

Programiranje 1

4. predavanje — 1. dodatak

Saša Singer

PMF – Matematički odsjek, Zagreb

Sadržaj predavanja — dodatka

- Cijeli brojevi — binarni prikaz i prikaz u računalu:
 - Prikaz cijelih brojeva u bazi 2.
 - Algoritam za nalaženje znamenki u prikazu.
 - Prikaz brojeva bez predznaka u računalu.
 - Prikaz brojeva s predznakom u računalu, komplementiraj i dodaj 1.
- Realni brojevi — binarni prikaz:
 - Prikaz realnih brojeva u bazi 2 — cjelobrojni i razlomljeni dio.
 - Algoritam za nalaženje znamenki razlomljenog dijela.
 - Normalizirani prikaz realnih brojeva u bazi 2.
 - Algoritam za nalaženje normaliziranog prikaza.

Sadržaj predavanja — dodatka (nastavak)

- Realni brojevi — prikaz u računalu:
 - Izgled prikaza i prikazivi brojevi.
 - Zaokruživanje realnih brojeva u računalu za prikaz.
 - Prikaz u tipovima `float` i `double` — primjeri.
- Prikaz realnih brojeva — zaokruživanje i greške (još nije sređeno!):
 - Nalaženje najbližih prikazivih brojeva.
 - Greške zaokruživanja u prikazu.

Binarni prikaz cijelih brojeva, algoritmi i primjeri

Sadržaj

- Cijeli brojevi — binarni prikaz i prikaz u računalu:
 - Prikaz cijelih brojeva u bazi 2.
 - Algoritam za nalaženje znamenki u prikazu.
 - Prikaz brojeva bez predznaka u računalu.
 - Prikaz brojeva s predznakom u računalu, komplementiraj i dodaj 1.

Zapis prirodnog broja u bazi 2

Neka je $B \in \mathbb{N}$ neki prirodan broj.

Tzv. pozicioni zapis broja B u bazi $b = 2$ ima sljedeći oblik:

$$B = b_k \cdot 2^k + \dots + b_1 \cdot 2 + b_0, \quad b_i \in \{0, 1\},$$

gdje su b_i binarne znamenke ili bitovi broja B .

Da bismo dobili jednoznačnost prikaza, koristimo da je $B > 0$ i

ne dozvoljavamo da vodeće znamenke budu nule,

tj. zahtijevamo da vodeća znamenka b_k mora biti pozitivna

$$b_k > 0.$$

Normalizirani zapis prirodnog broja u bazi 2

Prikaz oblika

$$B = b_k \cdot 2^k + \dots + b_1 \cdot 2 + b_0, \quad b_i \in \{0, 1\}, \quad b_k > 0,$$

zovemo **normalizirani** pozicioni prikaz broja B u bazi $b = 2$.

Skraćeni zapis ovog prikaza je

$$B = (b_k b_{k-1} \dots b_1 b_0)_2.$$

Broj $k \in \mathbb{N}_0$ **jednoznačno** je određen zahtjevom $b_k > 0$ i vrijedi

$$k = \lfloor \log_2 B \rfloor.$$

Broj binarnih znamenki (“duljina”) broja B je

$$k + 1 = \lfloor \log_2 B \rfloor + 1.$$

Računanje znamenki broja u bazi 2

Binarne znamenke b_i zadanog broja B

$$B = b_k \cdot 2^k + \dots + b_1 \cdot 2 + b_0 = (b_k b_{k-1} \dots b_1 b_0)_2,$$

dobivamo dijeljenjem s bazom 2.

🔴 Preciznije, koristimo cjelobrojni kvocijent i ostatak.

Kako to ide? Direktno iz Euklidovog teorema, zapisom

$$B = (b_k \cdot 2^{k-1} + \dots + b_1) \cdot 2 + b_0 = \text{oznaka} = B_1 \cdot 2 + b_0.$$

Zadnju znamenku b_0 dobivamo kao ostatak pri dijeljenju broja B s bazom 2

$$b_0 = B \bmod 2,$$

pa je $b_0 \in \{0, 1\}$, tj. b_0 je korektna binarna znamenka.

Računanje znamenki broja u bazi 2 (nastavak)

Cjelobrojni kvocijent broja B s bazom 2 je “novi” broj

$$B_1 = B \operatorname{div} 2 = b_k \cdot 2^{k-1} + \dots + b_1.$$

Njegov zapis u bazi 2 je

$$B_1 = (b_k b_{k-1} \dots b_1)_2,$$

tj. dobiva se “brisanjem” znamenke b_0 iz zapisa broja B .

Znamenku b_0 smo upravo našli, pa je dovoljno naći binarni zapis broja B_1 , a taj broj ima jednu znamenku manje.

Naravno, njegovu zadnju znamenku b_1 nalazimo

🔴 ponavljanjem opisanog postupka, ali na broju B_1 .

Računanje znamenki broja u bazi 2 (nastavak)

Čitav postupak možemo zapisati na sljedeći način.

Neka je, na početku, $B_0 := B$.

U općem — i -tom koraku, krećemo s brojem B_i i računamo:

● ostatak — izdvoji (trenutnu) zadnju znamenku u B_i

$$b_i = B_i \bmod 2,$$

● cjelobrojni kvocijent — “obriši” (trenutnu) zadnju znamenku u B_i

$$B_{i+1} = B_i \operatorname{div} 2,$$

tako da uvijek vrijedi

$$B_i = B_{i+1} \cdot 2 + b_i, \quad i = 0, 1, \dots$$

Računanje znamenki broja u bazi 2 (nastavak)

Pitanje je samo — kad treba **stati** (jer **k** ne znamo unaprijed)?

Odgovor: kad “**obrišemo**” sve znamenke iz broja **B**, ostaje nam **nula**.

Uočite da je

$$B_i = b_k \cdot 2^{k-i} + \dots + b_i = (b_k b_{k-1} \dots b_i)_2.$$

Za $i = k$ imamo $B_k = b_k > 0$. Nakon dijeljenja dobivamo

$$B_k = B_{k+1} \cdot 2 + b_k, \quad B_{k+1} = 0.$$

Dakle, postupak **staje** kad **prvi** put dobijemo kvocijent **nula**,

$$B_{k+1} = B_k \operatorname{div} 2 = 0.$$

Znamenke broja u bazi 2 — algoritam

Ovaj postupak možemo zapisati u obliku **algoritma**.

Algoritam 1. Binarne znamenke prirodnog broja.

Ulaz: Prirodni broj B .

Izlaz: Broj $k \geq 0$ i znamenke $b_k, \dots, b_0$ broja B u bazi 2.

```

$$\begin{aligned} & i \leftarrow 0; B_0 \leftarrow B; \\ & \text{sve dok je } B_i > 0 \text{ radi } \{ \\ & \quad b_i \leftarrow B_i \bmod 2; \\ & \quad B_{i+1} \leftarrow B_i \operatorname{div} 2; \\ & \quad i \leftarrow i + 1; \\ & \} \\ & k \leftarrow i - 1; \end{aligned}$$

```

Korektnost algoritma za računanje znamenki

Zadatak. Neka je $\ell \in \mathbb{N}$. Dokažite da **nakon** ℓ koraka opisanog postupka vrijedi (korake brojimo od **jedan**)

$$B = B_\ell \cdot 2^\ell + (b_{\ell-1} \cdot 2^{\ell-1} + \dots + b_1 \cdot 2 + b_0).$$

Drugim riječima, vrijedi

$$B \operatorname{div} 2^\ell = B_\ell,$$

$$B \operatorname{mod} 2^\ell = b_{\ell-1} \cdot 2^{\ell-1} + \dots + b_1 \cdot 2 + b_0.$$

Dodatno, možemo uzeti ove tvrdnje vrijede i za $\ell = 0$,

🔴 tj. na **početku** cijelog postupka — **prije** prvog koraka.

Algoritam **staje** kad **prvi puta** dobije $B_{k+1} = 0$. Dokažite da tada vrijedi $B \operatorname{mod} 2^{k+1} = B$ i $b_k = 1$.

Zapis cijelog broja u bazi 2

Neka je $B \in \mathbb{Z}$ neki **cijeli** broj (smije biti i negativan).

Dogovor za prikaz broja $B = 0$.

- 🕒 U **svakodnevnom** životu pišemo **jednu** znamenku:
 $0 = (0)_2$, tj. $k = 0$ i $b_0 = 0$.
- 🕒 **Normaliziranom** prikazu bi “prije” odgovaralo da **nula**
nema znamenki: $0 = ()_2$, uz, na pr. $k = -1$.

Na primjer, tako radi **Algoritam 1** za ulaz $B = 0$.

Izbor ovisi o potrebi u “okolnom” algoritmu.

Za **negativne** brojeve $B < 0$, prikaz dobivamo tako da

- 🕒 nađemo prikaz **pozitivnog** broja $|B| \in \mathbb{N}$ i
- 🕒 **dodamo** predznak $-$ (minus).

Prikaz broja 1717 u bazi 2

Primjer. Prikaz broja 1717 u bazi 2 dobivamo ovako:

i	B_i	$b_i = B_i \bmod 2$	$B_{i+1} = B_i \operatorname{div} 2$
0	$1717 = 858 \cdot 2 + 1$	$1717 \bmod 2 = 1$	$1717 \operatorname{div} 2 = 858$
1	$858 = 429 \cdot 2 + 0$	$858 \bmod 2 = 0$	$858 \operatorname{div} 2 = 429$
2	$429 = 214 \cdot 2 + 1$	$429 \bmod 2 = 1$	$429 \operatorname{div} 2 = 214$
3	$214 = 107 \cdot 2 + 0$	$214 \bmod 2 = 0$	$214 \operatorname{div} 2 = 107$
4	$107 = 53 \cdot 2 + 1$	$107 \bmod 2 = 1$	$107 \operatorname{div} 2 = 53$
5	$53 = 26 \cdot 2 + 1$	$53 \bmod 2 = 1$	$53 \operatorname{div} 2 = 26$
6	$26 = 13 \cdot 2 + 0$	$26 \bmod 2 = 0$	$26 \operatorname{div} 2 = 13$
7	$13 = 6 \cdot 2 + 1$	$13 \bmod 2 = 1$	$13 \operatorname{div} 2 = 6$
8	$6 = 3 \cdot 2 + 0$	$6 \bmod 2 = 0$	$6 \operatorname{div} 2 = 3$
9	$3 = 1 \cdot 2 + 1$	$3 \bmod 2 = 1$	$3 \operatorname{div} 2 = 1$
10	$1 = 0 \cdot 2 + 1$	$1 \bmod 2 = 1$	$1 \operatorname{div} 2 = 0$

Prikaz broja 1717 u bazi 2 (nastavak)

Dobili smo da je $B_{11} = 0$, pa je $k = 10$.

Vodeća znamenka b_{10} je na dnu tablice, tj. znamenke treba pisati odozdo prema gore, da ih dobijemo u standardnom poretku $b_{10}, \dots, b_0$.

Rješenje. Prikaz broja 1717 u bazi 2 je

$$(1717)_{10} = (11010110101)_2$$

i ima 11 binarnih znamenki.

Primjer. Prikaz broja -1717 u bazi 2 je

$$(-1717)_{10} = (-11010110101)_2$$

i (dogovorno, također) ima 11 binarnih znamenki. Predznak minus smijemo napisati i ispred zgrade (svejedno je).

Prikaz broja 1717 kao int

Primjer. Prikaz broja 1717 kao `int` i `unsigned int`.

☞ Za tipove `int` i `unsigned int`, broj bitova za prikaz brojeva je $n = 32$.

