Universitatea din Pitești
Facultatea de Matematică-Informatică

Reprezentarea informației în sistemele de calcul. Coduri

Cuprins

Reprezentarea informației folosind coduri

- Informația este transmisă între sistemele de calcul, prin rețele de calculatoare sub formă de șiruri de biți.
- Pentru un nivel corespunzător de securitate și corectitudine a informației transmise se utilizează reprezentarea datelor în diferite coduri.
- În prezent există numeroase metode de codificare a informației, printre care:
    - reprezentarea numerelor în cod binar;
    - reprezentarea caracterelor în coduri precum ASCII sau UNICODE;
    - reprezentarea șirurilor de biți în coduri care sunt detectoare și corectoare de erori (codul Gray, coduri de tip Hamming, etc).

Fie $A = \{0, 1\}$ (numit alfabet binar). Se notează prin A^n produsul cartezian $\underbrace{A \times A \times \ldots \times A}_{n \text{ ori}}$.

Definiție

Se numește **cod de tipul** (n, k), $n \geq k$ *imaginea funcției injective* $C : A^k \to A^n$. *Elementele codului (adică elementele mulțimii* $Im(C)$*) se numesc cuvinte codate.*

Exemplu

Dacă $s = a_1 \ldots a_k \in A^k$ *este un șir de elemente astfel încât* $a_1, \ldots, a_k \in A$ *(*s *se numește cuvânt necodat), atunci* $C(s) = x_1 \ldots x_n \in A^n$ *va fi cuvântul codat.*

- În general, există o submulțime de k biți din cuvântul codat $x_1 \dots x_n$ care reprezintă biții de informație inițiali care trebuie transmiși. Ceilalți biți se numesc biți de control și sunt utilizați pentru detectarea și corectarea unor eventuale erori.

Definiție

Se numește codificare transformarea unui cuvânt necodat $s \in A^k$, într-un cuvânt codat $C(s) \in A^n$.
Operația inversă se numește decodificare.

- Dacă în codificarea unui cuvânt apar biți eronați, atunci codificarea este greșită. În caz contrar, codificarea este corectă.

- Pentru codificarea unui mesaj format din mai mult de k simboluri, acesta se împarte în cuvinte de k simboluri, fiecare dintre aceste cuvinte este codificat obținând un cuvânt format din n simboluri cuvinte care reunite reprezintă codificarea mesajului inițial.

Exemplu

Funcția $C : A \to A^3$, $C(0) = 000$ *și* $C(1) = 111$ *reprezintă un cod de tip* $(3, 1)$.
Dacă mesajul inițial este $m = 1010$, *atunci mesajul codat este* $C(m) = 111000111000$.
Regula de detectare și corectare a erorilor este aceea de a alege bitul majoritar din fiecare secvență de trei biți.

Coduri detectoare și corectoare de erori

Definiție

Se numește distanță Hamming între două șiruri de biți a și b de aceeași lungime numărul biților de valori diferite situați în aceeași poziție în cele două șiruri.

Astfel, dacă $a, b \in A^n$, $a = a_1 \dots a_n$ și $b = b_1 \dots b_n$ atunci $d_H(a, b) = card\{i | a_i \neq b_i, 1 \leq i \leq n\}$.

Exemplu

Pentru $a = 1000$, $b = 1001$, $d_H(a, b) = 1$.

Observație

Distanță Hamming se poate determina utilizând operatorul $^\wedge$ *(sau exclusiv) pe biți astfel:*

$$d_H(a,b) = \sum_{i=1}^{n} a_i \,^\wedge b_i$$

Definiție

Distanța Hamming minimă a unui cod $C \subseteq A^n$ *este:*

$$d_H(C) = min\{d_H(a,b) | a,b \in C, \ \ a \neq b\}$$

Observație

Oricare două cuvinte distincte ale codului C *diferă în cel puțin* $d_H(C)$ *poziții.*

Coduri detectoare și corectoare de erori

- Numărul de erori pe care un cod le poate corecta se numește capacitatea de corectare a codului și este dată de egalitatea:

$$e(C) = \left[\frac{d_H(C) - 1}{2}\right]$$

- Astfel, pentru ca un cod C să poată corecta k erori într-un șir de biți, este necesar ca $d_H(C)$ să fie $2k + 1$.
- Numărul de erori pe care un cod C le poate detecta se numește capacitatea de detecție a codului și este egal cu $d_H(C) - 1$.

