Diskrete Mathematik

Die wichtigsten Kenntnisse in diskreter Mathematik, die einen Bezug zur Informatik haben

Leon Muscat

Inhaltsverzeichnis

0 Einführung

In dieser Vorlesung werden die wichtigsten Kenntnisse in diskreter Mathematik, die einen Bezug zur Informatik haben, behandelt. Dabei geht es sowohl um die mathematische Theorie über die natürlichen Zahlen, endliche Körper, Kombinatorik und Graphentheorie, sowie deren Anwendungen in Informationstheorie, Kryptographie und Komplexitätstheorie.

0.1 Lernziele

Studierende sollten durch diesen Kurs…

- Grundlagen der Zahlentheorie kennen, und mit dieser Kenntnis fehlerkorrigierende Prüfziffern sowie das RSA-Verschlüsselungsverfahren erstellen können,
- Polynome und endliche Körper kennen, und mit diesen fehlerkorrigierende Codes für digitale Kommunikation erstellen können,
- mit kombinatorischen Argumenten diskrete Wahrscheinlichkeiten berechnen können,
- Grundlagen in Graphentheorie verstanden haben, und diese im Algorithmendesign anwenden können.

0.2 Prüfung

Es gibt jede Woche während der Vorlesungszeit Hausaufgaben, die insgesamt 20% der finalen Note ausmachen.

Die Zentrale Prüfung macht 80% der finalen Note aus und ist 120 Minuten lang. Als Hilfsmittel erlaubt ist ein Taschenrechner und ein selbsterstelltes DIN A4 Blatt, dass beidseitig beschrieben ist.

0.3 Literatur

Die Vorlesung orientiert sich an dem 3. Kapitel des Buches “Mathematik für Informatiker Band 1 (4. Auflage)” von Teschl.

1 Zahlensysteme

$$\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$$

Die natürlichen Zahlen:

$$\mathbb{N} = \{1, 2, 3, 4, ... \}$$

Die ganzen Zahlen:

$$\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$$

Die rationalen Zahlen:

$$\mathbb{Q} = \{\frac{p}{q} | q \neq 0, \quad p, q \in \mathbb{Z}\}$$

Alle rationalen Zahlen können als jene Dezimalzahlen dargestellt werden, die endlich darstellbare Nachkommastellen haben (endlich heißt endlich viele oder periodisch).

z.B.:

$$\frac{5}{27} = 0.\overline{185} = \frac{185}{999}$$

$$0.1\overline{6} = \frac{1}{10} + \frac{1}{10} * 0.\overline{6} = \frac{1}{10} + \frac{1}{10} * \frac{6}{9} = \frac{15}{90}$$

Daraus folgt, dass $0.\overline{9} = \frac{9}{9} = 1$

Die reellen Zahlen:

$$\mathbb{R} = \{\pm a_n a_{n-1} \dots a_0 . a_{-1} a_{-2} \dots | a_j \in \{0, 1, \dots, 9\}\}$$

Die reellen, aber nicht rationalen Zahlen, wie $\sqrt{2}$, nennt man irrational.

Die komplexen Zahlen:

$$\mathbb{C} = \{x + iy | x, y \in \mathbb{R}, i^2 = -1\}$$

1.1 Generalisierung des Dezimalsystems

Zahlen im Dezimalsystem sind dargestellt durch

$$a_n \dots a_0 . a_{-1} \dots a_{-m} = \sum_{j=-m}^{n} a_j 10^j.$$

Wir können aber auch jede andere Basis nehmen, daher kann für $b \in \mathbb{N}$ die b-äre Zahl dargestellt werden durch

$$(x_n \dots x_0)_b := \sum_{j=0}^{n} x_j b^j. \quad (x_i \in \{0, 1, \dots, b-1\})$$

(Wir verzichten auf Nachkommastellen, der Einfachheit halber.)

In der Informatik benutzen wir meistens das Binärsystem ($b = 2$), Oktalsystem ($b = 2^3 = 8$) oder das Hexadezimalsystem ($b = 2^4 = 16$).

Für die Umrechnung zwischen diesen Basen, siehe hier.

2 Moduloarithmetik

Dieser Teil orientiert sich an dem Buch Kapital 3.

Definition: Seien $a \in \mathbb{Z}, m \in \mathbb{N}$, dann gibt es eindeutige Zahlen $q, r \in \mathbb{Z}$ mit $0 \leq r < m$, so dass

$$a = q \cdot m + r.$$

Man nennt m den **Modul**, r den **Rest modulo m** und schreibt

$$a = r \mod m \quad \text{oder} \quad a \equiv r \mod m$$

Wenn zwei ganze Zahlen $a, b \in \mathbb{Z}$ bei Division durch m denselben Rest haben, so sind a und b **kongruent modulo m**. Man schreibt

$$a \equiv b \mod m \quad \text{oder} \quad a = b \mod m$$

Theorem (Buch Satz 3.2):

$$\begin{aligned} a &= b \mod m \\ a - b &= 0 \mod m \\ a - b &= km \quad k \in \mathbb{Z} \end{aligned} \tag{1}$$

Definition: Alle Zahlen $c \in \mathbb{Z}$, die $c = b \mod m$ erfüllen, bilden die **Restklasse** von $b \mod m$. Z.B. Restklassen $\mod 5$ sind:

$$\{\dots, -15, -10, -5, 0, 5, 10, 15, \dots\}$$

$$\{\dots, -14, -9, -4, 1, 6, 11, 16, \dots\}$$

$$\{\dots, -13, -8, -3, 2, 7, 12, 17, \dots\}$$

$$\{\dots, -12, -7, -2, 3, 8, 13, 18, \dots\}$$

$$\{\dots, -11, -6, -1, 4, 9, 14, 19, \dots\}$$

Die Standardrepräsentanten der Restklassen sind die Zahlen $0, 1, \dots, m-1 \mod m$.

Theorem (Buch Satz 3.4): Sei $a = b \mod m$ und $c = d \mod m$, dann gilt

$$\begin{aligned} a + c &= b + d \mod m \\ a \cdot c &= b \cdot d \mod m \end{aligned} \tag{2}$$

Beweis:

$$\begin{aligned} a = b \mod m &\iff \exists l \in \mathbb{Z} : a = lm + b \\ c = d \mod m &\iff \exists k \in \mathbb{Z} : c = km + d \\ a \cdot c &= \underbrace{(lkm + dl + bk)}_{\in \mathbb{Z}} m + bd \end{aligned} \tag{3}$$

Das bedeutet, wir können während den Operationen jederzeit modulo m reduzieren und dann weiterrechnen.

2.1 Primzahlen

Definition: Eine ganze Zahl $a \in \mathbb{Z}$ heißt durch $b \in \mathbb{N}$ teilbar, falls es eine ganze $n \in \mathbb{Z}$ gibt, s.d. $b \cdot n = a$. Man schreibt dann $b|a$. Eine natürliche Zahl $p > 1$, die nur durch sich selbst und 1 teilbar ist, heißt **Primzahl**.

Jede natürliche Zahl lässt sich als Produkt von Primzahlen schreiben. Diese **Primfaktoren** sind (bis auf die Reihenfolge) eindeutig. Z.B.: $180 = 3 \cdot 60 = 2^2 \cdot 3^2 \cdot 5$

Theorem: Es gibt unendlich viele Primzahlen

Beweis: Wir nehmen in Gegenteil an, dass es endlich viele Primzahlen $p_1, p_2, \dots, p_n$ gibt, für welche ein fixes $n \in \mathbb{N}$. Dann betrachten wir die Zahl

$$x = p_1 \cdot p_2 \cdot \dots \cdot p_n + 1$$

Wenn wir x durch p_i (für $i = 1, \dots, n$) teilen, erhalten wir immer den Rest 1, d.h. x ist nicht durch $p_1, p_2, \dots, p_n$ teilbar. D.h. x ist durch keine Primzahl teilbar und muss selber prim sein. Da aber $x > p_i$ für alle i, erhalten wir einen Widerspruch dazu, dass $p_1, \dots, p_n$ alle Primzahlen sind. □

Definition: Wenn 2 natürliche Zahlen keinen gemeinsamen Teiler außer 1 haben, nennt man sie **teilerfremd**.

Wir beschreiben den größten gemeinsamen Teiler von $a, b \in \mathbb{N}$ als ggT(a, b) (englisch gcd(a, b)). Der ggT ist zudem immer der größte von beiden geteilte Wert in einer Primfaktorzerlegung. Sind a und b teilerfremd, dann ist der ggT(a, b) $= 1$.

2.2 Anwendungsbeispiele

2.2.1 Wochentageformel

$$0 \cong \text{Montag}, 1 \cong \text{Dienstag}, 2 \cong \text{Mittwoch}, \dots$$

Wenn der 1.1.1900 ein Montag war, was für ein Wochentag war der 15.5.1955?

$$\underbrace{\underbrace{\underbrace{55 \cdot 365}_{\text{komplette Jahre}} + \underbrace{13}_{\text{Schaltjahre}}}_{\text{bis 1.1.1955}} + \underbrace{31 + 28 + 31 + 30 + 14}_{\text{bis 1.5.1955}}}_{} \mod 7$$

$$= (-1) \cdot 1 + (-1) + 3 + 0 + 3 + 2 + 0 \mod 7$$

$$= 6 \cong \text{Sonntag}$$

2.2.2 Prüfziffern

Prüfnummern werden häufig bei Identifikationsnummern benutzt, z.B. IBAN, ISBN, … Bei ISBN gibt es das alte ISBN-10 und das neue ISBN-13 System. In beiden Systemen geben die ersten Ziffern Informationen über das Buch, aber die letzte Ziffer ist die Prüfziffer. Bei ISBN-13 wird diese wie folgt berechnet:

$$z_{13} = -(z_1 + z_3 + z_5 + \cdots + z_{11}) + 3 \cdot (z_2 + z_4 + \cdots + z_{12}) \mod 10$$

Der ISBN-13-Code sieht dann wie folgt aus:

$$z_1 z_2 z_3 - z_4 - z_5 z_6 - z_7 z_8 z_9 z_{10} z_{11} z_{12} - z_{13}$$

Zum Überprüfen, ob ein ISBN-13-Code gültig ist, muss man die folgende Gleichung stimmen:

$$(z_1 + z_2 + z_5 + \cdots + z_{11} + z_{13}) + 3 \cdot (z_2 + z_4 + \cdots + z_{12}) = 0 \mod 10$$

Der ISBN-13-Code kann die meisten Vertauschungen von 2 aufeinanderfolgenden Ziffern erkennen. Das liegt daran, dass die Zahlen nach gerade und ungerade aufteilt werden und einen anderen Multiplikator haben.

Bei ISBN-10 wird die letzte Ziffer wie folgt berechnet:

$$\begin{aligned} z_{10} &= -\sum_{i=1}^{9} (11 - i) z_i \mod 11 \\ &= -(10 z_1 + 9 z_2 + \cdots + 2 z_9) \mod 11 \end{aligned} \tag{4}$$

Die Prüfgleichung sieht daher wie folt aus:

$$\sum_{i=1}^{10} (11 - i) z_i = 0 \mod 11$$

Falls $z_{10} = 10 \mod 10$ ist, schreibt man "X".

Theorem: Der ISBN-10 Code kann jeden Einzelfehler erkenn (also wo eine Ziffer falsch ist) oder jede Vertauschung von zwei Ziffern erkennen.

Beweis:

1. **Einzelfehler**: Wir nehmen an, dass $z_1, z_2, z_3, \dots, z_{10}$ eine gültige ISBN-10 ist und dass für ein $i \in \{1, \dots, 10\}$ mit $y_i \neq z_i$, aber für alle anderen indizes j gilt $y_j = z_j$. Also ist $y_1.y_2, \dots, y_{10}$ die ISBN von oben mit einem Fehler: Dann setzen wir $y_1.y_2, \dots, y_{10}$ in die Prüfgleichung ein:

$$\begin{aligned}\sum_{j=1}^{10}(11-j)y_j \mod 11 &= \sum_{j\neq i}(11-j)y_j + y_i(1-i) \mod 11 \\ &= \sum_{j\neq i}(11-j)z_j + y_i(1-i) \mod 11 \\ &= -(11-i)z_i + y_i(11-i) \mod 11 \\ &= \underbrace{(11-i)}_{\neq 0}\underbrace{(y_i - z_i)}_{\neq 0} \mod 11\end{aligned} \tag{5}$$

Der letzte Schritt folgt aus Theorem 3.18, da 11 eine Primzahl ist und daher teilerfremd mit allen Zahlen in $\pm\{1, 2, \dots, 10\}$. □

2. **Vertauschungsfehler**: Falls wir z_l und z_i vertauschen, also $y_i = z_l$ und $y_l = z_i$ (und $y_1, \dots, y_{10}$ die falsche ISBN), dann:

$$\begin{aligned}\sum_{j=1}^{10}(11-j)y_j \mod 11 &= \sum_{j\neq i,l}(11-j)y_j + (11-l)y_l + (11j)y_j \mod 11 \\ &= -((11-l)z_l + (11-j)z_j) + (11-l)z_j + (11-j)z_l \mod 11 \\ &= (11-l)(z_j - z_l) + (11-j)(z_l - z_j) \mod 11 \\ &= (11-l-11+j)(z_j - z_l) \mod 11 \\ &= \underbrace{(j-l)}_{\neq 0}(z_j - z_l) \mod 11\end{aligned} \tag{6}$$

2.2.3 Crpytosysteme

Caeser-Verschlüsselung: Die Caesar-Verschlüsselung ist eine der einfachsten Verschlüsselungsmethoden und wurde bereits von Julius Caesar verwendet. Bei dieser Methode wird jeder Buchstabe des Klartexts um eine feste Anzahl von Positionen im Alphabet verschoben, um den verschlüsselten

Text zu erzeugen. Diese feste Anzahl wird als “Schlüssel” bezeichnet und kann beispielsweise 3 sein, was bedeutet, dass jeder Buchstabe im Klartext um 3 Stellen im Alphabet verschoben wird.

