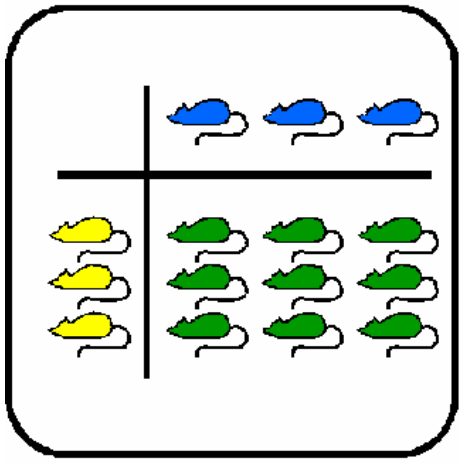
Multiplikation



Error-correcting codes

Beitrag zu "Werkstattunterricht Multiplikation"

Allgemeine Didaktik - Seminar SS95

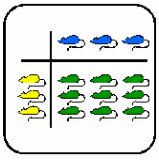
Autor:

Ernesto Ruggiano
Oberwiesenstr. 42
8050 Zürich

eruggian@inf.ethz.ch

Betreuer

Prof. Werner Hartmann



Thema: Error Correcting-Codes, Red-Muller Code

Schultyp: Mittelschule, technische Berufsschule, Fachhochschule

Vorkenntnisse: Begriffe Bit, BCD und Matrix

Bearbeitungsdauer: 45-60 Minuten

Fassung vom: 28.7.95

Schulerprobung: nein

Übersicht

Was heisst, eine Nachricht zu verschlüsseln? Wie kann ich eine fehlerhafte Nachricht erkennen? Wie kann ich in einer Nachricht einen Übermittlungsfehler korrigieren?
Dies sind einige Fragestellungen, die an diesem Postens behandelt werden: er wurde als Einführung in die Thematik der Codes und der Kryptologie konzipiert. Er kann zum Beispiel als Einstieg in die Hamming Codes benützt werden.

Lernziele

Nach der kompletten Ausführung des Postens, sollten folgende Ziele erreicht sein:

1) Die Schüler wissen, was ein Code ist und wie man ihn aufbaut. Sie kennen das Konzept der Hamming Distanz.
2) Sie wissen, welche Eigenschaften ein Code besitzen muss, um einen oder mehrere Fehler zu korrigieren.
3) Sie kennen eine Art von Code, der die Korrektur wenigstens eines Fehlers garantiert. (Hier wird die Reed-Muller Methode erklärt, aber andere Lehrer können zum Beispiel den berühmten Hamming Code vorstellen.)

Material

- PC-Raum
- Theorie: *Error Correcting-Codes*

Quellen

DewdneyA.K.: The Touring Omnibus - 61 Excursion in Computer Science. USA 1989 (Computer Science Press)

Hinweise, Lösungen

Lösung Auftrag 1

Für diesen Auftrag gibt es viele korrekte Lösungen. Es genügt, wenn der Schüler zwei Serien von Bit (eine um ”S.”, die andere um ”O.” darzustellen) mir einer Hamming Distanz ≥ 3 angeben.
Ein Beispiel:

S = 1001
O = 0111

Hinweis:

Die Theorie sollte nicht ausgeteilt werden, bevor der erste Auftrag beendet ist: die Schüler könnten ansonsten in Versuchung geraten, sie sofort zu lesen (vor allem wenn sie in Gruppen arbeiten).

Beschreibung des Programms:

Um die Simulation realistisch durchzuführen, prüfen die Schüler ihre Codes mit Hilfe eines Programms.
Das Programm soll die Definition eines Codes erlauben, also die Schülerinn kann einen Code einem Buchstaben zuweisen ('A'= 01).
Dazu soll das Programm eine Übermittlung eines Codes simulieren. Das bedeutet ein oder mehrere Bits des Code zu verändern, gemäss dem Erdabstand. Für Auftrag1 soll der Abstand so gesetzt, dass genau einen Fehler während der "Übertragung" stattfindet.
Welche Bit(s) wir verändern müssen, kann mit einer Zufallszahlengenerator bestimmt werden.

