Goethe-Gymnasium
Regensburg

Seminararbeit im Fach:
Mathematik

Einige dezimale Codes

XXXXXXXXX
Schuljahr: 2007/2008
bei
Markus Meiringer

Einige interessante Dezimalcodes

Vorwort

In der folgenden Seminararbeit beschäftigte ich mich intensiv mit einigen der wichtigsten Dezimalcodes.

Dezimalcodes haben eine gute und eine weniger gute Eigenschaft; denn einerseits kommen bei Dezimalcodes alle Ziffern, die es gibt, vor, was sie zu den größten (fast) buchstabenfreien Codes macht. Andererseits ist das Alphabet $Z/10Z$ kein Körper, da 10 keine Primzahl ist. Daher können gute Dezimalcodes nur konstruiert werden, indem man zuerst einen Code über das Alphabet $Z/11Z$ definiert und aus diesem dann alle Codewörter entfernt, bei denen an irgendeiner Stelle die Ziffer 10 steht.

Da ein wirklich guter Code abgesehen von einer hohen Fehlerkorrektur in erster Linie relativ groß sein muss, führe ich zu Beginn das Einschluss-Ausschluss-Prinzip ein, welches mir die Bestimmung der Größe eines Codes sehr erleichtert.

1 Das Einschluss-Ausschluss-Prinzip

1.1 Allgemeines

Bei dem Einschluss-Ausschluss-Prinzip (auch Prinzip von Inklusion und Exklusion) handelt es sich um eine nützliche Methode, um die Mächtigkeit einer Menge U zu bestimmen, wenn man von jeder Menge einer Folge U_n von Teilmengen von U, deren Vereinigung U ergibt, ihre Mächtigkeit weiß. Die einfachste Form dieses Prinzips erhält man, wenn man nur zwei Mengen A,B betrachtet. Dann gilt nämlich für die Mächtigkeit deren Vereinigung: $|A \cup B| = |A| + |B| - |A \cap B|$. Diese Formel ist nahe liegend, wenn man sich die einzelnen Schritte anschaut: Zuerst addiert man die Anzahl der Glieder von A und B; hierbei zählt man natürlich die Glieder, die sowohl in A als auch in B - also im Schnitt von A und B – enthalten sind, doppelt und zieht sie deshalb einmal wieder ab.
Das Einschluss-Ausschluss-Prinzip liefert nun eine Verallgemeinerung für eine beliebige Anzahl von Teilmengen.

Satz 1.1 *Es sei* $U_1,...,U_n$ *eine Folge von Mengen derart, dass* $\bigcup_{i=1}^{n} U_i = U$.

Dann gilt: $|U| = \sum_{i=1}^{n} |U_i| - \sum_{i<j} |U_i \cap U_j| + \sum_{i<j<k} |U_i \cap U_j \cap U_k| ... + (-1)^{n+1} |U_1 \cap U_2 ... \cap U_n|$.

Auf einen ausführlichen Beweis verzichte ich an dieser Stelle, da dies den Rahmen dieser Seminararbeit sprengen würde,

1.2 Anwendung in der Kodierungstheorie

In der Kodierungstheorie nutzt man dieses Prinzip um die Größe N eines n-ären Codes Y, der durch Reduktion eines m-ären Codes X mit der Größe M entstanden ist (m>n). Bei der Differenz U zwischen M und N handelt es sich um die Menge aller Codewörter von X, die an irgendeiner Stelle eine Zahl zwischen n und m-1 haben. Um U nun zu berechnen, bildet man die Folge der Mengen $U_1,U_2,...,U_n$, wobei es sich bei der Menge U_i um die Menge aller Codewörter von X handelt, bei denen an der i-ten Stelle eine Zahl wischen n und m-1 steht. Da die Vereinigung dieser Mengen U ergibt, kann man auf diese Mengen das Einschluss-Ausschluss-Prinzip anwenden, um U zu berechnen. Weiß man dann noch M, kann man N=M-U ermitteln.
In meiner Seminararbeit wird es immer darauf hinauslaufen, dass ich von einem bereits konstruierten 11-ären Code auf diese Weise die Anzahl der Codewörter ermittle, die an keiner Stelle eine 10 haben.
Zum Abschluss des Themas ein Beispiel:
Beispiel 1.1 Es sei der 11-äre Code R der Länge 4 mit der Prüfmatrix P=[1 1 1 1] gegeben. Man bestimme den Code S, der entsteht, indem man aus R alle Codewörter entfernt, die an irgendeiner Stelle eine 10 haben.
Lösung: Es sei U=R\S.
$|R|$ zu bestimmen ist trivial. Wenn ein Codewort die Form $x_1x_2x_3x_4$ hat, dann gibt es für jede Wahl von $x_1x_2x_3$ genau eine Möglichkeit für x_4, sodass $x_1x_2x_3x_4$ ein Codewort von R ist.
Also ist $|R| = 11^3 = 1331$.
Um $|U|$ zu bestimmen, benutze ich nun das Einschluss-Ausschluss-Prinzip:
Wenn U_1,U_2,U_3,U_4 die Mengen der Codewörter sind, die an der 1/2/3/4 Stelle eine 10 haben, dann gilt, wie sich der aufmerksame Leser leicht erschließen kann, dass $|U| = 4 \cdot 11^2 - 6 \cdot 11 + 4 = 422$. Also ist $|S| = 1331-422=909$.

