Jeden za osmnáct a druhý bez dvou za dvacet aneb Záporné cifry v zápisu čísel

Balková Ľubomíra[*], Bekrová Martina[†], Lukeš Jakub[‡]

Abstrakt

Valná většina z nás zapisuje přirozená čísla v desítkové (decimální) soustavě s ciframi od 0 do 9. Pro počítače je z technických důvodů výhodné reprezentovat čísla v dvojkové (binární) soustavě s ciframi 0 a 1. Ač jsou tyto dva způsoby nejrozšířenější, nejsou jediné možné a v četných úlohách nejsou ani nejvýhodnější. Podíváme se, co se stane, když v zápisech čísel připustíme i záporné cifry.

1 Historie užití záporných cifer v zápisu přirozených čísel

Už od první třídy základní školy se učíme počítat s přirozenými čísly $1, 2, 3, \ldots$ a zapisujeme je výlučně v desítkové (decimální) soustavě s ciframi od 0 do 9. Víme, že 923 je decimálním zápisem čísla $9 \cdot 100 + 2 \cdot 10 + 3 \cdot 1$. Připusťme nyní i záporné cifry, např. z množiny $\{-9, \ldots, 0, \ldots, 9\}$. **Pro přehlednost budeme záporné cifry v zápisu značit pruhem:** $\overline{2} = -2$. Pak číslo osmnáct má dva různé zápisy: 18 a $2\overline{2}$, kde $2\overline{2}$ je zápis čísla $2 \cdot 10 - 2 \cdot 1$, což je skutečně osmnáct. Soustavám, které připouští vícero zápisů stejného čísla, říkáme *redundantní*. Neznamená to ale, že by soustavy se zápornými ciframi musely být vždy redundantní. Například v *balancované desítkové soustavě*, kde cifry bereme z množiny $\{-4, \ldots, 0, \ldots, 5\}$, má každé číslo jediný zápis. Tuto soustavu oceníme při násobení, vystačíme totiž s malou násobilkou (do 5×5 [1]), jak ilustruje následující příklad.

$$14\overline{1} \times 1\overline{3} = \begin{array}{cccc} & 1 & 4 & (-1) \\ & & 1 & (-3) \\ \hline & (-4) & (-2) & 3 \\ 1 & 4 & (-1) & \\ \hline 1 & 0 & (-3) & 3 \end{array}$$

Obrázek 1: Uvažujme součin $139 \cdot 7 = 973$. Zapíšeme-li 139 a 7 v balancované desítkové soustavě, máme $14\overline{1}$ a $1\overline{3}$ (protože $1 \cdot 100 + 4 \cdot 10 + (-1) \cdot 1 = 139$ a $1 \cdot 10 + (-1) \cdot 3 = 7$). Násobení provádíme analogickým způsobem jako v klasické desítkové soustavě. Sami ověřte, že zápis $10\overline{3}3$, který vyšel při násobení v balancované soustavě, odpovídá hodnotě 973. A vyzkoušejte si těžší příklad $136 \cdot 257$.

První, kdo si uvědomil výhody balancované decimální soustavy, byl anglický profesor matematiky John Colson (1680 – 1760), který využíval záporných cifer k urychlení aritmetických operací, hlavně násobení (viz. článek [2]), jež se v jeho době provádělo ručně. Colson nebyl jen dobrým matematikem. Spíše je známý jako překladatel (angličtina, latina, francouzština, italština).

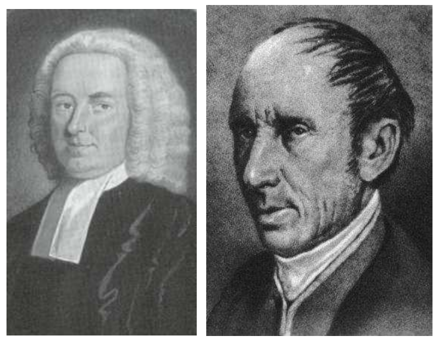
[*]Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze

[†]Gymnázium Trutnov

[‡]Gymnázium Českolipská, Praha 9

[1]Pro násobení používáme občas symbol $\cdot$ a občas symbol $\times$. Oba symboly značí to samé a střídavé použití se řídí jen naším vkusem.