Zum Beispiel würde das Wort “HELLO” bei einem Schlüssel von 3 zu “KHOOR” verschlüsselt werden. Der Empfänger der verschlüsselten Nachricht kann den Text wieder entschlüsseln, indem er jeden Buchstaben um die gleiche Anzahl von Positionen im Alphabet zurückverschiebt.

Es gibt noch weitere einfache Verschlüsselungsarten, wie Vignère-Verschlüsselung (siehe Folien).

2.2.4 Schnelle Mathematik (Exkurs, sehr wahrscheinlich nicht prüfungsrelevant)

Modulo erlaubt es schnell zu überprüfen, ob eine Zahl durch x teilbar ist. Schaue dafür dieses Video bis Minute 13.

2.3 Multiplikative Inverse modulo m

Für eine Zahl $x \in \mathbb{R}\backslash\{0\}$ bezeichnet x^{-1} die Zahl in $\mathbb{R}\backslash\{0\}$, die erfüllt:

$$x \cdot x^{-1} = 1$$

Man schreibt auch $x^{-1} = \frac{1}{x}$.

In $\mathbb{Z}$ gibt es, ausser für $x = 1$ und $x = -1$, keine multiplikativen Inverse, allerdings können wir modulo m “mehr finden”.

Definition: Sei $x \in \{1, \dots, m-1\}$. Dann ist das multiplikative Inverse modulo m die Zahl $x^{-1} \in \{1, \dots, m-1\}$, s.d. $x \cdot x^{-1} = 1 \mod m$.

Beispiel:

$$\begin{aligned} 2 \cdot 5 &= 1 \mod 9 \\ \text{also ist } 2^{-1} &= 5 \mod 9 \text{ und } 5^{-1} = 2 \mod 9 \end{aligned} \tag{7}$$

Das heißt, bei der modulo Rechnung ist **Division** $\cong$ **Multiplikation mit Inversen**. Z.B.:

$$\begin{aligned} 3 \cdot 2^{-1} = 3 * 5 &= 15 \mod 9 \\ &= 6 \mod 9 \end{aligned} \tag{8}$$

Theorem: Das multiplikative Inverse zu x modulo m existiert genau dann wenn $ggt(x, m) = 1$ (also x und m teilerfremd sind). Fall es existiert, ist es (zwischen 1 und $m-1$) eindeutig.

Beweis: Wir suchen das x^{-1} in der Gleichung $x \cdot x^{-1} = 1 \mod m$.

$$\begin{aligned} x \cdot x^{-1} = 1 \mod m &\iff \exists k \in \mathbb{Z} : x \cdot x^{-1} = 1 + km \\ &\iff \exists k \in \mathbb{Z} : x \cdot x^{-1} - km = 1 \end{aligned} \tag{9}$$

Falls $ggT(x, m) = l > 1$, dann ist $\frac{x}{l} \in \mathbb{Z}$ und wir können l ausklammern:

$$x \cdot x^{-1} - km = l \underbrace{(\frac{x}{l} x^{-1} - k\frac{m}{l})}_{=y \in Z} = 1,$$

also $l \cdot y = 1$ mit $l > 1$ und $l, y \in \mathbb{Z}$. $\lightning$

Falls $ggT(x, m) = 1$, dann sind alle Vielfachen von x ($x, 2x, 3x, \dots, (m-1)x$ unterschiedlich modulo m (ab Zahlen größer m-1 wiederholen sich die Reste); falls nämlich $ax = bx \mod m$, dann gibt es $k \in Z$:

$$ax = km + bx \iff (a - b)x = km$$

Da x und m teilerfremd sind, muss $(a - b)$ ein Vielfaches von m sein und damit

$$\begin{aligned} a - b = 0 \mod m \\ \iff a = b \mod m \end{aligned} \tag{10}$$

D.h. $ax = bx \mod m \to a = b \mod m$ und deshalb ist $ax \neq bx \mod m$, falls $a \neq b \mod m$. Da $ax \neq 0 \mod m$ sein muss (für $1 \leq a \leq m - 1$ mit $ggT(x, m) = 1$ kann ax kein Vielfaches von m sein), muss x^{-1} genau eine Zahl $\in \{1, 2, \dots, m - 1\}$ sein.

Theorem: Wenn $ggT(a, m) = 1$, dann besitzt $ax = b \mod m$ genau eine Lösung in $\mathbb{Z}$ für x. Die Lösung ist $x = ba^{-1} \mod m$. Aber: Falls $ggT(a, m) > 1$, dann gilt es keine oder mehrere Lösungen (nämlich $ggT(a, m)$ viele, falls $ggT(a, b)|b$.

2.4 Euklid-Algorithmus

Der Euklidische Algorithmus ist ein effizientes Verfahren zur Bestimmung des größten gemeinsamen Teilers (ggT) zweier ganzer Zahlen. Er wurde von dem antiken griechischen Mathematiker Euklid in seinem Werk “Die Elemente” beschrieben.

Der Algorithmus basiert auf der Eigenschaft, dass der ggT zweier Zahlen auch der ggT von einer der Zahlen und dem Rest der Division der anderen Zahl durch diese ist. Dies wird wiederholt, bis der Rest Null ist, wobei der ggT die letzte Nicht-Null-Restzahl ist.

Der Euklidische Algorithmus kann formalisiert werden als:

1. Gegeben seien zwei natürliche Zahlen (a) und (b) mit (a > b).
2. Teile (a) durch (b) und erhalte den Rest (r).
3. Wenn (r = 0), dann ist (b) der ggT von (a) und (b).
4. Andernfalls setze (a = b) und (b = r) und gehe zurück zu Schritt 2.

In mathematischer Notation kann dies ausgedrückt werden als:

$$a = bq + r, \quad 0 \leq r < b$$
$$\text{ggT}(a, b) = \text{ggT}(b, r)$$

wobei (q) der Quotient und (r) der Rest ist.

Beispiel für den Euklidischen Algorithmus, um den größten gemeinsamen Teiler (ggT) von 48 und 18 zu finden:

$$48 = 18 \cdot 2 + 12$$
$$18 = 12 \cdot 1 + 6$$
$$12 = 6 \cdot 2 + 0$$

In diesem Fall ist der ggT von 48 und 18 die letzte Nicht-Null-Restzahl, also 6.

2.4.1 Erweiterte Euklidische Algorithmus

Der erweiterte Euklidische Algorithmus berechnet zusätzlich zu dem ggT die Koeffizienten x und y von Bézout's Identität, d.h. er findet $x, y \in \mathbb{Z}$, so dass $ax + by = ggT(a, b)$. Im Kontext der Modulararithmetik ist er besonders nützlich, da er zur Berechnung von multiplikativen Inversen verwendet wird.

Dafür berechnen wir neben dem ggT auch noch u_i und v_i durch

$$u_i = u_{i-2} - u_{i-1}q_i$$
$$v_i = v_{i-2} - v_{i-1}q_i$$

Die Startwerte sind $u_{-1} = 1, u_0 = 0$ und $v_{-1} = 0, v_0 = 1$

Wie zuvor, sobald der Rest $r_k = 0$ für ein beliebiges k, dann gilt

$$u_{k-1}m + v_{k-1}n = ggT(m, n)$$

Beispiel für den eggT(48,39)

i	$q_i r_{i-1} + r_i$	u_i	v_i
-1		1	0
0		0	1
1	$48 = 1 \cdot 39 + 9$	1	-1
2	$39 = 4 \cdot 9 + 3$	-4	5
3	$9 = 3 \times 3 + 0$		

Der $eggT(48, 39) = (3, -4, 5)$. Der ggT ist also 3 und es gilt $-4 \cdot 48 + 5 \cdot 39 = 3$.

2.5 Zusammenfassung Modulo Rechnung

Man kann modulo m addieren, subtrahieren und multiplizieren (und an jeder Stelle mod m reduzieren).

Nur die Division ist schwieriger, aber Division $\cong$ Multiplikation mit Inversen:

$$x \cdot x^{-1} = 1 \mod m$$

Das Inverse x^{-1} existiert $\iff ggT(x, m) = 1$.

Eine effiziente Art das Inverse zu berechnen ist der erweiterte Euklidische Algorithmus, der generell s, t und $ggT(x, m)$ findet, s.d.

$$sx + tm = ggT(x, m)$$

Falls $ggT(x, m) = 1$, dann $sx + tm = 1$ und daher

$$sx + tm = 1 \mod m$$

$$sx = 1 \mod m$$

$$s = x^{-1} \mod m$$

Gleiches gilt auch für t: $t = m^{-1} \mod x$.

3 RSA Kryptographie

RSA ist eine asymmetrische Kryptographie-Methode, die zur Verschlüsselung und Entschlüsselung von Daten verwendet wird.

Es funktioniert, indem es ein Paar von Schlüsseln verwendet: einen öffentlichen und einen privaten Schlüssel. Der öffentliche Schlüssel kann frei verteilt werden und wird verwendet, um Daten zu verschlüsseln. Der private Schlüssel wird vom Empfänger gehalten und wird verwendet, um die verschlüsselten Daten zu entschlüsseln.

Zum Verschlüsseln von Daten mit RSA wird der öffentliche Schlüssel des Empfängers verwendet. Der Sender codiert die Daten mithilfe des öffentlichen Schlüssels, wodurch sie für jeden anderen unlesbar werden. Der Empfänger kann dann den verschlüsselten Text mit seinem privaten Schlüssel entschlüsseln, um die ursprünglichen Daten wiederherzustellen.

Die Sicherheit von RSA hängt von der Schwierigkeit der Faktorisierung großer Primzahlen ab. Je größer die Primzahlen sind, desto schwieriger wird es, den privaten Schlüssel zu berechnen, der zum Entschlüsseln der Daten verwendet wird.

3.1 Mathematischer Ansatz

Ein kryptographisches System basiert auf eine Rechnung, die einfach zu kalkulieren ist, aber sehr schwer zurückverfolgbar ist.

Das Ergebnis der Rechnung ist der Public Key, während das Urbild der Rechnung den Private Key bildet.

Eine solche Rechnung ist die Multiplikation von 2 Primzahlen. Es ist leicht das Produkt zu berechnen, aber bei sehr großen Primzahlen, ist es schwer die ursprünglichen Primzahlen des Produktes herauszufinden. Das ist der Ansatz von RSA Kryptographie.

3.2 Rechnung

- Wähle zwei große Primzahlen p und q.
- Berechne das Produkt $n = p \cdot q$.
- Wähle eine Zahl e, die coprime zu $(p-1)(q-1)$ ist.
- Berechne das Inverse mit dem Euklid-Algorithmus

$$d = e^{-1} \mod (p-1)(q-1)$$

Der Private Key ist d und der Public Key sind die Zahlen n und e.

Um eine Nachricht zu verschlüsseln muss sie zu einer Zahl m überführt werden und dann folgenderweise verschlüsselt werden:

$$c = m^e \mod n$$

Die Entschlüsselung funktioniert folgendermaßen:

$$m = c^d \mod n$$

Ein Video mit Beispielen hier.

3.3 Beweis von RSA

Siehe Folien RSA ab S. 24.

3.3.1 Chinesischer Restsatz (CRT)

Theorem: Sind zwei Zahlen n_1 und n_2 coprime, dann existiert eine einzige Lösung $\mod n_1 n_2$ für die Gleichungen

$$\begin{aligned} x &= a \mod n_1 \\ x &= b \mod n_2. \end{aligned} \tag{11}$$

Die Lösung kann gefunden werden durch

$$x = atn_2 + bsn_1 \mod n_2 n_1$$

. Die Zahlen s und t ergeben sich durch den erweiterten Euklidischen Algorithmen von $sn_1 + tn_2 = 1$.