2 Die ISBN-Codes

2.1 Allgemeines

Die Internationale Standardbuchnummer (kurz: ISBN) dient zur eindeutigen Kennzeichnung von Büchern. Sie wird von einer speziellen Behörde jedes Landes (in Deutschland z.B. der ISBN-Agentur für die Bundesrepublik Deutschland) an die dort ansässigen Verlage verteilt. Da die Beantragung einer ISBN-Nummer mit Kosten verbunden ist, verzichten einige Kleinverlage auf die Registrierung einer ISBN. Jedes Buch hat eine eigene ISBN-Nummer; selbst unterschiedliche Auflagen eines Buches haben andere ISBN-Nummern.
Das erste ISBN-Modell wurde 1972 von der Internationalen Organisation für Normung veröffentlicht, die sog. ISBN-10. Diese wurde bis 2006 verwendet. Ab dem 1.Januar 2007 stieg man aber auf die ISBN-13 um.

2.2 Der ISBN-10-Code

Bei dem ISBN-10-Code handelt es sich eigentlich nicht um einen echten Dezimalcode.
Er ist wie folgt definiert:

Definition 2.2 Der ISBN-10-Code ist der Code, der entsteht, wenn man aus dem 11-ären $(10, 11^9, 2)$-Code I mit der Prüfmatrix P=[1 2 3 4 5 6 7 8 9 10] alle Codewörter entfernt die an irgendeiner Stelle außer der letzten eine 10 haben.

Hat ein Codewort die Form $x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$, so ergibt sich, da -10=1 mod 11, folgende Prüfgleichung: $x_{10} = x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 + 7x_7 + 8x_8 + 9x_9 \mod 11$.
x_{10} nennt man auch Prüfziffer; ist diese gleich 10 so schreibt man dies als X, die römische 10. Man erlaubt für die Prüfziffer deshalb auch die 10, weil sich damit ganz leicht ISBN-Codewörter konstruieren lassen: man wählt x_1 bis x_9 beliebig und setzt diese dann in der obigen Prüfgleichung ein, um x_{10} zu erhalten. Außerdem lässt sich deswegen ganz leicht die Größe des ISBN-10-Codes angeben, nämlich 10^9. Die anderen Ziffern stehen der Reihe nach für die Sprache, den Verlag und die Titelnummer des Buches.
Das wirklich Interessante am ISBN-10-Code sind jedoch folgende zwei Eigenschaften:

Satz 2.2 *Der ISBN-10-Code ist 1-fehlererkennend.*

Bew.: Angenommen es wird ein Codewort c gesendet, aber ein Codewort x mit einem Fehler der Größe u an der i-ten Stelle wird empfangen, also $x=c+u e_i$. Folglich gilt:
$S(x)=S(c+u e_i)=S(c)+S(u e_i)=S(u e_i)=u \cdot i$
Damit der Fehler nicht erkannt wird, muss S(x)=0 gelten. Dies ist dann der Fall, wenn i oder u gleich 0 ist. Da i aber nicht 0 sein kann, müsste u gleich 0 sein. Ist u aber gleich 0, so liegt kein Fehler vor. Also ist der ISBN-10-Code 1-fehlererkennend.

q.e.d.

Satz 2.3 *Der ISBN-10-Code erkennt beliebige Vertauschungsfehler.*

Bew.: Angenommen es wird bei der Übertragung x_i und x_j vertauscht (i<j o.B.d.A.).
Dann gilt für das Fehlerwort e: $e=0\ldots0(x_j - x_i)0\ldots(x_i - x_j)0\ldots0$
Also ist das Syndrom $S=e P^t = i \cdot (x_j - x_i) + j \cdot (x_i - x_j) = (i-j)(x_j - x_i)$.
Damit S=0 ist, muss entweder i=j oder $x_i = x_j$ gelten. Dann liegt aber kein Fehler vor.
Also erkennt der ISBN-10-Code beliebige Vertauschungsfehler.

q.e.d.

2.3 Der ISBN-13-Code

Im Gegensatz zum ISBN-10-Code ist der ISBN-13-Code ein echter Dezimalcode. Er ist folgendermaßen definiert.

Definition 2.3 Der ISBN-13-Code ist der $(13,\ 2\cdot 10^9,\ 2)$-Dezimalcode, bei dem:

- an den ersten drei Stellen entweder 978 oder 979 steht und
- die Prüfmatrix P=[1 3 1 3 1 3 1 3 1 3 1 3 1] ist.

