Modulární aritmetika, Malá Fermatova věta.

Matematické algoritmy (K611MAG)

Jan Přikryl, Miroslav Vlček

3. přednáška K611MAG
úterý 9. října 2012

verze: 2012-10-10 15:06

1 Úvod

Modulární aritmetika je aritmetikou na množině celých čísel $\mathbb{Z}$ v níž se čísla opakují po dosažení určité hodnoty n, již nazýváme **modul**.

Na rozdíl od běžných celočíselných operací se zde po každé operaci provede ještě *celočíselné dělení* modulem n a výsledkem operace je *zbytek* po tomto dělení.

Příklad 1. V modulární aritmetice modulo 7 mají čísla 8 a 71 shodné reprezentace, protože $8 \bmod 7 = 1$ a zároveň $71 \bmod 7 = 1$.

Celočíselná aritmetika v počítačích je modulární.

Příklad 2 (Aritmetika osmibitových čísel). 250+10 je v osmibitové aritmetice rovno 4 (tedy $260 \bmod 2^8$). 12-16 je v osmibitové aritmetice rovno 252 (což je $-4 \bmod 2^8$).

Praktické aplikace modulární aritmetiky:

- **přenos zpráv** – ochrana zpráv proti chybám, komprese, zajištění integrity, utajování,
- **výpočetní technika** – hašovací funkce, pseudonáhodná čísla, dvojková komplementární reprezentace celých čísel, aritmetika s VELKÝMI celými čísly.

2 Dělitelnost a kongruence

2.1 Dělitelnost

Na množině celých čísel $\mathbb{Z}$ mějme definována dvě čísla: a, b. Říkáme, že a **dělí** b, pokud existuje libovolné $c \in \mathbb{Z}$ takové, že $b = ac$.

Pro zkrácený zápis toho vztahu používáme symbol $a|b$.

Pro **společný dělitel** c čísel a a b platí, že $c|a$ a zároveň $c|b$.

Největší společný dělitel

Číslo d označujeme jako **největšího společného dělitele** čísel a a b a zapisujeme $d = \gcd(a, b)$, pokud platí, že

- číslo d je společný dělitel a a b, a
- pokud existuje $c \neq d$ takové, že $c|a$ a zároveň $c|b$, pak také $c|d$.

Číslo $\gcd(a, b)$ je tedy největším kladným celým číslem jež dělí jak a, tak i b, s výjimkou $\gcd(0, 0) = 0$.

2.2 Kongruence

Uvažujme libovolný modul n takový, že $n \in \mathbb{N}$ a zvolme si dvě celá čísla $a, b \in \mathbb{Z}$.

Definice 3. Pokud v modulární aritmetice platí, že $a \bmod n$ a $b \bmod n$ jsou si rovny (mají stejný zbytek po dělení n), říkáme, že že *a je kongruentní s b modulo n* a zapisujeme

$$a \equiv b \,(\mathrm{mod}\, n).$$

Příklad 4. Je tedy $8 \equiv 71 \,(\mathrm{mod}\, 7)$, 8 je kongruentní s 71 modulo 7. Pozor na záporná čísla: $-1 \equiv 7 \,(\mathrm{mod}\, 8)$.

Označme si m onen zbytek po dělení $a \bmod n$ a $b \bmod n$. Bude potom platit, že

$$a = i \cdot n + m$$
$$b = j \cdot n + m$$

pro nějaká $i, j \in \mathbb{Z}$. V příkladu, uvedeném výše, je

$$8 = 1 \cdot 7 + 1$$
$$71 = 10 \cdot 7 + 1$$
$$-6 = -1 \cdot 7 + 1.$$

Příklad 5. Mějme abecedu velkých písmen české abecedy, $\{\mathsf{A}, \mathsf{Á}, \mathsf{B}, \ldots, \mathsf{Z}, \mathsf{Ž}\}$, reprezentovanou numerickými hodnotami $\{1, 2, \ldots, 42\}$. Nad touto abecedou provádíme všechny matematické operace modulárně, s modulem 42.