3.3.2 Fermats Theorem

Theorem: Wenn p eine Primzahl ist, dann gilt für jede Zahl a, dass

$$a^{p-1} = 1 \mod p$$

3.3.3 Eulers Theorem

Die Euler-Funktion $\varphi(n)$ gibt an wie viele Zahlen $< n$, die coprime zu n sind, existieren.

Wenn p eine Primzahl ist, dann ist $\varphi(p) = p - 1$.

Außerdem gilt für ein $e \in \mathbb{N}$, dass $\varphi(p^e) = p^{e-1}(p-1)$.

Wenn p und q coprime sind, dann ist $\varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$.

Eulers Theorem besagt, dass für 2 Zahlen a und n mit $gcd(a, n) = 1$ gilt

$$a^{\varphi(n)} = 1 \mod n.$$

4 Polynomringe und endliche Körper

Wir wollen (algebraische) Strukturen unterscheiden z.B. um festzulegen, ob ein multiplikatives Inverse immer existiert oder nicht.

Definition (Buch 3.22): Sei G eine Menge mit einer Verknüpfung $\circ$, die jedem Paar von Elementen a, b aus G ein Element $a \circ b \in G$ zuordnet. Dann wird $(G, \circ)$ eine **Gruppe** genannt, wenn

a) $(a \circ b) \circ c = a \circ (b \circ c)$ (assoziativ)
b) $\exists n \in G : \quad n \circ a = a \circ n = a \quad \forall a \in G$ (neutrales Element n ist 1 bei Multiplikation und 0 bei Addition)
c) $\forall a \in G \exists a^{-1} \in G : a \circ a^{-1} = a^{-1}a = n$ (inverse Element)

Gilt zusätzlich

d) $a \circ b = b \circ a \quad \forall a, b \in G$ (kommutativ),

dann nennt man $(G, \circ)$ eine **kommutative** (oder Abelsche) **Gruppe**.

Beispiele:

1. Additive Gruppen: $(\mathbb{Z}, +), (\mathbb{Z}_m, +), (\mathbb{Q}, +), (\mathbb{R}, +), H_n = (\{2n | n \in \mathbb{Z}\}, +)$ sind alles kommutative Gruppen. Aber $(\mathbb{N}, +)$ nicht, da c) nicht gegeben ist.
2. Multiplikative Gruppen: $(\mathbb{R}, \cdot), (\mathbb{Q}\backslash\{0\}, \cdot)$ sind kommutative Gruppen. Aber $(\mathbb{Z}, \cdot), (\mathbb{N}, \cdot)$ nicht, da c) nicht gegeben ist.

4.1 Körper

Definition: Eine Menge $\mathbb{K}$ mit zwei Verknüpfungen $+$ und $\cdot$, geschrieben $(\mathbb{K}, +, \cdot)$, heißt Körper, wenn

a) $(\mathbb{K}, +)$ ist eine kommutative Gruppe mit neutralem Element 0.
b) $(\mathbb{K}\backslash\{0\}, \cdot)$, ist eine kommutative Gruppe mit neutralem Element 1.
c) $\forall a, b, c \in \mathbb{K} : a \cdot b + a \cdot c = a \cdot (b + c)$ (Distributivgesetz).

Beispiele:

- $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ sind Körper

- $\mathbb{Z}_p$ mit p prim ist ein Körper
- $\mathbb{Z}, \mathbb{Z}_m$ (mit m nicht prim) sind keine Körper, aber Ringe

4.2 Ringe

Eine Menge $\mathbb{R}$ mit Verknüpfungen $+, \cdot$, heißt **Ring**, wenn:

a) $(R, +)$ ist eine kommutative Gruppe mit neutralem Element 0.
b) $\forall a, b, c \in R : (a \cdot b) \cdot c = a \cdot (b \cdot c)$ (assoziativ).
c) $\forall a, b, c \in R : a \cdot b + a \cdot c = a \cdot (b + c)$ (distributiv).

Gilt zusätzlich

d) $\forall a, b \in R : a \cdot b = b \cdot a$
e) $\exists 1 \in R : 1 \cdot a = a \cdot 1 = a \quad \forall a \in R$

dann ist R ein **kommutativer Ring mit Eins**.

Bemerkung: Ein Körper ist ein kommutativer Ring mit 1, wo jedes Element $\neq 0$ ein multiplikatives Inverse besitzt.

Beispiele:

a) $(\mathbb{Z}, +, \cdot)$ ist ein kommutativer Ring mit Eins
b) $(\mathbb{Z}_m, +, \cdot)$ mit m nicht prim, ist ein kommutativer Ring mit Eins
c) Die Menge der $(n \times n)$ Matrizen mit Einträgen aus $\mathbb{R}$ ist ein Ring mit Eins (der Ring ist also nicht-kommutativ, da d) nicht gilt).

4.3 Polynome

Eine Funktion $p : \mathbb{K} \to \mathbb{K}$ der Form

$$p(x) = \sum_{i=0}^{d} a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

mit $a \in \mathbb{K}$ und $d \in \mathbb{N}_0$ heißt Polynom. Wenn $a_d \neq 0$, dann ist $d = deg(p)$ der Grad von $p(x)$ (der Grad von Nullpolynomen ist als $-\infty$ festgelegt). Die a_i sind die Koeffizienten.

Ein Polynom mit $a_d = 1$ heißt normiert oder monisch.

Lemma: Die Menge $\mathbb{K}[x]$ (Koeffizienten sind aus K) mit

$$\mathbb{K}[x] = \{\sum_{i=0}^{j} a_i x^i | a_i \in \mathbb{K}\},$$

ist ein kommutativer Ring mit Eins (bezüglich der bekannten Polynomaddition und -multiplikation). Er wird auch Polynomring über $\mathbb{K}$ genannt. (Neutrales Element bzgl. $+$ ist $p(x) = 0$ und das neutrale Element bzgl. $\cdot$ ist $p(x) = 1$.)

Beispiele:

- $x^3 - 5x^2 + \sqrt{2} \in \mathbb{R}[x]$
- $x^4 + 2x^3 + 1 \in \mathbb{Z}_3[x]$ (Operationen werden in $\mathbb{Z}_3$, also mod 3 ausgeführt.)

4.3.1 Rechnen in $\mathbb{Z}_p[x]$ (p prim)

Beispiel in $\mathbb{Z}_2[x]$: $p(x) = x^3 + x$ und $q(x) = x + 1$

a) $p(x) + q(x) = x^3 + x + x + 1 = x^3 + 1 \mod 2$
b) $p(x)q(x) = (x^3 + x) \cdot (x + 1) = x^4 + x^3 + x^2 + x \mod 2$

Beispiel in $\mathbb{Z}_3[x]$: $p(x) = x^2 + 2x + 1$ und $q(x) = x + 2$

b)

$$\begin{aligned} p(x)q(x) &= x^3 + 2x^2 + x + 2x^2 + 4x + 2 \mod 3 \\ &= x^3 + 4x^2 + 5x + 2 \mod 3 \\ &= x^3 + x^2 + 2x + 2 \mod 3 \end{aligned} \tag{12}$$

4.3.2 Polynomdivision

Für zwei Polynome $p(x), q(x) \in \mathbb{K}[x]$ mit $deg(q) \leq deg(p)$ gibt es $s(x), r(x) \in \mathbb{K}[x]$, so dass

$$p(x) = s(x) \cdot q(x) + r(x)$$

und $deg(r) < deg(q)$. (Der Rest hat möglichst kleinen Grad.)

Beispiel:

$$\begin{array}{l} \Big(\quad 3x^4 + x^3 \qquad - 2x \Big) : \Big(x^2 + 1 \Big) = 3x^2 + x - 3 + \dfrac{-3x + 3}{x^2 + 1} \\ \underline{-3x^4 \qquad - 3x^2} \\ \qquad\quad x^3 - 3x^2 - 2x \\ \qquad \underline{-x^3 \qquad\quad - x} \\ \qquad\quad - 3x^2 - 3x \\ \qquad\quad \underline{3x^2 \qquad\quad + 3} \\ \qquad\qquad - 3x + 3 \end{array}$$

Wir würden in der Schreibweise von oben sagen, dass $q(x)$ $p(x)$ teilt, wenn $r(x) = 0$. Allerdings ist dann auch jedes skalare Vielfache von $p(x)$ ein Teiler von $q(x)$. Wir konzentrieren uns deshalb auf die normierten Teiler eines Polynoms (also mit $a_d = 1$, wenn $d = deg(r)$).

Beispiel: Die normierten Teiler von $x^2 - 8x + 15 = (x-5)(x-3)$ sind $1, (x-3), (x-5), x^2 - 8x + 15$ (in $\mathbb{R}[x]$).

Theorem (Buch 4.8): Ein Polynom $p(x) \in \mathbb{K}[x]$ ist genau dann durch $(x - a)$ teilbar, wenn $p(a) = 0$ (für $a \in \mathbb{K}$). (a ist eine Nullstelle von $p(x)$)

Definition: Für zwei Polynome $p(x), q(x) \in \mathbb{K}[x]$ ist der $ggT(p, q)$ das normierte Polynom von maximalem Grad, welches $p(x)$ und $q(x)$ teilt. Wenn $ggT(p, q) = 1$, dann nennt man $p(x)$ und $q(x)$ teilerfremd.

Beispiele:

a) $p(x) = 4(x-1)^2, q(x) = 8(x-1) \qquad ggT(p,q) = x - 1$
b) $p(x) = 5(x-1), q(x) = 5(x+1) \qquad ggT(p,q) = 1$

4.3.3 Der erweiterte Euklid'sche Algorithmus

Der erweiterte Euklid'sche Algorithmus funktioniert für Polynome genauso wie für ganze Zahlen.

Beispiel:

a) Wir berechnen $ggT(x^3 - 2x + 1, x^2 - 1)$:

$$\begin{aligned} x^3 - 2x + 1 &= x \cdot (x^2 - 1) + (-x + 1) \\ x^2 - 1 &= (-x - 1) \cdot (-x + 1) + 0 \end{aligned} \tag{13}$$

Das Ergegnis ist aber nicht $(-x + 1)$, da das nicht normiert ist $(-x)$, daher berechnen wir

$$\begin{array}{r} \left(\quad x^2 \qquad -1 \right) : \left(-x+1 \right) = -x-1 \\ \underline{-x^2 + x} \\ \qquad\quad x - 1 \\ \qquad \underline{-x + 1} \\ \qquad\quad 0 \end{array}$$

Also ist $ggT(x^3 - 2x + 1, x^2 - 1) = \begin{cases} -x + 1 \\ x - 1 \end{cases}$

b) Gleiche Polynome, aber mit Erweiterung, um $s(x), t(x) \in \mathbb{R}[x]$ zu finden, s.d. $s(x)a(x) + t(x)b(x) = ggT(p, q)$

4.4 Der Restklassenring $\mathbb{K}[x]_{m(x)}$

4.4.1 Kongruenz von Polynomen

Da wir durch den EEA mit Polynomen auch Division mit Rest für Polynome haben, können wir Kongruenz definieren.

Definition: Zwei Polynome $p(x), q(x) \in \mathbb{K}[x]$ heißen kongruent modulo $m(x)$, falls sie bei Division durch $m(x)$ den gleichen Rest $r(x) \in \mathbb{K}[x]$ haben.

Anders ausgedrückt: $a(x) = b(x)(\mod m(x))$ genau dann, wenn $a(x) - b(x)$ durch $m(x)$ teilbar ist, wenn also $a(x) - b(x) = q(x)m(x)$. Alle modulo m kongruenten Polynome bilden eine **Restklasse** modulo m.