Rješenje. Za početak, jer je $k + 1 = 11 < n = 32$, broj 1717 je prikaziv u oba tipa.

Prikazu broja 1717 u bazi 2 samo treba dodati potreban broj vodećih nula, do broja bitova predviđenog za prikaz u odgovarajućem tipu.

Dakle, prikaz broja 1717 u tim tipovima je

0000 0000 0000 0000 0000 0110 1011 0101

Prikaz broja -1717 kao `int`

Primjer. Prikaz broja -1717 kao `int`.

Krećemo od prikaza broja 1717 kao `int`:

```
0000 0000 0000 0000 0000 0110 1011 0101
```

Komplementiramo (bit-po-bit) i dodamo 1 (modulo 2^{32}):

```
1111 1111 1111 1111 1111 1001 0100 1010
+
                                     1
```

Rezultat je prikaz broja -1717 kao `int`:

```
1111 1111 1111 1111 1111 1001 0100 1011
```

Binarni prikaz realnih brojeva, algoritmi i primjeri

Sadržaj

- Realni brojevi — binarni prikaz:
 - Prikaz realnih brojeva u bazi 2 — cjelobrojni i razlomljeni dio.
 - Algoritam za nalaženje znamenki razlomljenog dijela.
 - Normalizirani prikaz realnih brojeva u bazi 2.
 - Algoritam za nalaženje normaliziranog prikaza.

Zapis realnog broja u bazi 2

Neka je $B \in \mathbb{R}$ neki **realan** broj.

Osnovni pristup prikazu **sličan** je onom za **cijele** brojeve.

Za **negativne** brojeve $B < 0$, prikaz dobivamo tako da

- nađemo prikaz **pozitivnog** broja $|B| \in \mathbb{R}_+$ i
- dodamo** predznak $-$ (minus).

Dakle, dovoljno je naći prikaz **nenegativnih** brojeva $B \geq 0$.
Njihov prikaz sastoji se iz 2 dijela:

- prikaza **cjelobrojnog** dijela $\lfloor B \rfloor$ — što znamo naći, i
- prikaza tzv. **razlomljenog** dijela $B - \lfloor B \rfloor \in [0, 1)$.

Kako se nalazi prikaz “**razlomljenog**” dijela broja?

Zapis razlomljenog dijela realnog broja u bazi 2

Neka je $B \in [0, 1)$ bilo koji razlomljeni dio realnog broja.

Tzv. pozicioni zapis broja B u bazi $b = 2$ ima sljedeći oblik:

$$B = \sum_{i=1}^{\infty} b_{-i} \cdot 2^{-i}, \quad b_{-i} \in \{0, 1\},$$

gdje su b_{-i} binarne znamenke ili bitovi broja B .

U skraćenom zapisu, znamenke b_{-i} pišemo iza binarne točke

$$B = (0.b_{-1}b_{-2} \dots)_2.$$

Traži se postupak — koji, za zadani broj $B \in [0, 1)$,

- ☛ “generira” pripadni niz binarnih znamenki $(b_{-i}, i \in \mathbb{N})$, i to “član-po-član”, za $i = 1, 2, \dots$

Računanje znamenki broja u bazi 2 — uvod

Prije opisa postupka za računanje znamenki broja B u bazi 2, treba uočiti jednu “sitnicu” — vezanu uz

● jednoznačnost rezultata i dozvoljene nizove znamenki.

Naime, za bilo koji niz binarnih znamenki $(b_{-i}, i \in \mathbb{N})$, gdje je $b_{-i} \in \{0, 1\}$, pripadni red iz zapisa broja

$$\sum_{i=1}^{\infty} b_{-i} \cdot 2^{-i},$$

uvijek konvergira, i suma je neki broj iz zatvorenog intervala $[0, 1]$ (niz parcijalnih suma monotonno raste i ograničen je s 1).

Međutim, različiti nizovi znamenki mogu dati istu sumu pripadnog reda — dakle, zapis pripadnog broja (= sume reda) ne mora biti jednoznačan.

Dovoljeni nizovi znamenki u bazi 2

Problem nastaje kod onih nizova znamenki $(b_{-i}, i \in \mathbb{N})$ koji imaju **konačno** mnogo **nula**, tj. počev od nekog mjesta $k \in \mathbb{N}$, imaju **beskonačni niz** (uzastopnih) **jedinica** na “repu”. Zbog

$$\sum_{i=k}^{\infty} 1 \cdot 2^{-i} = 2^{-k+1} = 1 \cdot 2^{-k+1} + \sum_{i=k}^{\infty} 0 \cdot 2^{-i},$$

svaki takav niz znamenki može se “**skratiti**” na **konačan** = ima beskonačni **niz nula** na “repu”, počev od tog istog mjesta k (te **nule**, obično, **ne pišemo** — pa zato kažemo “**konačan**” niz).

Zbog ove **nejedinstvenosti** zapisa,

- 🚫 nizovi znamenki s **beskonačnim nizom jedinica** na “repu” **nisu dozvoljeni** kao pozicioni zapis nekog broja u bazi 2.

Slično je i u drugim bazama (**niz najvećih** znamenki).

Problem za $k = 1$ — same jedinice u nizu

U slučaju kad je $k = 1$, “problematični” niz ima same jedinice. Njegov skraćeni zapis je $(0.1111\dots)_2$, a pripadna suma reda (tj. broj kojeg prikazuje taj niz) je

$$B = \sum_{i=1}^{\infty} 1 \cdot 2^{-i} = 1.$$

Kod traženja prikaza razlomljenog dijela realnog broja, ovo je dodatno zabranjeno pretpostavkom na ulazni broj B .

Naime, pretpostavka $B \in [0, 1)$ povlači $B < 1$, a to znači da je

❌ nemoguće da su sve znamenke b_{-i} jednake 1.

Drugim riječima,

❌ barem jedna znamenka b_{-i} je jednaka 0, za neki $i \in \mathbb{N}$.

Problem za $k > 1$ — nejedinstvenost zapisa

Dakle, u postupku kojeg tražimo, za “problematični” niz mora vrijediti $k > 1$, jer ima **bar jednu nulu**. No, onda je sigurno

 $b_{-k+1} = 0$ **zadnja** nula u tom nizu (jer $b_{-i} = 1$, za $i \geq k$).

Skraćeni zapis takvog “problematičnog” niza je

$$(0.b_{-1} \dots b_{-k+2} \mathbf{0} 111 \dots)_2.$$

Za pripadnu sumu reda (= broj kojeg prikazuje taj niz) vrijedi

$$B = \sum_{i=1}^{k-2} b_{-i} \cdot 2^{-i} + \sum_{i=k}^{\infty} 1 \cdot 2^{-i} = \sum_{i=1}^{k-2} b_{-i} \cdot 2^{-i} + \mathbf{2^{-k+1}},$$

pa broj B ima i “**kraći**” **konačni** zapis

$$(0.b_{-1} \dots b_{-k+2} \mathbf{1})_2.$$

Dovoljeni nizovi znamenki u bazi 2 (kraj)

Zadatak. Pokažite da samo navedeni “problematični” nizovi binarnih znamenki dovode do nejedinstvenosti zapisa u bazi 2, tj. svi ostali nizovi jednoznačno određuju pripadni broj B .

U terminima brojeva $B \in [0, 1]$, samo brojevi s konačnim binarnim zapisom imaju jedinstven pozicioni zapis, i to su svi brojevi oblika $B = m/2^n$, za $m, n \in \mathbb{N}$ i $m < 2^n$. ■

Ograničenje na **dozvoljene** nizove znamenki za pozicioni zapis

☞ nije bitno za postupak (algoritam) — on radi za svaki dozvoljeni ulaz, već samo objašnjava njegov izlaz.

U tom smislu, postupak kojeg ćemo opisati

☞ nalazi “**najkraći**” prikaz broja B u bazi 2,

☞ tj. **izbjegava** beskonačni niz jedinica na “repu” prikaza.

Računanje znamenki broja u bazi 2

Binarne znamenke b_{-i} zadanog broja $B \in [0, 1)$

$$\begin{aligned} B &= b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + b_{-3} \cdot 2^{-3} + \dots \\ &= (0.b_{-1}b_{-2}b_{-3}\dots)_2, \end{aligned}$$

dobivamo množenjem s bazom 2.

☞ Preciznije, koristimo množenje i najveće cijelo.

Kako to ide? Kad broj B pomnožimo s 2, dobijemo broj

$$\begin{aligned} B'_1 = \text{oznaka} &= 2B = b_{-1} + b_{-2} \cdot 2^{-1} + b_{-3} \cdot 2^{-2} + \dots \\ &= (b_{-1}.b_{-2}b_{-3}\dots)_2. \end{aligned}$$

Sad iskoristimo da je $B'_1 = 2B \in [0, 2)$, tj. $B'_1 < 2$.

Računanje znamenki broja u bazi 2 (nastavak)

Prvu znamenku b_{-1} dobivamo kao najveće cijelo od $B'_1 = 2B$

$$b_{-1} = \lfloor B'_1 \rfloor.$$

Zbog $B'_1 \in [0, 2)$, mora biti $b_{-1} \in \{0, 1\}$, pa je b_{-1} korektna binarna znamenka.

Kad od B'_1 oduzmemo cijeli broj b_{-1} , dobivamo “novi” broj

$$B_1 = B'_1 - b_{-1} = B'_1 - \lfloor B'_1 \rfloor = 2B - \lfloor 2B \rfloor.$$

Njegov zapis u bazi 2 je

$$B_1 = (0.b_{-2}b_{-3} \dots)_2,$$

tj. dobiva se “brisanjem” znamenke b_{-1} iz zapisa broja B .

Računanje znamenki broja u bazi 2 (nastavak)

Naravno, **prvu** znamenku b_{-2} broja B_1 nalazimo

🔴 **ponavljanjem** opisanog postupka, ali na broju B_1 .

Po definiciji funkcije **najveće cijelo**, za novi broj B_1 , također, vrijedi $B_1 \in [0, 1)$, pa **smijemo nastaviti** postupak s brojem B_1 .

Napomena. U prethodnom opisu, zadan je **samo** broj B .

Njegove znamenke $b_{-1}, b_{-2}, \dots$ **nisu** zadane ili **poznate**, već ih treba **naći**.

🔴 Napisane su samo za **lakši** opis postupka.

Ono što **stvarno vrijedi** nakon opisanog **prvog koraka** je:

$$B = b_{-1} \cdot 2^{-1} + B_1 \cdot 2^{-1},$$

s tim da smo **našli** $b_{-1} \in \{0, 1\}$ i $B_1 \in [0, 1)$.

Računanje znamenki broja u bazi 2 (nastavak)

Čitav postupak možemo zapisati na sljedeći način.

Neka je, na početku, $B_0 := B$.

U općem — i -tom koraku, krećemo s brojem B_{i-1} i računamo:

● produkt — pomakni (trenutnu) prvu znamenku u B_{i-1} ispred binarne točke

$$B'_i = 2B_{i-1},$$

● najveće cijelo — izdvoji tu prvu znamenku u B'_i

$$b_{-i} = \lfloor B'_i \rfloor,$$

● razlika — “obriši” (trenutnu) prvu znamenku u B_{i-1} (B'_i)

$$B_i = B'_i - b_{-i} = 2B_{i-1} - \lfloor 2B_{i-1} \rfloor.$$

Računanje znamenki broja u bazi 2 (nastavak)

Onda uvijek vrijedi

$$B_{i-1} = b_{-i} \cdot 2^{-1} + B_i \cdot 2^{-1}, \quad i = 1, 2, \dots,$$

s tim da je

- $b_{-i} \in \{0, 1\}$ nova **binarna** znamenka i
- $B_i \in [0, 1)$ novi “**razlomljeni dio**” za nastavak postupka.