V takové modulární aritmetice jsou si rovny například reprezentace celých čísel -41, 43 a 320328919, protože zbytek po dělení 42 je vždy 1:

$$\begin{aligned} -41 \equiv 43 \,(\mathrm{mod}\ 42) &\quad\Leftrightarrow\quad -41 = 43 + 42 \cdot (-2), \\ -41 \equiv 320328919 \,(\mathrm{mod}\ 42) &\quad\Leftrightarrow\quad -41 = 320328919 + 42 \cdot (-7626880), \\ 320328919 \equiv 43 \,(\mathrm{mod}\ 42) &\quad\Leftrightarrow\quad 320328919 = 43 + 42 \cdot 7626878. \end{aligned}$$

Znak A může tedy reprezentovat libovolné z čísel -41, 43 a 320328919.

Třída kongruence

Množinu všech celých čísel, která jsou kongruentní s nějakým m modulo n je zvykem nazývat **třída kongruence** a zapisovat ji $\overline{m}$, bez uvedení modulu kongruence, nebo $[m]_n$.

Příklad 6. Například číslo 3 v modulu 5 může zastupovat i všechna čísla s ním kongruentní $(\dots, -7, -2, 3, 8, 13, \dots)$. V textech bude tato třída kongruence označována jako $[3]_5$ nebo jako $\overline{3}$.

Vlastnosti kongruence modulo n umožňují počítat pouze se zbytky po dělení tímto modulem a výsledek pak zobecnit na všechna čísla.

3 Vlastnosti čísel v modulární aritmetice

Modulární aritmetika je uzavřená vůči operacím sčítání a násobení:

$$\begin{aligned}[a]_n + [b]_n &= [a+b]_n,\\ [a]_n - [b]_n &= [a-b]_n,\\ [a]_n \cdot [b]_n &= [a\cdot b]_n.\end{aligned}$$

Příklad 7. V aritmetice modulo 7 by mělo platit $[2]_7 + [6]_7 = [1]_7$. Pro $9 \in [2]_7$ a $-1 \in [6]_7$ je výsledek $9 - 1 = 8 \in [1]_7$. Zcela obecně je

$$\begin{aligned}a &= i \cdot 7 + 2,\\ b &= j \cdot 7 + 6\end{aligned}$$

a potom

$$a + b = (i \cdot 7 + 2) + (j \cdot 7 + 6) = (i + j) \cdot 7 + 8 = (i + j + 1) \cdot 7 + 1.$$

Podobně zkuste v aritmetice modulo 7 ověřit $[2]_7 \cdot [6]_7 = [5]_7$.

Sčítání a násobení v modulární aritmetice je komutativní a asociativní:

$$\begin{aligned}[a]_n + [b]_n &= [b]_n + [a]_n,\\ [a]_n \cdot [b]_n &= [b]_n \cdot [a]_n,\\ ([a]_n + [b]_n) + [c]_n &= [a]_n + ([b]_n + [c]_n)\,,\\ ([a]_n \cdot [b]_n) \cdot [c]_n &= [a]_n \cdot ([b]_n \cdot [c]_n)\,.\end{aligned}$$

Pro sčítání a násobení v modulární aritmetice existuje identita, pro sčítání i inverze:

$$\begin{aligned}[0]_n + [a]_n &= [a]_n,\\ [a]_n + [-a]_n &= [0]_n,\\ [1]_n \cdot [a]_n &= [a]_n.\end{aligned}$$

Příklad 8 (Dva příklady). V modulární aritmetice modulo 7 je $28 \in [0]_n$ a $15 \in [1]_n$. Pro jejich součet platí $(28+15) \bmod 7 = 43 \bmod 7 = 1$.[0.2ex] V modulární aritmetice modulo 3 je $10 \in [1]_n$ a $8 \in [2]_n$. Pro jejich součin platí $(10 \cdot 8) \bmod 3 = 80 \bmod 3 = 2$.

Jak dopadne součet 57 a -73 v aritmetice modulo 8?