Die Einführung der ISBN-13 diente in erster Linie dazu, um den Zahlenraum des ISBN-Codes zu verdoppeln und die ISBN-Nummer an die EAN anzupassen. Die ersten drei Ziffern einer ISBN-13 bilden ein Präfix, die anderen zehn stimmen in ihrer Funktion mit den Ziffern der ISBN-10 überein. Der ISBN-13 erkennt, genau wie der ISBN-10-Code, einfache Fehler, allerdings nicht beliebige Vertauschungsfehler, sondern nur Vertauschungen von einer 2k-ten auf eine 2n+1-te Stelle und auch nur dann, wenn die Fehlergröße ungleich 5 ist. Da die Beweise analog zur ISBN-10 verlaufen, verzichte ich an dieser Stelle auf sie.

3 Ein 1-fehlerkorrigierender Dezimalcode

3.1 Konstruktion und Eigenschaften

Nach den ISBN-Codes beschäftige ich mich mit einem Dezimalcode, der nicht nur 1-fehlerkorrigierend ist, sondern auch eine Korrekturalternative zur Syndrom-Dekodierung bietet.
Um diesen Code zu konstruieren, betrachte ich zuerst den linearen 11-ären Hammingcode $H_{11}(2)$, welcher die Prüfmatrix $P = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}$ hat.
Bei dieser Prüfmatrix streiche ich nun die erste und letzte Koordinate und erhalte einen neuen linearen 11-ären Code C mit der Prüfmatrix $P' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}$.
Mein gesuchter Code D ist derjenige, der entsteht, indem man aus C alle Codewörter entfernt, die an irgendeiner Stelle eine 10 haben. D ist nichtlinear, da beim Entfernen dieser Codewörter die Abgeschlossenheit verloren geht.
Bevor ich mich mit der Fehlerkorrektur beschäftige, untersuche ich D zuvor noch auf seinen Minimalabstand und seine Größe:
Da der Hammingcode $H_{11}(2)$ einen Minimalabstand von 3 hat, ist es nahe liegend, dass auch d(D)=3 gilt. Um dies zu beweisen muss ich zeigen, dass

a) d(D)>2 und
b) d(D)<4 gilt.

Zu a) Angenommen, es unterscheiden sich zwei Codewörter d_1, d_2 aus D nur an der i-ten Stelle um a und an der j-ten Stelle um b (i<j o.B.d.A.). Hat d_1 die Form $x_1...x_i...x_j...x_{10}$, dann hat d_2 folglich die Form $x_1...(x_i + a)...(x_j + b)...x_{10}$. Setzt man d_1, d_2 nun jeweils in die aus der ersten Zeile entstehende Prüfgleichung ein und zieht die beiden Gleichungen voneinander ab, erhält man a+b=0 mod 10⇔a=-b mod 11.
Als Nächstes betrachte ich die aus der zweiten Zeile entstehende Prüfmatrix. Da diese die Form $x_2 + 2x_3 ... + zx_{z+1} ... + 9x_{10} = 0$ mod 11 hat, ergibt sich, wenn man d_1, d_2 jeweils einsetzt und die beiden Gleichungen voneinander abzieht, dass (i-1)a+(j-1)b=0 mod 11. Setzt man nun die obige Erkenntnis ein, so erhält man (i-1)a-(j-1)a=0 mod 10⇔a(i-j)=0 mod 11. Da 11 aber eine Primzahl ist, ist dies nur möglich, wenn a=0 mod 11 oder i-j=0 mod 11⇔i=j mod 11.
In beiden Fällen sind d_1 und d_2 aber identisch. Also ist d(D)>2.

Zu b) Um dies zu zeigen, genügt es als Beispiel die Codewörter a=2511111111 und b=0321111111 zu erwähnen. Beide sind Codewörter aus D, da sie die Prüfgleichungen erfüllen und d(a,b)=3. Also ist d(D)<4.

q.e.d.

Nachdem ich D auf seinen Minimalabstand geprüft habe, beschäftige ich mich nun mit seiner Größe:
Die Größe von C zu bestimmen, ist sehr leicht. Wählt man x_3 bis x_{10} beliebig so ergibt sich für x_2 gemäß der zweiten Prüfgleichung genau eine Lösung; setzt man nun wiederum x_2 bis x_{10} in die erste Prüfgleichung ein, ergibt sich auch für x_1 eine eindeutige Lösung. Also ist $|C|$ $=11^8$. Kniffliger ist es U=$|C| - |D|$ zu ermitteln; hierbei nutze ich das zu Beginn eingeführte Einschluss-Ausschluss-Prinzip.
Es sei S_i die Menge aller Codewörter, die an der i-ten Stelle eine 10 haben. Um das Ganze ein wenig zu beschleunigen, beweise ich folgenden Satz.

