

Theoretische Grundlagen der Informatik

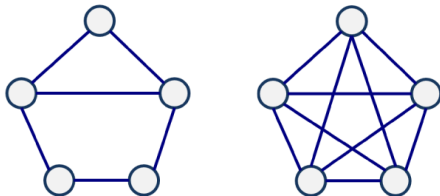
Vorlesung am 7. Februar 2019

INSTITUT FÜR THEORETISCHE INFORMATIK



- Vorlesung “*Algorithmen für planare Graphen*”
- Proseminar “*Algorithmen für \mathcal{NP} -schwere Probleme*”
- ICPC Praktikum

Ein planarer Graph ist ein Graph, der in der Ebene gezeichnet werden, ohne dass die Kanten sich kreuzen.



- besonders einfache, schnelle und schöne Algorithmen
- manche Probleme, die auf allgemeinen Graphen (\mathcal{NP} -)schwer sind, können auf planaren Graphen sehr effizient gelöst werden
- 5 ECTS-Punkte

Proseminar “Algorithmen für \mathcal{NP} -schwere Probleme”

- Fixed-Parameter Algorithmen
- Laufzeiten der Form $f(k) \cdot \text{poly}(n)$
- Vertieft und erweitert diese Vorlesung
 - eingeschränkte Graphenklassen
 - Independent Set, Vertex Cover
 - Baumzerlegungen, dyn. Programmieren
- Prüfbar im Bachelorstudium
- 3 ECTS-Punkte
- Anmeldung bei torsten.ueckerdt@kit.edu

Problem	Good news	Bad news
BAR FIGHT PREVENTION	$O(2^k \cdot k \cdot n)$ -time algorithm	\mathcal{NP} -hard (probably not in \mathcal{P})
CLIQUE with Δ	$O(2^\Delta \cdot \Delta^2 \cdot n)$ -time algo.	\mathcal{NP} -hard (probably not in \mathcal{P})
CLIQUE with k	$n^{O(k)}$ -time algorithm	$W[1]$ -hard (probably not FPT)

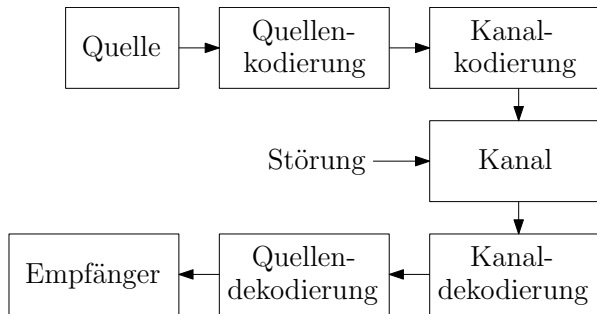
ACM International Collegiate Programming Contest (ICPC)

- kleine algorithmische Aufgaben (< 60 Min)
 - Aufgabe verstehen
 - Algorithmus entwerfen
 - Algorithmus implementieren (C++ oder Java)
- Funktionalität und Laufzeit des Codes wird bewertet
 - Kompiliert & übersteht Testdaten \implies Aufgabe gelöst

Vorbereitungspraktikum im Sommersemester

- 4 ECTS-Punkte
- Interessant für Leute, die:
 - beim ICPC mitmachen wollen
 - C++ üben wollen
 - Spaß an kleinen algorithmischen Aufgaben haben
- Details: <http://icpc.iti.kit.edu> **wenig Plätze, schnell anmelden!**

Kodierung zum Schutz gegen Übertragungsfehler



Kodierung zum Schutz gegen Übertragungsfehler

- Qualität einer digitalen Übertragung wird häufig als gemessene Bitfehlerquote bzw. Bitfehlerwahrscheinlichkeit angegeben.
- Beherrschung von Übertragungsfehlern:
 - Fehlerkorrektur (beim Empfänger),
 - Fehlererkennung und Wiederholungsanforderung.
- Trade-off: Wahrscheinlichkeit unentdeckter Fehler vs. Datendurchsatz.



Quelle: Wikipedia

- Paritätscode der RS232-Schnittstelle.
- Neunpoliges Kabel ermöglicht die parallele Übertragung von 8 Bits.
- Dabei werden nur sieben Bits b_1, \dots, b_7 für die Nachrichtenübertragung genutzt.
- Das achte Bit b_8 wird Paritätsbit genannt.