Analog zu den ganzen Zahlen verhalten sie Addition und Multiplikation gleich: Wenn $a(x) = b(x) \mod m$ und $s(x) = t(x) \mod m$, dann gilt

$$a(x) + s(x) = b(x) + t(x) \mod m(x) \quad \text{und} \quad a(x) \cdot s(x) = b(x) \cdot t(x) \mod m(x)$$

Tipp: Bei der Berechnung des Rests bzw. der Reduktion $\mod m(x)$ können wir benutzen, dass $m(x) = 0 \mod m(x)$:

$$\begin{aligned} &\iff \sum_{i=0}^{deg(m)} m_i x^i = 0 \mod m(x) \\ \iff x^{deg(m)} &= -\sum_{i=0}^{deg(m)-1} m_i x^i \mod m(x) \end{aligned} \tag{14}$$

Beispielrechnung für $x^{20} \mod x^4 - x - 1$:

$$x^4 = x + 1 \mod x^4 - x - 1$$

$$\begin{aligned} x^{20} &= (x^4)^5 \mod x^4 - x - 1 \\ &= (x+1)^5 \mod x^4 - x - 1 \\ &= x^5 + 1 \mod x^4 - x - 1 \quad \text{(Freshmen's Dream)} \\ &= x \cdot x^4 + 1 \mod x^4 - x - 1 \\ &= x \cdot (x+1) + 1 \mod x^4 - x - 1 \\ &= x^2 + x + 1 \mod x^4 - x - 1 \end{aligned} \tag{15}$$

Tipp: Freshmen's Dream:

$$(a+b)^p = a^p + b^p \mod p$$

4.4.2 Standardrepräsentanten

Definition: Hat $m(x)$ Grad k, dann sind die möglichen Reste alle Polynome in $\mathbb{K}[x]$ mit Grad $< k$. Dies sind die Standardrepräsentanten der Restklassen $\mod m(x)$. Die Menge dieser Reste wird geschrieben als $\mathbb{K}[x]_{m(x)}$ (analog zu $\mathbb{Z}_m$):

$$\mathbb{K}[x]_{m(x)} = \{\sum_{i=0}^{k-1} m_i x^i | m_i \in \mathbb{K}\}$$

Diese Menge enthält also alle Reste, die bei $\mod m(x)$ auftreten können. Allgemein gilt, dass $\mathbb{Z}_p[x]_{m(x)}$ genau p^k Reste hat.

4.4.3 Restklassenring

Für die Reste in $\mathbb{K}[x]_{m(x)}$ definieren wir nun Addition bzw. Multiplikation, indem wir einfach die vertraute Addition bzw. Multiplikation von Polynomen durchführen und am Ende, falls notwendig, den Rest modulo $m(x)$ nehmen, um nicht aus $\mathbb{K}[x]_{m(x)}$ hinauszufallen. Mit dieser Addition und Multiplikation wird $\mathbb{K}[x]_{m(x)}$ ein Ring, der so genannte Restklassenring $(\mathbb{K}[x]_{m(x)}, +, \cdot)$.

Beispiel: $\mathbb{Z}_2[x]_{x^2+x+1} = \{0, 1, x, x+1\}$.

Für die Addition und Multiplikation lassen sich Tabellen erstellen. Bei den Tabellen wird nach der Addition oder Multiplikation, wie oben definiert, immer modulo $m(x)$. Dies ist aber nicht nötig, wenn der Grad der Summe bzw. des Produktes kleiner ist als $m(x)$. In diesem Fall muss nur modulo $\mathbb{K}$ gerechnet werden.

Beispiel: $\mathbb{Z}_2[x]_{x^2+x+1} = \{0, 1, x, x+1\}$

$+$	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$x+1$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

$\cdot$	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	$x+1$	1
$x+1$	0	$x+1$	1	x

Zur linken Tabelle: Zum Beispiel ist $(x+1)+(x+1) = 2x+2 = 0$, da $2 = 0$ in $\mathbb{Z}_2$. Rechte Tabelle: Es ist zum Beispiel $(x+1) \cdot (x+1) = x^2 + 2x + 1 = x$.

Bei der Addition fällt auf, dass es sich um eine XOR Verknüpfung handelt:

$+$	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

Bei der Multiplikation sehen wir, dass für jedes Element $a(x)$ ($\neq 0$) ein anderes Element $b(x)$ existiert, s.d. deren Produkt $a(x)b(x) = 1 \mod m(x)$ ergibt. Das bedeutet, dass für jedes Element ein Inverses existiert. Damit ist $\mathbb{Z}_2[x]_{x^2+x+1}$ ein Körper.

Theorem: Für $p(x) \in \mathbb{K}[x]_{m(x)}$, gibt es ein Multiplikatives Inverses $p^{-1}(x) \in \mathbb{K}[x]_{m(x)}$, genau dann wenn $ggT(p(x), m(x)) = 1$.

Falls $p^{-1}(x)$ existiert, dann kann es mit dem EEA berechnet werden.

4.5 Endliche Körper

Wie oben erklärt, bilden nur manche Restklassenringe einen Körper (genau dann, wenn für jede Restklasse ein Inverses existiert). Da stellt sich die Frage: Mit $m(x)$ lässt sich ein Körper bilden?

Definition: Ein Polynom $p(x) \in \mathbb{K}[x]$ vom Grad > 1 heißt irreduzibel über $\mathbb{K}$, falls es kein Polynom $q(x) \in \mathbb{K}[x]$ mit $0 < deg(q) < deg(p)$ gibt, das $p(x)$ teilt. Andernfalls heißt es reduzibel. Um zu überprüfen, ob ein Polynom $p(x)$ des Grades 2 oder 3 in $\mathbb{Z}_n$ reduzibel ist, kann man alle Standardrepräsentanten in $p(x)$ einsetzen. Kommt bei einem Ergebnis 0 raus, dann ist das Polynom reduzibel.

Irreduzible Polynome sind also das äquivalent von Primzahlen in $\mathbb{Z}$.

Theorem: Sei $p(x) \in \mathbb{K}[x]$ ein normiertes Polynom von Grad > 1. Dann gibt es normierte, irreduzible Polynome $q_1(x), ..., q_n(x) \in \mathbb{K}[x]$, s.d.

$$p(x) = \prod_{i=1}^{n} q_i(x).$$

Bis auf die Reihenfolge ist diese Faktorisierung für jedes $p(x)$ eindeutig.

Ein Polynom vom Grad 1 ist immer irreduzibel. Für ein Polynom $p(x)$ höheren Grades gilt: Gibt es einen Teiler mit Grad m, dann gibt es automatisch auch einen Teiler mit Grad $deg(p) - m$.

Ein Polynom ist also genau dann irreduzibel, wenn es keine Teiler mit Grad $\leq deg(p)/2$ hat. Das liegt daran, dass die obere Hälfte ($\geq deg(p)/2$) nicht relevant ist, da falls ein Teiler in dieser Spanne existiert,

es auch automatisch einen Teiler $\leq deg(p)/2$ gibt.

Für Polynome vom Grad 2 oder 3 gilt: Sie sind irreduzibel, genau dann wenn sie keine Nullstellen haben (α Nullstellen $\iff$ $(x-a)$ Teiler).

Das wichtigste Resultat ist nun:

Theorem: $\mathbb{K}[x]_{m(x)}$ ist genau dann ein Körper, wenn $m(x)$ irreduzibel ist.

5 Kodierungstheorie

Bei der Übertragung von Daten können Bit Flips (Vertauschungen von Bits) auftreten.

Sagen wir A möchte eine Nachricht (ja/nein) an B schicken. Nutzt man nur ein Bit mit $1 =$ ja und $0 =$ nein, dann kann durch einen einzigen Bit Flip die Nachricht geändert werden.

Nutzen wir daher 3 Bits für die Übertragung mit den Codewörtern $111 =$ ja und $000 =$ nein. Selbst wenn ein Bit sich bei der Übertragung ändert, können wir klar entscheiden, ob die Nachricht "ja" oder "nein" ist. Allerdings brauchen wir so 3 Mal so viele Bits.

Die Effizienz von solchen Codewörtern lässt sich mit der Hamming-Metrik bestimmen.

5.1 Hamming-Metrik

Die Hamming-Metrik $d_H : \mathbb{F}_q^n \to \{0, 1, ..., n\}$ zählt in wie vielen Koordinaten sich zwei Vektoren unterscheiden, also

$$d_H(u, v) = |\{i | u_i \neq v_i\}|,$$

für $u, v \in \mathbb{F}_q^n$.

Das Hamming-Gewicht wt_H ist definiert als die Distanz zum Nullvektor, also $wt_H(u) = d_H(u, 0)$.

Ein Code über $\mathbb{F}_q$ der Länge n ist eine Teilmenge $C \subseteq \mathbb{F}_q^n$. Falls C ein Untervektorraum von $\mathbb{F}_q^n$ ist, spricht man von einem linearen Code. Im Rahmen dieses Moduls arbeiten hauptsächlich mit linearen Codes.

Die Minimaldistanz $d_H(C)$ eines Codes C ist das Minimum aller Distanzen zwischen den Codewörtern, also

$$d_H(C) = min\{d_H(u, v) | u, v \in C, u \neq v\}$$

5.1.1 Fehlerkorrekturkapazität

Die Fehler, die bei einer Übertragung entstehen, können wie folgt dargestellt werden:

$$\underbrace{r}_{\text{empfangenes Wort}} = \underbrace{c}_{\text{Codewort}} + \underbrace{e}_{\text{Fehlervektor}},$$

wobei $c, e, r \in \mathbb{F}_q^n$. Es gilt

$$d_H(r, c) = wt_H(r - c) = wt_H(e)$$

Sei $C \subseteq \mathbb{F}_q^n$ ein Code mit $d_H(C) = d$, dann kann C bis zu $d-1$ Fehler erkennen und bis zu $\lfloor (d-1)/2 \rfloor$ Fehler korrigieren (= Fehlerkorrekturkapazität).

Den Vorgang das nächste Codeword zu einem fehlerhaften Wort zu finden wird Dekodieren genannt.

5.2 Lineare Codes

Da ein linearer Code C ein Untervektorraum von $\mathbb{F}_q^n$ ist, erfüllt C viele schöne Eigenschaften, wie

- Nullvektor: Der Nullvektor aus dem Vektorraum $\mathbb{F}_q^n$ muss auch im Untervektorraum C enthalten sein. Symbolisch: $0 \in C$.
- Abgeschlossenheit bezüglich der Vektoraddition: Wenn zwei Codewörter u und v im Untervektorraum C liegen, muss auch ihre Summe $u + v \in C$.
- Abgeschlossenheit bezüglich der Skalarmultiplikation: Wenn ein Codewort u im Untervektorraum C liegt und ein Skalar $a \in \mathbb{F}_q$, muss auch das Produkt $a \cdot u \in C$.

Da der Nullvektor $\in C$ ist, gilt

$$d_H(C) = min\{wt_H(c) | c \in C\{0\}\}$$

5.2.1 Generatormatrix

Sei $C \subseteq \mathbb{F}_q^n$ ein k-dimensionaler linearer Code. Dann ist jede Matrix $G \in \mathbb{F}_q^{k \times n}$, deren Zeilen eine Basis von C bilden, eine **Generatormatrix** von C.

Die Zeilen der Generatormatrix sind linear unabhängig, und jedes Codewort in C kann als lineare Kombination dieser Zeilen dargestellt werden. Die Generatormatrix wird verwendet, um das Nachrichtenvektor m in das Codewort $c \in C$ umzuwandeln, indem die folgende Gleichung angewendet wird:

$$c = mG$$

Hierbei ist c ein $1 \times n$ Vektor und m ein $1 \times k$ Vektor.

Beispiel für eine Generatormatrix:

Wir betrachten einen $(6, 3)$-linearen Code über dem Körper $\mathbb{F}_2$, d.h. $q = 2$, $n = 6$ und $k = 3$. Eine mögliche Generatormatrix G könnte dann wie folgt aussehen:

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

Die Zeilen dieser Matrix bilden eine Basis für unseren Code. Jedes Codewort im Code kann durch eine lineare Kombination dieser Zeilen erzeugt werden.

Nun nehmen wir an, dass wir eine Nachricht $m = (1, 1, 0)$ codieren wollen. Dieser Vektor kann nun mit unserer Generatormatrix multipliziert werden, um das Codewort zu erhalten:

$$c = mG = (1, 1, 0) \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = (1, 1, 0, 1, 0, 1)$$

Dieses Codewort c könnte nun über einen Kanal übertragen werden, wobei der Empfänger die eingebauten Redundanzen (die zusätzlichen drei Bit) verwenden kann, um mögliche Übertragungsfehler zu erkennen und zu korrigieren.

5.2.2 Kontrollmatrix

Sei $C \subseteq \mathbb{F}_q^n$ ein k-dimensionaler linearer Code. Dann ist jede Matrix $H \in \mathbb{F}_q^{(n-k)\times n}$, deren Kern C ist, eine **Kontrollmatrix** von C.

Die Kontrollmatrix ist so konstruiert, dass sie orthogonal zur Generatormatrix ist, d.h. für jedes gültige Codewort $c \in C$ gilt:

$$c * H^T = 0$$

Z.B.: Die ISBN-10-Nummern erfüllen die Gleichung

$$\sum_{i=1}^{10}(11-i)z_i = 0 \mod 11.$$

Dies entspricht einem Block-Code in $\mathbb{F}_{11}^{10}$ mit Kontrollmatrix.

$$H = \begin{pmatrix} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

5.2.2.1 Berechnung In den meisten Fällen ist die Generatormatrix G in der standardisierten oder systematischen Form gegeben, $G = [I_k|P]$, wobei I_k die $k \times k$ Einheitsmatrix und P eine $k \times (n-k)$ Matrix ist. In diesem Fall kann die Kontrollmatrix H direkt durch den Austausch von I_k und P und Transposition von P erstellt werden, also $H = [-P^T|I_{n-k}]$.

