Niespotykane pozycje

*Piotr PIKUL**

* Wydział Matematyki i Informatyki, Uniwersytet Jagielloński

Pozycyjny system zapisywania liczb uchodzi za największe osiągnięcie matematyki w ogóle – arytmetyczny odpowiednik wynalezienia koła. Łatwość prowadzenia rachunków niewątpliwie popchnęła cywilizację do przodu. Tzw. *działania pisemne* są prawdopodobnie pierwszymi abstrakcyjnymi algorytmami, jakie współczesny człowiek poznaje w swoim życiu! Dla (zbyt) wielu ludzi może to być główne skojarzenie z hasłem „matematyka".

Użyłem tu bardziej rozpoznawalnego terminu – „wynalezienie koła", ale tak naprawdę chodzi o wynalezienie *osi*.

Pomimo tego wzniosłego wstępu Czytelnik może mieć wrażenie, że zapis pozycyjny nie kryje w sobie żadnych wielkich tajemnic. Mamy 10 symboli (cyfr) dla liczb od 0 do $9 = 10 - 1$ i kolejne wartości w zapisanym ciągu (czytając od prawej) mnożymy przez kolejne potęgi liczby 10 (podstawy systemu): 1, 10, 100 itd. Możemy wprawdzie wybrać inną podstawę niż powszechnie stosowane 10 i otrzymać nieco mylące zapisy, np. 101 oznaczające 5 w systemie binarnym (o podstawie 2) lub $210_3 = 21_{10}$ (liczba w indeksie dolnym oznacza podstawę systemu), ale co to za ciekawostka... Oprócz samej znakowej reprezentacji liczb, wraz ze zmianą podstawy zmieniają się także charakteryzacje podzielności. Na przykład w systemie szóstkowym podzielność jakiejś liczby przez 5 jest równoważna podzielności sumy jej cyfr przez 5. W systemie dziesiętnym działa tak trójka i dziewiątka. Zawsze „największa cyfra" jest związana z cechą podzielności, ponieważ suma cyfr przy podstawie d przystaje do samej liczby modulo $(d-1)$: $\sum c_j d^j - \sum c_j 1^j = (d-1)\sum c_j(1 + \ldots + d^{j-1})$. Oznacza to, że przejście do sumy cyfr zachowuje nie tylko samą podzielność, ale ogólniej – resztę z dzielenia. Uniwersalny schemat występuje również dla k dzielącego d^n: aby sprawdzić podzielność przez k, sprawdzamy podzielność liczby złożonej z ostatnich n cyfr (czyli cyfr najmniej znaczących). „U nas" tę własność wspomina się głównie dla 2, 5, 4 i 8. Szczególnym przypadkiem jest także podzielność przez potęgi 10 (zauważmy, że nie ma tu znaczenia podstawa, czyli to, ile właściwie wynosi „10"). Istnieje więcej interesujących cech podzielności związanych z zapisem dziesiętnym (np. liczba jest podzielna przez 11, gdy różnica sum cyfr na parzystych i nieparzystych pozycjach jest podzielna przez 11) i pewnie jeszcze więcej przy innych podstawach, ale wcale nie o tym jest ten artykuł.

W starożytnym Egipcie, choć oficjalnie do zapisywania liczb nie używano systemu pozycyjnego, znano metody rachunkowe oparte na zapisie binarnym.

O różnych cechach podzielności w systemie dziesiętnym pisał Paweł Bieliński w Δ^{11}_{21}.

Wybór podstawy systemu zmienia też radykalnie sytuację zapisu liczb niecałkowitych. Jak wiadomo, rozwinięcie liczby wymiernej jest skończone wtedy i tylko wtedy, gdy mianownik dzieli d^n dla pewnego $n \geqslant 1$. W przeciwnym wypadku jest nie skończone, ale okresowe (od pewnego miejsca). Ponieważ podstawa dziesięć jest uboga w dzielniki pierwsze (choć mogło być gorzej), takie *podstawowe* liczby jak $\frac{1}{3}$ lub $\frac{1}{6}$ mają *brzydką* postać dziesiętną. W systemie szóstkowym małe ułamki proste przyjmują postać $\frac{1}{2} = 0{,}3_6$, $\frac{1}{3} = 0{,}2_6$, $\frac{1}{4} = 0{,}13_6$, $\frac{1}{5} = 0{,}(1)_6$, $\frac{1}{6} = 0{,}1_6$, $\frac{1}{7} = 0{,}(05)_6$. Nie mamy skończonego rozwinięcia $\frac{1}{5}$, ale być może dzielenie czegoś na pięć części byłoby o wiele mniej modne, gdybyśmy nie używali systemu dziesiętnego... Nie będziemy się tu jednak rozwodzić nad wyborem podstawy systemu. Uniwersalną wadą systemów o innej podstawie niż 10 jest problem z odczytywaniem (nazywaniem) tak zapisanych liczb – dziesiętny zapis zakorzenił się bardzo mocno w znacznej części współczesnych języków.

