Grundkurs Codierung

Lösungsvorschläge zu den Fragen in den Unterkapiteln „Was blieb?“
Stand 06.01.2007

Unterkapitel 3.7.11 Seite 160

Zu Frage 1:

Das Multiplikationsschema zeigt, dass sich, abhängig von den 0/1-Stellen des Generatorpolynoms, in den ersten k Stellen des Info-Polynoms weitere Beiträge zum Info-Polynom hinzu addieren können. Damit werden die ersten k Stellen des Codewortes verändert, was die direkte Erkennung des Info-Teils verhindert.

Zu Frage 2:

Nein. Es sind nur diejenigen irreduziblen Polynome geeignet, die als Nullstellen primitive Elemente enthalten. Also solche, die die maximale Ordnung ergeben.

Zu Frage 3:

- Entweder müssen für jede der n möglichen 1-Bit-Fehlerpositionen bei der MOD g(x) - Division des Empfangswortpolynoms w(x) eindeutige Syndrompolynome s(x) erzeugt werden, d. h. alle

  $s(x) = w(x) \text{ MOD } g(x) = [v(x) + e(x)] \text{ MOD } g(x) = v(x) \text{ MOD } g(x) + e(x) \text{ MOD } g(x) = 0 + e(x) \text{ MOD } g(x)$

  sind für die n möglichen 1-Bit-Fehlerpositionen in e(x) verschieden,

- oder die Syndromwerte

  $s(x=\alpha) = w(x=\alpha) = v(x=\alpha) + e(x=\alpha) = 0 + e(\alpha) = e(\alpha)$

  müssen an der Stelle des primitiven Elements α, $g(\alpha) = 0$, für jede der n möglichen 1-Bit-Fehlerpositionen verschieden sein.

Es ist dabei **hinreichend**, wenn es für jede Fehlerposition einen eindeutigen Satz von Syndromwerten gibt. Einem 1-Bitfehler könnten also auch mehrere Syndromwerte zugeordnet sein, nicht jedoch ein Syndromwert mehreren Fehlerpositionen!

Zu Frage 4:

Das irreduzible Polynom $f(x) = x^8 + x^7 + x^2 + x + 1$ definiert für das primitive Element ψ über

$f(\psi) = \psi^8 + \psi^7 + \psi^2 + \psi + 1 = 0$

die Struktur

$\psi^8 = -\psi^7 - \psi^2 - \psi - 1 = \psi^7 + \psi^2 + \psi + 1$ (Koeffizienten im Z_2 !).

Daraus baut sich ein Galoisfeld $GF(2^8)$ mit 256 Elementen (= Potenzen von ψ) auf. Die ersten 10 sind:

0	=	0	= 0000 0000
ψ^0	=	1	= 0000 0001
ψ^1	=	ψ	= 0000 0010
ψ^2	=	ψ^1 ψ	= 0000 0100
ψ^3	=	ψ^2 ψ	= 0000 1000
ψ^4	=	ψ^3 ψ	= 0001 0000

ψ^5 = $\psi^4\,\psi$ = 0010 0000
ψ^6 = $\psi^5\,\psi$ = 0100 0000
ψ^7 = $\psi^6\,\psi$ = 1000 0000

ψ^8 = $\psi^7 + \psi^2 + \psi + 1$ = 1000 0111 (Definition !)

ψ^9 = $\psi^8\,\psi$
= $\psi^8 + \psi^3 + \psi^2 + \psi$
= $(\psi^7 + \psi^2 + \psi + 1) + \psi^3 + \psi^2 + \psi$
= $\psi^7 + \psi^3 + (\psi^2 + \psi^2) + (\psi + \psi) + 1$
= $\psi^7 + \psi^3 + 1$
= 1000 1001

usw. Hierbei wird wieder

ψ^{255} = $\psi^0 = 1$ = 0000 0001.

Das primitive Element hat also die Ordnung **ord = 255**.

Übrigens: Da sich jedes Element dieses Galoisfeldes in Kurzform als Polynom 7-ten Grades mit 8 "0/1"-Koeffizienten schreiben lässt, entspricht es jeweils einem der 256 möglichen Bytes. Ein Feld GF(2^8) ist daher besonders für Codierungszwecke geeignet – ohne allerdings zwingend notwendig zu sein.