Wenn die Generatormatrix G nicht in dieser Form vorliegt, muss sie erst in diese Form gebracht werden, zum Beispiel durch Zeilenoperationen (dies ändert den von G erzeugten Code nicht). Sobald G in der Form $[I_k|P]$ ist, kann man wie oben beschrieben fortfahren, um die Kontrollmatrix zu erstellen.

5.2.2.2 Fehlerkorrektur Die Kontrollmatrix H spielt eine wesentliche Rolle bei der Fehlerkorrektur. Mit ihrer Hilfe kann ein Syndromvektor berechnet werden, der die Positionen von Fehlern in einem empfangenen Codewort angibt. Dies geschieht durch Multiplikation des empfangenen Vektors r (der das gesendete Codewort c plus etwaige Fehler e ist) mit der transponierten Kontrollmatrix H^T:

$$s = rH^T = (c+e)H^T = eH^T$$

Der resultierende Syndromvektor s ist gleich dem Nullvektor, wenn keine Fehler aufgetreten sind (denn dann ist $e = 0$ und $cH^T = 0$). Ist s jedoch nicht gleich Null, dann liegt ein Fehler vor.

Um den Fehler zu korrigieren, muss die Position des Fehlers ermittelt werden. Dies kann erreicht werden, indem der Syndromvektor s mit den Spalten der Kontrollmatrix H verglichen wird. Wenn s gleich der i-ten Spalte von H ist, deutet dies darauf hin, dass an der i-ten Position des empfangenen Vektors r ein Fehler aufgetreten ist. Dann kann der Fehler korrigiert werden, indem das Bit an dieser Position invertiert wird.

Wichtig: Das funktioniert nur, wenn genau 1 Fehler im Code enthalten ist.

5.3 Hamming-Codes

Hamming-Codes sind eine Familie von linearen Fehlererkennungs- und Fehlerkorrekturcodes, die nach ihrem Erfinder, Richard Hamming, benannt sind. Sie korrigieren bis zu 1 Bit, indem sie zusätzliche Paritätsbits zu den übertragenen Daten hinzufügen.

Ich empfehle, die folgenden zwei Videos von 3Blue1Brown anzusehen:

- Einführung in Hamming-Codes
- Die Eleganz der Hamming-Codes

5.3.1 Wie sie funktionieren

1. **Paritätsbits**: Hamming-Codes fügen mehrere Paritätsbits zu den Datenbits hinzu. Die Position dieser Paritätsbits ist entscheidend. Sie werden an den Positionen platziert, die Potenzen von 2 sind (1, 2, 4, 8, usw.).
2. **Fehlererkennung und -korrektur**: Wenn ein Fehler in einem einzelnen Bit auftritt, wird das fehlerhafte Bit durch die Kombination von Paritätsbits eindeutig identifiziert. Somit können Hamming-Codes Einzelbitfehler korrigieren.

5.3.2 Einschränkungen

Hamming-Codes sind hervorragend für die Korrektur von Einzelbitfehlern und die Erkennung von Doppelbitfehlern geeignet. Sie sind jedoch nicht effektiv für die Korrektur von mehreren Bitfehlern, die gleichzeitig auftreten.

5.4 Kugelpackungs-Schranke

Sei $C \subseteq \mathbb{F}_q^n$ mit $d_H(C) = d$, dann gilt

$$|C| \leq \frac{|\mathbb{F}_q^n|}{|B_H(\lfloor \frac{d-1}{2} \rfloor, \mathbb{F}_q^n)|} = \frac{q^n}{\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i}(q-1)^i},$$

wobei $B_H(\lfloor \frac{d-1}{2} \rfloor, \mathbb{F}_q^n) = \{v \in \mathbb{F}_q^n | wt_H(v) \leq \lfloor \frac{d-1}{2} \rfloor\}$.

Einen Code, der die Kugelpackungs-Schranke erreicht, nennt man perfekt. Bei perfekten Codes, liegt jeder Vektor im Raum $\mathbb{F}_q^n$ in genau einer Kugel mit dem Radius $(d-1)/2$ um ein Codewort. Das bedeutet, dass jeder Vektor (Nachricht oder Codewort) einem eindeutigen nächsten Codewort zugeordnet werden kann. Diese Eigenschaft ermöglicht es, Fehler effizient zu erkennen und zu korrigieren, da es

keine Überschneidungen zwischen den Kugeln gibt und jeder Vektor einem eindeutigen Codewort zugeordnet ist.

Es gibt nur sehr wenige perfekte Codes, ein Beispiel ist der Hamming-Code in $\mathbb{F}_2^7$ mit Kontrollmatrix (jeder Hamming-Code erfüllt die Kugelpackungs-Schranke):

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Dieser Code hat Länge 7, Dimension 4 (also $2^4 = 16$ Codewörter) und Minimaldistanz 3. Die Kugelpackungs-Schranke ergibt in diesem Fall ebenfalls 16, deshalb ist es ein perfekter Code.

5.5 Singleton-Schranke

Sei $C \subseteq \mathbb{F}_q^n$ mit $d_H(C) = d$, dann gilt

$$|C| \leq q^{n-d+1}$$

Falls C ein linearer Code mit Dimension k ist, gilt also

$$k \leq n - d + 1$$

Falls ein Code die Singleton-Schranke erreicht, nennt man ihn maximum distance seperable (MDS) Code.

5.6 Reed-Solomon Codes

Reed-Solomon (RS) codes sind eine weitere Klasse von fehlerkorrigierenden Codes. Sie wurden nach den Hamming Codes entwickelt und sind in der Lage mehrere Fehler zu erkennen und zu korrigieren.

Das folgende Video erklärt Reed-Solomon Codes sehr gut:

What are Reed-Solomon Codes?

6 Kombinatorik

Die Kombinatorik beschäftigt sich grundlegend mit der Untersuchung und Zählung von Möglichkeiten, wie bestimmte Elemente arrangiert oder ausgewählt werden können. Im Kern unterscheidet sie

dabei zwischen Situationen mit und ohne Berücksichtigung der Reihenfolge sowie mit und ohne Zurücklegen der gezogenen Elemente.

	mit Reihenfolge	ohne Reihenfolge
ohne Zurücklegen	$\frac{n!}{(n-k)!}$	$\binom{n}{k}$
mit Zurücklegen	n^k	$\binom{n+k-1}{k}$

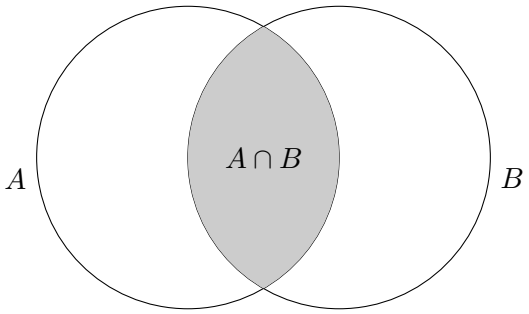
6.1 Kardinalität

Die Kardinalität einer Menge ist die Anzahl der Objekte, die sie enthält. Die Kardinalität einer Menge S wird als $|S|$ oder $\#S$ geschrieben.

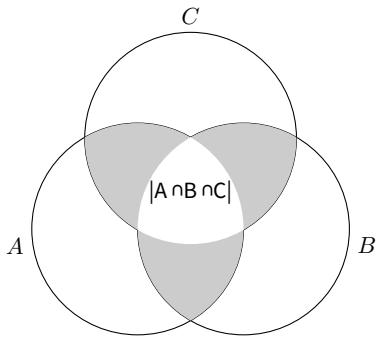
6.2 Das Prinzip von Inklusion und Exklusion

Das Prinzip von Inklusion und Exklusion ist eine mathematische Methode, um die Größe der Vereinigung mehrerer Mengen zu berechnen, indem die Größen der einzelnen Mengen und ihrer Schnittmengen berücksichtigt werden.

Denn $|S \cup T| \neq |S| + |T|$, sondern $|S \cup T| = |S| + |T| - |S \cap T|$, da die Überlappung von S und T ($|S \cap T|$) ansonsten zwei Mal gezählt wird.



Gleiches gilt für 3 Mengen: $|S \cup T \cup U| = |S| + |T| + |U| - |S \cap U| - |S \cap T| - |T \cap U| + |S \cap T \cap U|$, da die Überschneidung von S, T und U drei Mal hinzugefügt wird und entfernt wird, muss sie wieder hinzugefügt werden).



Was ist die Kardinalität von $S \setminus T$? Es gilt, dass

$$S \setminus T = S \setminus (S \cap T)$$

und deshalb

$$|S \setminus T| = |S \setminus (S \cap T)| = |S| - |(S \cap T)|.$$

6.3 Kartesisches Produkt

Seien S und T zwei Mengen. Dann wird das Kartesische Prodult von S und T definiert als

$$S \times T = \{(s,t) | s \in S, t \in T\}.$$

Beispiel: Sei $A = \{a, b, c\}$ und $B = \{\text{Tom}, \text{Andrea}\}$.

Das kartesische Produkt von A und B ist:

$$A \times B = \{(a, \text{Tom}), (a, \text{Andrea}), (b, \text{Tom}), (b, \text{Andrea}), (c, \text{Tom}), (c, \text{Andrea})\}$$

Die Kardinalität von $S \times T$ ist

$$|S \times T| = |S| \cdot |T|$$

Generell ist die Definition des Kartesischen Produkts:

Seien $S_1, S_2, \dots, S_n$ Mengen. Dann wird das Kartesische Produkt dieser Mengen definiert als

$$S_1 \times S_2 \times \cdots \times S_n = \{(s_1, s_2, \dots, s_n) \mid s_i \in S_i \text{ für } i = 1, \dots, n\}$$

Bezüglich der Kardinalität gilt

$$|S_1 \times S_2 \times \cdots \times S_n| = \prod_{i=1}^{n} |S_i|$$

6.4 Potenzmenge

Die Potenzmenge ist die Menge aller möglichen Teilmengen einer Menge. Wenn wir eine Menge A haben, wird die Potenzmenge als $\mathcal{P}(A)$ geschrieben. Zum Beispiel, wenn $A = \{1, 2\}$, dann ist die Potenzmenge $\mathcal{P}(A) = \{\{\}, \{1\}, \{2\}, \{1, 2\}\}$.

Sei S eine Menge mit Kardinalität n, dann gilt

$$|\mathcal{P}(S)| = 2^n.$$

6.5 Anzahl von Teilmengen mit gegebener Kardinalität

Sei S eine Menge mit Kardinalität n, dann hat S $\binom{n}{k}$ (= Binomialkoeffizient) viele Teilmengen der Größe k, wobei

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} = \frac{n(n-1)(n-2)\ldots(n-k+1)}{k!}$$

Die Fakultät gibt die Anzahl an Permutationen (Anordnung von Elementen in einer bestimmten Reihenfolge) an. Für k Elemente gibt es $k!$ Permutationen.

Der Binomialkoeffizient $\binom{n}{k}$ gibt an, wie viele Möglichkeiten es gibt aus n Elementen k zu wählen ohne Beachtung der Reihenfolge. Es berechnet also die Anzahl Kombinationen, welche weniger als die Anzahl Permuationen sind, da es z.B. von den Elementen {a} und {b} nur eine Kombination ({a,b}), aber zwei Permutationen ({a,b} und {b,a}) gibt. Beispiel: Betrachten wir die Menge $S = \{a, b, c, d\}$. Die zwei-elementigen Teilmengen sind

$$\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}$$

und der Binomialkoeffizient ist

$$\binom{4}{2} = 6.$$

Schauen wir uns nochmal die Potzenmenge an. Die Potenzmenge gibt alle Teilmengen an. Wir wissen auch, dass die Anzahl der Teilmengen mit k Elementen $\binom{n}{k}$ ist. Wenn wir diese für alle möglichen k addieren, sollten wir die Anzahl aller Teilmengen erhalten, also 2^n. Also ist

$$|\mathcal{P}(S)| = \sum_{k=0}^{n} \binom{n}{k} = 2^n$$

6.6 Rechenregeln des Binomialkoeffizientens

6.6.1 Symmetrie

Für jedes $0 \leq k \leq n$ gilt

$$\binom{n}{k} = \binom{n}{n-k},$$

da

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n!}{(n-k)!(n-(n-k))!} = \binom{n}{n-k}$$

Interpretation für Teilmengen: Beim Zählen der Anzahl von Teilmengen mit der Kardinalität k ist es dasselbe, darüber nachzudenken, wie viele Elemente man einschließt oder wie viele Elemente man ausschließt.

6.6.2 Additivität

Für $k \geq 1$ gilt

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

Interpretation: Angenommen, wir wollen k aus n Elementen auswählen. Von den k Elementen wählen wir ein spezifisches Elemenent.

