Floating Point

汪之立

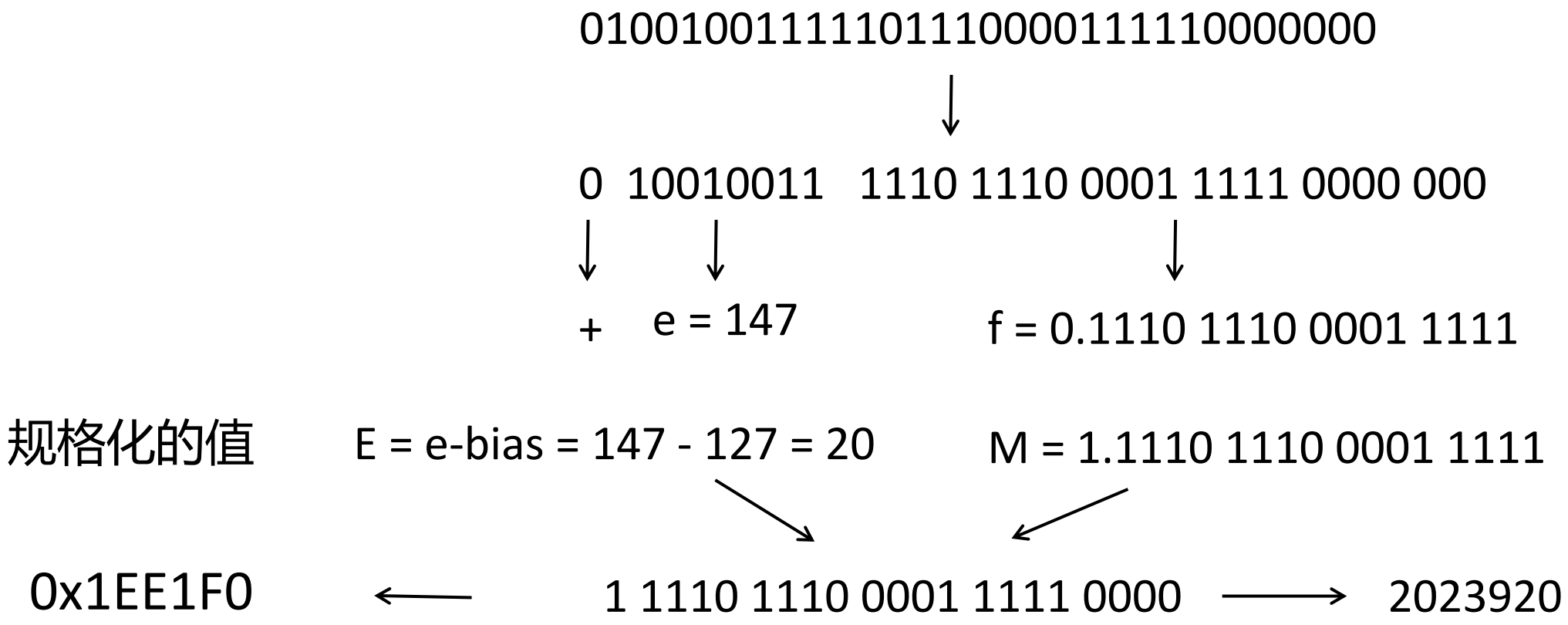
课本章节

- 引言　　浮点数的应用场景和规范历史
- 二进制小数　就是把进制变为2，换汤不换药
- IEEE浮点标准 本节重心
  - 编码格式和取值
  - 边界值
  - 整数转浮点的细节
- 舍入　　向偶数（最接近值）舍入和其他舍入方式
- 浮点运算　　次重点
  - 满足和不满足的运算性质
  - 特殊值的运算细节
  - 加法乘法实现
- C中浮点数转换的规则

目录

IEEE浮点标准　举个栗子

将下列单精度浮点数转换为整数



IEEE浮点标准　反过来捏

将下列数转换为单精度浮点数

-0.0007367835 ⟶ $-2^{-11} * 1.1000\ 0010\ 0100\ 1001\ 0110\ 1$

E = -11

M = 1.1000 0010 0100 1001 0110 1

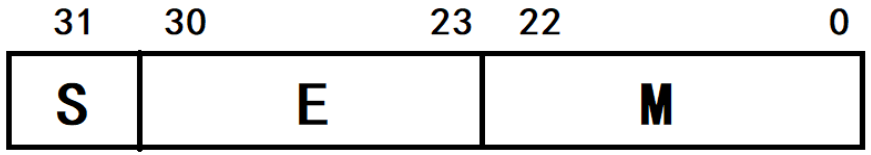
规格化的值　e = E + bias = - 11 + 127 = 116　f = 0.1000 0010 0100 1001 0110 1