Zadatak. Neka je $\ell \in \mathbb{N}$. Dokažite da **nakon** ℓ koraka opisanog postupka vrijedi (korake brojimo od **jedan**)

$$B = b_{-1} \cdot 2^{-1} + \dots + b_{-\ell} \cdot 2^{-\ell} + B_\ell \cdot 2^{-\ell}.$$

Dodatno, možemo uzeti ove tvrdnje vrijede i za $\ell = 0$,

- tj. na **početku** cijelog postupka — **prije** prvog koraka.

Računanje znamenki broja u bazi 2 (nastavak)

U principu, zadani broj B može imati **beskonačno** dug prikaz,
— pa ovaj postupak možemo ponavljati “do u **nedogled**”.

Da bismo dobili **algoritam**, potrebno je zadati

— $j = \text{maksimalni}$ broj znamenki koje želimo izračunati.

S druge strane, kako se **prepoznaje** da B ima **konačan** prikaz?

— Ako u nekom koraku k **prvi puta** dobijemo da je $B_k = 0$,
onda je prikaz **konačan**

$$B = (0.b_{-1}b_{-2} \dots b_{-k})_2.$$

Sve znamenke **iza** b_{-k} su jednake 0, pa ih, obično, **ne pišemo**.
To se može dogoditi i na **početku**, ako je $B = 0 = (0.)_2$.

Znamenke broja u bazi 2 — algoritam

Algoritam 2. Binarne znamenke razlomljenog dijela realnog broja.

Ulaz: Realni broj $B \in [0, 1)$ i cijeli broj $j \geq 0$.

Izlaz: Broj $k \geq 0$ i znamenke $b_{-1}, \dots, b_{-k}$ broja B u bazi 2.

```

$$\begin{aligned} & i \leftarrow 0; B_0 \leftarrow B; k \leftarrow j; \\ & \text{sve dok je } B_i > 0 \text{ i } i < k \text{ radi } \{ \\ & \quad i \leftarrow i + 1; \\ & \quad B'_i \leftarrow 2 \cdot B_{i-1}; \\ & \quad b_{-i} \leftarrow \lfloor B'_i \rfloor; \\ & \quad B_i \leftarrow B'_i - b_{-i}; \\ & \} \\ & \text{ako je } B_i = 0 \text{ onda } k \leftarrow i; \end{aligned}$$

```

Prikaz broja 0.46875 u bazi 2

Primjer. Nađimo prikaz broja 0.46875 u bazi 2.

Počinjemo s $B_0 = 0.46875$, a zatim izlazi

i	$B'_i = 2 \cdot B_{i-1}$	$b_{-i} = \lfloor B'_i \rfloor$	$B_i = B'_i - b_{-i}$
1	$0.46875 \cdot 2 = 0.9375$	$\lfloor 0.9375 \rfloor = 0$	$0.9375 - 0 = 0.9375$
2	$0.9375 \cdot 2 = 1.875$	$\lfloor 1.875 \rfloor = 1$	$1.875 - 1 = 0.875$
3	$0.875 \cdot 2 = 1.75$	$\lfloor 1.75 \rfloor = 1$	$1.75 - 1 = 0.75$
4	$0.75 \cdot 2 = 1.5$	$\lfloor 1.5 \rfloor = 1$	$1.5 - 1 = 0.5$
5	$0.5 \cdot 2 = 1.0$	$\lfloor 1.0 \rfloor = 1$	$1.0 - 1 = 0.0$

Dobili smo da je $B_5 = 0$, pa broj 0.46875 ima **konačan** prikaz u bazi 2.

Prikaz broja 0.46875 u bazi 2 (nastavak)

Prva znamenka b_{-1} je na vrhu tablice, tj. znamenke treba pisati odozgo prema dolje, da ih dobijemo u standardnom poretku $b_{-1}, b_{-2}, \dots$.

Rješenje. Prikaz broja 0.46875 u bazi 2 je

$$(0.46875)_{10} = (0.01111)_2$$

i ima 5 binarnih znamenki iza binarne točke.

Prikaz broja 0.1 u bazi 2

Primjer. Nađimo prikaz broja 0.1 u bazi 2.

Počnemo s $B_0 = 0.1$, a zatim izlazi

i	$B'_i = 2 \cdot B_{i-1}$	$b_{-i} = \lfloor B'_i \rfloor$	$B_i = B'_i - b_{-i}$
1	$0.1 \cdot 2 = 0.2$	$\lfloor 0.2 \rfloor = 0$	$0.2 - 0 = 0.2$
2	$0.2 \cdot 2 = 0.4$	$\lfloor 0.4 \rfloor = 0$	$0.4 - 0 = 0.4$
3	$0.4 \cdot 2 = 0.8$	$\lfloor 0.8 \rfloor = 0$	$0.8 - 0 = 0.8$
4	$0.8 \cdot 2 = 1.6$	$\lfloor 1.6 \rfloor = 1$	$1.6 - 1 = 0.6$
5	$0.6 \cdot 2 = 1.2$	$\lfloor 1.2 \rfloor = 1$	$1.2 - 1 = 0.2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Dobili smo da je $B_5 = B_1$. Zato se “**blok**” od 4 reda tablice (od **drugog** do **petog**) nadalje **ponavlja** do u beskonačnost.

Prikaz broja 0.1 u bazi 2 (nastavak)

To znači da 0.1 ima **beskonačan periodički** prikaz u bazi 2, a znamenke $b_{-2}, b_{-3}, b_{-4}, b_{-5}$ čine **jedan period**.

Rješenje. Prikaz broja 0.1 u bazi 2 je

$$(0.1)_{10} = (0.0\ 0011\ 0011\ \dots)_2 = (0.0\ \dot{0}01\dot{1})_2.$$

Time smo **završili** priču o prikazu realnih brojeva $B \in [0, 1]$, tj. sad znamo naći i prikaz

razlomljenog dijela bilo kojeg **nenegativnog** realnog broja.

Vratimo se onda prikazu “općih” realnih brojeva $B \in \mathbb{R}$.

Zapis bilo kojeg realnog broja u bazi 2

Neka je $B \in \mathbb{R}$ neki **realan** broj.

Prikaz broja B u **bazi** $b = 2$ dobiva se tako da nađemo prikaz njegove apsolutne vrijednosti — **nenegativnog** broja $|B|$.

Prikaz broja $|B|$ dobiva se **spajanjem** prikaza

- njegovog **cjelobrojnog** dijela $\lfloor |B| \rfloor$ i

- njegovog **razlomljenog** dijela $|B| - \lfloor |B| \rfloor \in [0, 1)$,

a **između** ta dva dijela pišemo **binarnu** točku.

Na kraju, ako je $B < 0$, dobivenom prikazu broja $|B|$

- **na početak** dodamo predznak $-$ (minus).

Zapis realnog broja u bazi 2 (nastavak)

Podloga je **pozicioni** zapis nenegativnog broja $|B|$ u **bazi** $b = 2$:

$$|B| = \underbrace{\sum_{i=0}^k b_i \cdot 2^i}_{\text{cijeli dio}} + \underbrace{\sum_{i=1}^{\infty} b_{-i} \cdot 2^{-i}}_{\text{razlomljeni dio}}, \quad b_i, b_{-i} \in \{0, 1\},$$

gdje su b_i, b_{-i} **binarne znamenke** ili **bitovi** broja $|B|$ i broja B , a **cijeli dio** broja je **normaliziran**. Ako je $\lfloor |B| \rfloor \neq 0$, onda je $b_k > 0$, a u protivnom pišemo znamenku $b_0 = 0$.

Skraćeni zapis, uz eventualni predznak broja B , je

$$B = \pm(b_k b_{k-1} \dots b_1 b_0 . b_{-1} b_{-2} \dots)_2.$$

Ako je B **cijeli** broj, tj. $B \in \mathbb{Z}$, onda se B prikazuje baš kao **cijeli** broj, a **razlomljeni** dio mu je jednak **nula** i ne piše se.

Prikaz broja -10.25 u bazi 2

Primjer. Nađimo prikaz broja -10.25 u bazi 2.

Cjelobrojni dio apsolutne vrijednosti broja B je

$$B_0 = \lfloor |B| \rfloor = 10.$$

Prikaz u bazi 2:

i	B_i	$b_i = B_i \bmod 2$	$B_{i+1} = B_i \operatorname{div} 2$
0	$10 = 5 \cdot 2 + 0$	$10 \bmod 2 = 0$	$10 \operatorname{div} 2 = 5$
1	$5 = 2 \cdot 2 + 1$	$5 \bmod 2 = 1$	$5 \operatorname{div} 2 = 2$
2	$2 = 1 \cdot 2 + 0$	$2 \bmod 2 = 0$	$2 \operatorname{div} 2 = 1$
3	$1 = 0 \cdot 2 + 1$	$1 \bmod 2 = 1$	$1 \operatorname{div} 2 = 0$

Dakle, prikaz cjelobrojnog dijela je $(10)_{10} = (1010)_2$.

Prikaz broja -10.25 u bazi 2 (nastavak)

Razlomljeni dio apsolutne vrijednosti broja B je

$$B_0 = |B| - \lfloor |B| \rfloor = 0.25.$$

Prikaz u bazi 2:

i	$B'_i = 2 \cdot B_{i-1}$	$b_{-i} = \lfloor B'_i \rfloor$	$B_i = B'_i - b_{-i}$
1	$0.25 \cdot 2 = 0.5$	$\lfloor 0.5 \rfloor = 0$	$0.5 - 0 = 0.5$
2	$0.5 \cdot 2 = 1.0$	$\lfloor 1.0 \rfloor = 1$	$1.0 - 1 = 0.0$

Dakle, prikaz razlomljenog dijela je $(0.25)_{10} = (0.01)_2$.

Kad spojimo ove prikaze i dodamo predznak, izlazi

$$(-10.25)_{10} = (-1010.01)_2.$$

Normalizirani zapis realnog broja u bazi 2

Normalizirani prikaz realnog broja B u bazi $b = 2$ odgovara tzv. “znanstvenom” zapisu tog broja u obliku

$$B = \text{predznak} \cdot \text{mantisa} \cdot 2^{\text{eksponent}},$$

s tim da je

• $\text{predznak} \in \{-1, 1\}$, a značenja su, redom, minus i plus.

Umjesto predznaka, mogli bismo koristiti i funkciju sign , no standardno je $\text{sign}(0) = 0$, što nepotrebno komplicira stvar.

Čisto matematički, bez obzira na prikaz brojeva u računalu, da bismo osigurali jednoznačnost ovog zapisa, moramo:

- precizirati oblik mantise i eksponenta
- i uvesti neki dogovor za prikaz broja $B = 0$.

Normalizirani zapis realnog broja u bazi 2

Dogovor za “normalizirani” prikaz nule je: $0 = 1 \cdot 0.0 \cdot 2^0$, tj.

• predznak(0) = 1, mantisa(0) = 0.0, eksponent(0) = 0.

Standardna ograničenja na mantisu i eksponent su:

• eksponent je cijeli broj,

• mantisa je pozitivan realni broj s jednoznamenkastim cjelobrojnim dijelom, tj. nalazi se u intervalu

$$\text{mantisa} \in [1, 2).$$

Uz ova ograničenja, svaki realni broj $B \neq 0$

• ima jednoznačan normalizirani prikaz u bazi 2.