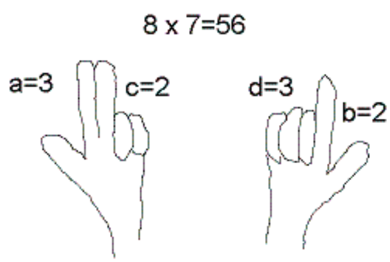
Jeho nejvýznamnějším počinem je překlad Newtonových Principií z latiny do angličtiny, cenným přínosem jsou zejména jeho vysvětlující komentáře. Zasadil se také o zpřístupnění počtů ženám, které neměly v Anglii 18. století právo na vzdělání.



Obrázek 2: John Colson a Augustin Louis Cauchy

Také vynikající francouzský matematik Augustin Louis Cauchy (1789 – 1857) [2] využíval záporných cifer, a to s cílem předcházet chybám ve výpočtech (viz. článek [3]). O jeho charakteru sice nekolují nejlepší zvěsti – jsou známé případy, kdy si přivlastnil objevy jiných matematiků– jeho přínos matematice je ale nepopiratelný. Učebnice Cours d'analyse (1821), v níž sepsal své přednášky na prestižní Ecole polytechnique, je dodnes základem vysokoškolské matematické analýzy. Byl také plodným autorem matematických článků. Pyšní se 789 publikacemi, více jich v jeho době měli na svém kontě pouze Arthur Cayley a Leonhard Euler.

Historie užití záporných cifer má ovšem ještě hlubší kořeny. Ač nevědomky, využívali jejich výhod již středověcí obchodníci. Dochovaly se záznamy o tzv. cikánské násobilce, která umožňuje pomocí prstů násobit do 9×9. Její princip pochopíte z následujícího obrázku.



Obrázek 3: Pokud chceme násobit 8×7, zeptáme se: „Osm a kolik je deset¿' „A dva." Skrčíme dva prsty na první ruce ($c = 2$). To samé uděláme s číslem sedm. Schováme tedy tři prsty na druhé ruce ($d = 3$). Na pozici desítek napíšeme součet vztyčených prstů ($a + b = 3 + 2 = 5$) a na pozici jednotek napíšeme součin skrčených prstů ($c \cdot d = 2 \cdot 3 = 6$). A obdržíme správný výsledek 56.

Bystrý čtenář si snadno rozmyslí, že algoritmus je použitelný jen násobení čísel, která jsou obě větší než 5, a že funguje díky následujícím rovnostem (a také si všimne, že při násobení některých čísel narazíme na jistá úskalí!)

$$\begin{aligned}(10-c)(10-d) &= 100-(c+d)10+cd \\ &= 10(10-c-d)+cd \\ &= 10(a+b)+cd.\end{aligned}$$

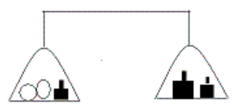
[2] Zmiňme pro zajímavost, že Cauchy pobýval krátce v Praze (1833 – 35). Dostal se sem jako učitel vnuka krále Karla X., který byl svržen z trůnu při červencové revoluci roku 1830 a byl nucen opustit Francii.

Naše historické putování zakončíme zmínkou o římských číslicích. Myšlenka záporných cifer se totiž objevuje i v římském zápisu. Předchází-li nižší cifra vyšší, znamená to odečítání: $IV = 5-1$ nebo $XC = 100-10$. Pro přesnost dodejme, že takové „odečítací“ zápisy jsou až vymožeností středověku. Římané sami psali 4 jako $IIII$ a 90 jako $LXXXX$.

2 Balancovaná ternární soustava – vážení a peníze

Trojková (ternární) soustava nemá praktické uplatnění – nepočítáme s ní v běžném životě a ani počítače s ní nepracují. [3] Přesto se k ní vážou dvě úlohy, které stojí za zmínku (detaily a další úlohy najdete v [4]).

1) Jakou nejmenší sadu závaží je třeba zvolit, chceme-li na rovnoramenných vahách vážit předměty o hmotnostech $1, 2, \ldots, 40$ kg?