Observație

Noțiunile prezentate anterior se pot extinde pentru un alfabet A cu q simboluri, situație în care vorbim despre un cod q-ar.

Codul BCD (Binary-Coded Decimal)

- Codul BCD presupune codificare fiecărei cifre a unui număr scris în baza 10 în secvența de 4 biți corespunzătoare.
- Pentru a controla transportul corect al informației codificate se poate introduce un *bit de control*. Acesta poate fi reprezentat de către un bit care redă paritatea valorilor 1 (sau a valorilor 0). Astfel se poate conveni ca bitul de control să fie 0 dacă numărul biților 1 este par și 1 în caz contrar.

cifra	codificare	cifra	codificare
0	0000	5	0101
1	0001	6	1010
2	0010	7	1011
3	0011	8	1000
4	0100	9	1001

Exemplu

Dacă $m = 938$ *atunci* $BCD(m) = 1001\ 0011\ 1000$, *iar dacă se adaugă biți de control:* $BCD(m) = 1001\textcolor{red}{0}\ 0011\textcolor{red}{0}\ 1000\textcolor{red}{1}$

Observație

- *Dacă* a, b *două codificări folosind BCD fără biți de paritate, atunci* $d_H(a,b) \geq 1$, *deci* $d_H(BCD) = 1$ *În consecință acest cod nu poate detecta (și corecta) erori.*
- *Dacă se adaugă un bit de paritate, atunci, se observă că pentru orice două codificări* x, y, $d_H(x,y) \geq 2$, *deci* $d_H(BCD) = 2$. *Prin urmare, acest cod poate detecta* 1 *eroare, dar* $e(C) = \left[\frac{2-1}{2}\right] = 0$, *deci codul nu poate corecta erorile detectate.*

Codul Gray

- Codul Gray realizează codificarea utilizând operatorul sau exclusiv ^ pe biți.
- Dacă $a[n]$ este o reprezentare în cod binar binar a unui număr, atunci codul Gray corespunzător, notat cu $g[n]$, se obține astfel:

$a[0] = g[0]$, iar pentru $i = \{1, ..., n-1\}$, $g[i] = a[i-1]\,^\wedge a[i]$.

Exemplu

Dacă $a = 1000_2$. *Atunci,* $g = 1100$.

Exemplu

Dacă $a = 13_{10}$, *atunci* $a = 8 + 4 + 1 = 1101_2$ *și se obține* $g = 1011$.

Codul Gray

Pentru $n = 4$, se pot scrie codificările:

Exemplu

Binar	*Gray*
0000	*0000*
0001	*0001*
0010	*0011*
0011	*0010*
....	....
1111	*1000*

Observație

Dacă a, b *două codificări din exemplul de mai sus, atunci* $d_H(a, b) = 1$, *deci codul nu poate detecta sau corecta erori.*

Codul ASCII (American Standard Code for Information Interchange)

- Pentru reprezentarea și transportul unui caracter (literă, simbol, cifră) utilizând codul ASCII standard se folosesc 7 biți (caracterele sunt reprezentate folosind valorile numerice între 0 și 127).

Observație

În limbajul de programare C++ tipul de dată char se reprezintă pe 1 octet (8 biți).

- Al optulea bit se poate folosi drept bit de control, fiind inițializat cu suma modulo 2 a primilor 7 biți care reprezintă în cod binar caracterul curent.

- Codurile ASCII coresepunzătoare cifrelor sistemului zecimal sunt cuprinse între 48 și 57, cele ale literelor mari ale alfabetului latin între 65 și 90, iar cele ale litelelor mici între 97 și 122.

Exemplu

- *Pentru codul ASCII a lui* $A = 65 = 64 + 1 = 2^6 + 2^0 = 1000\ 001_2$ *se calculează suma* $S = 1 + 0 + 0 + 0 + 0 + 0 + 1 = 2 = 0 (mod\ 2)$. *Deci reprezentarea lui 'A' va fi* $1000\ 0010$.
- *Dacă din diferite motive de transport, primul bit zero devine unu, această literă este transmisă greșit sub forma* $1100\ 0010$. *La destinație se va realiza verificarea sumei biților* $1 + 1 + 0 + 0 + 0 + 0 + 1 = 3 = 1 (mod\ 2)$. *Acest bit este diferit de bitul de control transmis, prin urmare se va afișa eroare.*

- Deși acest cod detectează erorile apărute pentru un număr impar de biți eronați, nu poate detecta care biți sunt greșit transmiși și deci nu se poate realiza corecția lor.
- De asemenea, dacă un numar par de biți au valorile modificate, bitul de control nu detectează erori.
- De exemplu, pentru transmiterea caracterului A prin șirul eronat $0100\ 0010$ se va recepta la destinație o codificare corectă corespunzătoare numărului 33, adică se va afișa caracterul '!'.

