Reed-Solomon kódok

Kasza Péter

2009. október 16.

Tartalomjegyzék

1. Bevezetés

A Reed-Solomon kódolás egy olyan hibajavító kódolás, amely viszonylag nagy bitszám meghibásodása esetén is képes az eredeti adatokat visszaállítani. A kódolás az eredeti adatok mellé paritás kódokat helyez. A Reed-Solomon kódolás az ún. BCH *(Bose-Chaudhuri-Hocquenghem)* kódok egy speciális nembináris esetének tekinthető. A Reed-Solomon kódokat előszeretettel alkalmazzák az adattárolási és továbbítási technológiák körében, ahol viszonylag zajos bemenetekkel kell dolgozni, mint például az optikai adattárolás *(CD-ROM, DVD, stb)*, telekommunikáció *(digitális televízió, stb)*.

A Reed-Solomon kódok nembináris ciklikus kódok, ahol a szimbólumok ω bitszámú sorozatok, és ω értéke legalább 2. A Reed-Solomon kódolás a legnagyobb lehetséges minimális kódtávolságot éri el, bármely lineáris kód esetén, azonos bemeneti és kimeneti blokkméretek mellett.

Reed-Solomon kódok esetén ismeretes a minimális kódtávolságra az alábbi formula [1]

$$d_{min} = k + 1$$

A minimális kódtávolságból adódik, hogy a kódolás bármilyen $\lfloor k/2 \rfloor$-nél kevesebb bitben meghibásodott kódszót korrigálni képes.

2. Véges testek

A Reed-Solomon kódolás megértéséhez szükséges a véges testek elméletét ismerni, hiszen az algoritmus által használt polinomok valamilyen $GF(2^{\omega})$ test felett vannak definiálva.

2.1. Definíció (test)**.** Az $(F, +, \cdot)$ algebrai struktúrát, amelyben az alábbi axiómák teljesülnek testnek nevezzük.

i) Az összeadásra és a szorzásra nézve teljesül a **zártság**

$$\forall a, b \in F \textit{ esetén } a + b \in F \textit{ és}$$
$$a \cdot b \in F$$

ii) Az összeadás és a szorzás műveletére teljesül az **asszociativitás**

$$\forall a, b, c \in F \textit{ esetén } (a + b) + c = a + (b + c) \textit{ és}$$
$$(a \cdot b) \cdot c = a \cdot (b \cdot c)$$

iii) Az összeadás és a szorzás műveletére teljesül a **kommutativitás**

$$\forall a, b \in F \textit{ esetén } a + b = b + a \textit{ és}$$
$$a \cdot b = b \cdot a$$

iv) Az összeadás és a szorzás műveletére teljesül a **disztributivitás**

$$\forall a, b, c \in F \textit{ esetén } a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

v) Létezik az additív és a multiplikatív egységelem

$$\exists \varepsilon^+, \varepsilon^* \in F, \forall a \in F, \quad \varepsilon^+ + a = a$$
$$\varepsilon^* \cdot a = a$$

vi) Minden elemnek létezik addtiív és multiplikatív inverze

$$\forall a \in F, \exists b \in F, \quad a + b = \varepsilon^+$$
$$\forall a \in F, \exists b \in F, \quad a \cdot b = \varepsilon^*$$

2.2. Definíció (véges test)**.** A véges test, vagy más néven Galois mező egy olyan test, amely csak véges számú elemmel rendelkezik.

Jelölés. A p elemmel rendelkező véges testet $GF(p)$-vel jelöljük, ahol $p \in \mathbb{P}$ prímszám.

Megjegyzés. A $GF(p)$ test kibővíthető egy p^ω elemű testté. Nyílván ekkor a $GF(p)$ része a kibővített $GF(p^\omega)$ testnek. A Reed-Solomon kódolás a bináris $GF(2)$ test $GF(2^\omega)$ kibővítésén értelmezett polinomokkal dolgozik. A kibővített testben a $\{0, 1\}$ szimbólumokon kívül léteznek még egyedi elemek melyeket az α szimbólummal jelölünk.