Rozmysleme si nejprve, že tři závaží nestačí. Kdybychom měli 3 závaží o hmotnostech A, B, C, pak moci zvážit předmět o hmotnosti m znamená najít čísla $a, b, c \in \{-1, 0, 1\}$ taková, že

$$m = a \cdot A + b \cdot B + c \cdot C,$$

kde

- $a = 1$ znamená, že závaží hmotnosti A je na opačné misce váhy než vážený předmět,
- $a = -1$ znamená, že závaží hmotnosti A je na stejné misce váhy jako předmět,
- $a = 0$ znamená, že závaží hmotnosti A jsme k vyvážení předmětu nepoužili.

Analogická je role b a c. Když dosadíme za a, b, c všechny možné trojice z $-1, 0$ a 1 (např. $a = 0,\ b = 1,\ c = -1$ nebo $a = 1,\ b = 1,\ c = -1$ atd.), dostaneme nejvýše $3 \cdot 3 \cdot 3 = 27$ různých hodnot pro m, což je méně než 40.

Řešením úlohy je sada závaží hmotností $1, 3, 9$ a 27 kg. Ověřte sami, že každé číslo m od 1 do 40 se dá psát ve tvaru

$$m = a \cdot 1 + b \cdot 3 + c \cdot 9 + d \cdot 27,$$

kde $a, b, c, d \in \{-1, 0, 1\}$. Například $5 = (-1) \cdot 1 + (-1) \cdot 3 + 1 \cdot 9 + 0 \cdot 27$.

Platí dokonce následující rovnost, z níž správnost řešení přímo plyne (omlouváme se za nefér hru – vypůjčíme si totiž na chvíli spíše vysokoškolské značení):

$$\{a_k 3^k + a_{k-1} 3^{k-1} + \cdots + a_1 3^1 + a_0 3^0 \mid a_i \in \{-1, 0, 1\}\} = \tag{1}$$
$$= \{-\tfrac{3^{k+1}-1}{2}, \ldots, 0, \ldots, \tfrac{3^{k+1}-1}{2}\}.$$

Tedy dosadíme-li za $k = 3$, dostaneme

$$\{a_3 3^3 + a_2 3^2 + a_1 3^1 + a_0 3^0 \mid a_i \in \{-1, 0, 1\}\} = \{-40, \ldots, 0, \ldots, 40\}.$$

Vidíme proto, že pomocí cifer $-1, 0, 1$ a mocnin trojky $1, 3, 9, 27$ vyrobíme jakékoliv číslo od -40 do 40.

2) Rozmyslete si, že ze vztahu (1) plyne i následující tvrzení: Představte si, že máme bankovky v hodnotách mocnin tří, tj. $1, 3, 9, 27, 81, \ldots$, a že máme v peněžence od každé bankovky jeden kus. Kdybychom pozvali instalatéra a on si řekl za opravu topení o jakoukoliv částku, byli bychom mu schopni zaplatit za předpokladu, že i on by měl od každé bankovky jeden kus.

[3] V počátcích počítačové historie existovaly počítače reprezentující čísla v ternární soustavě.

3 Redundantní aritmetika

Opusťme nyní soustavy, ve kterých jsou zápisy čísel jednoznačné, a přesvědčme se nejprve, že existence více různých zápisů stejného čísla může přinést jisté výhody.

3.1 Binární soustava

Jak jsme zmínili v abstraktu, počítače používají binární soustavu s ciframi 0 a 1. V takové soustavě je násobení vlastně jen opakovaným sčítáním. Spočtěme součin čísel 11 a 5, které mají binární zápisy 1011 (protože $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 11$) a 101 (protože $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$). Snadno si algoritmus vymyslíte, je podobný jako v decimální soustavě.

$$1011 \times 101 \quad = \quad \begin{array}{cccccc} & & 1 & 0 & 1 & 1 \\ & & & 1 & 0 & 1 \\ \hline & & 1 & 0 & 1 & 1 \\ & 0 & 0 & 0 & 0 & \\ 1 & 0 & 1 & 1 & & \\ \hline 1 & 1 & 0 & 1 & 1 & 1 \end{array}$$