Satz 3.4. *Beim Code C gilt für n∈ {1…8}, dass die Anzahl aller Codewörter aus C, die an mindestens n Stellen eine 10 haben, gleich* $\binom{10}{n} \cdot 11^{8-n}$ *ist.*

Bew.: Als Erstes betrachte ich die Anzahl der Möglichkeiten, wie man die n-mal vorkommende Ziffer 10 auf 10 mögliche Stellen verteilen kann. Da ich bei dieser Art der Betrachtung alle von 10 verschiedenen Ziffern als gleich ansehe, gilt, dass diese Anzahl gleich $\frac{10!}{n!(10-n)!} = \binom{10}{n}$ ist. Da es bei jeder dieser Kombinationen genau 11^{8-n} mögliche Codewörter gibt, weil wegen der zwei Prüfgleichungen ein Codewort schon dann genau bestimmt ist, wenn man 8 Ziffern von ihm kennt, gibt es also $\binom{10}{n} \cdot 11^{8-n}$ Codewörter, die an mindestens n Stellen eine 10 haben. Dabei kann n nicht größer als 8 sein, da ansonsten für diese Anzahl keine natürliche Zahl herauskäme.

q.e.d.

Mit diesem Satz kann ich nun ganz leicht die Formel für U gemäß dem Prinzip von Inklusion und Exklusion aufstellen:

$$U=\binom{10}{1}11^7 - \binom{10}{2}11^6 + \binom{10}{3}11^5 \ldots - \binom{10}{8}$$

Wie [R] behauptet, ist dies gleich 131 714 252; folglich ist $|D| = |C| - U = 11^8 - 131714252 =$ =82644629.

3.2 Fehlerkorrektur

Nachdem ich nun die wichtigsten Eigenschaften von D herausgearbeitet habe, beschäftige ich mich nun mit der besonders praktischen Fehlerkorrektur von D.
Dazu nehme ich an, dass bei einem Codewort an der i-ten Stelle ein Fehler der Größe u gemacht wird, sodass das neu entstandene Wort u=x+ue_i ist. Dann gilt für das Syndrom von u:

$$[\ s\ \ t\ ]=S(u)=S(x+u e_i)=[\ u\ \ u(i-1)\ ]$$

Also weiß ich, wenn ich das Syndrom von einem Codewort gegeben habe, bei dem ein einfacher Fehler gemacht wurde, dass die erste Koordinate der Größe des gemachten Fehlers

entspricht. Um dann noch die Stelle i, an der der Fehler gemacht wurde, zu ermitteln, löse ich einfach, wenn t die zweite Koordinate des Syndroms ist, die aus t=u(i-1) mod 11 resultierende Gleichung $i=t\cdot u^{-1}+1=t\cdot s^{-1}+1 \mod 11$.
Im Gegensatz zur normalen Syndrom-Dekodierung ist diese Variante sehr schnell und ohne großes Fehlerrisiko gelöst. Allerdings kann man auf diese Art noch etwas mehr, als einfache Fehler zu korrigieren:
Das der Code D Vertauschungsfehler genau wie der ISBN-Code erkennt, ist klar, da ein Vertauschungsfehler auch nur ein besonderer zweifacher Fehler ist und d(D)=3; im Gegensatz zum ISBN-Code, der, wenn ein Vertauschungsfehler gemacht wurde, nur registriert hat, dass irgendein Fehler gemacht wurde, ist der Code D aber auch in der Lage, wenn ein Vertauschungsfehler gemacht wird, explizit zu erkennen, dass ein Vertauschungsfehler gemacht wurde, auch wenn er ihn nicht korrigieren kann. Dies liegt daran, dass im Falle eines Vertauschungsfehlers die erste Koordinate des Syndroms 0 mod 11 sein muss, da die Koeffizienten der ersten Prüfgleichung alle 1 sind.
Also kann man, wenn von einem Codewort x das Syndrom S(x)=[s t] gegeben ist, nach folgendem Dekodierungsmuster vorgehen:
1.Fall) s=0 und t=0
In diesem Fall nimmt man an, dass kein Fehler gemacht wurde.
2.Fall) s=0 und t≠0
In diesem Fall nimmt man an, dass ein Vertauschungsfehler gemacht worden ist.
3.Fall) s≠0 und t≠0
In diesem Fall nimmt man an, dass an der Stelle $t\cdot s^{-1}+1 \mod 11$ ein Fehler der Größe s gemacht worden ist.

