Cyklické redundantní součty a generátory pseudonáhodných čísel

Definice

Řekneme, že polynomy $a(x)$, $b(x)$ jsou kongruentní modulo $m(x)$, (značení $a(x) \equiv b(x) \pmod{m(x)}$), pokud existuje polynom $k(x)$ tak, že $a(x) - b(x) = k(x) \cdot m(x)$.

Tvrzení

Kongruence modulo $m(x)$ je relace ekvivalence na množině $\mathbb{K}[x]$, která respektuje sčítání a násobení polynomů. Proto můžeme na třídách ekvivalence modulo $m(x)$ zavést sčítání a násobení:

$$[a(x)]_{m(x)} \oplus [b(x)]_{m(x)} = [a(x) + b(x)]_{m(x)}$$

$$[a(x)]_{m(x)} \odot [b(x)]_{m(x)} = [a(x) \cdot b(x)]_{m(x)}$$

Výsledná struktura, značená $\mathbb{K}[x]/m(x)$, je komutativní okruh s jednotkou.

Relax

Místo

$$[x]_{x^2+1} \odot [x+1]_{x^2+1} = [x+1]_{x^2+1} \quad \text{v } \mathbf{Z}_2[x]/(x^2+1)$$

píšeme

$$x \cdot (x+1) = x+1 \quad \text{v } \mathbf{Z}_2[x]/(x^2+1)$$

(vynásobení a nahrazení zbytkem po dělení).

Atd.

CRC

- Cyklické redundantní součty (zkratka CRC z anglického Cyclic Redundancy Check) slouží k detekci chyb, ale neopravují je. Zobecňují paritní bit.
- Optimální n-bitové CRC umí detekovat jakoukoli 2-bitovou chybu v čísle $2^n - 1$ (data). Běžné hodnoty n jsou 12, 16 a 32.

Data jako polynomy

Data vyjádřené ve dvojkové soustavě lze chápat jako polynomy nad tělesem $\mathbf{Z}_2$, tzv. datové polynomy. Bity v datech chápeme jako koeficienty polynomu. Např.

$$1100010100 \quad \rightarrow \quad x^9 + x^8 + x^4 + x^2 .$$

Algoritmus počítající CRC

Algoritmus počítající CRC je specifikován jeho tzv. generujícím polynomem $g(x) \in \mathbf{Z}_2[x]$. Např.

$$g(x) = x^3 + x + 1\,.$$

CRC hodnota datového polynomu $f(x) \in \mathbf{Z}_2[x]$ se spočítá jako zbytek po dělení datového polynomu $f(x)$ generujícím polynomem $g(x)$.

Přenos dat

Spolu s daty tedy posíláme také CRC hodnotu. Při příjmu spočítáme opět zbytek po dělení generujícím polynomem. Pokud vyjde stejná CRC hodnota prohlásíme data za správná, v opačném případě za chybná.

Příklad

Mějme datový polynom $f(x) = x^9 + x^8 + x^4 + x^2$ a generující polynom $g(x) = x^3 + x + 1$. Dělící algoritmus pro polynomy dává

$$f(x) = (x^6 + x^5 + x^4) \cdot g(x) + x^2 .$$

CRC hodnota pro data 1100010100 je tedy 100.

Poznámky

- CRC s generujícím polynomem $x + 1$ odpovídá výpočtu paritního bitu.
- CRC lze chápat jako lineární kód.
- Dělící algoritmus lze implementovat chytře, takže výpočet je rychlý i pro datové polynomy velkých stupňů.
- Pro generující polynom stupně n máme 2^n možných CRC hodnot, protože zbytek po dělení může být jakýkoliv polynom nad $\mathbf{Z}_2$ stupně $n - 1$ nebo méně.

Typické užívané generující polynomy

- $x^{12} + x^{11} + x^3 + x + 1$ (telekomunikace),
- $x^{16} + x^{12} + x^5 + 1$ (Bluetooth, SD),
- $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (Ethernet, MPEG-2, PNG).

Poznámka

Proč právě tyto polynomy? Protože jsou založeny na tzv. primitivních polynomech a jsou schopné detekovat 2-bitovou chybu ve velmi dlouhém datovém řetězci. Např. CRC stupně 16 je schopen detekovat 2-bitové chyby v binárních datech délky $2^{15} - 1$.

Jaké chyby umí CRC detekovat?

- Mějme CRC s generujícím polynomem $g(x)$. Předpokládejme, že vyšleme datový polynom $f(x)$ a přijmeme polynom $f'(x)$ s nějakými chybami.
- Chybový polynom je polynom $e(x) = f(x) - f'(x)$.
- Nechť $r(x)$ je CRC hodnota polynomu $f(x)$ a $r'(x)$ polynomu $f'(x)$, tj.

$$f(x) = h(x) \cdot g(x) + r(x)\,, \qquad f'(x) = h'(x) \cdot g(x) + r'(x)\,.$$

- Pak $e(x) = (h(x) - h'(x)) \cdot g(x) + (r(x) - r'(x))$, tj. $r(x) - r'(x)$ je zbytek po dělení $e(x)$ polynomem $g(x)$.
- CRC nedetekuje chybu, pokud $r(x) = r'(x)$, tj. pokud $g(x)|e(x)$.

1-bitové chyby

Pokud $f'(x)$ obsahuje 1-bitovou chybu na i-té pozici, pak $e(x) = x^i$. CRC s generujícím polynomem $g(x)$ nedetekuje 1-bitovou chybu, pokud $g(x)|x^i$. To je ale možné jen pro $g(x) = x^j$, $j \leq i$. Tedy $g(x) = x + 1$ detekuje 1-bitové chyby.