Poniżej 6 pierwszych cyfr rozwinięcia liczby π w systemach pozycyjnych od 2 do 12:

11,001001...$_2$
10,010211...$_3$
3,021003...$_4$
3,032322...$_5$
3,050330...$_6$
3,066365...$_7$
3,110375...$_8$
3,124188...$_9$
3,141592...$_{10}$
3,161507...$_{11}$
3,184809...$_{12}$

Zwolennicy stosowania systemu dwunastkowego (prawdopodobnie najbardziej praktycznego, łączącego wciąż niewielką liczbę cyfr i mnogość dzielników podstawy) mają swoje amerykańskie stowarzyszenie: `https://dozenal.org/` Znalazłoby się też pewnie stowarzyszenie na rzecz odrzucenia układu klawiatury QWERTY...

To było przypomnienie podstawowych faktów o *normalnych* systemach pozycyjnych, w których cyfry reprezentują możliwe reszty z dzielenia przez całkowitą podstawę $d > 1$, czyli wartości $0, 1, \ldots, d-1$. Liczby ujemne oznaczamy dodatkowym znakiem minus. Poza takimi prostymi uogólnieniami systemu dziesiętnego istnieją warianty nieco bardziej oryginalne, i to właśnie im się przyjrzyjmy.

Informatycy z pewnością kojarzą „kod uzupełnień do dwóch", powszechnie stosowany do reprezentacji liczb całkowitych w komputerach. Ustalamy liczbę cyfr (bitów) i najwyższy rząd uznajemy za ujemny. Na przykład dla 8 bitów liczba `10111111` jest równa $-128 + 63 = -65$ (pierwsza cyfra oznacza -2^7, a kolejne już $+2^6$, $+2^5$ itd. aż do $2^0 = 1$). Okazuje się, że standardowe, „pisemne" działania na takich liczbach funkcjonują zupełnie poprawnie tak długo, jak wynik mieści się w ograniczonym przedziale wartości (trudno o poprawny

wynik, kiedy nie da się go zapisać). Zauważmy, że aby zakodować liczbę $a < 0$, ustawiamy bit o wartości -2^{k-1} na 1, a pozostałe bity kodują wartość $a - (-2^{k-1}) = a + 2^{k-1}$. Tym samym całość liczby a zapisanej w kodzie uzupełnień, kiedy ją odczytamy w zwykłym systemie binarnym, będzie miała wartość $2^{k-1} + 2^{k-1} + a = 2^k + a$, czyli modulo 2^k (a tak działają pisemne działania na k bitach) nadal pracujemy z liczbą a zapisaną binarnie!

To tylko część argumentu za poprawnością działań w kodzie uzupełnień. Nie będziemy tu jednak rozpisywać się o własnościach *działań modulo*.

Co ciekawe, przy podstawie większej niż 2 to podejście już nie działa. Gdyby było inaczej, to rozważając zapis dziesiętny z trzema cyframi, przy czym trzecią cyfrą „ujemną”, mielibyśmy: $(-1) \cdot (-1) = 199_{\text{u10}} \cdot 199_{\text{u10}} = [39]601_{\text{u10}} = (-599)$! W ogólności liczbę $a < 0$ kodowalibyśmy teraz jako -10^{k-1} oraz $10^{k-1} + a$, co po odczytaniu z pominięciem minusa przy najstarszej cyfrze daje $2 \cdot 10^{k-1} + a$ i nie przystaje do a modulo 10^k, czego potrzebujemy, aby stosować standardowe działania dziesiętne. Aby uzyskać dziesiętny system uzupełnień pozbawiony tej wady, powinniśmy od standardowej interpretacji zapisu k-cyfrowego odejmować 10^k, jeśli wiodąca cyfra jest większa od 4. Przykładowe mnożenie wygląda następująco: $(-1) \cdot (-499) = 999_{\text{U10}} \cdot 501_{\text{U10}} = [500]499_{\text{U10}} = 499$.