Hierzu ein paar Beispiele:
Beispiel 3.2 Man dekodiere das Codewort 5634674319!
Da 5+6+3+4+6+7+4+3+1+9=4 und $6+2\cdot 3+3\cdot 4+4\cdot 6+5\cdot 7+6\cdot 4+7\cdot 3+8+9\cdot 9=$ =2 mod 11 ist, ist S(5634674319)=42. Also wurde an der Stelle $2\cdot 4^{-1}+1=2\cdot 3+1=7$ ein Fehler der Größe 4 gemacht. Daher ist das Fehlerwort gleich $4e_7$ und man dekodiert mit 5634674319-4e_7=5634670319.
An diesem Beispiel sieht man gut, dass man beim Dekodieren gelegentlich Stammbrüche in Zahlen aus $Z/11Z$ umwandeln muss. Deshalb sollte man sich merken, dass $\frac{1}{2}=\frac{12}{2}=6$, $\frac{1}{3}=\frac{12}{3}=4$, $\frac{1}{4}=\frac{12}{4}=3$, $\frac{1}{5}=\frac{45}{5}=9$, $\frac{1}{6}=\frac{12}{6}=2$, $\frac{1}{7}=\frac{45}{7}=5$, $\frac{1}{8}=\frac{56}{8}=7$, $\frac{1}{9}=\frac{45}{9}=5$ und $\frac{1}{10}=\frac{100}{10}=10$ in $Z/11Z$ gilt.
Beispiel 3.3 Man dekodiere das Codewort 1478264001!
Da 1+4+7+8+2+6+4+1=0 ist, hat das Syndrom von 1478264001 die Form 0a; folglich geht man davon aus, dass ein Vertauschungsfehler gemacht wurde.
Beispiel 3.4 man dekodiere das Codewort 8934726113!
Da 8+9+3+4+2+6+1+1+3=0 und 9+6+12+28+10+36+7+8+27=0, ist S(8934726113)=0.
Also nimmt man an, dass kein Fehler gemacht wurde!

4 Ein 2-fehlerkorrigierender Dezimalcode
4.1 Die Vandermonde Matrix

Bevor ich den nächsten Dezimalcode konstruiere, beschäftige ich mich ein bisschen mit einer Matrix, die für diesen und für viele andere gute Codes wichtig ist, der Vandermonde Matrix.

Definition 4.4 Es sei $m \in N$ und $a_1, a_2, ..., a_{m-1}, a_m$ Elemente eines Körpers.
Dann ist die durch $a_1, a_2, ..., a_{m-1}, a_m$ bestimmte Vandermonde Matrix V gleich:

$$V(a_1, a_2, ..., a_{m-1}, a_m) = \begin{bmatrix} 1 & 1 & ... & 1 \\ a_1 & a_2 & ... & a_m \\ a_1^2 & a_2^2 & ... & a_m^2 \\ \vdots & \vdots & \vdots & \vdots \\ a_1^{m-1} & a_2^{m-1} & ... & a_m^{m-1} \end{bmatrix}.$$

Diese Matrix ist besonders praktisch für meine Zwecke, da die Zeilen und Spalten linear unabhängig sind, wenn alle a_i unterschiedlich sind.
Um dies zu beweisen, betrachte ich zunächst eine Formel zur Berechnung der Determinante einer Vandermonde Matrix:

Satz 4.5 *Die Vandermonde Matrix $V(a_1, a_2, ..., a_{m-1}, a_m)$ hat die Determinante:*
$D(a_1, a_2, ..., a_{m-1}, a_m) = \prod_{i<j} (a_j - a_i)$.

Bew.: durch vollständige Induktion.

Induktionsanfang: Für die 2×2 Matrix $\begin{pmatrix} 1 & 1 \\ a_0 & a_1 \end{pmatrix}$ ist $\begin{vmatrix} 1 & 1 \\ a_0 & a_1 \end{vmatrix} = a_1 - a_0 = \prod_{i<j} (a_i - a_j)$.

Induktionsschritt: Für die $n \times n$ Vandermonde Matrix $\begin{pmatrix} 1 & 1 & ... & 1 \\ a_1 & a_2 & ... & a_n \\ \vdots & \vdots & ... & \vdots \\ a_1^{n-1} & a_2^{n-1} & ... & a_n^{n-1} \end{pmatrix}$

sei $D(a_1 ... a_{n-1}) = \prod_{i<j<n} (a_j - a_i)$.

Induktionsschluss: Für ein a_0 ergänze man nun diese $n \times n$ Vandermonde Matrix zu der entsprechenden (n+1)×(n+1) Vandermonde Matrix, indem man a_0 vor a_1 schiebt. Für dessen Determinante gilt dann:

$$D(a_0 ... a_n) = \begin{vmatrix} 1 & 1 & ... & 1 \\ a_0 & a_1 & ... & a_n \\ \vdots & \vdots & ... & \vdots \\ a_0^n & a_1^n & ... & a_n^n \end{vmatrix} = \begin{vmatrix} 1 & 1 & ... & 1 \\ 0 & a_1 - a_0 & ... & a_n - a_0 \\ \vdots & \vdots & ... & \vdots \\ 0 & a_1^n - a_1^{n-1} a_0 & ... & a_n^n - a_n^{n-1} a_0 \end{vmatrix} =$$

(Nun entwickelt man nach der ersten Zeile und wendet dann die Multilinearität an!)