Modulární krácení

Pokud

$$a \cdot d \equiv b \cdot d \pmod n$$

obecně neplatí, že také

$$a \equiv b \pmod n$$

Příklad 9. Kongruenci $8 \equiv 14 \pmod 6$ sice můžeme rozložit na $4 \cdot 2 \equiv 7 \cdot 2 \pmod 6$, ale rozhodně neplatí, že $4 \equiv 7 \pmod 6$. Všimněte se ale, že

$$\begin{aligned} 8 &\equiv 14 \pmod 6 \\ 4 \cdot 2 &\equiv 7 \cdot 2 \pmod{3 \cdot 2} \\ 4 &\equiv 7 \pmod 3. \end{aligned}$$

Zdá se tedy, že v případech, kdy je modul kongruence také dělitelný d, je třeba vykrátit d nejenom z ekvivatelntních tříd, ale i z modulu kongruence.

Jsou dvě varianty

1. Pro $\gcd(d, n) = 1$ je opravdu $a \equiv b \pmod n$.
2. Pro $\gcd(d, n) = k, k > 1$ je $d = k \cdot x$ a $n = k \cdot y$ a kongruence se postupně změní na

$$\begin{aligned} akx &\equiv bkx \pmod{ky} \\ ax &\equiv bx \pmod y \\ a &\equiv b \pmod y. \end{aligned}$$

Příklad 10 (Modulární krácení pro d a n nesoudělná). Pro $170 \equiv 35 \pmod 3 \longrightarrow 5 \cdot 34 \equiv 5 \cdot 7 \pmod 3$ je $34 \equiv 7 \pmod 3$, protože 3 a 5 jsou nesoudělná čísla.

Příklad 11 (Modulární krácení pro obecné $d \neq 0$). Z kongruence $10 \equiv 6 \pmod 4 \longrightarrow 5 \cdot 2 \equiv 3 \cdot 2 \pmod{2 \cdot 2}$ plyne $5 \equiv 3 \pmod 2$.

Co vyjde pro $10 \equiv 6 \pmod 3$?

4 Malá Fermatova věta

Definice 12. Pro $a \in \mathbb{Z}$ a prvočíslo $p \in \mathbb{N}$ takové, že $p \nmid a$ platí

$$a^{p-1} \equiv 1 \pmod p$$

a v alternativním tvaru

$$a^p \equiv a \pmod p$$

Ve skutečnosti je $a^{\phi(n)} \equiv 1 \pmod n$, kde $\phi(n)$ je takzvaná **Eulerova funkce**.

Malá Fermatova věta je základním stavebním kamenem algoritmu generování šifrovacího klíče asymetrické šifry RSA. Je také nutnou podmínkou pro prvočísla a základním kamenem **Fermatova testu prvočíselnosti**.

Jak už bylo naznačeno výše, na množině celých čísel nelze použít operaci dělení tak, jak ji známe z množiny reálných čísel – celá čísla umíme dělit pouze se zbytkem po dělení. Pokud řešíme v $\mathbb{R}$ rovnici

$$5x = 6$$

můžeme proměnnou x osamostatnit tak, že levou i pravou stranu rovnice vynásobíme **inverzním prvkem** čísla 5 vůči operaci násobení, tedy zlomkem $1/5$.

V modulární aritmetice, definované na množině celých čísel, ale se zlomky pracovat neumíme. Přesto i kongruenci

$$5x \equiv 6 \pmod{11}$$

umíme převést na tvar

$$x \equiv 6 \cdot 5^{-1} \pmod{11},$$

kde 5^{-1} je tak zvaná multiplikativní inverze čísla 5 v aritmetice modulo 11.

Multiplikativní inverze

Pro $a \in \mathbb{Z}$ a $n \in \mathbb{N}$ je celé číslo x **multiplikativní inverzí** a, pokud splňuje podmínku

$$a \cdot x \equiv 1 \pmod{n}. \tag{1}$$

Pro **nejmenší multiplikativní inverzi** platí, že x je nejmenší možnou kladnou multiplikativní inverzí k a a označujeme ji a^{-1}.

Z Malé Fermatovy věty přitom plyne, že

$$a^{-1} \equiv a^{p-2} \pmod{p} \tag{2}$$

pro $a \in \mathbb{Z}$ a prvočíselná $p \in \mathbb{N}$ taková, že $p \nmid a$.

