)$

Andere Richtung: $a \rightarrow \begin{cases} a_1 = R_{m_1}(a) \\ a_2 = R_{m_2}(a) \end{cases}$

Lemma aus dem CRS:
Betrachte die Menge $\mathbb{Z}_m^*$ mit $m = pq$. Es gelten folgende äquivalente Aussagen:

- $a \in \mathbb{Z}_m^*$
- $ggT(a,m) = 1$
- $ggT(a,p) = 1 \ \wedge \ ggT(a,q) = 1$
- $a \leftrightarrow \binom{a_1}{a_2} \ ; \ ggT(a_1,p) = 1 = ggT(a_2,q)$

Es folgt direkt: $\varphi(m) = |\mathbb{Z}_m^*| = \left|\mathbb{Z}_p^*\right| \cdot |\mathbb{Z}_q^*|$

Allgemein für $n = \prod_{i=1}^r p_i^{l_i}$

- $|\mathbb{Z}_n^*| = \left|\mathbb{Z}_{p_1^{l_1}}^*\right| \cdot \dots \cdot \left|\mathbb{Z}_{p_r^{l_r}}^*\right|$
- $|\mathbb{Z}_n| = \left|\mathbb{Z}_{p_1^{l_1}}\right| \cdot \dots \cdot \left|\mathbb{Z}_{p_r^{l_r}}\right|$

VORSICHT: $|\mathbb{Z}_4^*| \neq |\mathbb{Z}_2^*| \cdot |\mathbb{Z}_2^*| \ \ ; \ \ |\mathbb{Z}_4| \neq |\mathbb{Z}_2| \cdot |\mathbb{Z}_2|$

Neuformulierung des CRS:
PFZ von $n = \prod_i p_i^{l_i}$ dann gilt:

- $\mathbb{Z}_n^* \equiv \mathbb{Z}_{p_1^{l_1}}^* \times \mathbb{Z}_{p_2^{l_2}}^* \times \dots$
- $\mathbb{Z}_n \equiv \mathbb{Z}_{p_1^{l_1}} \times \mathbb{Z}_{p_2^{l_2}} \times \dots$

VORSICHT:
$\mathbb{Z}_4^* \not\equiv \mathbb{Z}_2^* \times \mathbb{Z}_2^*$ weil 2 mal selber Primfaktor
$\mathbb{Z}_4 \not\equiv \mathbb{Z}_2 \times \mathbb{Z}_2$ weil 2 mal selber Primfaktor

6. Algebra

Gruppen:

Sei G eine Menge und $* : G \rightarrow G$ eine Operation auf besagte Menge, so ist $(G,*)$ eine Gruppe, falls:

- **Abgeschlossen** bezüglich $*$
- **Assoziativ:** $\pi_1 * (\pi_2 * \pi_3) = (\pi_1 * \pi_2) * \pi_3$
- **Neutralelement:** $\exists e: \ \pi * e = \ e * \pi = \pi$
- **Invertierbar:** $\forall \pi \exists \pi^{-1}: \ \pi * \pi^{-1} = \ e$

Eine Gruppe heisst **Abelsch**, falls sie kommutativ ist.

Eine Gruppe ist **zyklisch**, falls $\exists g \in G$ mit $ord(g) = |G|$
Beispiel: $\langle a \rangle = \{e, a, a^2, a^3, \dots, a^{ord(a)-1}\}$

Es gilt: g Generator $\leftrightarrow ord(g) = |G|$

Es gilt: $|G|$ prim $\rightarrow$ G zyklisch und $\forall g \neq e: \quad ord(g) = p$

Eigenschaften einer zyklischen Gruppe G, $|G| = \prod_i p_i^{l_i}$:

- $G \equiv \left(\mathbb{Z}_{|G|}, +\right)$
- ihre Untergruppen sind zyklisch
- sie ist Abelsch (kommutativ)
- g Generator von G, falls $\forall i = 1 \dots n: \quad g^{|G|/p_i} \neq e$
- G besitzt genau $\varphi(|G|)$ Generatoren

Symmetriegruppe einer Figur: Drehungen und Spiegelungen, die die Figur invariant lassen (bei Quadrat $i \cdot 90°$ -Drehungen). Die Symmetriegruppe eines regelmässigen n-Ecks enthält immer $2n$ Elemente.

Zwei Gruppen heissen **isomorph**, falls eine bijektive Abbildung existiert, die verträglich mit $*$ ist.
Beispiel: $G_1 = (\mathbb{Z}_2, +)$ und $G_2 = (\{T, F\}, xor) \rightarrow G_1 \equiv G_2$

Gruppen von Ordnung p prim sind eindeutig aufgebaut.