Za $B = 0$ to ne vrijedi i zato nam treba raniji dogovor.

Normalizirani zapis $B \neq 0$ u bazi 2

U nastavku koristimo **skraćene** oznake

 s , m i e za predznak, mantisu i eksponent broja, po ugledu na oznake kod prikaza brojeva u računalu.

Ako je $B \neq 0$, onda njegov **normalizirani** prikaz u bazi $b = 2$ ima sljedeći oblik:

$$B = s \cdot \underbrace{\left(b_0 + \sum_{i=1}^{\infty} b_{-i} \cdot 2^{-i} \right)}_m \cdot 2^e, \quad b_0, b_{-i} \in \{0, 1\}, \quad b_0 > 0,$$

gdje su b_0 , b_{-i} **binarne znamenke** ili **bitovi** mantise m broja B .

Posebno, u bazi 2, **prva** (cjelobrojna) znamenka mantise **mora** biti $b_0 = 1$.

Normalizirani zapis $B \neq 0$ u bazi 2 (nastavak)

Skraćeni zapis ovog prikaza je

$$B = s \cdot (b_0.b_{-1}b_{-2} \dots)_2 \cdot 2^e.$$

Dogovorno, kad **mantisu** zapisujemo **znamenkama**, uvijek pišemo njezin **najkraći** oblik (to ne vrijedi u računalu).

Ponekad se za **bitove** mantise koriste **nenegativni** indeksi, uz oznaku $m_i := b_{-i}$ (na pr., kod prikaza brojeva u računalu).

Nadalje pretpostavljamo da je $B \neq 0$. Sljedeći problem je:

- 🔴 Kako naći **normalizirani** prikaz broja B u bazi 2, tj. njegovu **mantisu** i **eksponent**?

S predznakom je **lako**.

Normalizirani zapis $B \neq 0$ u bazi 2 (nastavak)

Odgovor ovisi o tome što već znamo:

- samo broj B ,
- ili već imamo njegov “obični” prikaz u bazi 2.

Ako znamo “obični” prikaz broja B u bazi 2, onda je lako:

- pomaknemo binarnu točku za odgovarajući broj mjesta na pravu stranu,
- tako da dobiveni broj ima jednoznamenasti cjelobrojni dio, tj. “padne” u traženi interval $[1, 2)$ za mantisu.

Eksponent je broj mjesta za koje smo pomakli binarnu točku:

- pomak udesno daje negativan eksponent, a
- pomak ulijevo daje pozitivan eksponent.

Normalizirani prikaz iz običnog — primjeri

Primjer. Nađimo **normalizirani** prikaz broja 1717 u bazi 2.

Obični prikaz broja 1717 u bazi 2 je (v. ranije)

$$(1717)_{10} = (11010110101)_2$$

i ima 11 binarnih znamenki. Binarna točka **ne piše**, jer je broj **cijeli**, ali nalazi se odmah **iza zadnje** znamenke.

Očito je da binarnu točku treba pomaknuti odmah **iza prve** znamenke 1, tj. za 10 mjesta **ulijevo**.

Rješenje. **Normalizirani** prikaz broja 1717 u bazi 2 je

$$(1717)_{10} = 1 \cdot (1.1010110101)_2 \cdot 2^{10}.$$

Po dijelovima: $s = 1$, $m = (1.1010110101)_2$ i $e = 10$.

Normalizirani prikaz iz običnog — primjeri

Primjer. Nađimo **normalizirani** prikaz broja -10.25 u bazi 2 .

Obični prikaz broja -10.25 u bazi 2 je (v. ranije)

$$(-10.25)_{10} = (-1010.01)_2.$$

Binarnu točku opet treba pomaknuti odmah **iza prve** napisane znamenke 1 , tj. za 3 mjesta **ulijevo**.

Rješenje. **Normalizirani** prikaz broja -10.25 u bazi 2 je

$$(-10.25)_{10} = -1 \cdot (1.01001)_2 \cdot 2^3.$$

Po dijelovima: $s = -1$, $m = (1.01001)_2$ i $e = 3$.

Normalizirani prikaz iz običnog — primjeri

Primjer. Nađimo **normalizirani** prikaz broja 0.1 u bazi 2.

Obični prikaz broja 0.1 u bazi 2 je **beskonačan** (v. ranije)

$$(0.1)_{10} = (0.0\ 0011\ 0011\ \dots)_2 = (0.0\ \dot{0}01\dot{1})_2.$$

Binarnu točku sad treba pomaknuti odmah **iza prve** napisane **jedinice**, tj. za 4 mjesta **udesno**.

Rješenje. **Normalizirani** prikaz broja 0.1 u bazi 2 je

$$\begin{aligned}(0.1)_{10} &= 1 \cdot (1.1001\ 1001\ \dots)_2 \cdot 2^{-4} \\ &= 1 \cdot (1.\dot{1}00\dot{1})_2 \cdot 2^{-4}.\end{aligned}$$

Po dijelovima: $s = 1$, $m = (1.\dot{1}00\dot{1})_2$ i $e = -4$.

Normalizirani zapis $B \neq 0$ u bazi 2 (nastavak)

Uzmimo sad da znamo samo broj B .

U tom slučaju, ranije operacije:

- pomakni binarnu točku za odgovarajući broj mjesta na pravu stranu,
 - tako da dobiveni broj ima jednoznamenasti cjelobrojni dio, tj. “padne” u traženi interval $[1, 2)$ za mantisu,
- treba obaviti “bez gledanja prikaza”,
- koristeći samo operacije na brojevima.

Već smo rekli da se predznak s lako nalazi — testom $B > 0$, pa je dovoljno raditi s brojem $|B| > 0$. Idemo redom.

Normalizirani zapis $B \neq 0$ u bazi 2 (nastavak)

Uočimo da **pomak** binarne točke u broju za **jedno** mjesto

- **udesno** — odgovara **množenju** broja s **2**,
- **ulijevo** — odgovara **dijeljenju** broja s **2**.

Cilj pomicanja binarne točke u broju $|B|$ je **dobiti** mantisu,

- tj. treba “**natjerati**” broj u **traženi** interval $[1, 2)$.

Usput, da bi polazni broj $|B|$ ostao **isti**, za svaki **pomak** točke, **eksponent** baze **2** treba **pomaknuti** na “**suprotnu**” stranu:

- točka **udesno** — **smanji** eksponent za **1**,
- točka **ulijevo** — **povećaj** eksponent za **1**.

Dakle, kostur algoritma se nazire.

Normalizirani zapis $B \neq 0$ u bazi 2 (nastavak)

Ovaj postupak, zapravo, rastavlja $|B|$ u produkt dva faktora

$$|B| = \text{broj} \cdot 2^e, \quad \text{broj} = |B| \cdot 2^{-e},$$

mijenjajući broj i eksponent e , sve dok ne dobije broj $\in [1, 2)$.

Fali nam još samo početak, tj. inicijalizacija postupka.

Krećemo s broj $= |B|$. Onda, očito, e treba staviti na nulu.

I zadnja stvar — “pravu” stranu za pomak binarne točke nalazimo usporedbom broja $|B|$ s rubovima željenog intervala:

- 🔴 $|B| \geq 2$ — ulijevo: smanjui broj (dijeljenje), povećavaj e ,
- 🔴 $|B| < 1$ — udesno: povećavaj broj (množenje), smanjui e ,
- 🔴 inače — onda je $|B| \in [1, 2)$ pa smo odmah gotovi.

Normalizirani zapis broja u bazi 2 — algoritam

Algoritam 3. Normalizirani binarni zapis realnog broja.

Ulaz: Realni broj B .

Izlaz: Predznak s , mantisa m i eksponent e u normaliziranom zapisu broja B u bazi 2.

```
/* Prvo sredimo  $B = 0$ . */  
ako je  $B = 0$  onda {  
     $s \leftarrow 1$ ;  $m \leftarrow 0.0$ ;  $e \leftarrow 0$ ;  
} inače {  
    /* Ovdje je  $B \neq 0$ . */  
    ako je  $B > 0$  onda  $s \leftarrow 1$ ; inače  $s \leftarrow -1$ ;  
     $m \leftarrow |B|$ ;  $e \leftarrow 0$ ;
```

Normalizirani zapis broja u bazi 2 — algoritam

```
ako je  $m \geq 2$  onda {  
    /* Smanjuj  $m$  dijeljenjem, povećavaj  $e$ . */  
    ponavljaj {  
         $m \leftarrow m/2$ ;  $e \leftarrow e + 1$ ;  
    } sve dok je  $m \geq 2$ ;  
} inače ako je  $m < 1$  onda {  
    /* Povećavaj  $m$  množenjem, smanjuj  $e$ . */  
    ponavljaj {  
         $m \leftarrow m \cdot 2$ ;  $e \leftarrow e - 1$ ;  
    } sve dok je  $m < 1$ ;  
}  
}
```

Ovdje, na kraju, brojevi s , m i e imaju tražene vrijednosti.

Normalizirani zapis broja u bazi 2 — komentar

Ovaj algoritam radi samo s brojevima, pa su i

🔴 njegovi rezultati — “obični” brojevi.

Ako je $B \neq 0$, onda je $B = s \cdot m \cdot 2^e$, uz

$$s \in \{-1, 1\}, \quad m \in [1, 2), \quad e \in \mathbb{Z}.$$

Drugim riječima, ne dobivamo binarne prikaze brojeva m i e

🔴 kao nizove njihovih znamenki u bazi 2.

U ovom obliku normaliziranog zapisa realnog broja B , baza 2 se pojavljuje samo na dva mjesta:

🔴 kao baza za eksponent broja, $B = s \cdot m \cdot 2^e$,

🔴 kao desni rub intervala $[1, 2)$ za mantisu, ako je $B \neq 0$.

Normalizirani zapis broja u bazi 2 — znamenke

Ako želimo prikaz dobivenih brojeva m i e u bazi 2, tj.

ako želimo naći i nizove njihovih znamenki, onda ima još posla.

Mantisa $m = (b_0.b_{-1}b_{-2} \dots)_2$:

- prvo izdvojimo cjelobrojnu znamenku b_0 iz broja m ,
 $b_0 = \lfloor m \rfloor$ (ako je $B \neq 0$, u bazi 2 je $b_0 = 1$),
- pozovemo **Algoritam 2** na razlomljenom dijelu mantise, tj. na broju $m - b_0 \in [0, 1)$.

Eksponent $e = (\pm e_\ell \dots e_0)_2$:

- izdvojimo predznak i pozovemo **Algoritam 1** na broju $|e|$.

Normalizirani prikaz broja -10.25 u bazi 2

Primjer. Nađimo **normalizirani** prikaz broja -10.25 u bazi 2.

Prvo nađemo predznak: iz $B = -10.25 < 0$ slijedi $s = -1$.

U nastavku radimo s apsolutnom vrijednošću $|B| = 10.25$.

Na početku je $m = |B|$ i $e = 0$. Zbog $m = 10.25 \geq 2$,

🔴 binarnu točku u broju m treba pomicati **ulijevo**, tj.

🔴 m treba **dijeliti** s 2, a e **povećavati** za 1.

korak	$m \leftarrow m/2$	$e \leftarrow e + 1$
1	$10.25 / 2 = 5.125 \geq 2$	$0 + 1 = 1$
2	$5.125 / 2 = 2.5625 \geq 2$	$1 + 1 = 2$
3	$2.5625 / 2 = 1.28125 < 2$	$2 + 1 = 3$

Normalizirani prikaz broja -10.25 u bazi 2

Rješenje. Normalizirani prikaz broja -10.25 u bazi 2 je

$$(-10.25)_{10} = -1 \cdot 1.28125 \cdot 2^3.$$

Po dijelovima: $s = -1$, $m = 1.28125$ i $e = 3$.