1. $\binom{n-1}{k-1}$ betrachtet den Fall, in dem das spezifisches Element in der Auswahl enthalten ist. Da dieses Element bereits ausgewählt wurde, müssen wir nur noch k-1 Elemente aus den verbleibenden n-1 Elementen auswählen.
2. $\binom{n-1}{k}$ betrachtet den Fall, in dem das spezifische Element nicht in der Auswahl enthalten ist. In diesem Fall müssen wir alle k Elemente aus den verbleibenden n-1 Elementen auswählen.

6.6.3 Vandermonde'sche Identität

$$\sum_{i=0}^{k} \binom{m}{i}\binom{n}{k-i} = \binom{m+n}{k}$$

Interpretation: Angenommen, wir wollen k Elemente aus den Mengen n und m auswählen.

Auf der rechten Seite steht $\binom{m+n}{k}$, was genau das repräsentiert: die Anzahl der Möglichkeiten, k Elemente aus einer kombinierten Menge von $m+n$ Elementen auszuwählen.

Die linke Seite der Gleichung zerlegt diesen Prozess in verschiedene Szenarien, basierend auf wie viele Elemente aus jeder Menge ausgewählt werden. Für jede Zahl i von 0 bis k berechnen wir die Anzahl der

Möglichkeiten, i Elemente aus der ersten Menge (repräsentiert durch $\binom{m}{i}$) und $k-i$ Objekte aus der zweiten Menge (repräsentiert durch $\binom{n}{k-i}$) auszuwählen. Die Summe über alle diese Szenarien gibt die Gesamtzahl der Möglichkeiten, k Elemente aus beiden Mengen zusammen auszuwählen.

6.6.4 Pascal'sches Dreieck

Das Pascalsche Dreieck ist eine tabellarische Anordnung von Zahlen, die die Binomialkoeffizienten repräsentieren.

Im Pascalschen Dreieck entspricht der Eintrag in der n-ten Zeile und der k-ten Spalte dem Binomialkoeffizienten $\binom{n}{k}$ (Zeilen werden von 0 auf gezählt).

Hier sind die ersten 5 Zeilen des Dreiecks:

$n=0$					1					
$n=1$				1		1				
$n=2$			1		2		1			
$n=3$		1		3		3		1		
$n=4$	1		4		6		4		1	
$n=5$	1		5		10		10		5	1

Darüber hinaus hat das Pascalsche Dreieck viele interessante Eigenschaften und Muster. Zum Beispiel:

1. Die Summe der Zahlen in jeder Zeile ist 2^n (analog zur Potenzmenge!), wobei n die Zeilennummer ist (wieder beginnend mit 0).
2. Die Zahlen in jeder Zeile sind symmetrisch: Die k-te Zahl von links in der n-ten Zeile ist die gleiche wie die k-te Zahl von rechts.
3. Jede Zahl ist die Summe der beiden Zahlen direkt über ihr. Dies entspricht der rekursiven Formel für binomiale Koeffizienten: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$.
4. Das Pascalsche Dreieck enthält viele andere Zahlenfolgen als Teil seiner Struktur, einschließlich der Fibonacci-Folge, der Dreieckszahlen und der Tetraederzahlen.

6.6.5 Der binomische Lehrsatz

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Interpretation:

Der binomiale Koeffizient $\binom{n}{k}$ repräsentiert die Anzahl der Möglichkeiten, k Elemente aus einer Menge von n Elementen auszuwählen. In diesem Zusammenhang entspricht dies der Anzahl der Möglichkeiten, wie man k Mal y und die restlichen $n-k$ Mal x in den n Produkten erhält, die durch die Ausklammerung des Ausdrucks $(x+y)^n$ erzeugt werden.

x^{n-k} ist die Potenz von x, die entspricht, wie oft x in diesen n Produkten erscheint, und ähnlich ist y^k die Potenz von y, die entspricht, wie oft y erscheint.

Die Summation über k von 0 bis n berücksichtigt alle möglichen Verteilungen von x und y in den n Produkten.

7 Wahrscheinlichkeitstheorie

7.1 Zufallsexperiment

Ein Zufallsexperiment ist ein nicht-deterministischer Vorgang (= ein zufälliger Vorgang), der beliebig oft unter gleichartigen Bedingungen wiederholt werden kann.

Ein Zufallsexperiment mit Ergebnisraum Ω heißt Laplace-Experiment, falls die folgenden Punkte zutreffen:

- $|\Omega| < \infty$. D.h. Es gibt endliche viele Ergebnisse und
- $P(\{\omega\}) = \frac{1}{|\Omega|}$ für alle $\omega \in \Omega$. Das bedeutet alle Ergebnisse sind gleich wahrscheinlich

In einem Laplace-Experiment gilt

$$P(A) = \frac{|A|}{|\Omega|}.$$

7.2 Ereignisse

Die Menge Ω aller möglichen Ergebnisse wird Ergebnismenge oder Ergebnisraum genannt.

Eine Teilmenge $A \subseteq \Omega$ wird Ereignis genannt.

- Falls $A = \{\omega\}$ für ein $\omega \in \Omega$, nennt man es ein Elementarereignis.
- $A = \Omega$ ist das sichere Ereignis, was auf jeden Fall eintritt.
- $A = \{\}$ ist das unmögliche Ereignis.

Für ein Ereignis $A \subseteq \Omega$ ist sein Gegenereignis das Komplement $\overline{A} = \Omega \setminus A$.

Für zwei Ereignisse $A, B \subseteq \Omega$ ist das Ereignis, dass A und B eintreten genau $A \cap B$ (falls $A \cap B = \{\}$, dann sind A und B unvereinbar).

Für das Ereignis, dass A oder B eintreten, ist die Wahrscheinlichkeit genau $A \cup B$.

Die Menge aller Ereignisse ist die Potenzmenge der Ergebnismenge, also $\mathcal{P}(\Omega)$. Das liegt daran, dass ein Ereignis jede Kombination (inklusive leere Menge) der Elemente von Ω ist.

Bei einem Zufallsexperiment, bei dem k aus n ohne Zurücklegen gezogen werden, ist die Kardinalität von Ω

$$|\Omega| = \binom{n}{k}$$

Für ein Zufallsexperiment, bei dem k aus n mit Zurücklegen gezogen werden, ist die Kardinalität von Ω

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

(Siehe Kombinatorik)

7.2.1 Wahrscheinlichkeiten von Ereignissen

Sei (Ω, P) ein Wahrscheinlichkeitsraum und $A, B \subseteq \Omega$, dann gilt für die Wahrscheinlichkeit

- $P(\overline{A}) = 1 - P(A)$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ (siehe Prinzip von Inklusion und Exklusion)
- Wenn $A \subseteq B$, dann $P(A) \leq P(B)$

7.2.2 Unabhängige Ereignisse

Gegeben ist ein Wahrscheinlichkeitsraum (Ω, P) und $A, B \in \Omega$, dann sind A und B unabhängig, wenn

$$P(A \cap B) = P(A)P(B).$$

Daraus folgt für unabhängige Ereignisse, dass

$$P(A|B) = P(A) \quad \text{und} \quad P(B|A) = P(B).$$

7.3 Diskrete Wahrscheinlichkeitsräume

Sei $\Omega = \{\omega_1, \omega_2, \dots\}$ eine diskrete Ergebnismenge (diskret heißt abzählbar), dann ist die Funktion $P : \mathcal{P}(\Omega) \to \mathbb{R}$ eine Wahrscheinlichkeit(sfunktion), wenn:

1. $P(A) \in [0, 1]$ für alle $A \in \mathcal{P}(\Omega)$.
2. Falls $A \cap B = \{\}$, dann $P(A \cup B) = P(A) + P(B)$.

3. $P(\Omega) = \sum_{i=1}^{|\Omega|} P(\omega_i) = 1$

Das Paar (Ω, P) wird (diskreter) Wahrscheinlichkeitsraum genannt.

Aus dem 3. Punkt folgt, dass $P(A) = \sum_{a \in A} P(a)$ für jedes $A \in \Omega$.

7.4 Bedingte Wahrscheinlichkeit

Betrachte einen Wahrscheinlichkeitsraum (Ω, P) und zwei Ereignisse $A, B \in \Omega$, dann ist die bedingte Wahrscheinlichkeit von A, gegeben, dass B eingetreten ist,

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Falls $P(B) = 0$, ist $P(A|B)$ nicht definiert.

Bsp.: Angenommen, es gibt 100 Schüler in einer Klasse, und 30 von ihnen mögen Schokolade. 20 Schüler mögen sowohl Schokolade als auch Vanille. Die Wahrscheinlichkeit, dass ein Schüler Schokolade mag, ist P(Schokolade) = 30/100. Die Wahrscheinlichkeit, dass ein Schüler Schokolade und Vanille mag, ist P(Schokolade ∩ Vanille) = 20/100. Die bedingte Wahrscheinlichkeit, dass ein Schüler Vanille mag, wenn er Schokolade mag, ist:

$$\begin{aligned} P(Vanille|Schokolade) &= P(Schokolade \cap Vanille)/P(Schokolade) \\ &= (20/100)/(30/100) \\ &= 20/30 \\ &= 2/3 \end{aligned}$$

Wichtig: Es kann sein, dass es auch Schüler gibt, die Vanille, aber keine Schokolade mögen. Diese Schüler sind nicht miteinbezogen.

7.4.1 Einfacher Satz von Bayes

Oftmals ist es leichter die andere Richtung der Bedingung zu berechnen. Dies ermöglicht der Satz von Bayes:

$$P(S|T) = \frac{P(T|S)P(S)}{P(T)}$$

Bsp.: Angenommen, in einer Stadt sind 1% der Menschen krank (A). Es gibt einen Test, der mit einer Wahrscheinlichkeit von 90% korrekt erkennt, ob jemand krank ist (B|A), und mit einer Wahrscheinlichkeit von 90% korrekt erkennt, ob jemand gesund ist. Nun hat eine Person einen positiven Test (B). Wie hoch ist die Wahrscheinlichkeit, dass die Person tatsächlich krank ist (A|B)?

$$P(A) = 0.01$$

$$P(B|A) = 0.9$$

$$P(B|\overline{A}) = 0.1$$

Um P(B) zu berechnen, verwenden wir die totale Wahrscheinlichkeit:

$$\begin{aligned} P(B) &= P(B|A) * P(A) + P(B|\overline{A}) * P(\overline{A}) \\ P(B) &= 0.9 * 0.01 + 0.1 * 0.99 \\ P(B) &= 0.108 \end{aligned}$$

Jetzt können wir den Satz von Bayes anwenden:

$$\begin{aligned} P(A|B) &= (P(B|A) * P(A))/P(B) \\ P(A|B) &= (0.9 * 0.01)/0.108 \\ P(A|B) &= 0.08\overline{3} \end{aligned}$$

Die Wahrscheinlichkeit, dass die Person tatsächlich krank ist, beträgt etwa 8,33%. (Dies liegt an der geringen Prävalenz der Krankheit in der Bevölkerung (1%). Dieses Phänomen ist als "False-Positive-Paradoxon" bekannt.)

7.4.2 Genereller Satz von Bayes

Sei $\{E_1, \dots, E_n\}$ eine Partition von Ω. Eine Partition heißt eine Unterteilung von Ω, wobei gilt, dass $\Omega = \bigcup_{i=1}^{n} E_i$ (Vereinigung von allen E_i) und $E_i \cap E_j = \{\}$ für $i \neq j$.

Für ein beliebiges Ereignis $A \subseteq \Omega$ gilt

$$P(A) = \sum_{k=1}^{n} P(A \cap E_k) = \sum_{k=1}^{n} P(E_k)P(A|E_k)$$

Außerdem gilt für ein beliebiges Ereignis $A \subseteq \Omega$

$$P(E_j|A) = \frac{P(E_j)P(A|E_j)}{P(A)} = \frac{P(E_j)P(A|E_j)}{\sum_{k=1}^{n} P(E_k)P(A|E_k)}.$$

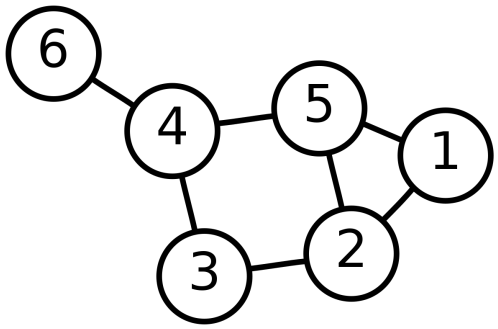


Abbildung 1: Graph Beispiel

8 Graphentheorie

Ein Graph $G(V, E)$ besteht aus einer endlichen Menge V von Knoten und einer Menge E von Kanten $\{a, b\}$ mit $a, b \in V$ und $a \neq b$. Statt $\{a, b\}$ schreiben wir auch verkürzt ab.