$(H,*)$ heisst **Untergruppe** von $(G,*)$, falls $H \subseteq G$ und H auch eine Gruppe ist bezüglich $*$. Es gilt: $|H| \mid |G|$

Produkt zweier Gruppen $(G,\circ_G)$ und $(K,\circ_K)$:
für $(G \times K,*)$: $(g,k) * (g',k') = (g \circ_G g',\ k \circ_K k')$

Ordnung eines Elements: $ord(a) = \min\{i > 0 \mid a^i = e\}$
$\langle a \rangle = \{e, a, a^2, a^3, \dots, a^{ord(a)-1}\} \rightarrow ord(a) = |\langle a \rangle|$
Es gilt: $a^i * a^j = a^{R_{ord(a)}(i+j)}$; $\left(a^i\right)^{-1} = a^{ord(a)-i}$

Satz von Lagrange:
G beliebige Gruppe: $a \in G \rightarrow a^{|G|} = e$; $ord(a) \mid |G|$

Der diskrete Logarithmus

Geg: $G = \langle g \rangle$, Zahl g^x → Ges: dazugehöriges x
Bsp: Gruppe $\mathbb{Z}_{11}^*$, $2^x \equiv 5$ → $x = 4$

Schwieriges Problem: → nützlich für Kryptographie

Baby-Step, Giant-Step Algorithmus $O\left(\sqrt{|G|} \cdot log(|G|)\right)$:

1. speichere: $g^{x+1}, g^{x+i}, \dots, g^{x+M}$ $\left(M = \sqrt{|G|}\right)$ als Tupel: $(0, g^x)$; $(1, g^{x+1})$; …; (M, g^{x+M}) und sortiere diese nach dem zweiten Wert
2. Berechne $g^0, g^M, g^{2M}, \dots, g^{jM}$ bis man im Intervall der gespeicherten Werte landet
3. Assoziiere g^{jM} mit einem i ,dann: $x = jM - i$

Ringe und Körper

$(K, +, *)$ heisst **Körper**, falls:

- $(K, +)$ abelsche Gruppe mit Neutralelement 0
- $(K \setminus \{0\}, *)$ abelsche Gruppe mit Neut. 1
- Distributivität: $a(b + c) = ab + ac$

Für Körper gilt:

- Charakteristik: $\chi(K) = ord_{(K,+)}(1)$
- $0 \cdot a = 0$
- Nullteilerfrei: $ab = 0 \rightarrow (a = 0 \vee b = 0)$
- $(a + b)^2 = a^2 + (1 + 1)ab + b^2$
- Die multiplikative Gruppe ist immer zyklisch

Jeder endliche Körper besitzt genau p^n Elemente. Jedem p^n kann man einen eindeutigen Körper $GF(p^n)$ zuordnen.

Körper, dessen Multiplikation keine Inverse besitzt: **Ring**

Moduloarithmetik mit Polynomen über Körper

Die Menge $K[x]$ aller Polynome über einen Körper $K = GF(p^n)$ bilden einen **Ring**:

$$K[x] = \left\{\sum_{i=0}^{\infty} a_i x^i \mid a_i \in K\right\} \cup \{0\}$$

Es gilt: $\deg(0) = -\infty$; für $a_r \neq 0$: $\deg\left(\sum_{i=0}^{r} a_i x^i\right) = r$

Es gelten analoge Sätze zur normalen Moduloarithmetik:

- $\forall a(x), b(x) \in K[x], \exists q(x), r(x)$: $a = q \cdot b + r$
- $a(x) \mid b(x) \rightarrow \exists c(x)$: $a(x) \cdot c(x) = b(x)$
- $p(x)$ **irreduzibel** $\leftrightarrow (p(x) = s(x) \cdot t(x) \rightarrow \deg(s) = 0 \vee \deg(t) = 0)$
- $\exists c$: $p(c) = 0 \rightarrow p(x)$ nicht irreduzibel
  Umkehrung gilt nur bis und mit $r = 3$

Über $GF(p)$ existieren irreduzible Polynome jeden Grades.

Definition von Inverse und ggT: gleich wie bei den Zahlen.
Berechnung des ggT zweier Polynome mittels EEA:

In $GF(3)$	$x^3 + 2x + 1$	$2x^2 + x + 2$
$x^3 + 2x + 1$	1	0
$2x^2 + x + 2$	0	1
$2x$	1	$x + 1$
2	$2x + 1$	$1 + (2x + 1)(x + 1)$*
1	$x + 2$	$x^2 + 1$

*weil: $1 \cdot (2x^2 + x + 2) + (2x + 1) \cdot (2x) = 2$

Sei allgemein $q(x)$ irreduzibel in $K = GF(p^n)$, dann bilden die Polynome $K[x]$ über K **modulo $q(x)$ einen Körper**.