Állítás. A test $GF(2^\omega)$ test bármely nemzérus eleme felírható α valamilyen hatványaként.

2.1. A $GF(2^\omega)$ test elemeinek generálása

A test elemeinek meghatározásához induljunk ki a $\{0, 1, \alpha\}$ elemekből generált végtelen halmazból

$$F = \{0, 1, \alpha, \alpha^2, \dots, \alpha^j, \dots\} = \{0, \alpha^0, \alpha, \alpha^2, \dots, \alpha^j, \dots\}$$

A $GF(2^\omega)$ elemeit úgy kaphatjuk meg az F halmazból, ha az alábbi feltételt szabjuk ki

$$\alpha^{(2^\omega - 1)} + 1 = 0$$

vagy másképp

$$\alpha^{(2^\omega - 1)} = 1 = \alpha^0$$

Az így kapott halmaz már zárt a szorzásra és csak véges sok elemből áll. Az előbbi polinomiális feltétel alkalmazásával bármely testbeli elem, amely $2^\omega - 1$ vagy annál nagyobb hatvánnyal rendelkezik $2^\omega - 1$ hatványnál kisebb hatványú elemmé redulkáható a következő képpen

$$\alpha^{(2^\omega + n)} = \alpha^{(2^\omega - 1)} \alpha^{n+1} = \alpha^{n+1}$$

A $GF(2^\omega)$ test elemei tehát

$$GF(2^\omega) = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^\omega - 2}\}$$

2.1.1. Az elemek algoritmikus generálása

A $GF(2^\omega)$ test elemeit $GF(2)[x]$-beli polinomokkal reprezentálhatjuk. A $GF(2)$ test feletti polinomok együtthatói csak a $\{0, 1\}$ halmaz elemei lehetnek, ezért célszerű ezeket az együtthatókat egy egész szám bitjeinek tekinteni. Ekkor megfigyelhető, hogy az első $\omega - 1$ darab α^i alakban kifejezett elemek a 2^i számoknak felelnek meg.

Az α^ω elem az $f(\alpha) = 0$ egyenlőségből fejezhető ki

$$\alpha^\omega = \sum_{i=0}^{deg(f)-1} f_i 2^i$$

Megjegyzés. Tulajdonképpen α^ω az $f(x)$ irreducibilis polinomhoz rendelt egész számból, annak legnagyobb helyiértékű bitjének kinullázásával kapható.

A többi elem az alábbi módon adható meg

$$\alpha^i = \begin{cases} \alpha^\omega \oplus 2(\alpha^{i-1} \oplus 2^{deg(f)-1}), & \textit{ha } \alpha^i > 2^\omega \\ \alpha^{i-1} & \textit{egyébként.} \end{cases} \quad (\forall i > \omega)\textit{-ra}$$

1. $\alpha^0 \leftarrow 1$
2. $\alpha^\omega \leftarrow \sum_{i=0}^{deg(f)-1} f_i 2^i$ *[f_i az f polinom i-edik együtthatója]*
3. **FOR** $i \leftarrow 1$ **TO** $\omega - 1$ **DO**
4. $\quad \alpha^i \leftarrow 2\alpha^{i-1}$
5. **ENDFOR**
6. **FOR** $i \leftarrow \omega + 1$ **TO** 2^ω **DO**
7. $\quad$ **IF** $\alpha^{i-1} > 2^{deg(f)-1}$ **THEN**
8. $\quad\quad \alpha^i \leftarrow \alpha^\omega \oplus 2(\alpha^{i-1} \oplus 2^{deg(f)-1})$
9. $\quad$ **ELSE**
10. $\quad\quad \alpha^i \leftarrow 2\alpha^{i-1}$
11. $\quad$ **ENDIF**
12. **ENDFOR**