$$= \begin{vmatrix} a_1 - a_0 & a_2 - a_0 & ... & a_n - a_0 \\ a_1(a_1 - a_0) & a_2(a_2 - a_0) & ... & a_n(a_n - a_0) \\ \vdots & \vdots & ... & \vdots \\ a_1^{n-1}(a_1 - a_0) & a_2^{n-1}(a_2 - a_0) & ... & a_n^{n-1}(a_n - a_0) \end{vmatrix} = \begin{vmatrix} 1 & 1 & ... & 1 \\ a_1 & a_2 & ... & a_n \\ \vdots & \vdots & ... & \vdots \\ a_1^{n-1} & a_2^{n-1} & ... & a_n^{n-2} \end{vmatrix} \cdot \prod_{i=1}^{n} (a_i - a_0) =$$

$$= \prod_{i<j<n} (a_j - a_i).$$

Um die Induktion abzuschließen, substituiert man noch $a_0 := a_1 ... a_n := a_{n+1}$.

q.e.d.

Nun brauche ich noch eine allgemeine Eigenschaft von Determinanten.

Satz 4.5 *Ist von einer* $n \times n$ *Matrix A die Determinante D(A) ungleich Null, so sind die Zeilen und Spalten von A linear unabhängig.*

Auf einen Beweis verzichte ich, da dieser in jedem guten Buch über lineare Algebra enthalten ist.
Wendet man diesen Satz nun auf einer Vandermonde Matrix an, so stellt man fest, dass die Zeilen und Spalten linear unabhängig sind, wenn die a_i alle verschieden sind, da $\prod_{i<j<n} (a_j - a_i)$ dann, und nur dann Null wird, wenn zwei a_i gleich sind. Dies wird gleich noch ziemlich nützlich sein.

4.2 Konstruktion des Dezimalcodes und Fehlerkorrektur

Man betrachte nun die Matrix

$$P = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & 10 \\ 1 & 2^2 & 3^2 & \dots & 10^2 \\ 1 & 2^3 & 3^3 & \dots & 10^3 \end{bmatrix}$$

Da man aus jeweils vier Spalten dieser Matrix eine Vandermonde Matrix mit unterschiedlichen a_i bilden kann, sind infolgedessen auch alle Kombinationen von vier Spalten linear unabhängig. Außerdem sind gemäß [R] die ersten fünf Spalten linear unabhängig. Damit ist die minimale Anzahl linear abhängiger Spalten also fünf, da weniger als fünf Spalten nicht linear abhängig sein können, da ja jeweils vier Spalten eine Vandermonde Matrix bilden und somit linear unabhängig sind. Des Weiteren sind auch die Zeilen von P linear unabhängig, denn wären sie es nicht, müsste es eine entsprechende nichttriviale Linearkombination geben; würden man aber nun alle Zeilen um sechs Dimensionen verkleinern, erhielte man wieder eine Vandermonde Matrix mit linear unabhängigen Zeilen. Die gefundene nichttriviale Linearkombination müsste aber auch für diese Matrix gelten, was ein Widerspruch wäre. Also sind die Zeilen von P linear unabhängig.
Aus diesen Eigenschaften folgt, dass P die Kontrollmatrix eines linearen [10,6,5] Codes über Z_{11} ist, da bei einem linearen Code die minimale Anzahl an linear abhängigen Spalten dem Minimalabstand entspricht.
Es sei nun E derjenige Code, der entsteht, wenn man aus diesem Code alle Codewörter entfernt, die an irgendeiner Stelle eine 10 haben. E ist natürlich nichtlinear, da offensichtlich die Abgeschlossenheit der Addition verloren geht. Aber E hat ebenfalls die Länge 10 und den Minimalabstand 5 und ist deshalb 2-fehlerkorrigierend. Um die Größe von E zu bestimmen, verwendet man erneut das Einschluss-Ausschluss-Prinzip, erhält eine ähnliche Formel wie beim Code C und erhält, wie [R] behauptet, 683 024 als Größe von E.
Auch E bietet eine interessante Alternative zur Syndrom-Dekodierung. Um diese zu verstehen, geht man davon aus, dass ein Codewort c aus E gesendet wird, aber bei der Übertragung ein Fehler e der Form $e=ue_i + ve_j$ gemacht wird, wobei $u,v \in Z_{11}$ und o.b.d.A. $i<j$ gilt. Für das Syndrom von $x=c+ue_i + ve_j$ gilt folglich:

$S(x)=xP^t = (ue_i + ve_j)P^t = \begin{bmatrix} u+v & ui+vj & ui^2+vj^2 & ui^3+vj^3 \end{bmatrix}$

Hat das Syndrom eines empfangenen Wortes also die Form $S(x) = [s_1 \quad s_2 \quad s_3 \quad s_4]$, so lässt sich folgendes Gleichungssystem aufstellen:

$$u + v = s_1$$
$$ui + vj = s_2$$
$$ui^2 + vj^2 = s_3$$
$$ui^3 + vj^3 = s_4$$

Als Nächstes multipliziert man die ersten drei Gleichungen mit i und subtrahiert dann die nächste Gleichung, wodurch man folgendes erhält:

$$v(i - j) = is_1 - s_2$$
$$vj(i - j) = is_2 - s_3$$
$$vj^2(i - j) = is_3 - s_4$$