Příklad 13 (Výpočet inverze). Chceme spočítat a^{-1} pro $n = 11$ a $a = -3$. Volíme postupně $x = 1, 2, \ldots$, první kladné číslo x splňující vztah (1) je $x = 7$: $-3 \cdot 7 \equiv 1 \pmod{11}$.

Příklad 14 (Výpočet inverze pomocí Malé Fermatovy věty). Použitím Malé Fermatovy věty (2) máme $a^{-1} \equiv (-3)^{11-2} \pmod{11}$, tedy $a^{-1} \equiv -19683 \pmod{11}$ což je to samé, jako $a^{-1} \equiv 7 \pmod{11}$ protože jde o stejnou třídu kongruence.

Zkuste si to nyní sami pro $n = 7$ a $a = 5$.

5 Příklady

5.1 Opice a kokosy

Na pustém ostrově ztroskotají tři námořníci. Jediná potrava, kterou během dne našli, je hromada kokosových ořechů.

V noci se první námořník probudí, spravedlivě rozdělí hromadu na tři díly, přičemž jeden kokos zbyde – ten dostane opice. Svou třetinu námořník ukryje, zbytek navrší zpátky a jde zase spát. Postupně hromadu stejným způsobem „třetina pro mne, jeden kokos opici, zbytek vrátit“ zmenší jeho oba druhové.

Ráno si hromadu rozdělí na třetiny, opět zbyde jeden kokos, ten dostane opice.

Kolik musí být v původní hromadě kokosů, aby to fungovalo?

První námořník začíná s hromadou obsahující $n \equiv 1 \pmod{3}$ kokosových ořechů.

Druhý námořník dělil hromadu s

$$m_1 = \frac{2(n-1)}{3} \equiv 1 \pmod{3}$$

ořechy, třetí námořník přerozděloval

$$m_2 = \frac{2(m_1 - 1)}{3} \equiv 1 \,(\mathrm{mod}\, 3)$$

ořechů a ve zbylé hromadě jich muselo zůstat

$$m_3 = \frac{2(m_2 - 1)}{3} \equiv 1 \,(\mathrm{mod}\, 3).$$

Hodnotu m_3 spočteme jako

$$m_3 = \frac{2}{3}m_2 - \frac{2}{3} = \cdots = \frac{8}{27}n - \frac{38}{27} \equiv 1 \,(\mathrm{mod}\, 3)$$

a rovnici v modulární aritmetice řešíme pro n

$$8n - 38 \equiv 27 \,(\mathrm{mod}\, 81) \Rightarrow 8n \equiv 65 \,(\mathrm{mod}\, 81).$$

Dělit osmi nemůžeme, můžeme ale násobit multiplikativní inverzí (pro jejíž výpočet nelze použít Fermatovu větu – proč?):

$$n \equiv 8^{-1} \cdot 65 \equiv 71 \cdot 65 \equiv 79 \,(\mathrm{mod}\, 81).$$

Nejmenší počet kokosů v hromadě je tedy 79 (ale může být i 160, 241, ...).

5.2 Kontrolní součty

Má ho každá kniha, identifikuje zemi či region původu, nakladatele a vydání. Existuje ve verzi ISBN-10 a ISBN-13. Na poslední pozici každého ISBN je *kontrolní cifra*.

Příklad 15 (Výpočet kontrolní cifry ISBN-10). Mějme ISBN 0-552-13105-9. Kontrolní cifra ISBN-10 se počítá v modulu 11, pro případ zbytku 10 se použije znak X. Kontrolní součet je $0 \cdot 10 + 5 \cdot 9 + 5 \cdot 8 + 2 \cdot 7 + 1 \cdot 6 + 3 \cdot 5 + 1 \cdot 4 + 0 \cdot 3 + 5 \cdot 2 + 9 \cdot 1 = 143 \bmod 11 = 9$. Uvedené ISBN je opravdu platné.

Což takhle 80-85609-70-3?