1 01110100 1000 0010 0100 1001 0110 100

IEEE浮点标准　条规表述

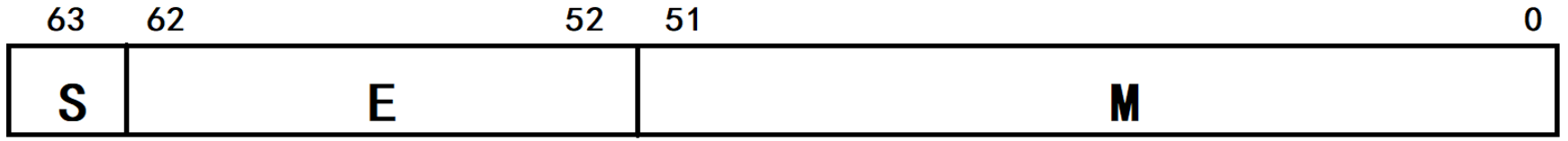
IEEE754标准

单精度(float)
32位浮点数



1：8：23

双精度(double)
64位浮点数



1：11：52

S：符号(sign),决定正负
E(exp)：阶码(exponent),表述权值
M(frac)：尾数(significand),具体数据的值

IEEE浮点标准　条规表述

计算公式：$(-1)^s * 2^E * M$

分类（按阶码位）：非规格化，特殊值和规格化

阶码部分有k位，尾数部分有n位，那么我们定义$bias = 2^{k-1} - 1$

32位为127

64位为1023

$f = 0.f_{w\_1}f_{w\_2}\ldots f_0$，$e = e_{k\_1}e_{k\_2}\ldots e_0$

①阶码部分全为0，非规格化的值，E=1-bias，M = f

②阶码部分全为1，特殊值

尾数部分全为0，则为inf（∞），按符号位决定+∞或-∞；

尾数部分不全为0，则结果称为NaN值;

③阶码部分不全为1或0，规格化的值，E = e - bias，M = 1 + f;

32位为 $-126 \sim +127$

64位为 $-1022 \sim +1023$

IEEE浮点标准　极限值

描　述	exp	frac	单精度		双精度	
			值	十进制	值	十进制
0	00 … 00	0 … 00	0	0.0	0	0.0
最小非规格化数	00 … 00	0 … 01	$2^{-23} \times 2^{-126}$	1.4×10^{-45}	$2^{-52} \times 2^{-1022}$	4.9×10^{-324}
最大非规格化数	00 … 00	1 … 11	$(1-\varepsilon) \times 2^{-126}$	1.2×10^{-38}	$(1-\varepsilon) \times 2^{-1022}$	2.2×10^{-308}
最小规格化数	00 … 01	0 … 00	1×2^{-126}	1.2×10^{-38}	1×2^{-1022}	2.2×10^{-308}
1	01 … 11	0 … 00	1×2^{0}	1.0	1×2^{0}	1.0
最大规格化数	11 … 10	1 … 11	$(2-\varepsilon) \times 2^{127}$	3.4×10^{38}	$(2-\varepsilon) \times 2^{1023}$	1.8×10^{308}