Durch Quadrieren der mittleren Gleichung erhält man $v^2 j^2(i - j)^2 = (is_2 - s_3)^2$, durch Multiplikation der ersten und dritten Gleichung $v^2 j^2(i - j)^2 = (is_1 - s_2)(is_3 - s_4)$. Setzt man nun beide Gleichungen gleich, ergibt sich:

$$(is_2 - s_3)^2 = (is_1 - s_2)(is_3 - s_4) \Leftrightarrow i^2 s_2^2 - 2is_2 s_3 + s_3^2 = i^2 s_1 s_3 - is_1 s_4 - is_2 s_3 + s_2 s_4 \Leftrightarrow$$
$$\Leftrightarrow (s_2^2 - s_1 s_3)i^2 + (s_1 s_4 - s_2 s_3)i + s_3^2 - s_2 s_4 = 0$$

Hätte man zu Beginn mit j anstatt mit i multipliziert, wäre man zu derselben Gleichung gekommen, nur mit einem j an jeder Stelle, wo jetzt ein i steht.
Um das Ganze etwas übersichtlicher zu machen, setzt man A=$(s_2^2 - s_1 s_3)$, B=$(s_1 s_4 - s_2 s_3)$ und C=$s_3^2 - s_2 s_4$.
Folglich sind die Stellen der Fehler gleich den Lösungen der quadratischen Gleichung

$$Ay^2 + By + C = 0$$

Es bleibt noch die Frage, was passiert, wenn nur ein Fehler beim Übertragen gemacht wird. Ist dies der Fall, so kann man die Größe j und Stelle v des zweiten Fehlers gleich null setzen. Dann ergibt sich für die Form des Syndroms von x: $S(x) = [u \quad ui \quad ui^2 \quad ui^3]$.
Setzt man dies in die Formeln für A,B und C ein, findet man heraus, dass in diesem Fall A= $(ui)^2 - u \cdot ui^2 = u^2 i^2 - u^2 i^2 = 0$, B=$u \cdot ui^3 - ui \cdot ui^2 = u^2 i^3 - u^2 i^3 = 0$ und C=$(ui^2)^2 - ui \cdot ui^3 =$ $= u^2 i^4 - u^2 i^4 = 0$. Also hat $Ay^2 + By + C = 0$ die Form 0=0, weshalb sich so die Stelle, an der der Fehler gemacht wurde, nicht errechnen lässt. Stattdessen benutzt man die Gleichungen $u + v = s_1$ und $ui + vj = s_2$, die ja in diesem Fall $u = s_1$ und $ui = s_2$ lauten. Somit befindet sich der Fehler an der Stelle $s_2 s_1^{-1}$ und hat die Größe s_1.
Interessant ist noch, dass zwei Fehler in keinem Fall wie ein Fehler korrigiert werden; denn in diesem Fall muss A ungleich null sein muss. Denn A=0 ist äquivalent zu
$s_2^2 - s_1 s_3 = 0 \Leftrightarrow (ui + vj)^2 - (u + v)(ui^2 + vj^2) = 0 \Leftrightarrow$
$\Leftrightarrow u^2 i^2 + 2uivj + v^2 j^2 - u^2 i^2 - vj^2 u - vui^2 - v^2 i^2 = 0 \Leftrightarrow vj^2 u - 2uivj + vui^2 = 0 \Leftrightarrow j^2 - 2ij + i^2 = 0 \Leftrightarrow$
$\Leftrightarrow (j - i)^2 = 0 \Leftrightarrow i = j$, was ein Widerspruch zur Annahme wäre.
Auf Grund dieser Erkenntnisse ist man nun in der Lage folgendes Dekodierungsmuster aufzustellen:
Gilt für ein empfangenes Wort x, dass $S(x) = [s_1 \quad s_2 \quad s_3 \quad s_4]$, so unterscheidet man folgende Fälle:

1) S(x)=0; in diesem Fall geht man davon aus, dass kein Fehler gemacht wurde.
2) S(x)≠0, aber A=0, B=0 und C=0; in diesem Fall geht man davon, dass ein Fehler gemacht wurde; dieser wurde an der Stelle $s_1^{-1} s_2$ gemacht und hat die Größe s_1.

3) S(x)≠0 und A ungleich null; in diesem Fall nimmt man an, dass ein Fehler der Form $ue_i + ve_j$ gemacht wurde. Die Stellen i,j der beiden Fehler entsprechen den Lösungen der quadratischen Gleichung Ay²+By+C=0. Hat man i und j ermittelt, sind die Größen u und v der beiden Fehler durch u+v=s_1 und ui+vj=s_2 bestimmt.
4) Besitzt die quadratische Gleichung aus Fall 3) keine Lösung über Z_{11}, so wurden bei der Übertragung mehr als zwei Fehler gemacht.