Lehrer-Lernkontrolle / Test

Aufgabe 1

Du befindest dich wieder in deinem Raumschiff. Deine Aufgabe besteht darin, folgende verschlüsselte Nachricht zu übermitteln:

”I’m OK”.

Finde einen Code, der es möglich macht, dass sie auf der Erde **wenigstens einen** Übertragungsfehler korrigieren können.
Deine Antwort ist in Ordnung, wenn es mit deinem Code möglich ist, wenigstens einen Fehler zu erkennen.

Aufgabe 2

In der Theorie hast du gelernt, welche Eigenschaften ein Code besitzen muss, um einen oder mehrere Übertragungsfehler zu korrigieren. Erkläre nun mit deinen Worten, weshalb die folgende Kodierung einen Fehler erkennt und korrigiert.

A = 010101
B = 101010
C = 111111

Die Aufgabe ist erfüllt, wenn du den Grund erklären kannst, ohne den Begriff der Hamming Distanz zu benützen.

Lösungen und Taxierung

Aufgabe 1

Eine mögliche Lösung ist:

" I " = 00000000
" ' " = 01010101
" m " = 00110011
" O " = 00001111
" K " = 11111111

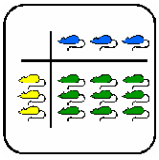
Um die Übung zu lösen, muss der Schüler die Theorie benützen. Das ist für ihn die erste Gelegenheit, die gelernte Methode (in diesem Fall die Reed-Muller Methode) in die Praxis umzusetzen. Die Übung gehört deswegens zu Stufe **K3**.

Aufgabe 2

Typische Antwort: "Da HD = 3 ist, kann ich (3 - 1) /2 = 1 Fehler korrigieren."
Verlangte Antwort:
Sie korrigiert einen Fehler, weil irgendein ausgetauschtes Bit in einer der drei Serien es verunmöglicht, die gesendete Serie zu erkennen.
Einige Beispiele:

001010 => B
000101 => A
101111 => C
usw.

Die Frage ”erkläre weshalb ...” ist ein typisches Beispiel für eine Aufgabe der Stufe **K2.**



Werkstatt Multiplikation Auftragblatt
Posten: **Error-correcting codes**

Was soll ich hier tun?

Für diesen Posten ist folgender Ablauf vorgesehen:

(1) Lese die Instruktionen des Captains und führe sie aus (in Gruppen oder allein). (35 Minuten)
(2) Entdecke das Ende der Instruktionen in der Theorie! (allein) (15 Minuten)

”What are my instructions, captain?”

Du befindest dich an Bord eines Raumschiffes, Lichtjahre von der Erde entfernt. Der Computer ist das einzige Mittel, das du und deine Kameraden besitzen, um mit der Erde zu kommunizieren. Öffne nun den Umschlag mit den geheimen Instruktionen des Captains, wie man Nachrichten mit dem Computer übermitteln kann.

Jede Nachricht besteht aus einer Serie von Bits. Der Buchstabe ”A” kann zum Beispiel der Serie 01 entsprechen und ”B” der Serie 00.
*Die Serien wie 01 heissen **Code**:*

01 ist der Code von A
00 ist der Code von B.

Deine Aufgabe besteht darin, einen Code zu erfinden, um mit der Erde zu kommunizieren.