Im Beispiel:

$$V = \{1, 2, 3, 4, 5, 6\}$$

$$E = \{\{1, 2\}, \{1, 5\}, \{2, 5\}, \{2, 3\}, \{5, 4\}, \{3, 4\}, \{4, 6\}$$

Ein Graph $H(V^*, E^*)$ mit $V^* \subseteq V$ und $E^* \subseteq E$ heisst Teilgraph von $G(V, E)$.

8.1 Knoten und Kanten

8.1.1 Inzidenz und Adjazenz

Wenn 2 Knoten $a, b \in V$ durch eine Kante verbunden sind, so heißen sie **adjazent** oder **benachbart**. Das bedeutet $\exists\{a, b\} \in E$.

Ein Knoten und eine Kante sind **inzident**, wenn der Knoten ein Endknoten der Kante ist (= die Kante berührt den Knoten). Zwei Knoten $a, b \in V$ sind also benachbart, wenn sie beide inzident zur gleichen Kante sind.

Wenn zwei Kanten einen gemeinsamen Endknoten haben, so heißen sie ebenfalls **inzident**.

Im Beispiel: Die Knoten 1 und 2 sind benachbart, da $\{1, 2\} \in E$. Außerdem sind sie beide inzident zu der Kante $\{1, 2\}$.

8.1.2 Grad eines Knotens

Der **Grad** $deg(a)$ eines Knotens a ist die Anzahl inzidenter Kanten zu a. Ist $deg(a) = 0$, dann heisst a **isoliert**.

Im Beispiel: Die beiden Kanten $\{1, 2\}, \{1, 5\}$ sind inzident zu dem Knoten 1. Daraus folt, der $deg(1) = 2$.

Für einen beliebigen Graphen $G(V, E)$ gilt

$$\sum_{a \in V} deg(a) = 2|E|.$$

Das heißt, die Summe der Grade aller Knoten ist gleich zweimal der Anzahl an Kanten.

8.1.3 Wege und Kreise

Eine Folge $ab, bc, cd, ...$ von inzidenten Kanten nennt man einen **Kantenzug**. Die Anzahl der durchlaufenen Knoten des Kantenzugs ist die **Länge**. Wenn man wieder an den Ausgangsknoten des Kantenzugs zurückkehrt (also Startknoten = Endknoten), nennt man es einen **geschlossenen** Kantenzug.

Ein Kantenzug, bei dem alle vorkommenden Knoten verschieden sind, wird ein **Weg** genannt. Ein Weg, bei dem wir zum Ausgangspunkt zurückkehren, heißt **Kreis** (nur der Startknoten = Endknoten, alle anderen Knoten müssen unterschiedlich sein). Ein Kreis ist also eine spezielle Form eines geschlossenen Kantenzugs.

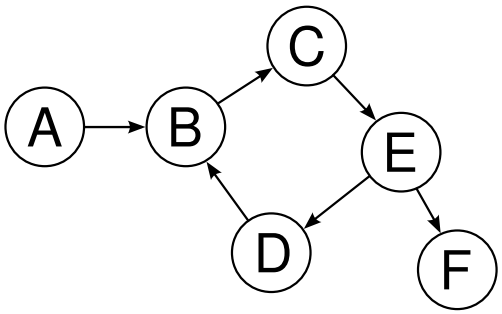


Abbildung 2: Graph mit Kreis

8.1.4 Euler-Zyklus

Ein Euler-Zyklus in einem Graphen ist ein Pfad, der jede Kante genau einmal verwendet und zum Ausgangspunkt zurückkehrt. Ein Graph enthält einen Euler-Zyklus, wenn er zusammenhängend ist und alle seine Knoten bis auf zwei einen geraden Grad haben.

Für einen stark zusammenhängenden gerichteten Graph gilt: Wenn für jeden Knoten bis auf zwei der Eingangsgrad gleich dem Ausgangsgrad ist, dann besitzt der Graph einen Euler-Zug. Für einen Knoten muss der Eingangsgrad eins höher als der Ausgangsgrad sein und vice versa für den letzten Knoten.

8.1.5 Hammilton-Kreis

Ein Hamilton-Zyklus in einem Graphen ist ein Pfad, der jeden Knoten genau einmal besucht und zum Ausgangspunkt zurückkehrt.

Wenn ein Graph mit n Knoten mindestens $\frac{(n-1)(n-2)}{2} + 2$ Kanten hat, dann besitzt er einen Hamilton-Kreis.

Im Gegensatz zum Euler-Zyklus gibt es für das Hamilton-Zyklus-Problem keinen effizienten Algorithmus, da es zu den NP-vollständigen Problemen gehört. Das Problem ist verwandt mit dem Travelling Salesman Problem.

8.2 Graphen

8.2.1 Isomorphie

Zwei Graphen $G(V, E)$ und $H(V^*, E^*)$ sind **äquivalent** (oder **isomorph**), wenn es eine bijektive Abbildung $f : V \to V^*$ gibt, sodass

$$ab \in E \iff f(a)f(b) \in E^*.$$

Es folgt, dass äquivalente Graphen die gleiche Anzahl an Kanten und Knoten von Grad i haben (für alle i).

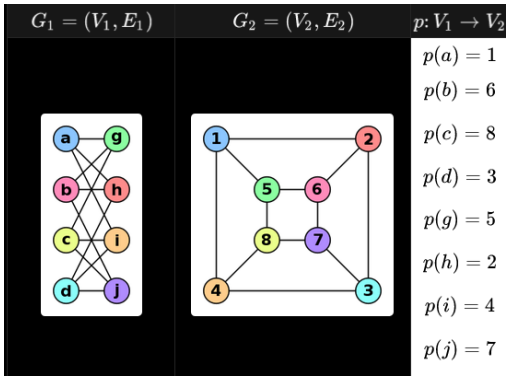


Abbildung 3: Isomorphie von 2 Graphen

8.2.2 Gerichtete Graphen

Ein gerichteter Graph (oder Digraph) ist ein Graph, in dem jede Kante eine Richtung besitzt. Das heißt ab bedeutet nicht, dass ba gegeben ist. In den bisherigen Beispielen sind die Graphen ungerichtet. Eine Kante eines gerichteten Graphens wird als geordnetes Paar (a, b) (von a nach b) geschrieben.

Beispiel: Siehe Figure 2.

8.2.3 Gewichtete Graphen

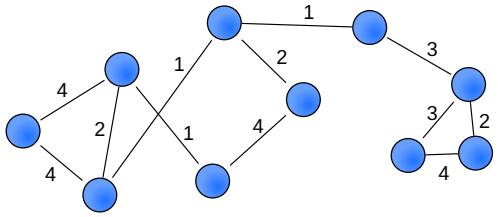


Abbildung 4: Gewichteter Graph

Ein gewichteter Graph ist ein Graph, bei dem jede Kante ein Gewicht besitzt. Dieses Gewicht kann eine Vielzahl von Bedeutungen haben, abhängig von dem Kontext, in dem der Graph verwendet wird. Beispielsweise kann es die Entfernung zwischen zwei Punkten auf einer Karte, die Zeit, die benötigt wird, um von einem Punkt zu einem anderen zu gelangen, oder die Kosten für die Verbindung zwischen zwei Punkten in einem Netzwerk repräsentieren.

Das Gewicht einer Kante von a nach b wird geschrieben als $w(a, b)$.

Siehe Abbildung 4

8.2.4 Bäume und Wälder

Ein **Baum** ist ein zusammenhängender Graph ohne Kreise und ein **Wald** ist ein nicht zusammenhängender Graph, deren Komponenten Bäume sind.

Für einen Baum G gilt:

- G hat genau $n - 1$ Kanten.
- Entfernt man eine Kante, ist G nicht mehr zusammenhängend.
- Es gibt genau einen Weg zwischen je zwei Kanten.

8.2.4.1 Aufspannender Baum Ein **aufspannender Baum** ist ein Teilgraph eines zusammenhängenden Graphens G mit n Knoten. Der Baum verbindet minimal alle n Knoten.

Ein minimal aufspannender Baum $T(V^*, E^*)$ von G ist ein minimaler aufspannender Baum, falls

$$\sum_{ab \in E^*} w(a,b) \leq \sum_{ab \in E^{**}} w(a,b)$$

für alle aufspannende Bäume $S(V^{**}, E^{**})$

8.2.4.2 Wurzelbaum Ein Baum, bei dem wir einen Knoten als Wurzel definieren, nennt man einen **Wurzelbaum**. Die Knoten von Grad 1 nennt man die Blätter des Baumes. Die Länge eines Wurzelbaums ist die maximale Länge eines WEgs von der Wurzel zu einem Blatt.

8.2.4.3 Binärer Baum Ein **binärer Baum** ist ein Baum, der maximal zwei Kinder hat.

8.2.4.4 Suchbaum Ein **Suchbaum** ist eine effiziente Methode zur Speicherung von sortierten Daten. Für alle Knoten (jeder Datenpunkt ist ein Knoten) y gilt: Alle Knoten $x < y$ sind im linken Unterbaum von y und alle Knoten z mit $z > y$ sind im rechten Unterbaum von y.

Operationen:

- **Suche** eines Knotens x: Beginne in der Wurzel. Ist der Knoten gleich x, dann stoppe (“gefunden”). Ansonsten gehe in den linken (resp. rechten) Unterbaum, falls x kleiner (resp. grösser) als der aktuelle Knoten ist. Falls der Unterbaum leer ist, stoppe (”nicht vorhanden”), ansonsten wiederhole den Schritt.
- **Einfügen** eines Knotens x: Suche nach x und füge ihn als Nachfolger des Knotens ein, wo die Suche abgebrochen wurde.
- **Löschen** eines Knotens x: Falls x ein Blatt ist, entferne es. Falls x nur einen Unterbaum hat, ersetze x durch die Wurzel des Unterbaums. Falls x zwei Unterbäume hat, suche das kleinste Element y im rechten Unterbaum und ersetze x durch y.

Vorteile eines Suchbaums gegenüber einer Liste:

- Suche/Finden ist effizienter. Im Worst Case brauchen wir Länge des Baumes + 1 viele Schritte. Falls der Baum gleichmäßig aufgebaut ist, ist die Länge $\lceil \log_2(|V|+1) \rceil - 1$.
- In einer Liste bräuchten wir Worst Case $|V|$ viele Schritte und im Durchschnitt $|V|/2$ viele Schritte.

Nachteil:

- Den Baum zu erstellen, Knoten hinzuzufügen oder zu entfernen ist komplizierter als in einer Liste.

8.2.5 Adjazenzmatrix

Die Adjazenzmatrix ist eine $n \times n$-Matrix, wobei n die Anzahl der Knoten des Graphen ist. Der Eintrag a_{ij} in der Matrix gibt an, ob es eine Kante zwischen Knoten i und Knoten j gibt. Wenn es eine Kante gibt, ist $a_{ij} = 1$, andernfalls ist $a_{ij} = 0$.

Seien die Knoten eines ungerichteten Graphens $G(V, E)$ durchnummeriert als $v_1, \dots, v_n$, dann ist die **Adjazenzmatrix** $A = (a_{ij})$ des Graphen gegeben durch

$$a_{ij} = \begin{cases} 1 & \text{wenn} \{i, j\} \in E \\ 0 & \text{sonst} \end{cases}.$$

Die Adjazenzmatrix ist für ungerichtete Graphen immer symmetrisch, also $a_{ij} = a_{ji}$.

Die Adjazenzmatrix eines gerichteten Graphens ist gegeben durch

$$a_{ij} = \begin{cases} 1 & \text{wenn} (i, j) \in E \\ 0 & \text{sonst} \end{cases}.$$

Gegeben ist der Graph $G(V, E)$ mit

$$V = \{1, 2, 3, 4, 5, 6\}$$

und

$$E = \{\{1, 2\}, \{1, 5\}, \{2, 5\}, \{2, 3\}, \{5, 4\}, \{3, 4\}, \{4, 6\}$$

(siehe Figure 1). Die Adjazenzmatrix von G ist

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

8.2.6 Adjazenz und Zusammenhang

Sei G ein Graph mit Adjazenzmatrix A. Der Koeffizient $a_{ij}^{(m)}$ von A^m gibt die Anzahl der Kantenzüge der Länge m von Knoten i zu Knoten j an. Die Matrix A^m ist das Ergebnis der Multiplikation von m Kopien der Matrix A.

Insbesondere sagt die Definition aus, dass der Koeffizient $a_{jj}^{(2)}$ gleich der Anzahl der Nachbarn von Knoten j ist. Dies folgt aus der Tatsache, dass ein Pfad der Länge 2 zwischen Knoten j und einem anderen Knoten k genau dann existiert, wenn j und k Nachbarn sind.