Zu Frage 5:

Eine "Brute-Force"-Methode zur Überprüfung eines Codes auf Korrekturfähigkeit für t Fehler wäre es, die Abstände sämtlicher Codewörter untereinander zu berechnen. Ist $d_{min} \geq 2 \cdot t + 1$, dann stellt dies eine hinreichende Bedingung dar – allerdings ohne damit bereits ein praktikables Verfahren für die Dekodierung zu haben. Ausserdem wäre dieser Weg nur für "kurze" Codes mit wenigen Info-Stellen gangbar. Ein Code mit k Info-Stellen hat 2^k Wörter, es wären also $2^k \cdot (2^k - 1)/2$ Abstandsberechnungen durchzuführen, der Aufwand steigt exponentiell mit k.

Bedeutend einfacher ist es, gemäß Unterkapitel 3.7.5 die k Basis-Codewörter zu berechnen, also diejenigen, die jeweils nur ein "1"-Bit im Info-Teil aufweisen, und deren Abstände

- zum Nullwort
- und aller untereinander

zu ermitteln. Die Abstände zum Nullwort entsprechen unmittelbar den Gewichten der k Basisworte, also der Zahl der "1"-Bits, für die Abstände untereinander sind dann $k \cdot (k-1)/2$ Berechnungen erforderlich. Ist die Anforderung für den Mindestabstand erfüllt, so gilt dies auch für alle übrigen Codewörter, da sich diese als Linearkombinationen aus den Basis-Wörtern ergeben und deren Eigenschaften übernehmen.

Ein Beispiel: Mit dem Generatorpolynom g(x) = 101 0011 0111 auf Seite 130 wird ein zyklischer Code der Länge n = 15 zur Korrektur von t = 3 Einbit-Fehlern aufgebaut. Die Bildungsvorschrift für systematische zyklische Codes auf Seite 103 erzeugt Codewörter mit 5 Info- und 10 Prüfstellen. Ist der Infoteil 00001, so besteht das (Basis-) Codewort v_{b1} aus dem Generatorpolynom, welches links mit 4 "0"-Bits aufgefüllt ist (bitte überprüfen) :

v_{b1} = 00001 01 0011 0111 (Gewicht 7).

Der Infoteil 00010 ergibt

v_{b2} = 00010 10 0110 1110 (Gewicht 7).

Zum Infoteil 00011 gehört das Codewort $v = v_{b1} + v_{b2}$

v = 00011 11 0101 1001 (Gewicht 8).

Es hat sowohl zum Nullwort als auch zu v_{b1} und v_{b2} wegen $v + v_{b1} = v_{b2}$ bzw. $v + v_{b2} = v_{b1}$ einen ausreichenden Abstand., usw.

Zu Frage 6:

Nein, die beliebige Verteilung der Fehler ist zwar auch beim RS-Code uneingeschränkt möglich, "Fehler" bezieht sich hier aber auf ein Fehlerelement des zugrunde liegenden Galoisfeldes $GF(2^m)$. Jedem Element wird ein physikalisches Binärmuster der Länge m (= Fehlerbündel) zugeordnet. Kann der gewählte RS-Code t Fehler korrigieren, so lassen sich damit maximal m Einbit-Fehler beheben, wenn diese entsprechend "weit" verstreut von einander im Empfangswort verteilt liegen. Genauer ausgedrückt heisst das, dass sie dann auf m verschiedene Elemente verteilt wären.

Andererseits würde ein RS-Code über $GF(2^m)$ für die Korrektur eines einzigen Fehlerbündels (t = 1) mit m kompakt beieinander liegenden "1"-Elementen bereits zur vollständigen Fehlerbeseitigung ausreichen – sofern das physikalische Bündel genau in das Raster der Breite m passt, eine unrealistische Forderung!

Beispiel: Der RS-Code der Länge n = 15 über das Galoisfeld $GF(2^4)$ zur Korrektur von t = 3 Fehlerbündeln der Breite m = 4 ($m \cdot n = 60$ Binärstellen) "verkraftet" minimal folgende 3 Einbitfehler:

e = 0000 0010 0000 0000 0001 1000 0000 ... 0000 0000,

maximal aber auch

e = 0000 1111 0000 0000 1111 1111 0000 ... 0000 0000.

Man erkennt hieran, dass sich der RS-Code ganz besonders für die Korrektur von Fehlerbündeln eignet, während Einbitfehler in einem m-Bit breiten Codewortraster die Leistungsfähigkeit des RS-Codes gar nicht richtig ausreizen.

Zum Vergleich:

- Ein BCH-Code der Länge n = 63 hat zur Korrektur von 3 Einbitfehlern ein Generatorpolynom des Grades 18, jedes Codewort besteht aus einem 45-Bit Infoteil und 18 Prüfbits, die Rate beträgt 71%. Er kann alle der 41727 möglichen 1-, 2- und 3-Bitfehler-Kombinationen korrigieren.