5.3 Čísla bankovních účtů

Číslo bankovního účtu v ČR má tvar 123456-1234567890/1234, kde první a druhá část čísla účtu jsou chráněny proti překlepům opět algoritmem váženého ciferného součtu mod 11. Kontroluje se pouze tzv. předčíslí a vlastní číslo účtu, váhy jednotlivých číslic v předčíslí jsou zleva $10, 5, 8, 4, 2, 1$ a váhy cifer v čísle účtu jsou zleva $6, 3, 7, 9, 10, 5, 8, 4, 2, 1$. Těmito vahami se vynásobí jednotlivé číslice předčíslí, resp. čísla účtu, výsledek se sečte a určí se zbytek po dělení 11. Pokud tento zbytek vyjde 0, jedná se o korektní číslo účtu.

Příklad 16 (Kontrola čísla účtu). Mějme číslo účtu 19-2000145399/0800, jež pro účely kontroly přepíšeme na 000019-2000145399/0800. Kontrolní součet první části je $0 \cdot 10 + 0 \cdot 5 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 9 \cdot 1 = 11 \bmod 11 = 0$. Kontrolní součet druhé části je $2 \cdot 6 + 0 \cdot 3 + 0 \cdot 7 + 0 \cdot 9 + 1 \cdot 10 + 4 \cdot 5 + 5 \cdot 8 + 3 \cdot 4 + 9 \cdot 2 + 9 \cdot 1 = 121 \bmod 11 = 0$.

5.4 Rodné číslo

Jednoznačný identifikátor občanů ČR a SR obsahující údaj o datumu narození, pohlaví a do roku 2004 i lokalitě porodnice.

Příklad 17 (Výpočet kontrolní cifry). Muž narozen 22. února 1959, rozlišující trojčíslí 177 (Zlín?). Poslední cifra rodného čísla zajišťuje dělitelnost ciferného součtu jedenácti, musí mít proto hodnotu $590222177 \bmod 11 = 6$. Odpovídající rodné číslo má tvar 590222/1776.

O kohopak asi jde?

5.5 Pseudonáhodná čísla

Pro různé druhy stochastických simulací a Monte Carlo výpočtů potřebujeme vhodný zdroj náhodných čísel z určitého rozdělení pravděpodobnosti – potřebujeme generovat sekvenci hodnot, kde hodnota prvku x_{k+1} nezávisí na žádném z prvků $x_0, x_1, \ldots, x_k$.

Takovou sekvenci v počítači vyrobit neumíme, umíme se jí ale docela věrně na počítači pro omezený počet hodnot nabpodobit pomocí generátoru pseudonáhodných čísel, jenž většinou generuje čísla z rovnoměrného rozdělení $U(0, 1)$. Ostatní rozdělení lze potom různými technikami simulovat pomocí přepočtu hodnot z rozdělení rovnoměrného.

Jednou z možností, jak pseudonáhodná čísla v $U(0, 1)$ generovat, je **lineární kongruentní generátor** *(LCG, Linear Congruence Generator)*.

Příklad 18 (Jak funguje LCG). Uživatel zvolí x_0 (pevné nebo třeba odvozené od aktuálního času). Potom $x_{k+1} = (a \cdot x_k + b) \bmod m$, kde a, b a m jsou zvolené parametry určující kvality generátoru. Jedna z možných voleb je třeba $a = 1664525$, $b = 1013904223$ a $m = 2^{32}$.

LCG jsou *velmi citlivé na volby parametrů*. Pokud dodržíme jisté předpoklady, generátor pracuje s periodou m, ale i to je v mnoha případech statistických výpočtů (například u vícerozměrné Monte Carlo integrace) žalostně málo.

5.6 Aritmetika velkých čísel

Registry v dnešních procesorech jsou většinou 32 nebo 64 bitové:

- největší binární číslo, s nímž počítač dokáže *pohodlně* pracovat, je tedy 2^{32} respektive 2^{64},
- největší binární číslo, jež můžeme reprezentovat v 1GB operační paměti, je $2^{1099511627776}$ ... jak rychle s ním ale budeme schopni počítat?

Jak se ale algoritmy typu RSA efektivně vypořádávají se sčítáním či násobením celých čísel v aritmetice velkých modulů (třeba 340282366920938463463374607431768211507)? Jak provádět operace s třídami čísel, která se do paměti počítače prostě nevejdou?