图 2-36　非负浮点数的示例

- +0.0: 全零
- +1：0 01…1 00…0

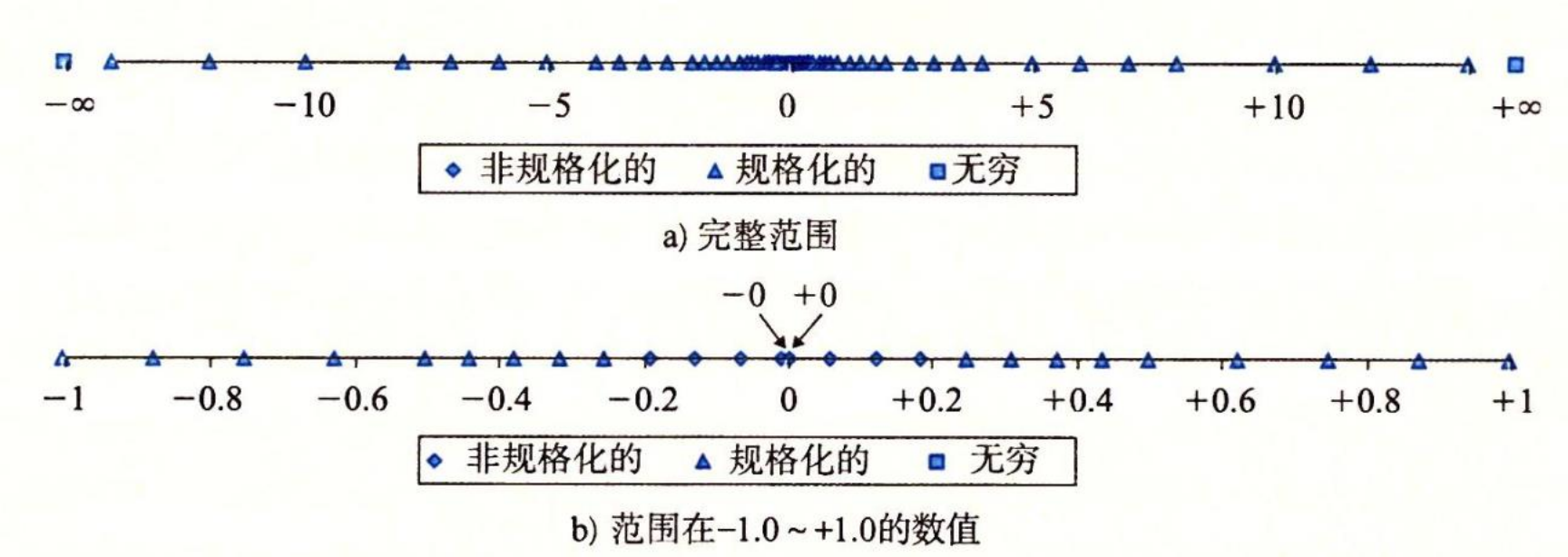
IEEE浮点标准　极限值

k位阶码，n位小数（由于正负对称，仅考虑正数）

- 最小非规格化值　0 0…00 0…01　$V = 2^{2-2^{k-1}} * 2^{-n}$
- 最大非规格化值　0 0…00 1…11　$V = 2^{2-2^{k-1}} * (1 - 2^{-n})$
- 最小规格化值　0 0…01 0…00　$V = 2^{2-2^{k-1}} * 1$
- 最大规格化值　0 1…10 1…11　$V = 2^{-1+2^{k-1}} * (2 - 2^{-n})$

- 非规格化值范围　$[-2^{2-2^{k-1}} * (1 - 2^{-n}), 2^{2-2^{k-1}} * (1 - 2^{-n})]$
- 规格化范围　$[-2^{-1+2^{k-1}} * (2 - 2^{-n}), -2^{2-2^{k-1}},] \cup$
  $[2^{2-2^{k-1}}, 2^{-1+2^{k-1}} * (2 - 2^{-n})]$

IEEE浮点标准　何为“逐渐溢出”