1. ábra. Algoritmus a $GF(2^\omega)$ test elemeinek generálására

3. Műveletek a $GF(2^\omega)$ testen

	X^0	X^1	X^2	
0	0	0	0	—
α^0	1	0	0	$\alpha^0 = 1$
α^1	0	1	0	$\alpha^1 = \alpha$
α^2	0	0	1	$\alpha^2 = \alpha \cdot \alpha$
α^3	1	1	0	$\alpha^3 = 1 + \alpha,\ f(\alpha) = 0$ *egyenlőségből kifejezve*
α^4	0	1	1	$\alpha^4 = \alpha \cdot \alpha^3 = \alpha \cdot (1 + \alpha) = \alpha + \alpha^2$
α^5	1	1	1	$\alpha^5 = \alpha \cdot \alpha^4 = \alpha \cdot (\alpha + \alpha^2) = \alpha^2 + \alpha^3 = 1 + \alpha + \alpha^2$
α^6	1	0	1	$\alpha^6 = \alpha \cdot \alpha^5 = \alpha \cdot (1 + \alpha + \alpha^2) = \alpha + \alpha^2 + \alpha^3 = 1 + \alpha^2$
α^7	1	0	0	$\alpha^7 = \alpha \cdot \alpha^6 = \alpha \cdot (1 + \alpha^2) = \alpha + \alpha^3 = \alpha + (1 + \alpha) = 1$

2. ábra. $1 + X + X^3$ polinommal definiált $GF(2^3)$ mező elemeinek együtthatói

3.1. Összegzés

Az elemek összege, az együtthatók 2-es maradékosztályon vett összegének felel meg. Az összegzés gépi implementációjakor ez megfelel az egyes elemek között vett XOR, vagy más néven kizáró vagy műveletnek.

$+$	α^0	α^1	α^2	α^3	α^4	α^5	α^6
α^0	0	α^3	α^6	α^1	α^5	α^4	α^2
α^1	α^3	0	α^4	α^0	α^2	α^6	α^5
α^2	α^6	α^4	0	α^5	α^1	α^3	α^0
α^3	α^1	α^0	α^5	0	α^6	α^2	α^4
α^4	α^5	α^2	α^1	α^6	0	α^0	α^3
α^5	α^4	α^6	α^3	α^2	α^0	0	α^1
α^6	α^2	α^5	α^0	α^4	α^3	α^1	0

$\cdot$	α^0	α^1	α^2	α^3	α^4	α^5	α^6
α^0	α^0	α^1	α^2	α^3	α^4	α^5	α^6
α^1	α^1	α^2	α^3	α^4	α^5	α^6	α^0
α^2	α^2	α^3	α^4	α^5	α^6	α^0	α^1
α^3	α^3	α^4	α^5	α^6	α^0	α^1	α^2
α^4	α^4	α^5	α^6	α^0	α^1	α^2	α^3
α^5	α^5	α^6	α^0	α^1	α^2	α^3	α^4
α^6	α^6	α^0	α^1	α^2	α^3	α^4	α^5

3. ábra. $GF(2^3)$ mező elemeinek összegző és szorzó táblázata

3.2. Szorzás

Az elemek közötti szorzás megfelel az α hatványaként megadott elemek kitevőinek $2^\omega - 1$ maradékosztályon vett összegével.

$$\alpha^i \cdot \alpha^j = \alpha^{(i+j) \mod (2^\omega - 1)}$$

Megjegyzés. Hasonlóan az osztást a kitevők különbségeként értelmezhetjük. Mivel a $GF(2^\omega)$ test véges sok elemmel rendelkezik, ezért megszerkeszthetőek a szorzó és összegző táblázatok, továbbá az elemek logaritmusainak és inverz logaritmusainak táblázata. Így az elemek szorzása, illetve osztása egyszerűen a megfelelő indexű táblázatbeli elemek kiolvasásával valósítható meg.

4. Reed-Solomon kódok

4.1. Definíció ((n, k) kód)**.** Az $A^n \to A^k, (k \geq n+1)$ kódolást (n, k) kódnak nevezzük.

4.2. Definíció ($RS(n, n+k)$ kód)**.** Az $RS(n, n+k)$ kód alatt a $GF(2^{\log_2(n+1)})$ testen értelmezett, k paritás szimbólummal rendelkező Reed-Solomon kódot értjük.