Kod uzupełnień, pomimo swoich praktycznych zastosowań, w teorii jest „ułomny”, ponieważ opisuje jedynie skończony zakres liczb całkowitych. Okazuje się jednak, że koncepcję „ujemnych rzędów” można zaprząc do pracy na całym zbiorze liczb całkowitych. Pozostając przy dwóch cyfrach, możemy rozważyć *system minus dwójkowy* (zwany też *negabinarnym*), czyli taki, w którym nieparzyste rzędy są ujemne (rząd jedności ma numer 0, odpowiadając potędze zerowej) i nadal używamy cyfr 0 i 1. Początkowe liczby naturalne wyglądają w tym systemie następująco:

$\geqslant 0$	0,	1,	110,	111,	100,	101,	11010,	11011,	11000,	11001,	11110, …
< 0	11,	10,	1101,	1100,	1111,	1110,	1001,	1000,	1011,	1010,	…

Liczba cyfr rośnie jeszcze szybciej niż w systemie binarnym, gdyż tylko zapis nieparzystej długości koduje wartości dodatnie. Jeśli spojrzymy na zapis liczb ujemnych, związek pomiędzy liczbami wzajemnie przeciwnymi może wydać się nieczytelny. Procedura zamiany znaku jest jednak dość prosta (choć oczywiście nie aż tak prosta jak podmiana jednego symbolu na początku liczby): przeglądamy zapis od prawej do lewej i zastępujemy cyfry według schematu $0 \mapsto 0$, $11 \mapsto 01$ oraz $01 \mapsto 11$ (jeśli zostaniemy z wiodącą jedynką, możemy oczywiście po jej lewej stronie dopisać zero i skorzystać z ostatniego przekształcenia).

Algorytm zapisu a przy podstawie d w pigułce: niech $a_0 = a$ oraz $a_i = da_{i+1} + c_i$, gdzie $0 \leqslant c_i < d$, $i = 0, 1, 2, \ldots$ (dzielimy z resztą przez d). Wówczas $a = (c_k c_{k-1} \ldots c_1 c_0)_d$, gdzie k jest najmniejszym indeksem spełniającym $a_{k+1} = 0$.

Algorytm konwersji liczby na zapis negabinarny (lub o innej ujemnej podstawie) jest taki sam jak dla podstawy dodatniej, tylko należy pamiętać, że dzielimy przez liczbę ujemną (podstawę), ale rozważamy reszty nieujemne. Na przykład dla 9 zaczynamy od reszty 1 (z dzielenia przez -2), pozostałe $+8$ dzielimy przez -2. Powstałe -4 daje resztę 0, a po podzieleniu zostaje $+2$, które również daje resztę 0. Kolejne dzielenie daje wynik -1, czyli po odjęciu reszty 1 otrzymujemy -2 do podzielenia, i na koniec zostaje 1. Zapisując wszystkie reszty w odwrotnej kolejności (pierwsza otrzymana reszta to oczywiście cyfra jedności), otrzymujemy $9 = 11001_{(-2)}$.

Prawie każdy system pozycyjny (kod uzupełnień nie do końca tutaj pasuje) używa tego samego algorytmu konwersji. Modyfikacje polegają na stosowaniu *mniej naturalnego* zestawu reszt lub właśnie ujemnej podstawy.

Zauważmy, że wykonując działania minusdwójkowe, w razie przepełnienia do następnego rzędu przenosimy -1. A z kolei -1 wymaga zapisania 1 i przeniesienia $+1$. Przeanalizowanie wszystkich niespodzianek związanych z działaniami minusdwójkowymi pozostawiam Czytelnikom Zainteresowanym. Niewątpliwie pewna trudność w porównywaniu wartości zapisanych nagabinarnie komplikuje algorytm dzielenia, zwłaszcza gdy spróbujemy dzielić liczby różnych znaków.

Choć system minusdwójkowy należy do najpopularniejszych systemów o ujemnej podstawie, możemy też wziąć na warsztat podstawę -10 (aby nie porzucać kompletu cyfr dziesiętnych). W takim systemie liczba 10 będzie trzycyfrowa, w końcu „10” musi oznaczać podstawę systemu, czyli -10.