规格化的值两端疏，中间密

非规格化的值为等差数列 公差为$2^{2-2^{k-1}} * 2^{-n}$

非规格化绝对值最大值到规格化绝对值最小值之间的差值为公差

$2^{2-2^{k-1}} * (1-2^{-n})$ <–> $2^{2-2^{k-1}} * 1$

IEEE浮点标准　浮点数比较

描述	位表示	指数			小数		值		
		e	E	2^E	f	M	$2^E \times M$	V	十进制
0	0 0000 000	0	−6	$\frac{1}{64}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{512}$	0	0.0
最小的非规格化数	0 0000 001	0	−6	$\frac{1}{64}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{512}$	$\frac{1}{512}$	0.001953
	0 0000 010	0	−6	$\frac{1}{64}$	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{2}{512}$	$\frac{1}{256}$	0.003906
	0 0000 011	0	−6	$\frac{1}{64}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{512}$	$\frac{3}{512}$	0.005859
	⋮								
最大的非规格化数	0 0000 111	0	−6	$\frac{1}{64}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{512}$	$\frac{7}{512}$	0.013672
最小的规格化数	0 0001 000	1	−6	$\frac{1}{64}$	$\frac{0}{8}$	$\frac{8}{8}$	$\frac{8}{512}$	$\frac{1}{64}$	0.015625
	0 0001 001	1	−6	$\frac{1}{64}$	$\frac{1}{8}$	$\frac{9}{8}$	$\frac{9}{512}$	$\frac{9}{512}$	0.017578
	⋮								
	0 0110 110	6	−1	$\frac{1}{2}$	$\frac{6}{8}$	$\frac{14}{8}$	$\frac{14}{16}$	$\frac{7}{8}$	0.875
	0 0110 111	6	−1	$\frac{1}{2}$	$\frac{7}{8}$	$\frac{15}{8}$	$\frac{15}{16}$	$\frac{15}{16}$	0.9375
1	0 0111 000	7	0	1	$\frac{0}{8}$	$\frac{8}{8}$	$\frac{8}{8}$	1	1.0
	0 0111 001	7	0	1	$\frac{1}{8}$	$\frac{9}{8}$	$\frac{9}{8}$	$\frac{9}{8}$	1.125
	0 0111 010	7	0	1	$\frac{2}{8}$	$\frac{10}{8}$	$\frac{10}{8}$	$\frac{5}{4}$	1.25
	⋮								
	0 1110 110	14	7	128	$\frac{6}{8}$	$\frac{14}{8}$	$\frac{1792}{8}$	224	224.0
最大的规格化数	0 1110 111	14	7	128	$\frac{7}{8}$	$\frac{15}{8}$	$\frac{1920}{8}$	240	240.0
无穷大	0 1111 000	—	—	—	—	—	—	∞	—

图 2-35　8 位浮点格式的非负值示例($k=4$ 的阶码位的和 $n=3$ 的小数位。偏置量是 7)

按同位的整数来排序
->整数原码

正数升序，负数降序

浮点运算运算性质

▶▶▶正常运算性质的产生都要求参与是数而不是NaN

▶ 加法

- 交换律 √　　x + y == y + x
- 结合率 ×　　x + **(y +** z) ？ (x + y) + z　　反例：舍入/溢出
- 单调性 √　　∀数x,a,b，若a>=b，则有 a + x >= b + x

▶ 乘法

- 交换律 √　　x * y == y * x
- 结合率 ×　　x * (y * z) ？ (x * y) * z　　反例：舍入/溢出
- 分配率 ×　　x * (y + z) ？ x * y + x * z　　反例：溢出
- 单调性 √　　∀数x,a,b，若a>=b且c>=0，则有 a * x >= b * x
  若a>=b且c>=0，则有 a * x >= b * x
- 平方非负 √　　∀数x, x * x >= 0

浮点运算运算性质　特殊值处理

- NaN
  - ∀数x，NaN + x = NaN
  - (+∞) + (−∞) = NaN
  - $\sqrt{-1}$ = NaN
  - ∞ * 0 = NaN
  - （C）0.0/0.0 —> NaN
- Inf
  - 1 / (-0.0) = −∞　1 / (+0.0) = +∞
  - 运算越界溢出产生

浮点运算运算性质　牛刀小试

D　2.对于 float 类型变量 a, b, c 下列说法正确的是:

A. 若 a > b，则 a + c > b + c

B. 若 a == b，则 a + c == b + c

C. 若 a + b + c == 0.0，则 c + b + a == 0.0

D. 若 a + b == 0.0，则 b + a == 0.0

向偶舍入中的误差

最小不能被float精确表示的整数值?

->超出23位的尾数

$$1.0000\ 0000\ 0000\ 0000\ 0000\ 0001 * 2^{24}$$

如果有a > b, 则一定有 a + 1 > b + 1?

->a,b相对于1来说太小了，被忽略

$a = 2b = 2 * 2^{2-2^{k-1}} * 2^{-n}$

向偶舍入中的误差

A 4. 给定一个实数，会因为该实数表示成单精度浮点数而发生误差。不考虑 NaN 和 Inf 的情况， 该绝对误差的最大值为

A. 2^{103} B. 2^{104} C. 2^{230} D. 2^{231}

C中浮点数转换

int -> double	一定可以精确表示
int-> float	可能需要进行舍入操作 x ？ (int)(float)x
floar/double -> int	向0取整 出现溢出：整数不确定（integer indefinite） C称为undefined behavior

Thanks

Litchi-w