***Achtung:**Während des Übermittlungsweges ist deine Nachricht Störungen unterworfen. Je weiter du dich von der Erde entfernt befindest, desto mehr verändern die Störungen deine Nachricht!*

Du musst deshalb einen Code schaffen, der es auf der Erde erlaubt, deine ursprüngliche Nachricht zu rekonstruieren. Ich gebe dir ein Beispiel.
Nehmen wir an, dass du einen Code für die vier mathematischen Operationen +, -, x, / schaffen willst. Du teilst also jedem Symbol eine Serie von Bits zu, zum Beispiel:

+ = 00
- = 01
x = 10
/ = 11

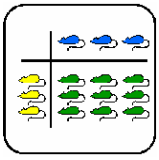
Du übermittelst + = 00. Allerdings findet ein Übertragungsfehler statt, der ein Bit verändert., sodass sie auf der Erde die Serie 10 empfangen. Diese wird als Multiplikation interpretiert !
Wenn du hingegen längere Serien definierst, wie z.B.:

+ = 0000
- = 0101
x = 1010
/ = 1111

dann wird ein Übertragungsfehler erkannt. Nehmen wir an,dass du ein + = 0000 sendest und auf der Erde erhalten sie 0010. Nun wissen die Leute der Erdbasis, dass ein

Übertragungsfehler stattgefunden hat. Sie können aber noch nicht auf die gesendete Serie schliessen:

$$0010 = \begin{cases} 1010 = \times\ ? \\ 0000 = +\ ? \end{cases}$$ *Wo liegt der Fehler?*



Werkstatt Multiplikation Auftragblatt
Posten: **Error-correcting codes**

Wir sehen , dass je länger den Code ist desto mehr Möglichkeiten es zu korrigierern gibt.
Als nächste Schritt definieren wir:

+ = 00000
- = 10101
x = 01110
/ = 11100

Jetzt kann ein Fehler entdeckt werden: du sendest 00000 und auf der Erde empfängen 00010. Sie wissen , dass mit einem Fehler 00010 nur als 00000 interpretiert werden kann.
Ziel ist es die Beziehung zwischen die Länge des Codes und die maximale Anzahl von korrigierbaren Fehler und einen Algorithmus, um einen Code herzustellen, zu finden.
Wir von der "Mariner series of Mars" haben die Beziehung berchnet und einen Algorithmus entwickelt .
Ich gebe dir die Beziehung zuerst.

ALARM!

Du riskierst den Zusamenstoss mit einem Meteor!

Du hast keine Zeit mehr, die Instruktionen des Captains zu lesen. Sie sind zu lang!
Du hast eine halbe Stunde Zeit, einen Code zu finden, um die Nachricht S.O.S zu übermitteln!
Denk daran, dass es bei deinem Code möglich sein muss, einen Fehler zu erkennen und zu korrigieren!
Nachdem du den Code geschaffen hast, benutze den Computer, um ihn zu übermitteln! Der Computer sagt dir, ob auf der Erde deinen Code richtig interpretiert würde.
Viel Glück !

Instructions ... second part !

Bemerkung: Die Theorie ist die Fortsetzung der im Auftragsblatt enthaltenen Instruktionen!

Ich gebe dir die Beziehung zuerst.

Hamming Distanz Beziehung (ohne Beweis)

Betrachten wir folgende Codierung als Beispiel:

\+ = 1111
\- = 1010
x = 1100
/ = 1001

Die Codes unterscheiden sich paarweise an genau zwei Stellen. Das bedeutet, dass 2 Übertragungsfehler ein Code in ein anderen verwandeln können. Eine 1111 Folge könnte als 1100 empfängt werden. Im Gegenteil wird 1 Fehler entdeckt: eine empfangene 1000 Folge wird als fehlerhafte erkennt. Es bleibt das Problem, dass das Fehler noch nicht korrigiert werden kann (war 1000 eine 1010 Folge, eine 1001 oder eine 1100 ?).

Definition:
der minimale Unterschied zwischen zwei Code-Zeilen heisst **Hamming Distance (HD).**
Im Beispiel ist HD = 2.

Theorem:

Ein Code mit einer HD = X , erkennt bis zu X-1 Fehler und korrigiert $\left\lfloor \frac{(X-1)}{2} \right\rfloor$ Fehler.

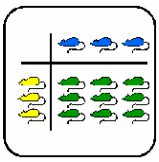
Im Beispiel HD = 2:

2 - 1 = 1 Fehler wird entdeckt und
$\left\lfloor \frac{2-1}{2} \right\rfloor = 0$ Fehler wird korrigiert.