Um $GF(p^d)$ aus $GF(p)$ zu konstruieren:

1. schreibe alle Polynome a_i mit $\deg(a_i) < d$ in die Additions- und Multipliationstabelle
2. wähle irreduzibles Polynom q_i d-ten Grades über $GF(p)$ und fülle Tabellen modulo q_i aus

Beispiel für Erweiterungskörper: $\mathbb{R}[x]\ mod(x^2 + 1) \equiv \mathbb{C}$

7. Kryptographie

RSA (Rivest, Shamir, Adleman)

RSA ist ein PK-SK Kryptosystem. Es benutzt:

p, q prim → $n = p \cdot q$ → $a^{(p-1)(q-1)} \equiv_n 1$

Vorgehen des Kryptosystems mit $m \in \mathbb{Z}_n^*$:

- B wählt 2 Primzahlen p, q und eine Zahl e mit $ggT\big(e, (p-1)(q-1)\big) = 1$
- B berechnet $n = pq$ und $d = e^{-1}\big(mod\ \varphi(n)\big)$
- B schickt den PK (n, e) und behält SK (n, d)
- A verschlüsselt Mitteilung $m < n$: $c = R_n(m^e)$
- A schickt c rüber, B entschlüsselt: $m = R_n(c^d)$

Potenzieren mit Square & Multiply: $e = (e_r, e_{r-1}, \dots, e_0)_2$
$m^e = ((((m^{e_r})^2 \cdot m^{e_{r-1}})^2 \cdot m^{e_{r-2}})^2 \dots)^2 \cdot m^{e_o}$

RSA – Digitale Signaturen

Vertrag $v \in \mathbb{Z}_n^*$ mit RSA so signieren, dass die Information verifizierbar nur vom wahren Absender kommen kann.

Vorgehen beim Signieren (A hat den Vertrag):

- PK und SK genau wie beim Kryptosystem
- A berechnet $s = R_n(v^d)$ und schickt v, s rüber
- Für die Verifikation muss gelten: $R_n(s^e) = v$

RSA Schwachstelle: „(p - 1) Algorithmus“

Wenn für eine der beiden Primzahlen gilt: $p - 1$ hat nur kleine Primfaktoren, dann:

- für ein relativ kleines B: $(p-1) \mid B!$
- es gilt: $a^{B!} = a^{k(p-1)} \rightarrow (a^{p-1})^k \equiv_p 1$ und

$$a^{B!} \not\equiv_q 1 \rightarrow \left[p \middle| \left(a^{B!}-1\right) \wedge \neg \left(q \middle| \left(2^{B!}-1\right)\right)\right]$$

- also muss gelten: $p = ggT(2^{B!}-1\ ,\ n)$

Diffie Hellman Verschlüsselung

Dieses symmetrische Kryptosystem basiert auf die Schwierigkeit von diskreten Logarithmen:

1. A wählt eine zyklische Gruppe $G = \langle g \rangle$ und ein beliebiges y. Der SK besteht nur aus dem y.
2. A schickt den PK (G, g, g^y). B wählt ein beliebiges x und schickt den Tupel $(m \cdot g^{xy}, g^x)$ zurück
3. Nun kennen A und B g^{xy}, ein dritter aber nicht
4. Entschlüsselung: $m = (\, m \cdot g^{xy} \cdot (g^{xy})^{-1})$ in G

Das System funktioniert für alle zyklischen Gruppen, aber für $\mathbb{Z}_p^*$ sind sicher keine effizienten Algorithmen bekannt.

8. Lagrange-Interpolation

Theorie

Ein Polynom von Grad n ist mit $> n$ Stützstellen eindeutig bestimmt. Berechnung dieses Polynoms:

1. Bestimme für jede Stützstelle α_j ein Polynom u_i, sodass: $u_i(\alpha_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$
2. Bilde das gesuchte Polynom $p(x) = \sum_{i=0}^{n} \beta_i\, u_i(x)$

Es gilt: $u_i(x) = \frac{(x-\alpha_0)(x-\alpha_1)\ldots(x-\alpha_{i-1})(x-\alpha_{i+1})\ldots(x-\alpha_n)}{(\alpha_i-\alpha_0)(\alpha_i-\alpha_1)\ldots(\alpha_i-\alpha_{i-1})(\alpha_i-\alpha_{i+1})\ldots(\alpha_i-\alpha_n)}$

Anwendung: Secret Sharing

Geheimnis unter n Parteien so verteilen, dass mindestens k Parteien nötig sind, um das Geheimnis zu entziffern.