Skoro liczby ujemne sprawdzają się jako podstawy systemów liczbowych, to może równie dobrym pomysłem mogłyby być ujemne cyfry? Najpopularniejszym tego typu systemem jest *trójkowy system zrównoważony*, który używa podstawy

trzy, ale cyfry mają wartości $+1$, 0 oraz -1. Na minus jedynkę w tym zapisie proponuję „cyfrę nedej”: $\overline{1}$. Początkowe liczby naturalne zapisujemy jako

$$0,\ 1,\ 1\overline{1},\ 10,\ 11,\ 1\overline{1}\overline{1},\ 1\overline{1}0,\ 10\overline{1},\ 100,\ 101,\ 11\overline{1},\ 110,\ 111,\ 1\overline{1}\overline{1}\overline{1},\ 1\overline{1}\overline{1}0,\ \ldots$$

Zwyczajowo jako cyfr w trójkowym systemie zrównoważonym używa się $+$, 0 i $-$, ale znaki działań mogą się jeszcze przydać. Czasem zamiast „nedej” używa się także znaków „T” lub „θ”.

Mając ujemne cyfry, nie musimy stosować znaku „$-$”. Podobnie jak w kodzie uzupełnień, znak liczby determinuje wiodąca cyfra – wartości ujemne zaczynają się od nedej. Tym razem nie jesteśmy jednak ograniczeni do ustalonej z góry liczby cyfr. Podobnie jak w klasycznym systemie trójkowym, parzystość jest taka sama jak parzystość sumy cyfr. Aby ją wyznaczyć, wystarczy zatem policzyć niezerowe cyfry. Gdyby upierać się przy ograniczeniu do liczb dodatnich, można by pomijać przy zapisie początkową jedynkę – jaka by to była oszczędność!

Dodawanie, odejmowanie i mnożenie w systemie zrównoważonym zachowuje się „normalnie”, jeśli za normalne uznajemy przenoszenie do kolejnego rzędu cyfry ujemnej (możemy przenieść każdą z trzech cyfr). W szczególności, dodając do siebie dwie jedynki, zapisujemy nedej i przenosimy jedynkę na lewo. Nieco zaskakujące rzeczy dzieją się przy dzieleniu. Na początku system jest zachęcający, dzięki mało wymagającej tabliczce mnożenia. Pojawia się jednak drobna kontrowersja dotycząca porównywania liczb. Aby zobrazować te „osobliwości”, podzielmy 2024 przez 11. Odnośne liczby zapisujemy jako $10\overline{1}1000\overline{1}$ oraz $11\overline{1}$. Bierzemy pierwsze trzy cyfry i wykonujemy odejmowanie (czyli dodawanie liczby przeciwnej): $10\overline{1} + \overline{1}\overline{1}1 = \overline{1}0$. Zapisujemy cyfrę wyniku 1 (wynik powinien być dodatni, więc wszystko idzie zgodnie z planem). Teraz liczba zaczyna się od nedej, więc zmieniamy znak (dopisujemy do wyniku $\overline{1}$): $\overline{1}01 + 11\overline{1} = 10$. Nadal wszystko jest w porządku. W kolejnym kroku obliczamy $100 + \overline{1}\overline{1}1 = \overline{1}1$. Teraz trzy pierwsze cyfry to $\overline{1}10 = -6$, co do modułu mniejsze od 11. Nadgorliwy rachmistrz poczułby potrzebę zwiększenia liczby cyfr, czyli „opuszczenia” cyfry 0 i obliczenia $\overline{1}100 + 11\overline{1} = \overline{1}1\overline{1}$. „Opuszczając” teraz cyfrę $\overline{1}$, dostaniemy $\overline{1}1\overline{1}\overline{1}$ do podzielenia przez $11\overline{1}$, czyli mamy problem. Okazuje się, że należało mimo wszystko wykonać odejmowanie $-6 + 11 = 5$. Właściwie to już na początku liczyliśmy $8 - 11 = -3$, więc trochę za późno na zdziwienie. Całe dzielenie jest rozpisane na marginesie. Czytelnik może na własną rękę opisać szczegóły tego algorytmu (czym się różni od „zwykłego” dzielenia pisemnego) i uzasadnić, dlaczego działa. Można się zastanawiać, jak należy interpretować „resztę” pozostającą na koniec takiego dzielenia (z premedytacją wybrałem przykład z ilorazem całkowitym). Prostym acz dobitnym przykładem możliwych komplikacji jest próba podzielenia 13 przez 5.