4.3. Definíció (generátor polinom)**.** A $g(x) \in GF(2^\omega)[x]$ polinomot az $RS(n, n+k)$ kód generátor polinomjának nevezzük, ha

$$g(x) = \prod_{i=1}^{k} (x - \alpha^i)$$

4.1. Kódolás

Adott az alábbi üzenet

$$M = \{m_0, m_1, \ldots, m_{L_m}\}, \quad (m_i \in GF(2^\omega))$$

definiáljuk az ún. üzenet polinomot

$$m(x) = \sum_{i=0}^{L_m} m_i x^i$$

ekkor a paritás polinom és a keletkező kódszó

$$p(x) = x^{n-k} m(x) \mod g(x) \tag{1}$$

$$c(x) = x^{n-k} m(x) + p(x) \tag{2}$$

4.2. Dekódolás

Adott az alábbi fogadott üzenet

$$R = \{r_0, r_1, \ldots, r_{L_r}\}, \quad (r_i \in GF(2^\omega))$$

képezzük az üzenet polinomot

$$r(x) = \sum_{i=0}^{L_r} r_i x^i$$

A fogadott üzenet felírható az $e(x)$ hibapolinom segítségével

$$r(x) = c(x) + e(x)$$

Állítás. A kódpolinom felírható a $c(x) = q(x)g(x)$ alakban.

Bizonyítás. Az $x^{n-k}m(x) = q(x)g(x) + p(x)$ kifejezéssel helyettesítve

$$c(x) = x^{n-k}m(x) + p(x) = q(x)g(x) + \underbrace{p(x) + p(x)}_{0} = q(x)g(x)$$

□

Következmény. A generátor polinom gyökei a kódpolinomnak is gyökei, azaz

$$c(\alpha^i) = 0, \quad (1 \leq i \leq k)$$

4.4. Definíció (Szindrómák)**.** Az alább definiált elemeket szindrómáknak nevezzük

$$S_i = r(\alpha^i) \quad (1 \leq i \leq k)$$

Állítás. Az S_i szindrómák megegyeznek az $e(\alpha^i)$ értékkel.

Bizonyítás. $S_i = r(\alpha^i) = c(\alpha^i) + e(\alpha^i) = \underbrace{m(\alpha^i)g(\alpha^i)}_{0} + e(\alpha^i) = e(\alpha^i)$ □

Ebből következik ha nem történt hiba, azaz $e(x) = 0$, akkor

$$S_i = 0 \quad (1 \leq i \leq k)$$

4.2.1. Hibák pozíciójának meghatározása

Tegyük fel, hogy az üzenet továbbítása során v darab hiba történt a $j_1, j_2, \ldots, j_v$ indexű szimbólumokban.

Ekkor az $e(x)$ hibapolinom az alábbi alakban írható fel

$$e(x) = e_{j_1}x^{j_1} + e_{j_2}x^{j_2} + \cdots + e_{j_v}x^{j_v}$$

ahol e_{j_i} az i-edik pozicióban történt eltérés nagyságát fejezi ki.

Definiáljuk az ún. hibalokátor számot

$$\beta_l = \alpha^{j_l}$$

Ekkor az $S_1, S_2, \ldots, S_k$ szindrómákat meghatározva az alábbi egyenletrendszerhez jutunk [2]

$$\begin{cases} S_1 = r(\alpha) = e_{j_1}\beta_1 + e_{j_2}\beta_2 + \cdots + e_{j_v}\beta_v \\ S_2 = r(\alpha^2) = e_{j_1}\beta_1^2 + e_{j_2}\beta_2^2 + \cdots + e_{j_v}\beta_v^2 \\ \vdots \\ S_k = r(\alpha^k) = e_{j_1}\beta_1^k + e_{j_2}\beta_2^k + \cdots + e_{j_v}\beta_v^k \end{cases}$$

Az egyenletrendszer k darab egyenletből és $2^{\frac{k}{2}}$ ismeretlenből áll, így létezik pontos megoldása. Azonban az egyenletrendszer gyökei konvencionális módszerekkel nem határozhatóak meg, hiszen az egyenletrendszer β_i ismeretlenjeiben nemlineáris. Bármely olyan módszert, amely az ismertetett egyenletrendszert megoldja, Reed-Solomon dekódolási algoritmusnak nevezünk.

