Wprowadznie

Poniższy dokument jest sprawozdaniem z listy 1 z przedmiotu Obliczenia Naukowe

1 Zadanie 1

Zadanie 1 ma na celu rozpoznanie arytmetyki.

1.1 Epsilon maszynowy

Pierwszym podpunktem zadania było napisać program w języku Julia, który wyznacza iteracyjnie epsilony maszynowe dla typów zmiennoprzecinkowych: Float16, Float32, Float64. Epsilon to najmnijesza liczba > 0.0, która maszynowo następuje po 1.0. Zadanie zostało wykonane poprzez dzielenie przez 2 w pętli potencjalnego epsilona i sprawdzanie czy dodanie go do jedynki będzie skutkowało zwiększeniem wyniku. Oto output programu, porównujący wyznaczone doświadczalnie epsilony do wartości zwracanych przez funckje `eps(Float16)`, `eps(Float32)`, `eps(Float64)`.

```
Float16
iteracyjnie:   0.000977
funkcja eps:   0.000977

Float32
iteracyjnie:   1.1920929e-7
funkcja eps:   1.1920929e-7

Float64
iteracyjnie:   2.220446049250313e-16
funkcja eps:   2.220446049250313e-16
```

Wyniki doświadczenia są za każdym razem precyzyjne i zgodne z wyjściem funkcji `eps()`. Porównując to z danymi zawartymi w pliku `float.h`:

`FLT_EPSILON` $= 1.192092896 \times 10^{-7}$ (odpowiednik Float32)
`DBL_EPSILON` $= 2.2204460492503131 \times 10^{-16}$ (odpowiednik Float64)

Zauważamy, że nasze wyniki wyszły bardzo podobne. (Nie ma odpowiednika Float16)

1.2 Liczba macheps a precyzja arytmetyki

Precyzją arytmetyki nazywamy liczbę opisaną wzorem:

$$\varepsilon = \frac{1}{2}\beta^{1-t}$$

- β to podstawa systemu. W naszym przypadku jest to 2, ponieważ w komputerach liczby zmiennoprzecinkowe reprezentowane są w systemie binarnym.
- t to liczba cyfr mantysy. Mantysa w formacie zmiennoprzecinkowym ma określoną liczbę bitów:
  - `Float16` - 10
  - `Float32` - 23
  - `Float64` - 52

Precyzje arytmetyki wynoszą, więc odpowiednio:

`Float16:` $2^{-1} \cdot 2^{1-10} = 2^{-10}$
`Float32:` $2^{-1} \cdot 2^{1-23} = 2^{-23}$
`Float64:` $2^{-1} \cdot 2^{1-52} = 2^{-52}$

Gdy spradzimy w Julii wartości policzonych przed chwilą precyzji, z dokładnością do określonych typów, otrzymamy:

```
Float16: 0.000977
Float32: 1.1920929e-7
Float64: 2.220446049250313e-16
```

Powyższe wartości precyzji pokrywają się z wcześniej wyznaczonymi epsilonami maszynowymi, zatem nasuwa się prosty wniosek, że dla danej arytmetyki macheps jest równy jej precyzji.

1.3 Liczba maszynowa eta

Drugim podpunktem zadania było iteracyjne wyznaczenie liczby maszynowej eta. Jest to pierwsza liczba maszynowa większa od 0.0. Oto output programu, porównujący wyznaczone doświadczalnie liczby eta do wartości zwracanych przez funckje `nextfloat(Float16(0.0))`, `nextfloat(Float32(0.0))`, `nextfloat(Float64(0.0))`.

```
Float16
iteracyjnie:   6.0e-8
funkcja eps:   6.0e-8

Float32
```

```
iteracyjnie:   1.0e-45
funkcja eps:   1.0e-45

Float64
iteracyjnie:   5.0e-324
funkcja eps:   5.0e-324
```

Znów wyniki doświadczenia są zgodne z wartościami zwracanymi przez dedykowaną do tego funckję.

1.4 Liczba eta a MIN_{sub}

Liczba MIN_{sub} jest najmniejszą liczbą, która jest nieznormalizowana, ponieważ przedstawienie jej w postaci znormalizowanej wymagałoby użycia wykładnika mniejszego niż dopuszczalny. Wzór na nią przedstawiony jest poniżej.

$$MIN_{sub} = 2^{1-t} \cdot 2^{c_{min}}$$

Gdzie:

- t to klasycznie liczba cyfr mantysy,
- c_{min} to minimalna możliwa do zapisania cecha,

$$c_{min} = -2^{d-1} + 2$$

- d liczba bitów przeznaczona na zapis cechy.

Dla badanych przez nas typów powyższe wartości to:

`Float16`: $c_{min} = -14, MIN_{sub} = 2^{-24}$
`Float32`: $c_{min} = -126, MIN_{sub} = 2^{-149}$
`Float64`: $c_{min} = -1022, MIN_{sub} = 2^{-1074}$

Po sprawdzeniu tych wartości w Julii:

```
Float16: 6.0e-8
Float32: 1.0e-45
Float64: 5.0e-324
```

otrzymujemy wynik, którego mogliśmy się spodziewać. Mianowicie MIN_{sub} jest równe co do wartości liczbie eta dla danej arytmetyki.

2 Liczba MIN_{nor}

Liczba MIN_{nor} to najmniejsza liczba w postaci znormalizowanej, którą da się zapisać. Wyliczyć ją można ze wzoru:

$$MIN_{nor} = 2^{c_{min}}$$

Mamy zatem:

`Float16`: $MIN_{nor} = 2^{-14}$
`Float32`: $MIN_{nor} = 2^{-126}$
`Float64`: $MIN_{nor} = 2^{-1022}$

Program liczący wartości `floatmin()` oraz MIN_{nor} dla danych typów zwraca nam takie wyniki:

```
------Badanie floatmin------
Float16: 6.104e-5
Float32: 1.1754944e-38
Float64: 2.2250738585072014e-308
------Badanie MIN_nor------
Float16: 6.104e-5
Float32: 1.1754944e-38
Float64: 2.2250738585072014e-308
```

Wartości zwracane przez `floatmin()` pokrywają się z odpowiadającym mu MIN_{nor} w danej arytmetyce.