Dostáváme výsledek 1100111, což je zápis čísla $1{\cdot}2^5{+}1{\cdot}2^4{+}0{\cdot}2^3{+}1{\cdot}2^2{+}1{\cdot}2^1{+}1{\cdot}2^0 = 55$, tedy správná hodnota. Navíc pozorujeme, že rychlost této operace je přímo úměrná počtu jedniček v binárním zápisu násobitele (v našem případě 101). Právě tolik sčítání totiž musíme provést. Pravděpodobnost výskytu nuly v klasickém binárním zápisu je 1/2. Připusťme nyní v binární soustavě cifry $-1, 0, 1$. Tím se stává redundantní, protože například číslo 15 lze zapsat jako 1111 a také jako $1000\overline{1}$ (vskutku $1{\cdot}2^4{+}0{\cdot}2^3{+}0{\cdot}2^2{+}0{\cdot}2^1{+}(-1){\cdot}2^0 = 15$). Když z možných zápisů vybereme ten s maximálním počtem nul (rozmyslete si, jakým algoritmem to lze provést), pak pravděpodobnost výskytu nuly zvýšíme na 2/3. Z tohoto důvodu je výhodné násobení v redundantní binární soustavě. Poznamenejme, že nejčastěji používaná šifrovací metoda RSA je založena na násobení čísel velkých řádově 10^{100}.

3.2 Avizienisův algoritmus

Kromě násobení je nespornou výhodou redundantních soustav také možnost paralelního sčítání. Při klasickém sčítání se objevuje přenos, který znemožňuje provádět sčítání na každé pozici nezávisle na předchozích. Sami si napište na kus papíru algoritmus součtu 9999 a 1 a uvědomte si, jak si v každém kroku musíte pamatovat jedničku. Francouzský informatik Christophe Mazenc dokázal, že každý algoritmus pro paralelní sčítání vyžaduje redundantní soustavu. S prvním takovým algoritmem přišel v roce 1961 litevský matematik Algirdas Avizienis [1]. Jeho algoritmus funguje pro báze $b \geq 3$ a cifry od $-a$ do a, kde $a \leq b-1$ a $2a \geq b+1$. Jednotlivé cifry součtu $s_{n+1}s_n \dots s_1 s_0$ čísel $x_n \dots x_1 x_0$ a $y_n \dots y_1 y_0$ se získají takto $s_i = w_i + t_i$, kde pro $i = 0, 1, \dots, n$

$$t_{i+1} = \begin{cases} -1, & \text{když } x_i + y_i \leq -a, \\ +1, & \text{když } x_i + y_i \geq a, \\ 0, & \text{když } -a < x_i + y_i < a, \end{cases}$$

$$w_i = x_i + y_i - b t_{i+1},$$

$$t_0 = 0 = w_{n+1}.$$

Takové sčítání je vskutku paralelní, pokud máme k dispozici tolik procesorů, že každá pozice může být zpracována jedním z nich. Proběhne ne jeden, nýbrž tři paralelní kroky. Nejprve napočteme naráz všechna t_{i+1}. Stačí nám k tomu znát hodnoty x_i a y_i, speciálně si nepotřebujeme pamatovat x_j a y_j pro $j < i$. Poté napočítáme naráz všechna w_i, k čemuž nám stačí znalost x_i, y_i a v předchozím kroku napočteném t_{i+1}. Na závěr spočteme opět v jediném kroku všechna s_i.

Uvažujeme-li bázi $b = 10$, pak nejmenší rozsah cifer a, který vyhoví výše uvedeným podmínkám $a \leq 9$ a $2a \geq 11$, je $a = 6$. Součet čísel 2389 a 2672, která mají v redundantní decimální soustavě s ciframi od -6 do 6 po řadě zápisy $3\overline{6}\overline{1}\overline{1}$ a $3\overline{3}\overline{3}2$, se pomocí tohoto algoritmu vypočítá následovně:

i	4	3	2	1	0
x_i		3	-6	-1	-1
y_i		3	-3	-3	2
$x_i + y_i$		6	-9	-4	1
t_{i+1}		1	-1	0	0
w_i	0	-4	1	-4	1
s_i	1	-5	1	-4	1

Výsledek $1\overline{5}1\overline{4}1$ je správný, neboť $2389+2672 = 5061$ a $1\cdot10^4+(-5)\cdot10^3+1\cdot10^2+(-4)\cdot10^1+1\cdot10^0 = 5061$.