Für den 4.Fall ist es wichtig, sich zu merken, dass in Z_{11} folgendes gilt $\sqrt{1} = 1, \sqrt{3} = 5, \sqrt{4} = 2$, $\sqrt{5} = 4$ und $\sqrt{9} = 3$. Alle anderen Zahlen aus Z_{11} haben keine Quadratwurzeln.
Des Weiteren ist noch erwähnenswert, dass zwar dreifache und vierfache Fehler erkannt werden, aber bei der Korrektur nicht zwingend als solche berücksichtigt werden.
Einige von den dabei entstandenen Wörtern haben zu einem gültigen Codewort den Abstand zwei oder sogar noch geringer und werden deshalb entsprechend dekodiert.

4.3 Beispiele

Um das Verfahren zu verdeutlichen, folgen nun einige Rechenbeispiele:
Beispiel 4.5 Man dekodiere das Wort x=3235556411!
Da S(3235556411)=[2 8 10 7], ist A=64-20=44=0 mod 11, B=14-80=-66=0 mod 11 und C=100-56=44=0 mod 11; also geht man davon aus, dass ein Fehler gemacht wurde.
Dieser Fehler wurde an der Stelle $2^{-1} \cdot 8 = 6 \cdot 8 = 48 = 4$ gemacht und hat die Größe 2.
Folglich ist das Fehlerwort gleich $2e_4$ und man dekodiert mit c=x-$2e_4$ =3233556411.
Beispiel 4.6 Man dekodiere das Wort x=4739688119!
Da S(473968819)=[1 7 10 10] ist A=49-10=39=6, B=10-70=-60=6 und C=100-70=30=8.
Also ergibt sich folgende Gleichung:

$$6y^2+6y+8=0$$

Mit dieser lassen sich nun die beiden Stellen der Fehler berechnen.

$$i, j = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} = \frac{-6 \pm \sqrt{36-16}}{12} = 5 \pm \sqrt{9} = 5 \pm 3 = 2,8$$

Die Größen u,v der Fehler lassen sich durch das Lösen der Gleichungen u+v=1 und 2u+8v=7 ermitteln. Dabei ergibt sich u=2 und v=10. Also ist das Fehlerwort gleich $2e_2 + 10e_8$ und man dekodiert mit c=x-$2e_2 + 10e_8$ =4539688219.
Beispiel 4.7 Man dekodiere das Wort x=1111037407!
Da S(1111037407)=[5 10 4 4], ist A=100-20=80=3, B=20-40=-20=2 und C=16-40=-24= =9. Also ergibt sich folgende Gleichung:

$$3y^2+2y+9=0$$

Damit gilt für die Stellen der beiden Fehler:

$$i, j = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} = \frac{-2 \pm \sqrt{4-9}}{6} = \frac{9 \pm \sqrt{6}}{6}$$

Allerdings hat 6 in Z_{11} keine Quadratwurzel. Folglich wurden mehr als zwei Fehler gemacht und x ist nicht dekodierbar.

5 Ausblick auf BCH-Codes

Zum Abschluss meiner Seminararbeit will ich das Prinzip des 2-fehlerkorrigierenden Codes D ein wenig verallgemeinern, und zwar auf die so genannten BCH-Codes, benannt nach ihren Entdeckern R.C. Bose, D.K. Ray-Chaudhuri und A. Hoquenghem. Diese haben eine ähnliche Prüfmatrix wie D und lassen sich folgendermaßen konstruieren:

Definition 5.5 Es sei p prim, $d, n \in N$ und d≤n≤p-1. Der lineare, p-äre BCH-Code der Länge 10 und mit dem Minimalabstand d ist durch die Prüfmatrix P

$$P = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 2^2 & 3^3 & \dots & n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2^{d-2} & 3^{d-2} & \dots & n^{d-2} \end{bmatrix}$$

charakterisiert.

Besonders interessant sind diese Codes, da sie die Singeltonschranke erfüllen, es gilt also $A_p(n,d) = p^{n-d+1}$. Damit sind sie perfekte [n,n-d+1,d]-Codes über Z_p.
Im Rahmen dieser Seminararbeit findet sich allerdings für eine detailliertere Erörterung dieser Codes kein Platz mehr, weshalb ich an dieser Stelle aufhöre.

Literaturverzeichnis

[R] S. ROMAN *Introduction to coding and information theory,* Springer-Verlag, New York (1997)

$[W_1]$ *http://en.wikipedia.org/wiki/International_Standard_Book_Number* entnommen am 7.5.2008

$[W_2]$ *http://en.wikipedia.org/wiki/Linear_code* entnommen am 14.5.2008

$[W_3]$ *http://en.wikipedia.org/wiki/Square_root* entnommen am 20.5.2008

Erklärung

Ich erkläre hiermit, dass ich die Seminararbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe.

Regensburg, 28.5.2008