Erstelle ein Polynom von Grad $k-1$, dessen Auswertung bei $x = 0$ genau das Geheimnis ist. Verteile dann an jede der n Personen einen Share$(i) = (\alpha_i\ ,\ p(\alpha_i))\ ;\ \alpha_i \neq 0$

Codierung allgemein

Ziel: k-Bit-String über verrauschten Kanal verschicken, so dass der Empfänger die Nachricht eindeutig versteht.

Vorgehen:

- Bilde alle k-Strings auf n-Strings ab, wobei $n > k$. Diese bilden die Menge der Codewörter C.
- Schicke man den n-String zum Empfänger. Was bei ihm ankommt, ist wegen verrauschtem Kanal nicht zwingend Codewort. Der Empfänger nimmt dann das Codewort mit der kleinsten Hamming-Distanz zum Empfangenen und dekodiert diesen.
- Erwünschte Eigenschaft: Codewörter liegen möglichst weit auseinander.

Minimaldistanz: $d = \min\ \{\, d_H(u,v)\ \mid\ u, v \in C\ \}$

In einem beliebigen Code kann man:

- $\leq \frac{d-1}{2}$ Fehler korrigieren
- $\leq d-1$ Fehler detektieren

Codierungsstrategien

Lineare Codes:

Ein Code heisst linear, falls:

- symmetrisch
- translationsinvariant

Lineare Codes besitzen eine Generatormatrix $G \in \mathbb{R}^{k \times n}$, um die Strings abzubilden und eine Parity-Check Matrix $H \in \mathbb{R}^{x \times n}$ mit der Eigenschaft: $c \in C \leftrightarrow H \cdot c = 0$

Grundsätzlich kommt beim Empfänger das Wort $c + e$ an. Die Dekodierung erfolgt folgendermassen:

$$H(c+e) = Hc + He = 0 + He$$

He nennt man Syndrom. Im Allgemeinen ist es schwierig, vom Syndrom auf den Fehler zu schliessen, aber durch geschickte Parameterwahl der Codierung ist es möglich.

Beispiele von Codierungen

Hamming-Code (linearer Code über $GF(2)$):

Der Code-Raum ist ein 4-dim Unterraum von $GF(2)^7$.

Die H-Matrix ist: $\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$

Die Zeilen von H bilden eine Basis des 3-dim Orhtogonalraums auf den Code-Raum C.

Die Minimaldistanz ist die kleinste Hammingdistanz zweier Codewörter, also auch das kleinste Hamminggewicht eines Codeworts (aufgrund der Linearität und wegen $GF(2)$).

Da mindestens 3 Spalten von H linear abhängig sind, ist 3 auch die minimale Anzahl „Einsen“, die man baucht, um ein Codewort ($Hc = 0$) zu erzeugen. Somit gilt: $d = 3$

Nun fehlt nur noch die Generatormatrix G. Es ist erwünscht, dass die linke 4×4 Matrix die Einheitsmatrix ist, damit die ersten 4 Bits eines Codeworts die eigentliche Botschaft darstellen.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & ? & ? & ? \\ 0 & 1 & 0 & 0 & ? & ? & ? \\ 0 & 0 & 1 & 0 & ? & ? & ? \\ 0 & 0 & 0 & 1 & ? & ? & ? \end{pmatrix}$$

Die 4 Zeilen müssen eine Basis des Code-Raums sein, also braucht man 4 linear unabhängige Codes in den Zeilen. Deshalb muss für jede Zeile gelten: $H \cdot g_i = 0$
Man erhält:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Optimale Codes (Lagrange-Interpolation):

Die maximale erreichbare Minimaldistanz bei gegebenem n, k beträgt: $d \leq n - k + 1$. Optimale Codes sind Codes, die genau diesen Wert erreichen.

Ein optimaler Code entsteht durch Polynomauswertung. Man konstruiert aus einer k-Array-Botschaft $[a_0, \ldots, a_{k-1}]$ ein Polynom $p(x) = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1}$.

Die Codierung besteht darin, dieses Polynom an n verschiedene Stellen auszuwerten: $x_0 \ldots x_n$

Die Dekodierung ist allgemein NP-vollständig, aber durch leichte Abänderungen dieses Codes kann man es einfacher machen.

Ein Beispiel dafür ist der Reed-Solomon-Code.