- Paritätscode der RS232-Schnittstelle.
- Neunpoliges Kabel ermöglicht die parallele Übertragung von 8 Bits.
- Dabei werden nur sieben Bits b_1, \dots, b_7 für die Nachrichtenübertragung genutzt.
- Das achte Bit b_8 wird Paritätsbit genannt.

Wiederholung XOR-Verknüpfung \oplus

\oplus	0	1
0	0	1
1	1	0

- Es wird b_8 so gesendet, dass

$$b_8 = b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6 \oplus b_7.$$

Wiederholung XOR-Verknüpfung \oplus

\oplus	0	1
0	0	1
1	1	0

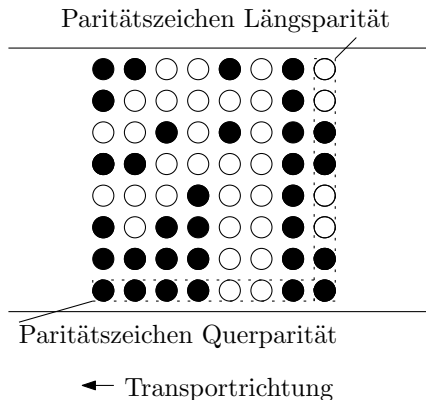
- Es wird b_8 so gesendet, dass

$$b_8 = b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6 \oplus b_7.$$

- Mit dem Paritätscode werden einfache Fehler im Codewort erkannt.
- Gleichzeitige Übertragung von 2 Fehlern wird nicht erkannt.
- Falls ein Fehler erkannt wird, kann die ursprüngliche Nachricht nicht rekonstruiert werden: Die Fehlerstelle ist unbekannt.

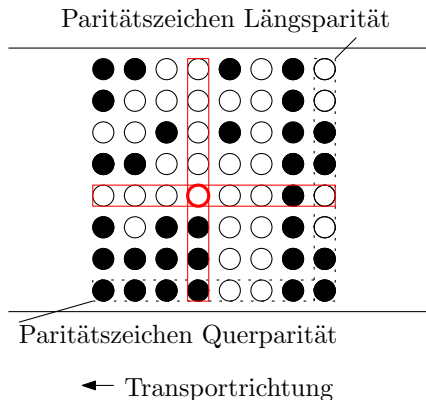
Kreuzsicherung

- Dient zum Schutz gegen Doppelfehler
- Erklärung am Beispiel Lochkarte



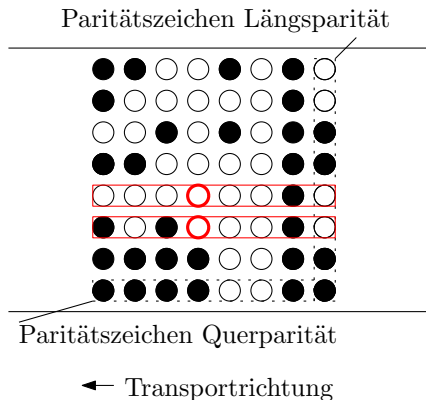
Kreuzsicherung

- Alle 1,2,3 fachen Fehler sind erkennbar
- Ab 4 Fehlern nicht zwingend erkennbar



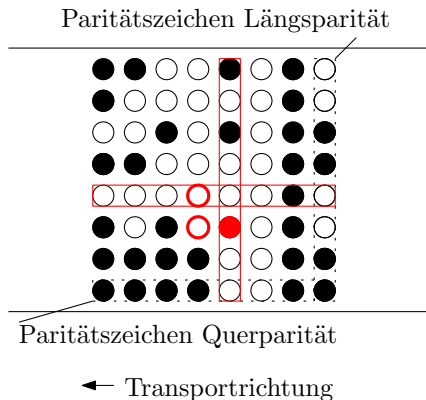
Kreuzsicherung

- Alle 1,2,3 fachen Fehler sind erkennbar
- Ab 4 Fehlern nicht zwingend erkennbar



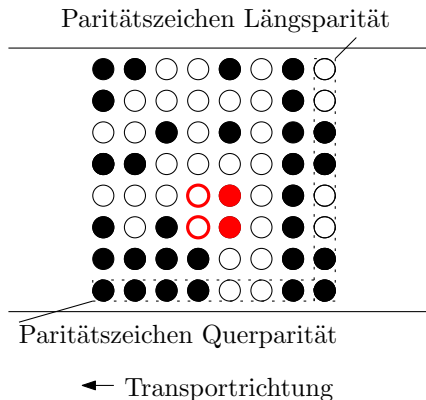
Kreuzsicherung

- Alle 1,2,3 fachen Fehler sind erkennbar
- Ab 4 Fehlern nicht zwingend erkennbar



Kreuzsicherung

- Alle 1,2,3 fachen Fehler sind erkennbar
- Ab 4 Fehlern nicht zwingend erkennbar



Paritätscodes

- Gegeben ein Alphabet $\Sigma = \{1, 2, \dots, q - 1\}$.
- Ein Code der Länge n zur Basis q sei eine Menge von Folgen mit Elementen aus Σ .
- Einzelne Folgen werden Codewörter genannt.