Binarni prikaz mantise:

☞ $b_0 = 1$, a Algoritam 2 na broju $m - b_0 = 0.28125$ daje

$$0.28125 = (0.01001)_2$$

(provjerite), pa je $m = 1.28125 = (1.01001)_2$.

Binarni prikaz eksponenta:

☞ Algoritam 1 na broju $|e| = 3$ daje $3 = (11)_2$, pa je
 $e = 3 = (11)_2$.

Normalizirani prikaz broja 0.1 u bazi 2

Primjer. Nađimo **normalizirani** prikaz broja 0.1 u bazi 2.

Prvo nađemo predznak: iz $B = 0.1 \geq 0$ slijedi $s = 1$.

Na početku je $m = |B|$ i $e = 0$. Zbog $m = 0.1 < 1$,

🔴 binarnu točku u broju m treba pomicati **udesno**, tj.

🔴 m treba **množiti** s 2, a e **smanjivati** za 1.

korak	$m \leftarrow m \cdot 2$	$e \leftarrow e - 1$
1	$0.1 \cdot 2 = 0.2 < 1$	$0 - 1 = -1$
2	$0.2 \cdot 2 = 0.4 < 1$	$-1 - 1 = -2$
3	$0.4 \cdot 2 = 0.8 < 1$	$-2 - 1 = -3$
4	$0.8 \cdot 2 = 1.6 \geq 1$	$-3 - 1 = -4$

Normalizirani prikaz broja 0.1 u bazi 2

Rješenje. Normalizirani prikaz broja 0.1 u bazi 2 je

$$(0.1)_{10} = 1 \cdot 1.6 \cdot 2^{-4}.$$

Po dijelovima: $s = 1$, $m = 1.6$ i $e = -4$.

Binarni prikaz mantise:

☞ $b_0 = 1$, a Algoritam 2 na broju $m - b_0 = 0.6$ daje

$$0.6 = (0.\dot{1}00\dot{1})_2$$

(provjerite), pa je $m = 1.6 = (1.\dot{1}00\dot{1})_2$.

Binarni prikaz eksponenta:

☞ Algoritam 1 na broju $|e| = 4$ daje $4 = (100)_2$, pa je
 $e = -4 = (-100)_2$.

Prikaz realnih brojeva u računalu i primjeri

Sadržaj

- Realni brojevi — prikaz u računalu:
 - Izgled prikaza i prikazivi brojevi.
 - Zaokruživanje realnih brojeva u računalu za prikaz.
 - Prikaz u tipovima `float` i `double` — primjeri.

Prikaz realnih brojeva u računalu — uvod

Prikaz realnih brojeva u računalu sličan je normaliziranom prikazu realnih brojeva u bazi 2

$$B = s \cdot m \cdot 2^e.$$

Bitne razlike:

- za svaki od tri dijela s , m i e imamo na raspolaganju konačan broj binarnih znamenki (bitova) za prikaz,
- umjesto eksponenta, pamti se karakteristika k , koja je pomaknuti eksponent, $k = e + bias$ (engl. bias = pomak),
- umjesto mantise, pamti se signifikand (oznaka je isto m), a dobiva se nakon zaokruživanja mantise,
- vodeći bit b_0 mantise se, najčešće, ne pamti (hidden bit).

Dodatno, neki brojevi se prikazuju u denormaliziranom obliku.

Realni brojevi u računalu — duljine dijelova

Svaki od **tri dijela** broja ima svoju **duljinu** — broj bitova predviđenih za prikaz tog dijela.

- **predznak** s — uvijek zauzima **jedan** bit, i to **najviši**;
- **karakteristika** k — zauzima sljedećih w bitova;
- **signifikand** m — zauzima sljedećih
 - t bitova, ako se pamti samo **razlomljeni** dio **mantise**,
 - $t + 1$ bit, ako se pamti i njezin **vodeći cjelobrojni** bit.

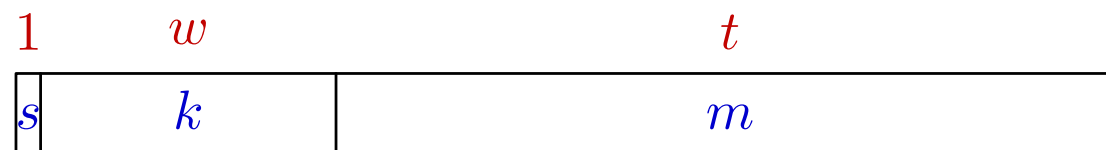
U nastavku, radi jednostavnosti, uzimamo da se u računalu

- sprema samo **razlomljeni** dio **mantise**,

kao u standardnim **IEEE 754** tipovima **binary32** i **binary64** (tipovi **float** i **double** u C-u).

Realni brojevi u računalu — izgled i predznak

Cijeli prikaz realnog broja u računalu onda ima sljedeći oblik:



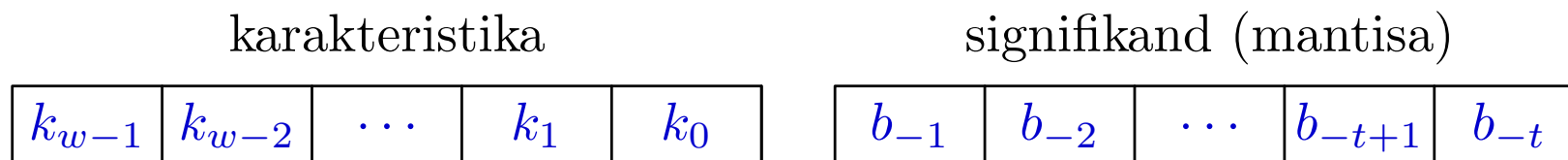
Bit predznaka s uvijek se kôdira tako da je

• $s = 0$ — za pozitivan broj ili $+0$,

• $s = 1$ — za negativan broj ili -0 .

tj., u ranijim oznakama je predznak $= (-1)^s \in \{-1, 1\}$.

Po bitovima, karakteristika i signifikand imaju sljedeći izgled:



gdje su $k_i, b_{-i} \in \{0, 1\}$. Precizan opis značenja ide u nastavku.

Karakteristika

Karakteristika k

karakteristika

k_{w-1}	k_{w-2}	$\dots$	k_1	k_0
-----------	-----------	---------	-------	-------

se interpretira kao cijeli broj bez predznaka,

$$k = \sum_{i=0}^{w-1} k_i \cdot 2^i,$$

tako da je $k \in \{0, \dots, 2^w - 1\}$.

“Rubne” vrijednosti za k označavaju tzv. posebna stanja:

🔴 $k_{\min} = 0$ — nula i denormalizirani brojevi,

🔴 $k_{\max} = 2^w - 1$ — beskonačno (Inf) i “nije broj” (NaN).

Karakteristika i stvarni eksponent

Sve ostale vrijednosti $k \in \{1, \dots, 2^w - 2\}$ koriste se za prikaz normaliziranih brojeva (različitih od nule).

U tom slučaju, veza između karakteristike k i stvarnog eksponenta e prikazanog broja je

$$k = e + bias, \quad bias = 2^{w-1} - 1.$$

Uočite da je $bias$ = polovina najveće dozvoljene karakteristike.

Dakle, eksponenti e prikazivih normaliziranih brojeva nalaze se između

$$e_{\min} = -(2^{w-1} - 2) \quad \text{i} \quad e_{\max} = 2^{w-1} - 1.$$

Stvarni eksponent e svih prikazivih denormaliziranih brojeva je $e = e_{\min} = -(2^{w-1} - 2)$, a pripadna karakteristika je $k = 0$.

Signifikand

Signifikand m ima oblik

signifikand (mantisa)

b_{-1}	b_{-2}	$\cdots$	b_{-t+1}	b_{-t}
----------	----------	----------	------------	----------

Katkad se ovi bitovi indeksiraju **pozitivnim** indeksima, tako da je $m_i = b_{-i}$

signifikand (mantisa)

m_1	m_2	$\cdots$	m_{t-1}	m_t
-------	-------	----------	-----------	-------

Ako je $k < k_{\max} = 2^w - 1$, onda se **signifikand** m interpretira  kao **razlomljeni** dio m_r stvarne **mantise** prikazivog broja

$$m_r = \sum_{i=1}^t b_{-i} \cdot 2^{-i} = \sum_{i=1}^t m_i \cdot 2^{-i} \in [0, 1).$$

Signifikand i stvarna mantisa

Stvarna mantisa dobiva se dodavanjem “skrivenog bita” b_0 (engl. “hidden bit”), kao cjelobrojnog dijela mantise broja:

$$m = b_0 + m_r = b_0 + \sum_{i=1}^t b_{-i} \cdot 2^{-i} = (b_0.b_{-1}b_{-2} \dots b_{-t})_2,$$

s tim da je “skriveni bit”

• $b_0 = 1$ — za normalizirane brojeve, tj. mantisa im je

$$m = 1 + \sum_{i=1}^t b_{-i} \cdot 2^{-i} = (1.b_{-1}b_{-2} \dots b_{-t})_2,$$

• $b_0 = 0$ — za denormalizirane brojeve, tj. mantisa im je

$$m = 0 + \sum_{i=1}^t b_{-i} \cdot 2^{-i} = (0.b_{-1}b_{-2} \dots b_{-t})_2.$$

Skupovi prikazivih realnih brojeva u računalu

Skup svih prikazivih realnih brojeva u računalu je konačan i parametriziran je s dva parametra:

- duljinom t signifikanda ili mantise,
- duljinom w karakteristike ili eksponenta.

Označavamo ga s $\mathbb{R}(t, w)$.

Možemo ga prikazati kao uniju dva disjunktna podskupa:

$$\mathbb{R}(t, w) = \mathbb{F}(t, w) \cup \mathbb{D}(t, w),$$

gdje je

- $\mathbb{F}(t, w)$ = skup svih normaliziranih prikazivih brojeva, ili skup svih prikazivih brojeva u normaliziranom rasponu,
- $\mathbb{D}(t, w)$ = skup svih denormaliziranih prikazivih brojeva.

Normalizirani prikazivi brojevi

Normalizirani prikazivi brojevi $B \in \mathbb{F}(t, w)$ prepoznaju se po karakteristici k za koju vrijedi $k_{\min} < k < k_{\max}$. Imaju oblik

$$B = (-1)^s \cdot m \cdot 2^e,$$

gdje je:

- bit predznaka $s \in \{0, 1\}$,
- stvarna mantisa $m = (1.b_{-1}b_{-2} \dots b_{-t})_2$,
- stvarni eksponent $e = k - 2^{w-1} + 1$, a nalazi se između $e_{\min}$ i $e_{\max}$, tj. vrijedi $e \in \{-(2^{w-1} - 2), \dots, 2^{w-1} - 1\}$.

Skup normaliziranih prikazivih brojeva $\mathbb{F}(t, w)$ ima točno

$$2 \cdot 2^t \cdot (2^w - 2) = 2^{p+1} \cdot bias$$

elemenata (uz oznaku $p = t + 1$).