Ein Graph G heißt **zusammenhängend**, wenn es zwischen allen zwei Knoten einen Weg gibt. Ein maximaler zusammenhängender Teilgraph von G heißt eine **Komponente** von G. Für einen zusammenhängenden Graphen gilt:

- Ein zusammenhängender Graph mit n Knoten muss mindestens $n-1$ Kanten haben.
- Ein Graph mit mehr als $\frac{(n-1)(n-2)}{2}$ Kanten ist zusammenhängend.

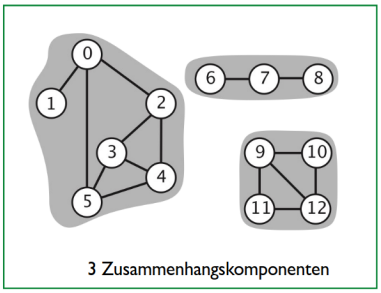
Es gibt verschiedene Methoden, um zu überprüfen, ob ein gegebener Graph mit n Knoten zusammenhängend ist:

- Mittels Matrixmultiplikation der Adjazenzmatrix. Die Laufzeit der Matrixmultiplikation is allerdings sehr hoch ($O(n^3)$)
- Breitensuche (ungerichtet): Starte mit einem beliebigen Knoten und markiere ihn. Schritt für Schritt markieren wir alle Knoten, die mit einer Kante zu einem bereits markierten Knoten verbunden sind. Wenn wir keine mehr markieren können, überprüfe, ob alle n Knoten markiert sind. Die Laufzeit ist $O(n)$,
- Breitensuche (gerichtet): Wie die ungerichtete Breitensuche, aber wir probieren so lange verschiedene nicht markierte Startknoten aus, bis alle Wege gefunden sind.

8.2.7 Zusammenhangskomponenten

Eine Zusammenhangskomponente ist eine maximale Menge an verbundenen Knoten. Es ist also ein Untergraphen eines ungerichteten Graphen, in dem jeder Knoten mit jedem anderen Knoten durch einen Pfad verbunden ist (Äquivalenzrelation, siehe Union Find). In anderen Worten, es gibt keine zwei Knoten in einer Zusammenhangskomponente, zwischen denen kein Pfad existiert. Darüber hinaus hat eine Zusammenhangskomponente keine Verbindung zu anderen Knoten außerhalb der Komponente.

2 Knoten v und w sind stark zusammenhängend, wenn es einen direkten Pfad von v nach w und einen direkten Pfad von w nach v gibt. Starker Zusammenhang ist also eine Äquivalenzrelation.

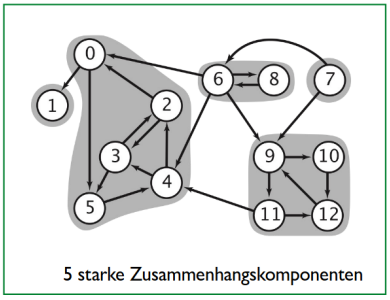


Zusammenhangskomponenten ID (leicht mit DFS zu berechnen)

	0	1	2	3	4	5	6	7	8	9	10	11	12
cc[]	0	0	0	0	0	0	1	1	1	2	2	2	2

Abbildung 5: Beispiel von Zusammenhangskomponenten eines Graphens

Eine starke Zusammenhangskomponente ist eine maximale Untermenge an stark zusammenhängenden Knoten eines Graphens.



starke Zusammenhangskomponenten ID (wie berechenbar?)

	0	1	2	3	4	5	6	7	8	9	10	11	12
scc[]	1	0	1	1	1	1	3	4	3	2	2	2	2

Abbildung 6: Beispiel von starken Zusammenhangskomponenten eines Graphens

9 Quellenkodierung und Kompression

Das Ziel von Quellenkodierung ist eine Nachricht mit möglichst wenig Bits zu übetragen. Es geht also um Kompression.

Man unterscheidet zwischen verlustfreier (lossless) und verlustbehafteter (lossy) Kompression. Wir behandeln nur lossless Kompression.

9.1 Präfixfreier Code

Ein **präfixfreier Code** ist ein Code, in dem kein Codewort ein Präfix eines anderen Codeworts ist. Das bedeutet, dass es in einem solchen Code kein Codewort gibt, das mit dem gleichen Muster beginnt wie ein anderes Codewort. Präfixfreie Codes sind sofort decodierbar, da jedes Codewort eindeutig ist und nicht mit einem anderen Codewort verwechselt werden kann. Sie werden häufig in der Datenkompression und der Informationsübertragung verwendet.

9.1.1 Beispiel

Ein einfaches Beispiel für einen präfixfreien Code könnte wie folgt aussehen:

- A -> 00
- B -> 01
- C -> 10
- D -> 11

In diesem Code ist kein Codewort ein Präfix eines anderen.

Ein Nicht-Beispiel (also kein präfixfreier Code) wäre:

- A -> 0
- B -> 01
- C -> 011
- D -> 0111

Hier ist das Codewort für A ein Präfix aller anderen Codewörter, und das Codewort für B ist ein Präfix der Codewörter für C und D, und so weiter. Dies kann zu Verwirrung führen, da eine Sequenz von Nullen und Einsen auf mehrere Arten interpretiert werden könnte.

9.2 Mittlere Codewortlänge

Gegeben sind die zu kodierenden Symbole $s_1, \dots, s_n$, die mit den Wahrscheinlichkeiten $p_1, \dots, p_n$ vorkommen. Für jedes Symbol s_i ist ein Codwort c_i mit der Länge l_i definiert.

Die gewichtete Summe ist die mittlere Codewortlänge und wird berechnet durch

$$L(X, C) = \sum_{i=1}^{n} p_i l_i$$

9.3 Huffman-Code

9.3.1 Grundgedanke

Der erste Gedanke bei jeglicher Art von Kodierung ist eine feste Codewortlänge zu definieren und jedem Symbol (oder was auch immer kodiert werden soll) ein festes Codewort zuzuweisen. Meistens kommt aber nicht jedes Symbol gleichhäufig vor. Das erkennt man sehr gut an (UTF-8 kodiertem) Text: Besonders die Vokale kommen häufig vor, aber dennoch sind deren Codewörter genauso lang, wie alle anderen Symbole.

Es wäre also sinnvoll, die Codewörter kürzer zu machen, die am häufigsten vorkommen. Dabei muss aber dennoch der Start und das Ende jedes Codeworts klar sein. Das geht durch einen präfixfreien Code.

Der Huffman-Code ist genau ein solcher Code. Durch den Algorithmus können wir einen optimalen präfixfreien binären Code mithilfe eines binären Baumes erstellen.

9.3.2 Algorithmus

Der Algorithmus zur Erstellung eines Huffman-Codes ist relativ einfach und besteht aus den folgenden Schritten:

1. Erstelle eine Prioritätsliste, in dem jedes Symbol und seine Wahrscheinlichkeit aufgelistet sind.
2. Wähle die beiden Symbole mit der geringsten Wahrscheinlichkeit aus und füge sie zu einem neuen Knoten zusammen. Die Summe ihrer Wahrscheinlichkeiten ist die Wahrscheinlichkeit des neuen Knotens.
3. Entferne die ursprünglichen Symbole aus der Liste und füge den neuen Knoten hinzu.
4. Wiederhole die Schritte 2 und 3, bis nur noch ein Knoten übrig ist. Dieser Knoten repräsentiert den Huffman-Baum.

Jeder Pfad von der Wurzel bis zu einem Blatt in diesem Baum stellt den Code für das entsprechende Symbol dar. Links geht man für eine 0 und rechts für eine 1. Die Symbole, die am häufigsten vorkommen, haben die kürzesten Pfade und somit die kürzesten Codes.

9.3.3 Beispiel

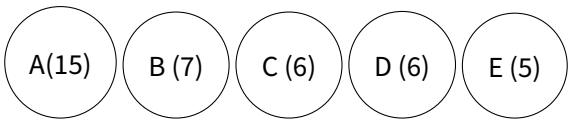
Angenommen, wir haben die folgenden Symbole und ihre Häufigkeiten:

- A: 15
- B: 7

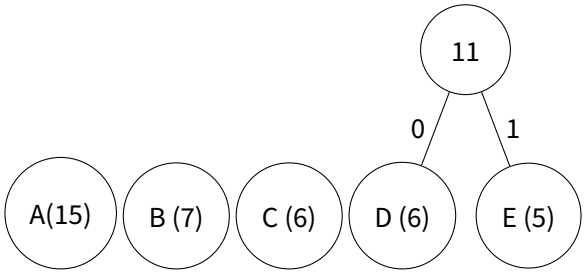
- C: 6
- D: 6
- E: 5

Der Huffman-Algorithmus würde dann wie folgt arbeiten:

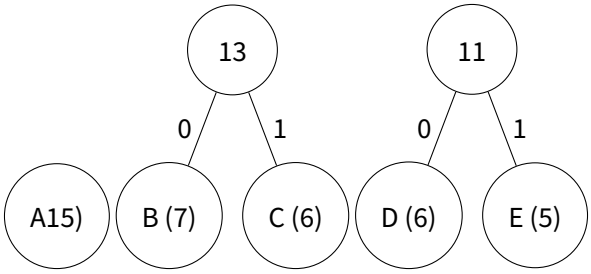
1. Beginne mit der Liste {A:15, B:7, C:6, D:6, E:5}.



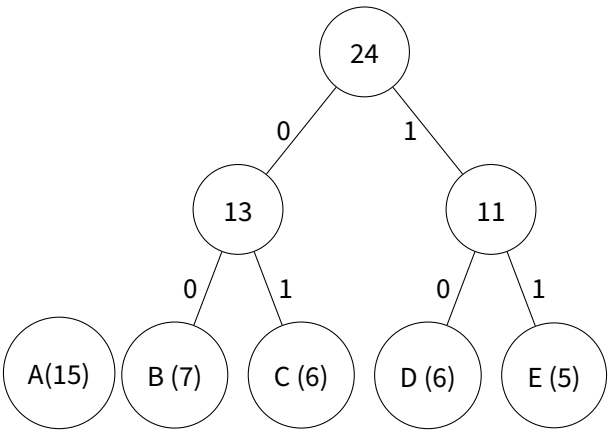
2. Wähle die Symbole mit der geringsten Häufigkeit aus (D und E) und füge sie zu einem neuen Knoten mit der Häufigkeit 11 zusammen. Die Liste wird zu {A:15, B:7, C:6, DE:11}.



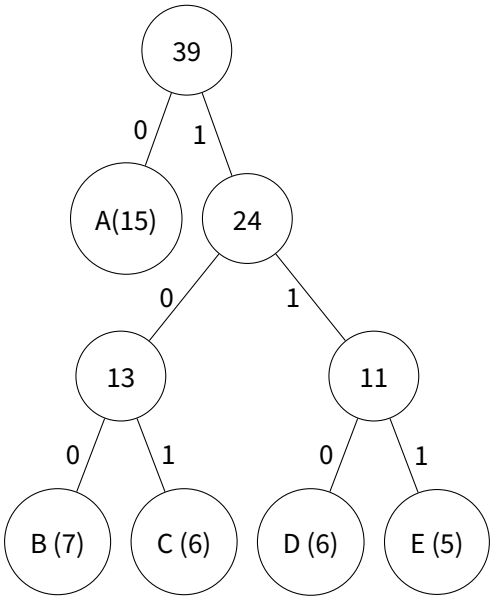
3. Wiederhole den Vorgang und wähle die Symbole mit der geringsten Häufigkeit aus (B und C) und füge sie zu einem neuen Knoten mit der Häufigkeit 13 zusammen. Die Liste wird zu {A:15, BC:13, DE:11}.



4. Wiederhole den Vorgang erneut und füge die Symbole mit der geringsten Häufigkeit (DE und BC) zu einem neuen Knoten mit der Häufigkeit 24 zusammen. Die Liste wird zu {A:15, BCDE:24}.



5. Schließlich füge die verbleibenden Symbole zusammen, um den finalen Huffman-Baum zu erstellen {ABCDE}.



Das Codewort für jedes Symbol ist also wie folgt:

- A -> 0
- B -> 100
- C -> 101
- D -> 110
- E -> 111

Der Huffman-Code ist optimal im Sinne, dass kein anderer präfixfreier Code eine kürzere mittlere Codewortlänge hat. Das macht ihn sehr effektiv für die Datenkompression.

9.4 Kompletter Kommunikationskanal

Ein “kompletter” Kommunikationskanal nutzt Quellenkodierung (Kompressionen) und Kanalkodierung (Redundanz).

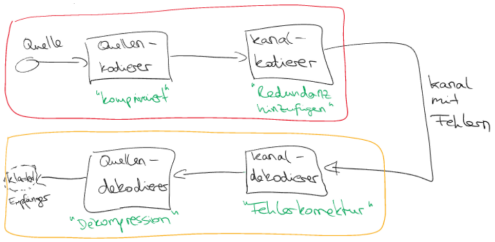


Abbildung 7: “Kompletter” Kommunikationskanal