Ein Paritätscode liegt vor, wenn für jedes Codewort $a_1 a_2 \cdots a_n$

$$(a_1 + a_2 + \cdots + a_n) \bmod q = 0$$

gilt.

Satz:

Jeder Paritätscode erkennt Einzelfehler.

- Sei $a_1 \cdots a_n$ das ursprüngliche Codewort.
- Annahme: Das i -te Element wurde fehlerhaft als \tilde{a}_i übertragen.
- Ein Übertragungsfehler liegt vor, wenn

$$(a_1 + a_2 + \cdots + \tilde{a}_i + \cdots + a_n) \bmod q = 0.$$

- Für das ursprüngliche Codewort gilt:

$$(a_1 + a_2 + \cdots + a_n) \bmod q = 0.$$

Also

$$\begin{aligned} 0 &= (a_1 + a_2 + \cdots + \tilde{a}_i + \cdots + a_n) \bmod q \\ &= (a_1 + a_2 + \cdots + a_n) \bmod q \end{aligned}$$

$$\begin{aligned} 0 &= (a_1 + a_2 + \cdots + \tilde{a}_j + \cdots + a_n) \pmod q \\ &= (a_1 + a_2 + \cdots + a_n) \pmod q \end{aligned}$$

Damit

$$\begin{aligned} 0 &= (a_1 + a_2 + \cdots + \tilde{a}_j + \cdots + a_n) \pmod q - \\ &\quad (a_1 + a_2 + \cdots + a_n) \pmod q \\ &= (\tilde{a}_j - a_j) \pmod q \end{aligned}$$

Um dies zu erfüllen muss $(\tilde{a}_j - a_j)$ durch q teilbar sein. Weil aber

$$0 \leq \tilde{a}_j, a_j < q$$

folgt

$$0 \leq |\tilde{a}_j - a_j| < q .$$

- Häufige Fehlerart bei manueller Eingabe: Vertauschungsfehler.
- Diese werden von gewöhnlichen Paritätscodes nicht erkannt.
- Sei wieder $a_1 \cdots a_n$ ein Codewort.
- **Paritätscode mit Gewichten:** Wir führen zusätzlich ganzzahlige Gewichte w_1, \dots, w_{n-1} ein, so dass

$$(w_1 \cdot a_1 + w_2 \cdot a_2 + \cdots + w_{n-1} \cdot a_{n-1} + a_n) \pmod q = 0$$

gilt.

- Zusatzbedingung: Alle Gewichte w_i müssen teilerfremd zu q sein.

- **Paritätscode mit Gewichten:** Wir führen zusätzlich ganzzahlige Gewichte w_1, \dots, w_{n-1} ein, so dass

$$(w_1 \cdot a_1 + w_2 \cdot a_2 + \dots + w_{n-1} \cdot a_{n-1} + a_n) \pmod q = 0$$

gilt.

- Zusatzbedingung: Alle Gewichte w_i müssen teilerfremd zu q sein.

Satz:

Jeder Paritätscode mit Gewichten erkennt Einzelfehler.

Beweis: Analog zu normalen Paritätscodes kann gezeigt werden, dass für jeden Einzelfehler $a_i \rightarrow \tilde{a}_i$ gilt

$$w_i \cdot (\tilde{a}_i - a_i) \pmod q = 0 .$$

- **Paritätscode mit Gewichten:** Wir führen zusätzlich ganzzahlige Gewichte w_1, \dots, w_{n-1} ein, so dass

$$(w_1 \cdot a_1 + w_2 \cdot a_2 + \dots + w_{n-1} \cdot a_{n-1} + a_n) \pmod q = 0$$

gilt.

- Zusatzbedingung: Alle Gewichte w_i müssen teilerfremd zu q sein.

Satz:

Ein Paritätscode mit Gewichten erkennt die Vertauschung an den Stellen i und j , falls die Zahl $w_i - w_j$ teilerfremd zu q ist.