Normalizirani prikazivi brojevi — primjeri

Primjer. Najveći pozitivni normalizirani broj je

$$\begin{aligned} v_{\max} &= (1.\overset{\uparrow}{1}111\dots111\overset{\uparrow}{t})_2 \cdot 2^{e_{\max}} \\ &= (1 + (1 - 2^{-t})) \cdot 2^{e_{\max}} = 2(1 - 2^{-t-1}) \cdot 2^{e_{\max}} \\ &= (1 - 2^{-p}) \cdot 2^{e_{\max}+1} = (1 - 2^{-p}) \cdot 2^{2^w-1}. \end{aligned}$$

Primjer. Najmanji pozitivni normalizirani broj je

$$\begin{aligned} v_{\min} &= (1.\overset{\uparrow}{1}000\dots000\overset{\uparrow}{t})_2 \cdot 2^{e_{\min}} \\ &= 2^{e_{\min}} = 2^{-(2^w-1-2)}. \end{aligned}$$

Normalizirani prikazivi brojevi — izbor pomaka

Već smo rekli da za **normalizirane** prikazive brojeve vrijedi sljedeća veza između **karakteristike** k i stvarnog **eksponenta** e

$$k = e + \textit{bias}, \quad \textit{bias} = 2^{w-1} - 1.$$

U ovoj vezi, pomak $\textit{bias}$ je **namjerno izabran** tako da vrijedi

$$e_{\min} + e_{\max} = 1,$$

odnosno, $e_{\max} = 1 - e_{\min} > -e_{\min}$, a **ne** obratno!

Razlog. Ovaj izbor povlači da je

$$\frac{1}{v_{\min}} = 2^{-e_{\min}} \in \mathbb{F}(t, w),$$

tj. **recipročna** vrijednost **najmanjeg** pozitivnog **normaliziranog** broja je, također, **normalizirani** broj.

Denormalizirani prikazivi brojevi

Denormalizirani prikazivi brojevi $B \in \mathbb{D}(t, w)$ prepoznaju se po karakteristici k za koju vrijedi $k = k_{\min} = 0$. Imaju oblik

$$B = (-1)^s \cdot m \cdot 2^e,$$

gdje je:

- bit predznaka $s \in \{0, 1\}$,
- stvarna mantisa $m = (0.b_{-1}b_{-2} \dots b_{-t})_2$, što uključuje i nule oba predznaka — za mantise $m = 0$,
- stvarni eksponent $e = e_{\min} = -(2^{w-1} - 2)$.

Skup denormaliziranih prikazivih brojeva $\mathbb{D}(t, w)$ ima točno

$$2 \cdot 2^t = 2^p$$

elemenata (uz oznaku $p = t + 1$).

Denormalizirani prikazivi brojevi — primjeri

Primjer. Najveći pozitivni denormalizirani broj je

$$\begin{aligned} v_{\max}^d &= (0.\underset{\substack{\uparrow \\ 1}}{1}11 \dots 11\underset{\substack{\uparrow \\ t}}{1})_2 \cdot 2^{e_{\min}} \\ &= (1 - 2^{-t}) \cdot 2^{e_{\min}} = (1 - 2^{-t}) \cdot 2^{-(2^{w-1}-2)}. \end{aligned}$$

Primjer. Najmanji pozitivni denormalizirani broj je

$$\begin{aligned} v_{\min}^d &= (0.\underset{\substack{\uparrow \\ 1}}{0}00 \dots 00\underset{\substack{\uparrow \\ t}}{1})_2 \cdot 2^{e_{\min}} \\ &= 2^{-t} \cdot 2^{e_{\min}} = 2^{-(t+2^{w-1}-2)}. \end{aligned}$$

Karakteristika $k_{\max}$ — beskonačno i “nije broj”

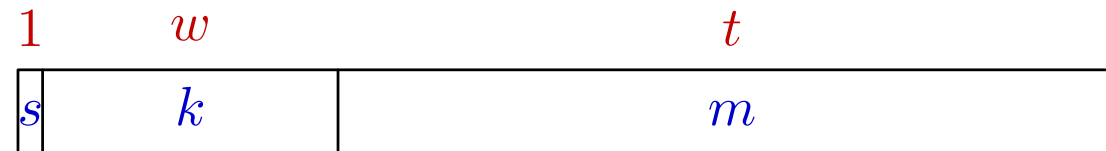
Maksimalna vrijednost karakteristike $k = k_{\max} = 2^w - 1$ označava posebna stanja koja ne odgovaraju “običnim” brojevima, a mogu se dogoditi prilikom zaokruživanja ili kao rezultat aritmetičkih operacija.

Interpretacija “stanja” ovisi o vrijednosti signifikanda m .

- $m = 0$, tj. svi bitovi su nula — spremljena vrijednost se interpretira kao beskonačno (Inf), a ovisi o predznaku
 - $s = 0$ — prikaz $+\infty$, skraćena oznaka +Inf,
 - $s = 1$ — prikaz $-\infty$, skraćena oznaka -Inf.
- $m \neq 0$, tj. postoji bit različit od nule — vrijednost se interpretira kao “nije broj” (NaN, engl. “Not a Number”). Nastaje kao rezultat nedozvoljene aritmetičke operacije i uvijek označava grešku.

Prikaz realnih brojeva u računalu — sažetak

Uzmimo da prikaz realnih brojeva u računalu ima oblik



Uz oznake

$$k_{\max} = 2^w - 1, \quad e_{\max} = bias = 2^{w-1} - 1, \quad e_{\min} = 1 - e_{\max},$$

vrijednost prikazanog broja je

$$v = \begin{cases} (-1)^s * 2^{(k-e_{\max})} * (1.m) & \text{ako je } 0 < k < k_{\max}, \\ (-1)^s * 2^{e_{\min}} * (0.m) & \text{ako je } k = 0 \text{ i } m \neq 0, \\ (-1)^s * 0 & \text{ako je } k = 0 \text{ i } m = 0, \\ (-1)^s * \text{Inf} & \text{ako je } k = k_{\max} \text{ i } m = 0, \\ \text{NaN} & \text{ako je } k = k_{\max} \text{ i } m \neq 0. \end{cases}$$

Prikaz realnih brojeva u računalu i zaokruživanje

Sada znamo točno kako izgledaju **svi prikazivi** brojevi u **računalu** — skup $\mathbb{R}(t, w)$.

Međutim, u praksi obično **ne** krećemo od **prikazivih** brojeva, već od standardnih **realnih** brojeva iz skupa $\mathbb{R}$, čak $\mathbb{R} \cup \{-0\}$ (v. stranicu iza). A onda nam fali još jedna, ali **ključna** stvar:

🔴 Što se događa kad imamo **zadan** realni broj $B \in \mathbb{R}$ — kako se on “**prikazuje**” u računalu?

Naime, broj B , naravno, **ne mora** biti **prikaziv** — pripadati skupu $\mathbb{R}(t, w)$!

Ukratko, ako broj B nije prikaziv, onda se u računalu sprema

- 🔴 njegova **najbolja** prikaziva **aproksimacija**, u oznaci $fl(B)$,
- 🔴 a dobiva se **zaokruživanjem** polaznog broja.

Proširenje polaznog skupa na $\mathbb{R} \cup \{-0\}$ i oznake

U nastavku, uzimamo da polazni broj B pripada većem skupu $\mathbb{R} \cup \{-0\}$. Zašto smo dodali i mogućnost da je $B = -0$?

Ako zamislimo da se broj B učitava, onda

• svakom realnom broju smijemo napisati predznak, pa to vrijedi i za nulu, tj. smijemo napisati i broj $B = -0$.

Ovo učitavanje zaista korektno radi, zbog načina pretvaranja zapisa broja na ulazu u prikazivi broj. Naime, ako napišemo

• predznak “ $-$ ” — to se uredno zapamti pri čitanju, i na kraju spremi kao predznak dobivenog prikazivog broja.

Kao rezultat toga, uvedimo još i sljedeće oznake:

• B ima predznak “minus” — $B \leq -0$,

• B ima predznak “plus” — $B \geq +0$.

Zaokruživanje realnih brojeva u računalu

Za precizniji opis, zgodno je **zaokruživanje** gledati kao funkciju

$$fl : \mathbb{R} \cup \{-0\} \rightarrow \overline{\mathbb{R}}(t, w) = \mathbb{R}(t, w) \cup \{-\text{Inf}, +\text{Inf}\}.$$

Prvi korak u dobivanju **prikazive** aproksimacije $fl(B)$ je

● nalaženje **najbližeg** lijevog i desnog **prikazivog** susjeda, tj. najbližih brojeva $B_-, B_+ \in \overline{\mathbb{R}}(t, w)$ za koje vrijedi

$$B_- \leq B \leq B_+.$$

Preciznije rečeno,

● $B_- =$ **najveći** broj iz $\overline{\mathbb{R}}(t, w)$ **manji** ili **jednak** B ,

● $B_+ =$ **najmanji** broj iz $\overline{\mathbb{R}}(t, w)$ **veći** ili **jednak** B .

Onda vrijedi: broj B je **prikaziv** ako i samo ako je $B_- = B_+$!

Zaokruživanje realnih brojeva u računalu (nast.)

Zbog toga što skup “prikazivih” brojeva $\overline{\mathbb{R}}(t, w)$ sadrži brojeve $+0$, -0 , $+\text{Inf}$ i $-\text{Inf}$, ova dva broja B_- , B_+

- uvijek postoje i jednoznačno su definirani,
- imaju isti predznak kao i broj B .

Za pozitivne brojeve može biti $B_- = +0$ ili $B_+ = +\text{Inf}$.

Analogno za negativne brojeve, samo na suprotnu stranu.

Kako se stvarno nalaze brojevi B_- , B_+ — o tome malo kasnije.

Kad imamo najbliže prikazive susjede B_- , $B_+ \in \overline{\mathbb{R}}(t, w)$, za prikazivu aproksimaciju broja B , tj. za zaokruženi broj B

- uvijek se uzima jedan od ta dva broja, ovisno o vrsti ili tipu zaokruživanja.

Vrste zaokruživanja realnih brojeva u računalu

Po IEEE 754 standardu, postoje četiri vrste zaokruživanja.

Standardno (engl. “default”) zaokruživanje za sve procesore je

- prema najbližem broju, a ako su dva najbliža, onda prema “parnom” (engl. “ties to even”) — oznaka fl ,

$$fl(B) = \text{“bliži” od brojeva } B_-, B_+.$$

Ako su oba susjeda B_- i B_+ jednako “udaljena” od B ,

- izabire se onaj koji ima parni zadnji bit, tj. zadnji bit mu je jednak 0 (tačno jedan od brojeva je takav).

Napomena. Ako je neki od brojeva B_- i B_+ jednak $\pm Inf$, “udaljenost” se mjeri po tome koliko treba dodati ili oduzeti od B u aritmetici računala da dobijemo $\pm Inf$ (v. malo iza).

Vrste zaokruživanja realnih brojeva (nastavak)

Preostale tri vrste zaokruživanja možemo “zatražiti” opcijama prevoditelju, ili postavljanjem kontrolnih bitova u procesoru:

— prema dolje, ili prema $-\infty$ — oznaka fl_-

$$fl_-(B) = B_-,$$

— prema gore, ili prema $+\infty$ — oznaka fl_+

$$fl_+(B) = B_+,$$

— prema nuli — oznaka fl_0

$$fl_0(B) = \begin{cases} B_-, & \text{ako je } B \geq +0, \\ B_+, & \text{ako je } B \leq -0. \end{cases}$$

Ovo odgovara odbacivanju “viška” znamenki u B .

Uvod u primjere za standardno zaokruživanje

Za potpuni opis zaokruživanja realnih brojeva trebalo bi još:

- opisati kako se za zadani broj $B \in \mathbb{R} \cup \{-0\}$,
 - stvarno nalaze najbliži prikazivi susjedi B_- i B_+ ,
- uključivo i pripadne algoritme.

