Floating Point Nubmers

Chia-Tien Dan Lo
Computer Science and Software Engineering
Southern Polytechnic State University

The Needs

- To represent very large or very small numbers
- In Chemistry, the number of atoms in a molecule is

$$6.022 \times 10^{23}$$

- In Electronics, the charge of an electron is

$$1.602176487 \times 10^{-19} C$$

  - C: coulomb

Scientific Notations

- Instead of writing 602200000000000000000000
- We write this: 6.022×10^{23}

- The general form: $r \times 10^{e}$
  - r is a real number and called significand or mantissa
  - e is an integer and called exponent

Scientific Notations

- A negative mantissa indicates a negative number.

$$-1.23 \times 10^{2}$$

- A negative exponent denotes a number close to zero.

$$1.0 \times 10^{-5} = 0.00001$$

Precision

- The mantissa determines digits of precision.

$$0.12345 \times 10^{8}$$

  - 5 digits of precision

$$0.123456789 \times 10^{8}$$

  - 9 digits of precision

Order of Magnitude

- The exponent indicates the range of magnitude.

$$0.12345 \times 10^{8}$$

  - Magnitude 8

$$0.12345 \times 10^{15}$$

  - Magnitude 15

Addition/Subtraction

- Adjust exponents if they are not the same

$$r_1 \times 10^e + r_2 \times 10^e$$
$$= (r_1 + r_2) \times 10^e$$
$$r_1 \times 10^e - r_2 \times 10^e$$
$$= (r_1 - r_2) \times 10^e$$

Addition

$$1\times10^{2}+2\times10^{1}=1\times10^{2}+0.2\times10^{2}=1.2\times10^{2}$$

Adjust
exponent

Multiplication/Division

- Operate directly on mantissas and exponents

$$r_1 \times 10^{e_1} \times r_2 \times 10^{e_2}$$
$$= (r_1 \times r_2) \times 10^{e_1+e_2}$$

$$\frac{r_1 \times 10^{e_1}}{r_2 \times 10^{e_2}} = (\frac{r_1}{r_2}) \times 10^{e_1-e_2}$$

Multiplication

Multiply mantissas

$$(1.2\times10^{2})\times(2\times10^{1}) = (1.2\times2)\times10^{(2+1)} = 2.4\times10^{3}$$

Sum exponents

Normalization

- Obviously, there are lots of scientific representations for a number.

$$1 \times 10^2 = 10 \times 10^1 = 0.1 \times 10^3 = \ldots$$

- Easier to compare two numbers via order of magnitude

$$r \times 10^e \text{ where } 1 \leq |r| < 10.$$

Binary Real Numbers

- A "binary point" is used to separate integral part from fraction part of a binary real number.

$$0.1_2 = 2^{-1} = 0.5$$

$$0.01_2 = 2^{-2} = 0.25$$

$$0.001_2 = 2^{-3} = 0.125$$

$$0.101_2 = 2^{-1} + 2^{-2} = 0.625$$

- Repeating decimals like 1/7 may not be represented.

IEEE 754 Standard

- Allows floating point data to be exchanged
- Defines floating point formats, including infinities and NaN (not a number)
- Exponents use excess-127 or excess-1023 representations
- Denormals for small numbers

Binary32 Single Precision Format

- Use 32 bits
- Normalized binary numbers

$$r \times 2^{e}, 1 \le r < 2$$

bits	1	8	23
	S	Exponent	Fraction

$$(-1)^{S} \times (1 + Fraction) \times$$
$$2^{Exponent\ -Bias} \text{ where } bias = 127$$

Decimal to Binary32 Example

- Convert to binary fraction
- Normalization
- Exponent +127

Normalization

$$-0.75 = -0.11_2 = -1.\boxed{1} \times 2^{\boxed{-1}}$$

bits	1	8	23
	S	Exponent	Fraction
values	1	0111 1110	100 0000 0000 0000 0000 0000

-1+127=126

Reserved Exponents

- All zeros
- All ones

Minimum of Single Precisions

- The sign is negative, i.e., the sign bit is 1,
- The exponent is minimum, i.e., 1, and
- The fraction bits are all zeros.

bits	1	8	23
	S	Exponent	Fraction
values	1	0000 0001	000 0000 0000 0000 0000 0000

The number represented is -1.0×2^{-126}.

Maximum of Single Precisions

- The sign is positive, i.e., the sign bit is 0,
- The exponent is maximum, i.e., 127, and
- The fraction bits are all one's.

bits	1	8	23
	S	Exponent	Fraction
values	0	1111 1110	111 1111 1111 1111 1111 1111

The number represented is

$$1.111\ 1111\ 1111\ 1111\ 1111\ 1111 \times 2^{127}.$$

Single Precision Range

- Minimum to maximum

$$1.2 \times 10^{-38} \sim 3.4 \times 10^{38}$$

Binary64 Double Precision

- Use 64 bits

bits	1	11	52
	S	Exponent	Fraction

$$(-1)^S \times (1 + Fraction) \times$$
$$2^{Exponent\ -Bias} \text{ where } bias = 1023$$

Decimal to Binary64 Example

- Convert to binary fraction
- Normalization
- Exponent +1023

Decimal to Binary64 Example

$$-12.625_{10} = -1100.101_2 = -1.\boxed{100101}\times 2^3$$

bits	1	11	52
	S	Exponent	Fraction
values	1	100 0000 0010	1001 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

3+1023=1026

Minimum of Doubles

- The sign is negative, i.e., the sign bit is 1,
- The exponent is the smallest, i.e., , and
- The fraction bits are all zero's.

bits	1	11	52
	S	Exponent	Fraction
values	1	000 0000 0001	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

The number represented is -1.0×2^{-1022}.