2-bitové chyby

Pokud $f'(x)$ obsahuje 2-bitovou chybu na i-té a j-té pozici $(i < j)$, pak $e(x) = x^i + x^j$ není detekováno, pokud $g(x)$ dělí

$$e(x) = x^i + x^j = x^i(1 + x^{j-i}).$$

Pokud má $g(x)$ absolutní člen 1, pak $g(x)$ nemá žádný společný faktor s x^i, a proto $g(x)|1 + x^{j-i}$.

Všiměte si, že $j - i$ je vzdálenost mezi chybami.

Příklad

Zjistěme jaké 2-bitové chyby by detekovat CRC s generujícím polynomem $g(x) = x^8 + 1 = x^8 - 1$ (odpovídá "XORování" jednotlivých bytů dat). Pro $n \geq 1$ platí:

$$x^n - 1 \quad = \quad (x-1) \cdot (x^{n-1} + x^{n-2} + \cdots + x + 1)$$

Substitucí

$$x^{8n} - 1 \quad = \quad (x^8 - 1) \cdot (x^{8(n-1)} + x^{8(n-2)} + \cdots + x^8 + 1)$$

Tedy $x^8 - 1 | x^{8n} - 1$ pro každé $n \geq 1$, tj. nebudou detekovány 2-bitové chyby, kde vzdálenost $j - i$ bude násobek 8.

Příklad (pokračování)

Nicméně chyby vzdálené o méně než 8, lze zjišťovat polynomem $x^4 + x + 1$, který nedetekuje 2-bitové chyby, které jsou od sebe vzdáleny násobek 15! Tj. $x^4 + x + 1$ dělí $x^n - 1$ pouze, pokud $15|n$. Z toho plyne, že polynom $x^8 - 1$ není optimální na detekci 2-bitových chyb.

Příklad

Polynom $x^5 + x^2 + 1$ je schopen detekovat 2-bitové chyby, které jsou od sebe vzdáleny o méně než 31. Nicméně malá změna, může tuto schopnost výrazně degradovat, např. $x^5 + x + 1$ nedekuje 2-bitové chyby vzdálené od sebe násobek 21.

Primitivní polynomy

Výše uvedené příklady ukazují, že generující polynom CRC je potřeba pečlivě vybírat. Jako rozumná volba se jeví tzv. primitivní polynomy.

Definice

Polynom $g(x) \in \mathbf{Z}_2[x]$ stupně d se nazývá primitivní, pokud nejmenší přirozené číslo n takové, že $g(x)|x^n - 1$, je rovno $2^d - 1$.

Věta

Polynom $g(x) \in \mathbf{Z}_2[x]$ stupně d je primitivní právě tehdy, když

1. $g(x)|x^{2^d-1} - 1$,
2. $g(x) \nmid x^{(2^d-1)/q} - 1$ pro každý prvočíselný faktor q čísla $2^q - 1$.

Detekce Burst Errors

Tzv. Burst Errors jsou chyby, které se vyskytují blízko sebe, tj. bloky chybných dat v datovém toku. CRC s generujícím polynomem $g(x)$ stupně n je schopen detekovat tyto chyby do délky $< n$.
Chybový polynom pro tyto chyby délky $< n$ lze zapsat takto:

$$e(x) = x^i \cdot p(x)\,,$$

kde $p(x)$ je polynom stupně $< n$ a $i \in \mathbb{N}$.

Potom $g(x)$ nedělí $e(x)$, protože $g(x)$ typicky neobsahuje faktor x a $\deg(g(x)) > \deg(p(x))$.

Příklad

Generující polynom CRC

$$g(x) = x^{16} + x^{15} + x^2 + 1 = (x+1) \cdot (x^{15} + x + 1)\,,$$

detekuje všechny 3-bitové chyby v datech o 32767 bitech. Primitivní polynom $x^{15} + x + 1$ zaručuje detekci 2-bitových chyb vzdálených od sebe méně než $2^{15} - 1 = 32767$, zatímco $x + 1$ detekuje chyby obsahující lichý počet bitů.

Generátor pseudonáhodných čísel

Generátor pseudonáhodných čísel je založen na rekurentní rovnici

$$z_{i+1} = f(z_i)\,,$$

kde $f : \mathbf{Z} \to \mathbf{Z}$ je nějaká funkce a z_0 je inicializační hodnota generátoru (seed).

Generátor založený na lineární diferenční rovnici nad $\mathbf{Z}_2$

Budeme chápat čísla z_i jako čísla vyjádřená ve dvojkové soustavě s pevným počtem bitů, tj. z_i odpovídá $(s_n, s_{n-1}, \ldots, s_1, s_0)$, kde $s_j \in \mathbf{Z}_2$. Pak jeden možný generátor můžeme definovat takto: Spočteme v tělese $\mathbf{Z}_2$ pro zadaná $a_0, \ldots, a_n \in \mathbf{Z}_2$

$$s_{n+1} = a_n s_n + a_{n-1} s_{n-1} + \cdots + a_1 s_1 + a_0 s_0\,.$$

Pak z_{i+1} vyjádřeno ve dvojkové soustavě je $(s_{n+1}, s_n, \ldots, s_2, s_1)$.

Definice

Charakteristický polynom diferenční rovnice
$s_{n+1} = a_n s_n + a_{n-1} s_{n-1} + \cdots + a_1 s_1 + a_0 s_0$ je polynom nad $\mathbf{Z}_2$

$$f(x) = x^{n+1} + a_n x^n + \cdots + a_1 x + a_0 \,.$$

Věta

1. Posloupnost vygenerovaná výše uvedeným generátorem je periodická.
2. Perioda je maximální právě tehdy, když charakteristický polynom je primitivní. V tom případě je perioda $2^{n+1} - 1$.