Zmiana znaku jest tak prosta, że wszystkie odejmowania w systemie zrównoważonym będę zapisywał jako dodawanie.

$$\begin{array}{r|l}
 & 1\,\overline{1}\,1\,\overline{1}\,1\,1 \\
\hline
1\,0\,\overline{1}\,1\,0\,0\,0\,\overline{1} & 11\overline{1} \\
+\,\overline{1}\,\overline{1}\,1 & \\
\hline
=\,\overline{1}\,0\,1 & \\
+\,1\,1\,\overline{1} & \\
\hline
=\,1\,0\,0 & \\
+\,\overline{1}\,\overline{1}\,1 & \\
\hline
=\,\overline{1}\,1\,0 & \\
+\,1\,1\,\overline{1} & \\
\hline
1\,\overline{1}\,\overline{1}\,0 & \\
+\;\;\overline{1}\,\overline{1}\,1 & \\
\hline
=\,0\,1\,1\,\overline{1} & \\
+\,\overline{1}\,\overline{1}\,1 & \\
\hline
=\,0 &
\end{array}$$

Zapis ułamków w systemie zrównoważonym także kryje w sobie niespodzianki. Zauważmy, że „maksymalny” ułamek, czyli $0{,}1111\ldots$, jest równy $\sum_{n=1}^{\infty} \frac{1}{3^n} = \frac{1}{2}$. Oznacza to, że zapis liczb spoza przedziału $[-\frac{1}{2}, +\frac{1}{2}]$ wymaga niezerowej cyfry jedności! Nie powinno to jednak dziwić, skoro zapis liczby $5 < 9$ wymaga użycia rzędu dziewiątek ($1\overline{1}\overline{1}$). Nie mamy też jednoznaczności zapisu, ponieważ $0{,}(1) = 1{,}(\overline{1})$. Co ciekawe, niejednoznaczność dotyczy tylko liczb niecałkowitych (dokładnie tych, które są postaci $\frac{k+1/2}{3^n}$ dla pewnych k i n).

W standardowych systemach pozycyjnych liczby całkowite posiadają alternatywne reprezentacje z nieskończonym rozwinięciem po przecinku. Na przykład $0{,}9999\ldots_{10} = 1$

Ujemne cyfry można też wykorzystać w sposób „niezrównoważony”. Na przykład w systemie dziesiętnym „zastąpić cyfrę 9 przez nedej” – wszystkie pozostałe cyfry zostają nieujemne. Oddaje to pewną powszechną preferencję dotyczącą znaku liczb. Liczbę dziewięć zapisywałoby się $1\overline{1}$ i nazywałaby się pewnie „nedejnaście”. A z 90 zrobiłoby się „sto nedejdziesiąt”. Gdyby kogoś gryzła nazwa „nedej”, może rozważyć określenia typu „za jeden dziesięć” i „za dziesięć sto”. Niezależnie od gustów widać, że ze wszystkich „szalonych” systemów ten dość łatwo zintegrować z polszczyzną. Łatwo też konwertować klasyczny zapis dziesiętny na „nedejowany” – przechodzimy od prawej do lewej i każdą dziewiątkę zastępujemy przez nedej, dodając jeden do kolejnego rzędu (może wystąpić przeniesienie). W ramach ciekawostki można wspomnieć, że cena $9^{\underline{99}}$ musiałaby ustąpić $10^{\underline{0\overline{1}}}$ lub $1\overline{1}^{\underline{10\overline{1}}}$...

A tak wygląda zrównoważone rozwinięcie trójkowe liczby π: $10{,}011\overline{1}111\overline{1}000\overline{1}0\ldots$ W „nedejowanym systemie dziesiętnym” wygląda ono zbyt zwyczajnie, aby się nim tutaj chwalić...

Jeśli Czytelnik doszedł do wniosku, że teraz pewnie zacznę opowiadać o systemie, w którym występuje cyfra o wartości $\frac{3}{4}$, a podstawa jest równa π, to znak, że nabrał już pewnego wyczucia i rozpoznaje, jak dalekie uogólnienia znanego systemu dziesiętnego można poczynić. Ułamkowe cyfry i podstawy to jednak temat na (nie)całkowicie inną opowieść.