- Der oben genannte RS-Code der "Binärlänge" $n \cdot m = 60$ hat 36 Info- und 24 Prüfstellen, die Rate beträgt 60%. Er kann alle der im ungünstigsten Fall auftretenden 30860 1-, 2- und 3-Bitfehler-Kombinationen korrigieren. Ein hypothetischer BCH-Code der Länge 60 könnte alle 36050 dieser Kombinationen korrigieren. Allerdings korrigiert der RS-Code alle 239800 Kombinationen mit 1-, 2-, 3-, ... 10-, 11- und 12-Bitfehlern im vorgegebenen 4-Bit breiten Raster. Alle möglichen Kombinationen wären 1835237017323 (1,8 Billionen), also 7.6 Millionen mal mehr.

Zu Frage 7:

- Nach Unterkapitel 3.7.10 muss das Generatorpolynom g(x) zunächst den Grad 6 haben, damit ein beliebig im Empfangswort w(x) liegendes Fehlerbündel der Breite m = 6 einen von Null verschiedenen Rest bei der Division $w(x)$ MOD $g(x) \neq 0$ erzeugt.

- Damit alle ungeraden Fehlermuster ebenfalls einen von Null verschiedenen Divisionsrest ergeben, muss das Generatorpolynom einen Anteil $x + 1$ enthalten.

- Damit alle 2-Bit-Fehler einen von Null verschiedenen Divisionsrest liefern, muss das g(x) ein über Z_2 irreduzibles Polynom möglichst hohen Grades enthalten (damit der Code eine möglichst große Länge erhält).

Damit wäre ein geeignetes Generatorpolynom nach Tabelle 3-24 auf Seite 131 z. B. g(x) = (100101)(11), Grad g(x) = 6. Das Teilpolynom 100101 (Grad = 5) teilt nach dem Fermat'schen Theorem (S. 46) das aus zwei "1"-Elementen bestehende Polynom $x^q + 1$ erst für $q = 2^5$, also $x^{32} +1$. Die Länge des Codewortes kann also n= q – 1 = 31 betragen.

Zu Frage 8:

Wenn der Code die Länge n hat, so können in einem Empfangswort

$$n_3 = \binom{n}{3} + \binom{n}{2} + \binom{n}{1}$$

Kombinationen von 1-, 2- und 3-Bitfehlern auftreten. Das Generatorpolynom g(x) muss also so viele Prüfstellen m erzeugen, dass die Anzahl der 2^m Kombinationen erfüllter und nicht erfüllter Prüfgleichungen mindestens gleich groß ist:

$$2^m \geq n_3 = \binom{n}{3} + \binom{n}{2} + \binom{n}{1} \quad .$$

Da für die Codewortlänge $n = 2^m - 1$ gilt, ist die nichtlineare (Un-) Gleichung

$$2^m \geq n_3 = \binom{2^m-1}{3} + \binom{2^m-1}{2} + \binom{2^m-1}{1}$$

in der Unbekannten m zu lösen. Das Generatorpolynom muss dann den Grad m + 1 besitzen. **Aber bitte beachten:** Dies ist nur eine notwendige Bedingung, sie reicht für die Eignung des Codes zur 3-Bitfehler-Korrektur nicht aus. Dafür müssen den 2^m Kombinationen erfüllter und nicht erfüllter Prüfgleichungen **eindeutig** n_3 unterschiedliche Syndrome zugeordnet werden können.

Zu Frage 9:

Vergleicht man für unterschiedliche Codes bei wachsenden Codewortlängen n die Paare der zugehörigen Inforaten R = k/n und Korrekturraten R_{kr} = t/n (also korrigierbare Fehler auf die Länge n bezogen), so sieht man, dass bei fester Korrekturrate die Inforate der BCH-Codes stetig sinkt, während sie bei RS-Codes konstant bleibt, siehe Unterkapitel 3.7.9, Tabellen 3-24 und 3-25, Seiten 154 und 155.

Da sinkende Inforaten einen immer schlechteren Störabstand bewirken, sinkt im Gegensatz zum RS-Code die Restfehlerrate EBR des BCH-Codes mehr, als durch die positive Wirkung vergrößerter Codewortlänge gewonnen wird, siehe auch Unterkapitel 3.5.4. Ausgehend von kleinen Codewortlängen n verbessert sich die Restfehlerrate des BCH-Codes mit wachsender Länge zwar zunächst, fällt dann aber wieder ab, siehe Tabelle 3-4 auf Seite 77.

Zu Frage 10:

Nach der Formel auf Seite 244, bzw. gemäß Bild 4.8 auf Seite 245 sind es für m = 6 genaue N_6 = 9 verschiedene irreduzible Polynome.