Codul ASCII

Exemplu

Pentru $s = Help!$ *codificarea corectă va fi:*

$H = 72 = 64 + 8 = 1001\ 000_2,$

$S = 2 = 0(mod\ 2) \rightarrow 1001\ 0000.$

$e = 101 = 64 + 32 + 4 + 1 = 1100\ 101_2,$

$S = 4 = 0(mod\ 2) \rightarrow 1100\ 1010.$

$l = 108 = 64 + 32 + 8 + 4 = 1101\ 100_2,$

$S = 4 = 0(mod\ 2) \rightarrow 1101\ 1000.$

$p = 112 = 64 + 32 + 16 = 1110\ 000_2,$

$S = 3 = 1(mod\ 2) \rightarrow 1110\ 0001.$

$! = 33 = 32 + 1 = 0100\ 001_2,$

$S = 2 = 0(mod\ 2) \rightarrow 0100\ 0010.$

Codul ASCII

Exemplu

Deci, șirul de biți care trebuie transportat este:
1001 0000 1100 1010 1101 1000 1110 0001 0100 0010.
Dacă din diferite motive de transport, se petrec următoarele modificări:
1001 0000 1100 1010 1101 1000 11**01** **1**01**1** **1**1**1**0 **1**01**1**,
atunci, decodificarea ar rezulta în: $Helou$.
Dacă ultimul bit ar rămâne corect, atunci pentru ultimul caracter, s-ar afișa eroare.

Codul ASCII

Observație

Fie a, b *două codificări folosind codul ASCII.*
Atunci, $d_H(a, b) \geq 2$*, deoarece există un bit de control care reprezintă paritatea și codul ASCII Standard conține, de exemplu valorile* $a = 000\ 1000$ *și* $000\ 1001$*, deci* $d_H(ASCII)$*=2.*
În consecință, codul ASCII poate detecta o eroare, dar nu are proprietatea de a corecta erorile detectate.

Coduri Hamming

Hamming a introdus, printre altele, tipurile de coduri care îi poartă numele. Printre proprietățile unui cod Hamming se număra:

- detectarea și corectarea unui bit eronat;
- introducerea unor biți de control poziționați pe pozițiile corespunzătoare puterilor lui 2, care sunt de fapt combinații între biții de informație și ajută la corectarea erorilor;
- numărul de biți suplimentari pentru o codificare de lungime n se determină după regula: pentru $k_0 = max\{k|2^k \leq n\}$ se vor introduce $k_0 + 1$ biți de control.

Coduri Hamming

Exemplu

Pentru $n = 4 \rightarrow k_0 = 2 \rightarrow$ *se vor introduce* 3 *biți de control.*
Pozițiile acestor biți vor fi date de:
$2^0 = 1, 2^1 = 2, 2^2 = 4$
Prin urmare, se vor codifica 4 *biți de informație în* 7 *biți de transport. Codul se numește din acest motiv cod Hamming* $(7, 4)$*.*

Numerotarea se începe de la stânga spre dreapta și primul indice va fi 1.

Exemplu

Pentru $n = 16 \rightarrow k_0 = 4 \rightarrow$ *se vor introduce* 5 *biți de control.*
Pozițiile acestor biți vor fi date de:
$2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16$

Coduri Hamming

Modalitatea de control (C) a biților de control $(C1, C2, C3, \ldots)$ asupra biților de informație $(I1, I2, I3, I4, \ldots)$.

Nr. Bit	Tip Bit	C1	C2	C3	C4 ...
1	C1	C	0	0	0
2	C2	0	C	0	0
3	I1	C	C	0	0
4	C3	0	0	C	0
5	I2	C	0	C	0
6	I3	0	C	C	0
7	I4	C	C	C	0
8	C4	0	0	0	C
...	...	...	...	...	...

