Assignment 1 - Floating Point Arithmetic **LE, Nhat Hung**
COMP 350 - Numerical Computing McGill ID: 260793376
Prof. Chang Xiao-Wen Date: September 14, 2018
Fall 2018 Due date: September 20, 2018

1. (2 points) Show that a real number cannot have finite binary representation but infinite (or non terminating) decimal representation.

Proof 1:

Any finite binary number can be represented as:

$$a_n a_{n-1} \dots a_1 a_0 \,.\, a_{-1} a_{-2} \dots a_{-(m+1)} a_{-m}$$

n, m positive integers
a_i is either 0 or 1

By definition, its decimal presentation is

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_0 \cdot 2^0 + a_{-1} \cdot 2^{-1} + \dots + a_{-m} \cdot 2^{-m}$$

or

$$\sum_{i=-m}^{n} a_i \cdot 2^i$$

Which is fairly obviously finite.

But if this is not enough, then **we will go through a more general proof:**

Proof 2:
A real number has a finite representation in the binary system if and only if it is of the form:

$$\pm \frac{m}{2^n}$$

where n, m positive integers (distinct from the m & n in the first proof).

A real number has a finite representation in the decimal system if and only if it is of form:

$$\pm \frac{k}{10^n}$$

where k, n are positive integers.

From above,

$$\pm \frac{m}{2^n} = \pm \frac{5^n m}{5^n 2^n} = \pm \frac{k}{10^n} \quad \text{; with } k = 5^n m\text{, m \& n positive integers}$$

Therefore and in conclusion, a real number cannot have a finite binary representation but an infinite decimal representation.

2. (2 points) Using a 32-bit word, how many different integers can be represented by

(a) sign and magnitude;

With 32 bits, sign & magnitude can represent

- 2^{31} - 1 positive integers
- 2^{31} - 1 negative integers
- And zero

Thus, the number of different integers sign & magnitude can represent is

$$\begin{aligned} 2(2^{31} - 1) + 1 &= 2^{32} - 2 + 1 \\ &= 2^{32} - 1 \text{ different integers} \end{aligned}$$

(b) 2's complement? Express the answer using powers of 2.

2CR solves sign & magnitude's problem of having 2 representations for zero, and thus can represent one more integer. The answer is

$$2^{32} \text{ different integers}$$

3. Suppose in IEEE single format, the width of the exponent field is 5, not 8, and the width of the fraction field is 5, not 23.

(a) (.5 point) What should the exponent bias be?

$$\begin{aligned} 2^{\text{exponent field width - 1}} - 1 &= 2^{5-1} - 1 \\ &= 2^4 - 1 \\ &= 15 \end{aligned}$$

(b) (.5 point) What is the machine epsilon of this system?

$$\begin{aligned} \epsilon &= 2^{\text{-fraction field width}} \\ &= 2^{-5} \\ &= 1/32 \end{aligned}$$

(c) (2 points) What are the smallest and largest positive normal floating point numbers in this system?

The largest positive normal FPN here, N_{max}, is

Sign: 0 | E field: 11110 | m: 11111

Which is

$$
\begin{aligned}
N_{max} &= m \times 2^{\text{E field - exponent bias}} \\
&= 1.11111_2 \times 2^{30-15} \\
&= [1\,(2^0) + 1\,(2^{-1}) + 1\,(2^{-2}) + 1\,(2^{-3}) + 1\,(2^{-4}) \\
&\quad + 1\,(2^{-5})] \times 2^{15} \\
&= \left(1 + \tfrac{31}{32}\right) \times 2^{15} \\
&= 63/32 \times 2^{15} \\
&= 1.96875 \times 2^{15}
\end{aligned}
$$

The smallest positive normal FPN here, N_{min}, is

Sign: 0 | E field: 00001 | m: 00000

Which is

$$
\begin{aligned}
N_{min} &= m \times 2^{\text{E field - exponent bias}} \\
&= 1 \times 2^{1-15} \\
&= 2^{-14}
\end{aligned}
$$

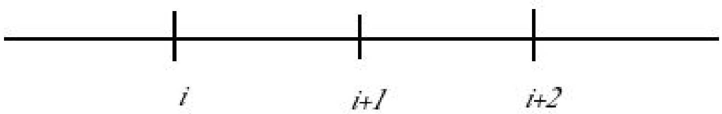
(d) (2 points) Can any integer number between the smallest and largest positive normal floating point numbers be stored exactly in this floating point system? Either prove it or give a counterexample.