Maximum of Doubles

- The sign is positive, i.e., the sign bit is 0,
- The exponent is the largest, i.e., , and
- The fraction bits are all one's.

bits	1	11	52
	S	Exponent	Fraction
values	1	111 1111 1110	1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

The number represented is 2.0×2^{1023}.

Range of Binary64

$$2.2 \times 10^{-308} \text{ to } 1.8 \times 10^{308}$$

Floating Point Precision

- All the fraction bits are significant.
- Binary32: 6 decimal digits of precision

$$23 \times \log 2 \cong 23 \times 0.3 \cong 6$$

- Binary64: 15 decimal digits of precision

$$52 \times \log 2 \cong 52 \times 0.3 \cong 15$$

Precision Errors

- Number 999,999,999 when converted to Binary32 is actually 999,999,936!

bits	1	8	23
	S	Exponent	Fraction
values	0	1001 1100	110 1110 0110 1011 0010 0111

$$999999999 = 1.1\ 1011\ 1001\ 1010\ 1100\ 1001\ 1111\ 1111 \times 2^{29}$$

The last 6 one's of the significand may not be stored in the fraction field due to the precision limit.

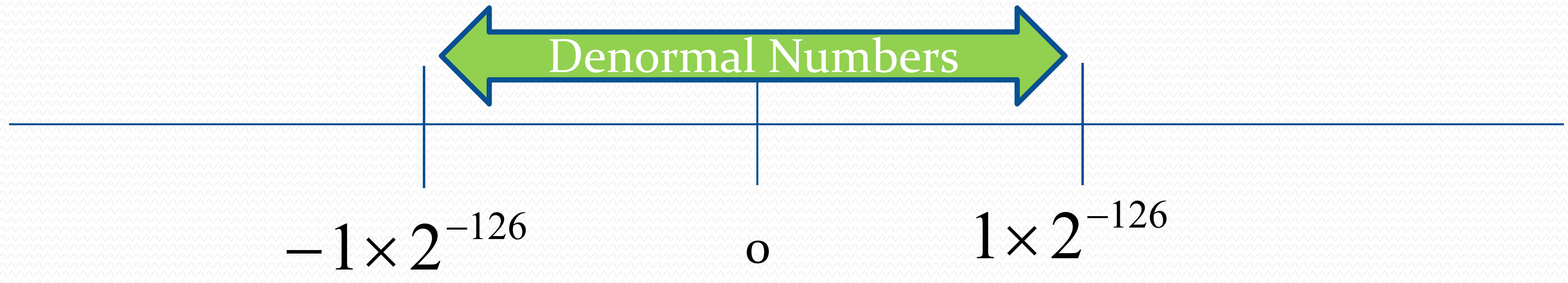
Special Values

- 0, 0/0, square root of negatives, etc.

Sign	Exponent	Fraction	Object Represented
0	All zero's	All zero's	+0
1	All zero's	All zero's	−0
0	All one's	All zero's	+∞
1	All one's	All zero's	−∞
-	All one's	Nonzero	NaN (Not a Number)

Denormal Numbers

- There is a gap between
  - 0 and smallest positive number
  - Largest negative number and 0
- For Binary32:
  - Exponent is all zeros but non-zero fraction



Denormal Binary32

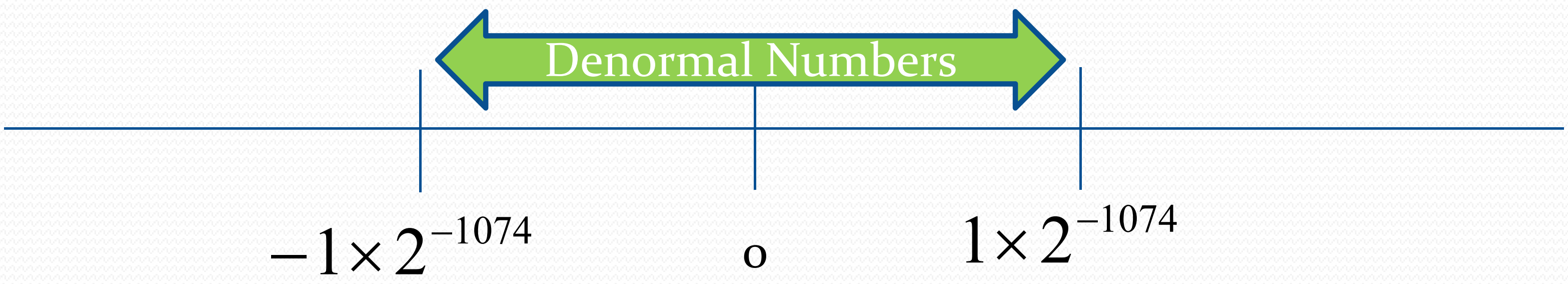
- E.g., 0.1×2^{-149}

bits	1	8	23
	S	Exponent	Fraction
values	0	0000 0000	000 0000 0000 0000 0000 0001

The number represented is $0.000\ 0000\ 0000\ 0000\ 0000\ 0001 \times 2^{-127}$.

Denormal Numbers

- Binary64



Questions