Definiáljuk a $\Lambda(x)$ hibalokátor polinomot

$$\begin{aligned} \Lambda(x) &= (1 + \beta_1 x)(1 + \beta_2 x)\ldots(1 + \beta_v x) = \\ &= 1 + \lambda_1 x + \lambda_2 x^2 + \cdots + \lambda_v x^v \end{aligned}$$

Az alábbiakban ismertetjük a $\Lambda(x)$ polinom, vagyis a β_i értékek meghatározására alkalmas Berlekamp-Massey és az e_{j_i} értékek meghatározásához a Forney algoritmust.

Megjegyzés. A hibalokátor polinom gyökei az $1/\beta_i$ értékek, azaz a hibalokátor számok inverzei. Ha a hibalokátor polinom nem négyzetmentes, vagy gyökei nem esnek a $GF(2^\omega)$ mezőbe, akkor a keletkezett hibákat nem tudjuk javítani.

4.2.2. A Berlekamp-Massey iteratív algoritmus

---

1. **Berlekamp-Massey**(S, L) *[S - szindrómák halmaza]*
2. $k \leftarrow 0,\ \Lambda(x) \leftarrow 1,\ T(x) \leftarrow x$
3. **FOR** $i \leftarrow 1$ **TO** $|S|$ **DO**
4. $\quad \delta \leftarrow 0$
5. $\quad$ **FOR** $j \leftarrow 1$ **TO** k **DO**
6. $\quad\quad \delta \leftarrow \delta + \lambda_j S_{i-j}$
7. $\quad$ **ENDFOR**
8. $\quad \delta \leftarrow \delta - S_i$
9. $\quad$ **IF** $\delta \neq 0$ **THEN**
10. $\quad\quad \Lambda(x) \leftarrow \Lambda(x) - \delta T(x)$
11. $\quad\quad$ **IF** $2k < i$ **THEN**
12. $\quad\quad\quad T(x) \leftarrow T(x) + \Lambda(x)/\delta$
13. $\quad\quad\quad k \leftarrow i - k$
14. $\quad\quad$ **ENDIF**
15. $\quad$ **ENDIF**
16. $\quad T(x) \leftarrow xT(x)$
17. **ENDFOR**
18. $L \leftarrow \Lambda(x)$
19. **END**

---

4. ábra. Berlekamp-Massey iteratív algoritmus

4.2.3. Forney algoritmus

Képezzük az S_i szindrómákból az alábbi végtelen fokszámú polinomot

$$S(x) = S_1 x + S_2 x^2 + \cdots + S_k x^k + S_{k+1} x^{k+1} + \ldots$$

Definiáljuk az $\Omega(x)$ hibaméret meghatározó polinomot

$$\Omega(x) = [1 + S(x)]\Lambda(x)$$

Mivel csak az $S(x)$ polinom első k darab együtthatóját ismerjük, a dekódolási probléma az alábbi egyenlőséget kielégítő $k/2$ fokú $\Lambda(x)$ polinom meghatározásával válik egyenlővé

$$\Lambda(x)[1 + S(x)] \equiv \Omega(x) \mod x^{k+1}$$

Ekkor a hibapolinom együtthatói az alábbi módon fejezhetőek ki

$$e_{j_i} = \frac{-\beta_{j_i}\Omega(\beta_{j_i}^{-1})}{\Lambda'(\beta_{j_i}^{-1})}$$

és a hibapolinom

$$e(x) = \sum_{i=1}^{v} e_{j_i} x^{\beta_{j_i}}$$

A hibapolinom meghatározása után az eredeti üzenetet a hibapolinom kivonásával kaphatjuk vissza

$$m(x) = r(x) - e(x)$$

Hivatkozások

[1] Robert G. Gallager. *Information Theory and Reliable Communication*. Wiley, January 1968.

[2] Bernard Sklar. Reed-solomon codes.