Codul Hamming(7,4)

Modalitatea de control (C) pentru codul (7,4) a biților de control ($C1, C2, C3$) asupra biților de informație ($I1, I2, I3, I4$) este:

- C1 controlează biții de informație de pe pozițiile 3, 5, 7.
- C2 controlează biții de informație de pe pozițiile 3, 6, 7.
- C3 controlează biții de informație de pe pozițiile 5, 6, 7.

Prin urmare:

- I1 (poziția 3) este controlat de biții de control $C1$, $C2$ (de pe pozițiile 1, 2).
- I2 (poziția 5) este controlat de biții de control $C1$, $C3$ (de pe pozițiile 1, 4).
- I3 (poziția 6) este controlat de biții de control $C2$, $C3$ (de pe pozițiile 2, 4).
- I4 (poziția 7) este controlat de toți cei 3 biți de control.

Codul Hamming(7,4)

- Dacă notăm cu $H[i]$ bitul aflat pe poziția i în mesajul codificat folosind codul Hamming(7,4), atunci:

H[1] = C1 = I1 + I2 + I4 (mod 2)
H[2] = C2 = I1 + I3 + I4 (mod 2)
H[3] = I1
H[4] = C3 = I2 + I3 + I4 (mod 2)
H[5] = I2
H[6] = I3
H[7] = I4

Codul Hamming(7,4)

Exemplu

Spre exemplu, fie $a = 1010$ *un cuvânt codificat. Atunci reprezentarea în codul Hamming (7,4) este:*

1	*2*	*3*	*4*	*5*	*6*	*7*
1	**0**	*1*	**1**	*0*	*1*	*0*

Cazuri posibile:

- *dacă bitul 3 este eronat (I1), adică*

1	*2*	*3*	*4*	*5*	*6*	*7*
1	*0*	**0**	*1*	*0*	*1*	*0*

  atunci din calcule, rezultă $C1 = 0(F), C2 = 1(F), C3 = 1(A)$, *deci biții* $C1, C2$ *sunt eronați, deci I1 este eronat.*

Codul Hamming(7,4)

Exemplu

- *dacă bitul 5 este eronat, adică*

1	2	3	4	5	6	7
1	0	1	1	**1**	1	0

atunci din calcule, rezultă $C1 = 0(F), C2 = 0(A), C3 = 0(F)$, *deci biții* $C1, C3$ *sunt eronați, deci I2 este eronat.*

- *dacă bitul 6 este eronat, adică*

1	2	3	4	5	6	7
1	0	1	1	0	**0**	0

atunci din calcule, rezultă $C1 = 1(A), C2 = 1(F), C3 = 0(F)$, *deci biții* $C2, C3$ *sunt eronați, deci I3 este eronat.*

Codul Hamming(7,4)

Exemplu

- *dacă bitul 7 este eronat, adică*

1	2	3	4	5	6	7
1	0	1	1	0	1	**1**

atunci din calcule, rezultă $C1 = 0(F), C2 = 1(F), C3 = 0(F)$, *deci biții* $C1, C2, C3$ *sunt eronați, deci I4 este eronat.*

Se poate însă, ca și un bit de semn să fie transportat greșit. Acest tip de cod poate detecta și o astfel de eroare:

- *dacă bitul C1 este eronat, adică*

1	2	3	4	5	6	7
0	0	1	1	0	1	0

atunci din calcule, rezultă $C1 = 0(F), C2 = 0(A), C3 = 1(A)$, *deci bitul* $C1$ *este eronat.*

Codul Hamming(7,4)

Exemplu

- *dacă bitul C2 este eronat, adică*

1	2	3	4	5	6	7
1	**1**	1	1	0	1	0

atunci din calcule, rezultă $C1 = 1(A), C2 = 1(F), C3 = 1(A)$, *deci bitul* $C2$ *este eronat.*

- *dacă bitul C3 este eronat, adică*

1	2	3	4	5	6	7
1	0	1	**0**	0	1	0

atunci din calcule, rezultă $C1 = 1(A), C2 = 0(A), C3 = 0(F)$, *deci bitul* $C3$ *este eronat.*

Codul Hamming(7,4)

Observație

Dacă există doi biți eronați:

1	2	3	4	5	6	7
0	**1**	1	1	0	1	0

atunci $C1 = 1(F), C2 = 0(F), C3 = 1(A)$, *corectarea erorii nu este posibilă.*