Beweis: Analog zu oben: Vertauschungsfehler wird nicht erkannt, falls

$$[(w_i a_i + w_j a_j) - (w_j a_i + w_i a_j)] \pmod q = [(w_i - w_j)(a_i - a_j)] \pmod q = 0.$$



Beschreibung ISBN-10 (oben)

- Alphabet $\Sigma = \{0, 1, \dots, 9\}$,
- Basis $q = 11$ (Primzahl!),
- Länge $n = 10$,
- Paritätscode,
- Für Code $a_1 \cdots a_{10}$ berechnet sich die Prüfziffer a_{10} aus

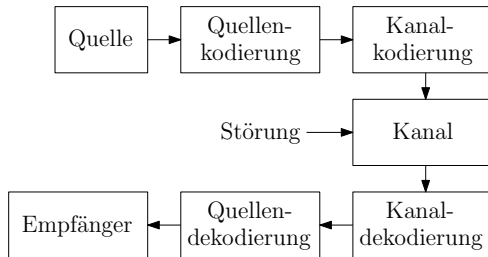
$$(10a_1 + 9a_2 + 8a_3 + \cdots + 2a_9 + a_{10}) \bmod 11 = 0.$$

- $10a_1 + \cdots + 2a_9 = 275 = 11 \cdot 25$
 $\Rightarrow a_{10} = 0$

Bsp: ISBN-10



- ISBN: International Standard Book Number (ISO Standard 2108)
- ISBN-10 war bis 2006 übliche Kodierung
- Seit 2007 EAN-13 (European Article Number)



- Man unterscheidet verschiedene Arten von Kanal-Codes.
- **Block-Codes:** Hier betrachtet man Codeworte fester Länge. Aufeinanderfolgende Blöcke werden unabhängig voneinander kodiert.
- **Faltungs-Codes:** Codeworte können beliebig lang sein. Die Zeichen sind vom Vorgeschehen abhängig.
- In TGI befassen wir uns ausschließlich mit Block-Codes.

Hamming-Distanz

Für $x, y \in \{0, 1\}^n$ ist

$$d(x, y) := \#\{i \mid i = 1, \dots, n, x_i \neq y_i\}$$

die Hamming-Distanz zwischen x und y .

Anschaulich: Die Hamming-Distanz zwischen x und y ist die Anzahl der Zeichen in x , die sich von denen in y unterscheiden.

Es sei $B_r(x)$ die Menge aller Worte y mit $d(x, y) \leq r$.

Anschaulich: B_r ist eine Kugel (die Hamming-Kugel) um x mit Radius r .

Hamming-Distanz

Für $x, y \in \{0, 1\}^n$ ist

$$d(x, y) := \#\{i \mid i = 1, \dots, n, x_i \neq y_i\}$$

die Hamming-Distanz zwischen x und y .

Anschaulich: Die Hamming-Distanz zwischen x und y ist die Anzahl der Zeichen in x , die sich von denen in y unterscheiden.

Maximum-Likelihood-Decoding

Sei eine Kodierung C gegeben und y ein empfangenes Wort. Dekodiere y als dasjenige Codewort $x \in C$, für das $d(x, y)$ minimal wird.

Block-Code

- Gegeben ist ein endliches Alphabet Σ .
- Ein Block-Code ist eine Teilmenge $C \subseteq \Sigma^n$ für ein $n \in \mathbb{N}$.
- Falls $\#C = 1$, so heißt C trivial, da es nur ein Codewort gibt.

Minimaldistanz

Die Minimaldistanz eines nichttrivialen Block-Codes C ist

$$m(C) := \min_{c_1, c_2 \in C, c_1 \neq c_2} d(c_1, c_2) .$$

Zeichen	Codeworte C
A	00000
H	10011
L	11100
O	01111

Maximum-Likelihood-Decoding

- Empfänger dekodiert 00010 als A, weil $d(00010, 00000) = 1$.
- Minimaldistanz ist $m(C) = \min\{3, 3, 4, 4, 3, 3\} = 3$.
- Es werden 2 Fehler erkannt.
- Es kann 1 Fehler korrigiert werden.

Ziel:

- Finde Codeworte C in $\{0, 1\}^n$ mit $m(C)$ groß, also paarweise großer Hamming-Distanz.

Satz:

Ein Block-Code C mit Minimaldistanz $m(C) = d$ kann bis zu $d - 1$ Fehler erkennen und bis zu $\lfloor \frac{d-1}{2} \rfloor$ Fehler korrigieren.

Beweisskizze