Counterexample:

Consider *a, b* two FPNs between N_{min} and N_{max}.
Let the exponent field of *a* be 6.
Then the gap between *a* and the next smallest FPN, *b*, after *a* is

$$
\begin{aligned}
\text{gap} &= \epsilon \times 2^6 \\
&= 2^{-5} \times 2^6 \\
&= 2
\end{aligned}
$$

Then, between *a* and *b* is at least 1 integer which cannot be stored exactly in this floating point system, as seen in the figure below:



***a* = *i* and b = *i* + 2:** *i+1* cannot be represented
***i* < *a* < *i+1* and *b* > *i+2*:** *i+1* and *i+2* cannot be represented

Therefore, not all integers between N_{min} and N_{max} that cannot be stored exactly in this floating point system.

(e) (2 points) What are the largest and smallest nonnegative subnormal floating point numbers in this system?

Largest subnormal: $0.11111_2 \times 2^{-14} = 31/32 \times 2^{-14}$

Smallest subnormal: $2^{-5} \times 2^{-14} = 2^{-19}$

(f) (1 point) What is the largest floating point number smaller than 2?

$2 = 1.0 \times 2^1$ Sign: 0 | E field: 10000 | m field: 00000

Largest FPN smaller than 2 = Sign: 0 | E field: 01111 | m field: 11111
$= 1.11111_2 \times 2^0$
$= 63/32$

(g) (2 points) Given number $-(10.110101)_2$. Round it using the four rounding modes.

$$-(10.110101)_2 = -(1.0110101)_2 \times 2^1$$

Rounding now gives:

Round up: $-(1.01101)_2 \times 2^1$
Round down: $-(1.01110)_2 \times 2^1$
Round to zero: $-(1.01101)_2 \times 2^1$ (same as round up)

Let $x = -(10.110101)_2$. Then from above: $x_+ = -(10.1101)_2$ and $x_- = -(10.1110)_2$.

$$|x - x_+| = 0.000001$$
$$|x - x_-| = 0.000011$$

Thus x_+ is nearer to x.

Therefore:

Round to nearest: $-(1.01101)_2 \times 2^1$ (same as round up)

4. Are the following statements true or false? If a statement is true, give a proof and if it's false, give a counterexample. We assume no overflow occurs in the calculations and the rounding mode used can be any of the four rounding modes.

(a) (2 points) If x is a nonzero finite floating point number, then $x \oplus x = 2x$.

$$
\begin{aligned}
x \oplus x &= \mathrm{round}(x + x) \\
&= \mathrm{round}(2x)
\end{aligned}
$$

Let $x = m + 2^E$
There is no overflow, so

$$2x = m + 2^{E+1}$$

which is a representable FPN.

Thus

$$\mathrm{round}(2x) = 2x$$

And

$$x \oplus x = \mathrm{round}(2x) = 2x.$$

In conclusion, if x is a nonzero finite floating point number, then $x \oplus x = 2x$.

(b) (2 points) If x and y are two finite floating point number, then $x \ominus y = -(y \ominus x)$.

Counterexample:

We will use the same IEEE format as in question 3: width of exponent and fraction fields = 5.
We will be **rounding up**.
The idea is to make a non FPN from the difference of x and y.

Take the following x, y
(x is an FPN, y a subnormal number):

$x = 1.00000_2 \times 2^1$
$y = 2^{-15}$

We will now calculate $x \ominus y$ and $-(y \ominus x)$:

$$
\begin{aligned}
x \ominus y &= \mathrm{round}(x - y) \\
&= \mathrm{round}(1.00000_2 \times 2^1 - 2^{-15}) \\
&= \mathrm{round}(10_2 - 0.000000000000001_2) \\
&= \mathrm{round}(1.111111111111111_2) \\
&= 1.11111_2 + 0.00001_2 \\
&= 10_2 \\
&= 1.00000_2 \times 2^1
\end{aligned}
$$

$$
\begin{aligned}
-(y \ominus x) &= -\mathrm{round}(y - x) \\
&= -\mathrm{round}(0.000000000000001_2 - 10_2) \\
&= -\mathrm{round}(-1.111111111111111_2) \\
&= -(-1.11111_2) \\
&= 1.11111_2
\end{aligned}
$$

Thus,

$$x \ominus y \neq -(y \ominus x)$$

In conclusion, there exists x, y finite FPNs such that $x \ominus y \neq -(y \ominus x)$.

5. (2 points) What are the values of the expressions $\infty/0$, $\infty/(-\infty)$, NaN−NaN, and −0/NaN?

$\infty/0 = \infty$
$\infty/(-\infty) = \text{NaN}$
$\text{NaN}-\text{NaN} = \text{NaN}$
$-0/\text{NaN} = \text{NaN}$