Algorithmus

Auf der Erde müssen sie einen oder mehrere Fehler korrigieren können (je nach der Entfernung, in der du dich befindest). Wir von der *"Mariner series of Mars"* haben deshalb folgende Kodierungsalgorithmus entwickelt.
Diese Methode wird am besten mit einem Beispiel erklärt. Nehmen wir an, dass du die Farbe eines Punktes auf dem Bildschirm (Pixel) kodieren musst:

0 = schwarz
1 = Farbe
…
30 = Farbe
31 = weiss



Werkstatt Multiplikation Theorie
Posten: **Error-correcting codes**

Um diese Nachricht zu übermitteln, genügen dir, falls es auf dem Weg keine Störungen auftreten, 5 bit:

$$\begin{aligned} 0 &= 00000 \\ 1 &= 00001 \\ &\dots \\ 30 &= 11110 \\ 31 &= 11111 \end{aligned}$$

Um einen Code zu schaffen, der wenigstens einem Fehler "standhält", kannst du nun unsere Methode benützen.
Sie basiert auf der folgenden rekursiven Formel:

$$H_0 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ und}$$

$$H_{n+1} = H_n \otimes H_0$$

wobei das Symbol $\otimes$ die folgende Operation definiert:
ersetze jede 1 von H_n mit der Matrix H_0 und jede -1 von H_n mit der Matrix $-H_0$.
Zum Beispiel:

$$H_1 = H_0 \otimes H_0 \;\Rightarrow\; H_1 = \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} \end{bmatrix} \Rightarrow H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

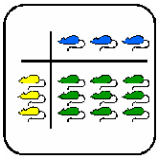
Jede Zeile von irgendeiner H_i entspricht einem Code (die -1's zählen als 0's).
Um nun den Code zu schaffen, brauchst du 32 verschiedenen Codes. Du muss also die erste Matrix H_i berechnen, die 32 Zeilen hat. Dafür brauchst du ein paar Überlegungen über das Aufbau der Matrizen H_i. H_0 hat zwei Zeilen, H_1 vier, H_2 acht und H_3 sechzehn. Die Operation $\otimes$ vergrössert die Matrix um den Faktor 2. Du kannst leicht überprüfen,dass die Anzahl von Zeilen der Matrix H_i gleich 2^{i+1} ist. Also ist H_4 hat $2^{4+1} = 32$ Zeilen: jede Zeile von H4 entspricht die 32 Codes der Farben.

H_4 :

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1
1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1
1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1
1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0

.....

Diese Methode wird **Reed-Muller Code** genannt.



Werkstatt Multiplikation Theorie
Posten: **Error-correcting codes**

Welches sind die Eigenschaften dieser Methode ?
Betrachten wir die Matrix H_1 und ihre Zeilen:

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

In H_1 unterscheiden sich die Zeilen paarweise an genau zwei Stellen.

Man kann nachweisen, dass sich für H_n die Zeilen paarweise an genau 2^n unterscheiden.
Für H_1 ist HD = 2^1, für H_4 ist HD = 2^4 =16, für H_5 ist HD = 2^5 = 32.
Was hat das für Konsequenzen ?
Für den Reed-Muller Code folgen daraus diese Eigenschaften:

Hn	HD =	erkennte Fehler	korrigierte Fehler	korrigiert mind. 1 Fehler
H1	2	1	0	nein
H2	4	2	1	ja
H3	8	4	3	ja
H4	16	8	7	ja
H5	32	16	15	ja
H6	64	32	31	ja
…	…	…	…	…

Letze Nachricht des Captains !

Achtung! :
Wenn du ein SOS übermitteln musst, genügt dir ein bit, falls es auf dem Weg keine Störungen gibt: S=0 O=1.
Wenn du aber sicher sein musst, dass wenigstens ein Fehler korrigiert wird, dann wähle für S bzw. für O irgendwelche zwei Zeilen von H2,H3,... aus!