Nažalost, precizan opis nije jednostavan i zahtijeva dosta “tehničkog” posla na detaljima.

Prije toga, zgodno je “neformalno” prikazati

- kako se radi standardno zaokruživanje fl u računalu, i
- ilustrirati to na primjerima standardnih tipova `float` i `double` u C-u.

Standardno zaokruživanje — kratki opis

Definiciju funkcije fl možemo ugrubo podijeliti u tri grupe. Podjela ovisi o odnosu broja $|B|$ i dozvoljenog raspona za normalizirani prikaz broja u računalu. Preciznije, o odnosu

ekspONENTA e i “graničnih” ekspONENTATA $e_{\min}$ i $e_{\max}$.

Imamo tri mogućnosti.

- Ako je $e > e_{\max}$, onda je $|B|$ prevelik (tzv. “overflow”), pa je $fl(B) = s \cdot \text{Inf}$.
- Ako je $e_{\min} \leq e \leq e_{\max}$, onda je $|B|$ unutar raspona za normalizirani prikaz, pa se stvarna mantisa m zaokružuje na t mjesta. Usput, onda definiramo $m' = m$.
- Ako je $e < e_{\min}$, onda je $|B|$ premali (tzv. “gradual underflow”), pa se broj denormalizira na eksponent $e_{\min}$, a tako dobivena mantisa $m' < 1$ zaokružuje na t mjesta.

Standardno zaokruživanje — malo neprecizno

Standardno zaokruživanje fl brojeva u računalu (do na pravilo o “dva najbliža”) možemo i ovako opisati.

Broj $fl(B)$ dobivamo zaokruživanjem iz broja B na sljedeći način: ako je prva odbačena znamenka b_{-t-1} mantise m'

- jednaka 1 — zaokruži m' nagore,
- jednaka 0 — zaokruži m' nadolje.

Kod zaokruživanja nagore, može se desiti da dobiveni broj treba “renormalizirati” (i tada možemo dobiti rezultat $\pm Inf$).

Ovo “neprecizno” pravilo može biti pogrešno samo ako je

- $b_{-t-1} = 1$ zadnja ne-nula znamenka mantise m' .

Tad vrijedi pravilo o dva najbliža: zaokruži prema “parnom”.

Prikaz broja 0.1 kao float

Primjer. Prikaz broja 0.1 kao float ($t = 23$, $w = 8$).

$$(0.1)_{10} = +(\underbrace{1}_{\uparrow 1}.1001 \ 1001 \ \dots \ 100\underbrace{1}_{\uparrow 23} \ 1001 \dots)_2 \cdot 2^{-4}.$$

Zaokruživanje iza $t = 23$ bita razlomljenog dijela (**naviše**) daje

$$s = 0$$

$$k = e + 2^{w-1} - 1 = -4 + 127 = (123)_{10} = 0111 \ 1011$$

$$m = 1001 \ 1001 \ 1001 \ 1001 \ 1001 \ 101$$

Prikaz broja 0.1 [float] u racunalu:

0 01111011 10011001100110011001101

Prikaz broja 0.1 kao float (nastavak)

Na IA-32 se **viši** bitovi nalaze na **višim** adresama, pa prikaz po **byteovima** (odnosno, **riječima** = 4 bytea) izgleda ovako:

Prikaz broja 0.1 [float] u racunalu:

- 1. byte: 1100 1101
- 2. byte: 1100 1100
- 3. byte: 1100 1100
- 4. byte: 0011 1101

1. rijec: 0011 1101 1100 1100 1100 1100 1100 1101

Prikaz broja 0.1 kao double

Primjer. Prikaz broja 0.1 kao double ($t = 52$, $w = 11$).

$$(0.1)_{10} = +(\underbrace{1}_{\uparrow 1}.1001 \ 1001 \ \dots \ 1001 \underbrace{1001}_{\uparrow 52} \dots)_2 \cdot 2^{-4}.$$

Zaokruživanje iza $t = 52$ bita razlomljenog dijela (**naviše**) daje

$$s = 0$$

$$k = e + 2^{w-1} - 1 = -4 + 1023 = (1019)_{10} = 011 \ 1111 \ 1011$$

$$m = 1001 \ 1001 \ 1001 \ 1001 \ 1001 \ 1001 \ 1001$$

$$1001 \ 1001 \ 1001 \ 1001 \ 1001 \ 1010$$

Prikaz broja 0.1 [double] u racunalu:

```
0 01111111011 10011001100110011001
   10011001100110011001100110011010
```

Prikaz broja 0.1 kao double (*nastavak*)

Prikaz po byteovima (odnosno, riječima) izgleda ovako:

Prikaz broja 0.1 [double] u racunalu:

1. byte: 1001 1010
2. byte: 1001 1001
3. byte: 1001 1001
4. byte: 1001 1001
5. byte: 1001 1001
6. byte: 1001 1001
7. byte: 1011 1001
8. byte: 0011 1111

1. rijec: 1001 1001 1001 1001 1001 1001 1001 1010
2. rijec: 0011 1111 1011 1001 1001 1001 1001 1001

Prikaz realnih brojeva — zaokruživanje i greške

Sadržaj

- Prikaz realnih brojeva — zaokruživanje i greške (još nije sređeno!):
 - Nalaženje najbližih prikazivih brojeva.
 - Greške zaokruživanja u prikazu.

Nalaženje prikazivih susjeda B_- i B_+

Na kraju, opišimo još **kako** se za zadani broj $B \in \mathbb{R} \cup \{-0\}$

🔴 stvarno **nalaze** najbliži prikazivi susjedi B_- i B_+ .

Polazna točka je **normalizirani** prikaz broja B u bazi 2.

Jednostavnim **proširenjem** ranije definicije, svaki “ulazni” broj $B \in \mathbb{R} \cup \{-0\}$ ima **jednoznačan normalizirani** prikaz u bazi 2

$$B = s \cdot |B|, \quad |B| = (1) \cdot m \cdot 2^e,$$

s tim da je predznak $s \in \{-1, 1\}$, a za **egzaktanu** mantisu m i eksponent e vrijedi

🔴 ako je $|B| > 0$, onda je $m \in [1, 2)$ i $e \in \mathbb{Z}$,

🔴 ako je $|B| = 0$, onda je $m = 0.0$ i $e = 0$.

Dodavanje broja -0 daje potpunu **simetriju** skupa oko **nule**!

Predznak prikazivih susjeda B_- i B_+

Ponovimo, najbliži prikazivi susjedi broja B su definirani kao

• B_- = najveći broj iz $\overline{\mathbb{R}}(t, w)$ manji ili jednak B ,

• B_+ = najmanji broj iz $\overline{\mathbb{R}}(t, w)$ veći ili jednak B .

Skup $\overline{\mathbb{R}}(t, w)$ sadrži brojeve -0 i $+0$, pa traženi susjedi

• B_- i B_+ uvijek imaju isti predznak s kao i broj B .

Osim toga, $\overline{\mathbb{R}}(t, w)$ je, također, potpuno simetričan oko nule.

Neka su $|B|_-$ i $|B|_+$ najbliži prikazivi susjedi broja $|B|$. Za sve “negativne” brojeve $B \leq -0$ onda vrijedi

$$B_- = -(|B|_+), \quad B_+ = -(|B|_-),$$

pa njihove apsolutne vrijednosti $|B_-|$ i $|B_+|$ možemo naći iz broja $|B|$.

Nalaženje B_- i B_+ iz $|B|_-$ i $|B|_+$

Zato nalaženje najbližih prikazivih susjeda B_- i B_+ broja B

🔴 kreće od broja $|B|$ i prvo se nalaze njegovi prikazivi susjedi $|B|_-$ i $|B|_+$.

To su, ujedno, i apsolutne vrijednosti najbližih prikazivih susjeda broja B . Njih nalazimo koristeći predznak od B .

🔴 Ako je $B \geq +0$, onda je

$$B_- = |B|_-, \quad B_+ = |B|_+.$$

🔴 Ako je $B \leq -0$, onda je

$$B_- = -|B|_+, \quad B_+ = -|B|_-.$$

Dakle, na samom kraju, ako je $B \leq -0$, nađenim brojevima $|B|_-$ i $|B|_+$ se dodaje negativan predznak i “okreće” značenje.

Nalaženje brojeva $|B|_-$ i $|B|_+$ iz $|B|$

Za svaki broj $B \in \mathbb{R} \cup \{-0\}$, njegova apsolutna vrijednost $|B|$ ima jednoznačan normalizirani prikaz u bazi 2 (uz $s = 1$)

$$|B| = m \cdot 2^e,$$

s tim da za egzaktnu mantisu m i eksponent e vrijedi

• ako je $|B| > 0$, onda je $m \in [1, 2)$ i $e \in \mathbb{Z}$,

• ako je $|B| = 0$, onda je $m = 0.0$ i $e = 0$.

Brojevi B i $|B|$ imaju iste mantise i iste eksponente, a mogu se razlikovati samo u predznaku s .

Za početak, ako je $|B| = 0$, tj. $m = 0.0$, onda je

$$|B|_- = |B|_+ = +0 \in \mathbb{D}(t, w).$$

Broj B je egzaktno prikaziv i tu nema greške zaokruživanja.

Nalaženje brojeva $|B|_-$ i $|B|_+$ iz $|B|$ (nastavak)

Pretpostavimo nadalje da je $|B| \neq 0$. Onda je

$$|B| = m \cdot 2^e, \quad m \in [1, 2), \quad e \in \mathbb{Z}.$$

Nalaženje brojeva $|B|_-$ i $|B|_+$ ovisi o odnosu eksponenta e i “graničnih” eksponenata $e_{\min}$ i $e_{\max}$. Imamo tri mogućnosti.

- 🔴 Ako je $e > e_{\max}$, onda je $|B|$ prevelik (tzv. “overflow”). Ovaj slučaj razmatramo na kraju.
- 🔴 Ako je $e_{\min} \leq e \leq e_{\max}$, onda je $|B|$ unutar raspona za normalizirani prikaz.
- 🔴 Ako je $e < e_{\min}$, onda je $|B|$ premali (tzv. “gradual underflow”), pa se broj denormalizira na eksponent $e_{\min}$

$$|B| = m' \cdot 2^{e_{\min}}, \quad m' = m \cdot 2^{e - e_{\min}} \in \langle 0, 1 \rangle.$$

Nalaženje brojeva $|B|_-$ i $|B|_+$ iz $|B|$ (nastavak)

Uzmimo da broj $|B|$ nije prevelik, tj. da je $e \leq e_{\max}$ i neka je $|B|$ već denormaliziran (ako je $e < e_{\min}$), tako da je

$$|B| = m' \cdot 2^{e'},$$

s tim da vrijedi:

- za $e \geq e_{\min}$ — $m' = m$ i $e' = e$,
- za $e < e_{\min}$ — $m' = m \cdot 2^{e-e_{\min}}$ i $e' = e_{\min}$.

Prikazive susjede $|B|_-$ i $|B|_+$ dobivamo iz ovog zapisa $|B|$,

- zaokruživanjem njegove “mantise” m' , odnosno, njezinog razlomljenog dijela, na t mjesta iza binarne točke,
- i to: nadolje — za $|B|_-$, a nagore — za $|B|_+$, uz eventualnu dodatnu normalizaciju broja $|B|_+$.

