Floating Point

Wen Hongyu

EECS

September 19, 2019

Fractional binary numbers

$$(12.345)_{10} = 1 \times 10^1 + 2 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2} + 5 \times 10^{-3}$$

$$(101.101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

Fractional binary numbers

$$(12.345)_{10} = 1 \times 10^1 + 2 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2} + 5 \times 10^{-3}$$
$$(101.101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

只有 $\frac{p}{q}(q = 2^k, k \in \mathbb{N})$ 的情况才能表达成有限小数。

IEEE Floating Point

$$(-1)^s M 2^E$$

IEEE Floating Point

$$(-1)^s M 2^E$$

- Normalized Number: E 不能取到全 0 或全 1.

IEEE Floating Point

$$(-1)^s M 2^E$$

- Normalized Number: E 不能取到全 0 或全 1.
- Denormalized Values: E 全 0，将默认首位为 1 改为默认首位为 0，使得可以取到更小的数字。

IEEE Floating Point

$$(-1)^s M 2^E$$

- Normalized Number: E 不能取到全 0 或全 1.
- Denormalized Values: E 全 0，将默认首位为 1 改为默认首位为 0，使得可以取到更小的数字。
- inf: E 全 1，M 全 0
- nan: E 全 1，M 非全 0

IEEE Floating Point

$$(-1)^s M 2^E$$

- Normalized Number: E 不能取到全 0 或全 1.
- Denormalized Values: E 全 0，将默认首位为 1 改为默认首位为 0，使得可以取到更小的数字。
- inf: E 全 1，M 全 0
- nan: E 全 1，M 非全 0

IEEE Floating Point

- M (Significand code) **决定可以有多精确**，E (Exponent code) **决定可以有多大。**E **反映的数值需要减去** Bias。

IEEE Floating Point

- M (Significand code) **决定可以有多精确**，E (Exponent code) **决定可以有多大。**E **反映的数值需要减去** Bias。
- **设** M **有** n **位，当** $E = m$ **时，相邻数的间隔为** 2^{m-n}.

IEEE Floating Point

- M (Significand code) 决定可以有多精确，E (Exponent code) 决定可以有多大。E 反映的数值需要减去 Bias。
- 设 M 有 n 位，当 $E = m$ 时，相邻数的间隔为 2^{m-n}.
- E 变化时（设由 $m-1$ 增加到 m），新数为 2^m，原数为 $2^{m-1}(1 + 2^{-1} + \cdots + 2^{-n})$，间隔为 2^{m-n-1}.

IEEE Floating Point

- M (Significand code) 决定可以有多精确，E (Exponent code) 决定可以有多大。E 反映的数值需要减去 Bias。
- 设 M 有 n 位，当 $E = m$ 时，相邻数的间隔为 2^{m-n}.
- E 变化时（设由 $m-1$ 增加到 m），新数为 2^m，原数为 $2^{m-1}(1 + 2^{-1} + \cdots + 2^{-n})$，间隔为 2^{m-n-1}.
- Denormalized Values 最终将以 2^{m-n} 为步长减到 0.

Operation

- Round to even: 保证期望误差为 0

Operation

- Round to even: 保证期望误差为 0
- 加法：封闭、可交换、不可结合，除 inf 和 nan 之外有逆元

Operation

- Round to even: 保证期望误差为 0
- 加法：封闭、可交换、不可结合，除 inf 和 nan 之外有逆元
- 乘法：封闭、可交换、不可结合，没有分配律

Operation

- Round to even: 保证期望误差为 0
- 加法：封闭、可交换、不可结合，除 inf 和 nan 之外有逆元
- 乘法：封闭、可交换、不可结合，没有分配律
- +inf 大于所有数，-inf 小于所有数

Operation

- Round to even: 保证期望误差为 0
- 加法：封闭、可交换、不可结合，除 inf 和 nan 之外有逆元
- 乘法：封闭、可交换、不可结合，没有分配律
- +inf 大于所有数，-inf 小于所有数
- nan 不大于任何数，不小于任何数，也不等于任何数

浮点误差

- 无法精确表示 $\frac{1}{3}$，$\frac{1}{5}$ 等数

浮点误差

- 无法精确表示 $\frac{1}{3}$, $\frac{1}{5}$ 等数
- $a + b = a$? b 小于精度下界

浮点误差

- 无法精确表示 $\frac{1}{3}$，$\frac{1}{5}$ 等数
- $a + b = a$? b 小于精度下界
- int 向 float 转换，long long 向 double 转换时丢失精度

浮点误差

- 无法精确表示 $\frac{1}{3}$，$\frac{1}{5}$ 等数
- $a + b = a$? b 小于精度下界
- int 向 float 转换，long long 向 double 转换时丢失精度
- 不断做除法导致得到 0

浮点误差

- 无法精确表示 $\frac{1}{3}$，$\frac{1}{5}$ 等数
- $a + b = a$? b 小于精度下界
- int 向 float 转换，long long 向 double 转换时丢失精度
- 不断做除法导致得到 0
- overflow to inf

浮点误差

- 无法精确表示 $\frac{1}{3}$，$\frac{1}{5}$ 等数
- $a + b = a$? b 小于精度下界
- int 向 float 转换，long long 向 double 转换时丢失精度
- 不断做除法导致得到 0
- overflow to inf
- 没有结合律、分配律