

Coduri Hamming

Considerand doua cuvinte de cod, (10001001 si 10110001), se poate determina prin câti biti difera. În acest caz, 3 biti sunt diferiti (numaram câti de 1 apar în rezultatul operatiei XOR pe bitii celor doua cuvinte de cod). Numarul de biti prin care difera cele doua cuvinte de cod se numeste distanta Hamming. Semnificatia este urmatoarea: daca doua cuvinte de cod sunt la distanta Hamming d , va fi nevoie de d erori de câte un bit pentru a-l converti pe unul în celalalt. Distanța Hamming minima este distanta Hamming a codului complet.

Puterea de detectare si corectare pentru un cod depind de distanta sa Hamming. Pentru a detecta d erori de câte un bit veti avea nevoie de un cod cu distanta $d + 1$, pentru ca un astfel de cod garanteaza ca d erori de câte un bit nu vor putea niciodata modifica un cuvânt de cod valid într-un alt cuvânt de cod valid. Similar, pentru a corecta d erori de câte un bit, veti avea nevoie de un cod cu distanta $2d + 1$, pentru ca în acest fel cuvintele de cod valide vor fi atât de departate, încât chiar cu d schimbări, cuvântul de cod original tot va fi mai aproape decât orice alt cuvânt de cod, astfel încât poate fi unic determinat.

Ca un exemplu simplu de cod detector de erori, sa consideram un cod în care la date se adauga un singur bit de paritate. Bitul de paritate este ales astfel încât numarul de biti egali cu 1 din cuvântul de cod sa fie par (sau impar). Un astfel de cod are distanta 2, deoarece orice eroare de un bit va produce un cuvânt de cod cu paritatea gresita. Cu alte cuvinte, este nevoie de doua erori de câte un bit pentru a trece de la un cuvânt de cod valid la alt cuvânt de cod valid. Poate fi folosit pentru a detecta erori singulare. Ori de câte ori un cuvânt cu paritatea gresita este citit din memorie se semnaleaza o eroare. Programul nu mai poate continua, dar cel puțin nu sunt calculate rezultate incorecte.

Ca un exemplu simplu de cod corector de erori, sa consideram un cod cu doar patru cuvinte de cod valide:

0000000000, 0000011111, 1111100000 si 1111111111

Acest cod are distanta 5, ceea ce înseamna ca poate corecta erori duble. Daca apare cuvântul de cod 0000000111, receptorul stie ca originalul trebuie sa fi fost 0000011111 (daca nu s-au produs mai mult de 2 erori). Daca totusi o eroare tripla a schimbat 0000000000 în 0000000111, eroarea nu poate fi corectata.

Într-un cod Hamming, cuvântului de m biti i se adauga r biti de paritate, formând un nou cuvânt, de lungime $r + m$ biti. Bitii sunt numerotati de la 1, nu de la 0, cu bitul 1 cel mai din stânga (cel mai semnificativ). Toti bitii al caror numar este o putere a lui 2 sunt biti de paritate; ceilalti sunt folositi pentru date. De exemplu, pentru un cuvânt de 16 biti, sunt adaugati 5 biti de paritate. Bitii 1, 2, 4, 8 si 16 sunt biti de paritate, iar toti ceilalti sunt biti de date. În total, cuvântul are 21 de biti (16 date, 5 paritate). În acest exemplu vom folosi (în mod arbitrar) paritatea para.

Fiecare bit de paritate verifica anumite pozitii ale bitilor; bitul de paritate este setat astfel încât numarul total de 1 din pozitiile verificate sa fie par. Pozitiile bitilor verificate de bitii de paritate sunt

Bitul 1 verifica bitii 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21.

Bitul 2 verifica bitii 2, 3, 6, 7, 10, 11, 14, 15, 18, 19.

Bitul 4 verifica bitii 4, 5, 6, 7, 12, 13, 14, 15, 20, 21.

Bitul 8 verifica bitii 8, 9, 10, 11, 12, 13, 14, 15.

Bitul 16 verifica bitii 16, 17, 18, 19, 20, 21.

În general, bitul b este verificat de acei biti b_1, b_2, \dots, b_j , astfel încât $b_1 + b_2 + \dots + b_j = b$. De exemplu, bitul 5 este verificat de bitii 1 și 4, pentru că $1 + 4 = 5$. Bitul 6 este verificat de bitii 2 și 4 pentru că $2 + 4 = 6$, și așa mai departe.

Exemplu pentru un cuvânt de 16 biti (1111000010101110). Cuvântul de cod de 21 de biti este 001011100000101101110. Pentru a vedea cum funcționează corectia erorilor, să vedem de se întâmplă dacă bitul 5 este inversat din cauza unei fluctuații de tensiune pe rețeaua de alimentare. Noul cuvânt de cod ar fi 001001100000101101110 în loc de 001011100000101101110. Cei 5 biti de paritate vor fi verificați, obținându-se următoarele rezultate:

Bitul de paritate 1 incorect (1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 conțin cinci de 1).

Bitul de paritate 2 corect (2, 3, 6, 7, 10, 11, 14, 15, 18, 19 conțin șase de 1).

Bitul de paritate 4 incorect (4, 5, 6, 7, 12, 13, 14, 15, 20, 21 conțin cinci de 1).

Bitul de paritate 8 corect (8, 9, 10, 11, 12, 13, 14, 15 conțin doi de 1).

Bitul de paritate 16 corect (16, 17, 18, 19, 20, 21 conțin patru de 1).

Numărul total de 1 din bitii 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 și 21 trebuie să fie un număr par, deoarece se folosește paritatea pară. Bitul incorect trebuie să fie unul dintre cei verificați de bitul 1, adică bitul 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 sau 21. Bitul de paritate 4 este incorect, însemnând că unul din bitii 4, 5, 6, 7, 12, 13, 14, 15, 20 sau 21 este incorect. Eroarea trebuie să fie într-unul din bitii care apar în ambele liste, adică 5, 7, 13 sau 21. Totuși bitul 2 este corect, eliminând bitii 7 și 15. Similar, bitul 8 este corect, eliminând bitul 13. În fine, bitul 16 este corect, eliminând bitul 21. Singurul bit rămas este bitul 5, care este cel eronat. De vreme ce a fost citit drept 1, ar trebui să fie un 0. În această mod se pot corecta erorile.

O metodă simplă pentru găsirea bitului eronat este să se calculeze mai întâi toți bitii de paritate. Dacă toți sunt corecți, atunci nu a fost nici o eroare (sau au fost mai multe). Apoi se adună toți bitii de paritate incorecți, considerând 1 pentru bitul 1, 2 pentru bitul 2, 4 pentru bitul 4 și așa mai departe. Suma rezultată indică poziția bitului eronat. De exemplu, dacă bitii de paritate 1 și 4 sunt incorecți, dar bitii 2, 8 și 16 sunt corecți, atunci bitul 5 ($1 + 4$) a fost inversat.