Nalaženje brojeva $|B|_-$ i $|B|_+$ iz $|B|$ (nastavak)

Za ovako dobivenu egzaktnu “mantisu” m' od $|B|$ vrijedi

• $m' \in [1, 2)$, za $e \geq e_{\min}$,

• $m' \in \langle 0, 1 \rangle$, za $e < e_{\min}$, nakon denormalizacije broja $|B|$.

U bazi 2, realni broj m' onda možemo prikazati u obliku

$$m' = b_0 + \sum_{i=1}^{\infty} b_{-i} \cdot 2^{-i}, \quad b_0, b_{-i} \in \{0, 1\}.$$

Ovaj zapis služi samo za ilustraciju, jer znamenke ne znamo (trebalo bi ih naći nekim algoritmom).

Stvarno, znamo samo cjelobrojni bit b_0 — iz granica za m' :

• $b_0 = 1$, za $e \geq e_{\min}$,

• $b_0 = 0$, za $e < e_{\min}$, tj. ako je $|B|$ bio denormaliziran.

Nalaženje brojeva $|B|_-$ i $|B|_+$ iz $|B|$ (nastavak)

Pogledajmo sad razlomljeni dio broja m'

$$m'_r = \sum_{i=1}^{\infty} b_{-i} \cdot 2^{-i} = (0.b_{-1}b_{-2} \dots)_2.$$

Tu imamo dvije mogućnosti, ovisno o broju znamenki u m'_r .

Ako m'_r ima najviše t binarnih znamenki iza točke, onda

🔴 zaokruživanje m'_r na t mjesta ništa ne mijenja,
tj. zaokruživanje nije potrebno i nema greške, pa je

$$|B|_- = |B|_+ = |B| = m' \cdot 2^{e'}.$$

Dakle, broj $|B|$ (a onda i B) je egzaktno prikaziv u računalu, a karakteristika i signifikand se dobivaju direktno iz e' i m'_r .

Nalaženje brojeva $|B|_-$ i $|B|_+$ iz $|B|$ (nastavak)

Ako m'_r ima više od t binarnih znamenki iza točke, onda

☞ broj $|B|$ nije egzaktno prikaziv u računalu, pa dolazi do zaokruživanja i pripadne greške.

Razlomljeni dio m'_r onda ima oblik

$$m'_r = \sum_{i=1}^t b_{-i} \cdot 2^{-i} + \sum_{i=t+1}^{\infty} b_{-i} \cdot 2^{-i} = (0.b_{-1} \dots b_{-t} b_{-t+1} \dots)_2.$$

s tim da je druga suma pozitivna

$$\sum_{i=t+1}^{\infty} b_{-i} \cdot 2^{-i} > 0.$$

Najbliže prikazive brojeve broju $|B|$ dobivamo zaokruživanjem m'_r nadolje i nagore na t mjesta iza binarne točke.

Prikazivi manji susjed $|B|_-$ za $|B|$

Zaokruživanjem m'_r nadolje na t mjesta iza binarne točke, tj. odbacivanjem drugog pozitivnog člana, dobivamo

$$(m'_r)_- := \sum_{i=1}^t b_{-i} \cdot 2^{-i} = (0.b_{-1} \dots b_{-t})_2 < m'_r.$$

Ovo je signifikand broja $|B|_-$, a pripadna prikaziva mantisa je

$$m'_- := b_0 + (m'_r)_- = (b_0.b_{-1} \dots b_{-t})_2 < m'.$$

Na kraju, najbliži prikazivi manji susjed $|B|_-$ broja $|B|$ je

$$|B|_- := m'_- \cdot 2^{e'} < |B|.$$

Njegova karakteristika se dobiva direktno iz eksponenta e' .

Primjer — Prikazivi manji susjed $|B|_- = +0$

Primjer. Može se dogoditi da je $|B|_- = +0$, iako je $|B| > 0$.
To se događa **ako i samo ako** je

$$0 < |B| < v_{\min}^d,$$

gdje je $v_{\min}^d \in \mathbb{D}(t, w)$ **najmanji** pozitivni **denormalizirani** broj.

Ranije smo pokazali da je $v_{\min}^d = 2^{-t} \cdot 2^{e_{\min}}$. Ako je $|B| < v_{\min}^d$,
onda je broj sigurno **premali** za normalizirani prikaz. Nakon
denormalizacije broja $|B|$ na eksponent $e_{\min}$, dobivamo

$$|B| = m' \cdot 2^{e_{\min}}, \quad 0 < m' < 2^{-t}, \quad b_0 = 0.$$

Onda je

$$m' = m'_r = (0.\underset{\substack{\uparrow \\ 1}}{0}000 \dots 000 \underset{\substack{\uparrow \\ t}}{0} \dots)_2,$$

odakle odmah slijedi $(m'_r)_- = m'_- = 0$, pa je $|B|_- = +0$.

Prikazivi veći susjed $|B|_+$ za $|B|$

Za zaokruživanje **nagore** koristimo ocjenu druge sume **odozgo**

$$\sum_{i=t+1}^{\infty} b_{-i} \cdot 2^{-i} < 2^{-t}.$$

Zaokruživanje m'_r **nagore** na t mjesta iza binarne točke odgovara **povećanju** “zadnjeg prikazivog” bita b_{-t} za **1**, pa je

$$(m'_r)_+ := \sum_{i=1}^t b_{-i} \cdot 2^{-i} + 2^{-t} > m'_r.$$

Nakon ovog **zbrajanja**, može se dogoditi da je $(m'_r)_+ = 1$, pa

 ovaj broj $(m'_r)_+$ više **nije signifikand** prikazivog broja.

Zato mu ne pišemo znamenke u bazi **2**. Međutim, $(m'_r)_+$ je uvijek **egzaktno** prikaziv pa nastavljamo konstrukciju.

Prikazivi veći susjed $|B|_+$ za $|B|$ (nastavak)

Analogno ranijem, pripadna “mantisa” broja $|B|_+$ je

$$m'_+ := b_0 + (m'_r)_+ > m'.$$

Na kraju, najbliži prikazivi veći susjed $|B|_+$ broja $|B|$ je

$$|B|_+ := m'_+ \cdot 2^{e'} > |B|.$$

Ako je $(m'_r)_+ = 1$, onda je $m'_+ = b_0 + 1$, pa su brojevi m'_+ i $|B|_+$ uvijek egzaktno prikazivi.

Posebno, ako dobijemo da je $m'_+ = 2$, onda broj $|B|_+$ treba dodatno normalizirati povećanjem eksponenta za 1

$$|B|_+ = 1 \cdot 2^{e'+1},$$

i tad se može dogoditi da je $|B|_+ = +\text{Inf}$.

Primjer — Gornji razlomljeni dio $(m'_r)_+ = 1$

Primjer. Nađimo za koje brojeve $|B| > 0$ se dobiva da je $(m'_r)_+ = 1$, odnosno, $m'_+ = b_0 + 1$.

To znači da u razlomljenom dijelu m'_r broja $|B|$

 **povećanjem** “zadnjeg prikazivog” bita b_{-t} za **1** dobivamo broj koji je **veći** od **1**, ili

$$1 = (m'_r)_+ = \sum_{i=1}^t b_{-i} \cdot 2^{-i} + 2^{-t} > m'_r > \sum_{i=1}^t b_{-i} \cdot 2^{-i}$$

pa je $m'_r > 1 - 2^{-t}$.

$$m'_r > 1 - 2^{-t} = (0.\underset{\substack{\uparrow \\ 1}}{1}111 \dots 111\underset{\substack{\uparrow \\ t}}{1})_2.$$

Primjer — Renormalizacija većeg susjeda $|B|_+$

Primjer — Prikazivi veći susjed $|B|_- = +\text{Inf}$

Primjer. Može se dogoditi da je $|B|_+ = +\text{Inf}$, iako je $|B|$ unutar raspona za **normalizirani** prikaz.

Nalaženje brojeva $|B|_-$ i $|B|_+$ iz $|B|$ (nastavak)

Zaokruživanje realnih brojeva — komentar (!)

Napomena. Binarne znamenke brojeva m' i m'_r ne znamo! Provjeru ima li m' ili m'_r više od t binarnih znamenki iza binarne točke treba napraviti

🔴 nekim **algoritmom** koji radi samo s brojevima.

Na primjer, formiramo **realni** broj $m'_r \cdot 2^t$ i njegovo “**najveće cijelo**”, a zatim provjeravamo razliku $m'_r \cdot 2^t - \lfloor m'_r \cdot 2^t \rfloor$.

Umjesto m'_r , možemo to isto napraviti i s brojem m' (samo je $m' \cdot 2^t$ malo veći).

Zaokruživanje realnih brojeva — zadatak (!)

Zadatak. Dokažite da kod **zaokruživanja** razlomljenog dijela m'_r broja $|B|$ **nadolje** i **nagore** na t mjesta iza binarne točke vrijede formule

$$(m'_r)_- = \lfloor m'_r \cdot 2^t \rfloor \cdot 2^{-t}, \quad (m'_r)_+ = \lceil m'_r \cdot 2^t \rceil \cdot 2^{-t}.$$

Obje formule uključuju i slučaj da je $|B|$ **egzaktno** prikaziv, što je ekvivalentno s tim da je $m'_r \cdot 2^t$ **cijeli** broj.

Greške zaokruživanja (nastavak) — *staro*

Za početak, ako je $B = 0$ onda je $fl(B) = +0 \in \mathbb{D}(t, w)$ i tu nema nikakve greške zaokruživanja.

Pogledajmo sad **razlomljeni** dio broja m'

$$m'_r = \sum_{i=1}^{\infty} b_{-i} \cdot 2^{-i}.$$

Tu imamo **dvije** mogućnosti, ovisno o **broju** znamenki u m' .

Ako m'_r ima **najviše** t binarnih znamenki, onda je

- broj B **egzaktno** prikaziv u računalu, pa je $fl(B) = B$, tj. zaokruživanje **nije potrebno** (nema greške).

Karakteristika i signifikand od B dobivaju se odmah iz e' i m' .

Ako m'_r ima **više** od t binarnih znamenki, onda

- broj B **nije egzaktno** prikaziv u računalu, pa dolazi do **zaokruživanja** (i pripadne greške)

Zaokruživanje realnih brojeva — komentar (!)

Napomena. Binarne znamenke brojeva m' i m'_r ne znamo! Provjeru ima li m' ili m'_r više od t binarnih znamenki iza binarne točke treba napraviti

🔴 nekim **algoritmom** koji radi samo s brojevima.

Na primjer, formiramo **realni** broj $m'_r \cdot 2^t$ i njegovo “**najveće cijelo**”, a zatim provjeravamo razliku $m'_r \cdot 2^t - \lfloor m'_r \cdot 2^t \rfloor$.

Umjesto m'_r , možemo to isto napraviti i s brojem m' (samo je $m' \cdot 2^t$ malo veći).

Primjer

Primjer. Recipročna vrijednost $1/v_{\max}$ najvećeg pozitivnog normaliziranog broja

— nije egzaktno prikaziva — zaokruživanjem dobivamo denormalizirani broj.

$$\begin{aligned} v_{\max} &= \underset{\substack{\uparrow \\ 1}}{1} . \underset{\substack{\uparrow \\ t}}{111 \dots 111}_2 \cdot 2^{e_{\max}} \\ &= (1 + (1 - 2^{-t})) \cdot 2^{e_{\max}} = 2(1 - 2^{-t-1}) \cdot 2^{e_{\max}} \\ &= (1 - 2^{-p}) \cdot 2^{e_{\max}+1} = (1 - 2^{-p}) \cdot 2^{2^w-1}. \end{aligned}$$