

Řešení úlohy č. 1

Konfigurační řetězec

Detailně rozebereme ukázkový vstup 2, tedy $n = 3$ skripty a $m = 2$ konfigurační řetězce. Víme, že první bit řetězce ovlivňuje první a třetí skript a druhý bit řetězce ovlivňuje druhý a třetí skript. Protože skripty nastřídačku fungují a padají, potřebujeme, aby počet bitů které ovlivňují daný skript, byl lichý. Nabízí se vyzkoušet všech 2^m možností konfiguračních řetězců a zkusit, zda fungují a zároveň umíme počítat, kolik jich bylo. To je ovšem pomalé.

Pro efektivnější řešení si problém trochu otočíme: bude nás zajímat, pro každý skript, kterými bity je ovlivňován. Označme si jednotlivé bity jako x_1, x_2 . To lze v tomto případě snadno zjistit: První skript je ovlivňován bitem x_1 , musí tedy platit, že x_1 je liché, protože ale x_i jsou bity, znamená to, že $x_1 = 1$. Druhý skript je ovlivňován druhým bitem řetězce, tedy stejně tak musí $x_2 = 1$. Poslední skript je ovlivňován oběma řetězci, takže musí být $x_1 + x_2$ liché, což v bitech znamená, že $x_1 \neq x_2$. Tady dostáváme spor a vidíme, že řešení neexistuje.

Problém jsme převedli na řešení soustavy lineárních rovnic modulo 2. V tomto případě soustava byla:

$$1 \cdot x_1 + 0 \cdot x_2 = 1$$

$$0 \cdot x_1 + 1 \cdot x_2 = 1$$

$$1 \cdot x_1 + 1 \cdot x_2 = 1$$

což lze maticově zapsat jako

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Vytvoříme tedy soustavu rovnic tím způsobem, že si pro každý skript napočítáme které bity jej ovlivňují. Stejně tak můžeme vzít pro každý bit řetězce, které on ovlivňuje a ty napsat po sloupečkách do matice a řešit soustavu lineárních rovnic $Ax = b$, kde b je vektor samých jedniček rozměru n , A je vyrobená matice rozměru $n \times m$ a $x = x_1 x_2 \dots x_n$ je vektor řešení. Je vidět, že každé řešení této soustavy odpovídá nějakému konfiguračnímu řetězci.

Už tedy umíme vyřešit otázku, zda existuje nějaký konfigurační řetězec a umíme i nějaký najít, pakliže umíme řešit soustavu rovnic. Tu lze řešit například Gaussovou eliminační metodou v čase $O(n^2 m)$, kde matice je rozměru $n \times m$ (n řádků a m sloupců). Pro výpočet počtu řešení použijeme trochu lineární algebry. Matice A v soustavě je rozměru $n \times m$. Frobeniova věta říká, že má-li soustava $Ax = b$ řešení, pak řešení tvoří prostor dimenze $m - h(A)$, kde m je počet proměnných v soustavě a $h(A)$ je hodnost matice A .

Víme, že jsme v prostoru \mathbb{Z}_2^m a tam je jakýkoli podprostor dimenze k isomorfní s prostorem k -tic nad \mathbb{Z}_2 , tedy má právě 2^k prvků. Hodnost matice lze spočítat opět pomocí Gaussovy eliminační metody a počet řešení je tedy, má-li soustava nějaké, $2^{m-h(A)}$. Toto číslo může být velké a nemusí se vlézt do standardních 64-bitových datových typů. Slušelo by se tedy počítat počet řešení modulo nějaké velké prvočíslo, např. $10^9 + 7$.

Trochu intuitivněji a méně formálně, proč je tohle pravda: Pokud má soustava $Ax = 0$ nějaké řešení x^* , pak pro libovolný násobek α je $\alpha \cdot x^*$ také řešení soustavy. Dá se ukázat, že pokud má soustava $Ax = b$ řešení, musí mít řešení i soustava $Ax = 0$ a navíc je počet řešení těchto dvou soustav stejný. Nyní, má-li soustava nějaké dvě řešení, např. x^* a y^* , pak i jejich součet $x^* + y^*$ je

řešením soustavy. V našem případě je sčítání bitových řetězců definováno po složkách, tedy např. $1010 + 1100 = 0110$. Tedy odpovídá to bitové operaci XOR. Z faktů výše plyne, že i libovolná *lineární kombinace* řešení dá také řešení. Tedy pro libovolné α, β a řešení x^*, y^* je také $\alpha x^* + \beta y^*$ řešení. Při řešení soustavy se jednou dostaneme do horního stupňovitého tvaru a ve své podstatě tolik, kolik je vedlejších sloupců, tolik existuje nezávislých řešení. Toto číslo je totéž jako $k = m - h(A)$. Tedy máme nějaká nezávislá řešení $z_1, z_2, z_3, \dots, z_k$. Každé další řešení se dostane jako lineární kombinace těchto řešení z_i . Koeficienty lineární kombinace mohou být pouze 0 a nebo 1, tedy pro každý z k vektorů máme dvě možnosti, který koeficient použijeme, tedy všech řešení je celkem 2^k .