Pro použití Avizienisova algoritmu potřebujeme převádět čísla z desítkové soustavy do redundantní desítkové s ciframi od -6 do 6 a naopak (hovoříme o *konverzi* čísel). Snadno nahlédneme jak postupovat podle následujícího příkladu.

- Chceme-li zapsat 2389 v redundantní soustavě, v prvním kroku přičteme 6666

  $$2389 + 6666 = 9055,$$

  v dalším kroku odečteme od posledních čtyř cifer součtu 6

  $$9-6=\mathbf{3} \quad 0-6=\mathbf{-6} \quad 5-6=\mathbf{-1} \quad 5-6=\mathbf{-1},$$

  výsledek: $3\overline{6}\overline{1}\overline{1}$.

- Chceme-li zapsat $3\overline{6}\overline{1}\overline{1}$ ve standardní desítkové soustavě, spočteme rozdíl kladné a záporné části

  $$3000 - 611 = 2389,$$

  výsledek: 2389.

Poznamenejme, že algoritmus pro paralelní sčítání existuje i v binární soustavě s ciframi $-1, 0, 1$.

3.3 Nevýhody záporných cifer

Zatím jsme viděli pouze výhody redundantních soustav (úspora času při násobení díky vyššímu počtu nul v zápisu i při sčítání díky paralelnímu algoritmu). Proč tedy osobní počítače založené na redundantní binární soustavě neexistují? Odpověď je jednoduchá: Záporné cifry mají i „drobné“ nevýhody. Kladou větší požadavky na paměť (zatímco pro reprezentaci čísel od -2048 do 2047 stačí při standardní binární reprezentaci 12 bitů, v redundantní soustavě s ciframi $-1, 0, 1$ je potřeba 24 bitů). Nejednoznačnost zápisu způsobuje pomalejší porovnávání čísel. Navíc i konverze je časově náročná (odpovídá náročnosti klasického – neparalelního – sčítání). Výhoda redundantní aritmetiky se projeví ve chvíli, kdy je nutné provést velká sčítání či násobení. Proto v praxi existují speciální počítače, které neprovádí operace pomalé v redundantní aritmetice, a také tzv. černé skříňky, které jsou součástí počítačů – vstupní a výstupní data se jim zadávají v klasickém formátu, ale výpočty provádějí v redundantní soustavě.

3.4 Budoucnost záporných cifer

První paralelní algoritmy se objevily koncem 70. let. Poté se dostaly do centra zájmu mnoha informatiků. Valérie Ménissier vytvořila knihovnu pro práci s reálnými čísly s libovolnou přesností založenou na redundantní aritmetice (součást jazyka CAML). Naofumi Tagaki a spol. v roce 1985 vyvinuli výkonný násobič (aplikace v šifrování). Jean-Michel Müller a spol. [5] programují „černé skříňky“ pro výpočet hodnot funkcí sin, cos, tg, cotg, exp, log, arctg pouze s využitím paralelního sčítání a dělení násobky dvou (což je jen posouvání binární zlomkové čárky). Zdá se tedy, že se s redundantní aritmetikou budeme v budoucnu setkávat čím dál tím častěji.

Reference

[1] Avizienis, A.: *Signed-digit number representations for fast parallel arithmetic*, IRE Transactions on Electronic Computers **10** (1961), 389–400

[2] Colson, J.: *A short account of negativo-affirmative arithmetic*, Philosophical Transaction of the Royal Society **34** (1726), 161–173

[3] Cauchy, A. L.: *Sur les moyens d'éviter les erreurs dans les calculs*, Oeuvres complètes **1** (November 1940), 431–442

[4] Hayes, B.: *The third base*, American Scientist (November/December 2001), 490–494

[5] Müller, J.-M.: *Ordinateurs en quête d'arithmétique*, La Recherche **26** (1995), 90–96

Prográmky provádějící konverzi čísel mezi desítkovou a redundantní desítkovou soustavou s ciframi od -6 do 6 a paralelní sčítání podle Avizienisova algoritmu v Javascriptu v HTML kódu (od M. Bekrové) a v Pascalu (od J. Lukeše) najdete na kmlinux.fjfi.cvut.cz/~balkolub/stredniskoly