§ 1.4 IEEE 浮点运算标准

浮点格式是一种数据结构，用于指定包含浮点数的字段、这些字段的布局及其算术解释。浮点存储格式指定如何将浮点格式存储在内存中。自计算机发明以来，曾出现许多中不同的浮点数表示方式，但目前最通用的是 IEEE 二进制浮点数算术标准 (IEEE Standard for Binary Floating-Point Arithmetic，简称 IEEE 754 标准)。

IEEE 754 标准的主要起草者是加州大学伯克利分校数学系的 William Kahan 教授，他帮助 Intel 公司设计了 8087 浮点处理器，并以此为基础形成了 IEEE 754 标准， Kahan 教授也因此获得了 1987 年的图灵奖。

1.4.1 IEEE 中的浮点数

- IEEE 754 标准中定义了表示浮点数的四种格式：
  - 两种基本的浮点数：单精确度 (32 位字长) 和双精确度 (64 位字长)。其中单精度格式具有 24 位有效数字，而双精度格式具有 53 位有效数字，相对于十进制来说，分别是 7 位 ($2^{24} \approx 10^7$) 和 16 位 ($2^{53} \approx 10^{16}$) 有效数字。
  - 两种扩展的浮点数：单精度扩展和双精度扩展。此标准并未规定扩展格式的精度和大小，但它指定了最小精度和大小：单精度扩展需 43 位字长以上，双精确度扩展需 79 位字长以上 (64 位有效数字)。单精度扩展很少使用，而对于双精确度扩展，不同的机器构架中有不同的规定，有的为 80 位字长 (X86)，有的为 128 位字长 (SPARC)。
- 通常一个浮点数由符号、尾数、基和指数组成，如：

$$-0.31415926_{10} \times 10^2, \quad 0.10101_2 \times 2^3.$$

  若尾数的首位数字不为 0 时，我们称其为**正规数** (或**规格化数**)，否则称为**次正规数** (或**非规格化数**)。如 $0.10101_2 \times 2^3$ 是正规数，而 $0.010101_2 \times 2^4$ 是次正规数。正规化表示方法可以使得每个浮点数的表示方式唯一，而且可以空出一个位置，使得表示精度更高。
- 描述一个浮点数的三个基本要素为：
  - 基
  - 尾数的位数：确定精度
  - 指数的位数：确定所能表示的数的范围
- 在 IEEE 754 标准中，浮点数由三部分组成：符号 (s)，指数 (e) 和尾数 (f)，见下图。

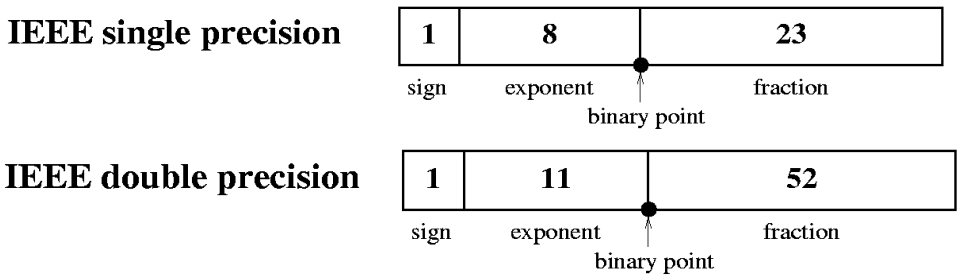


图 1.1 IEEE 单精度与双精度数的记录格式

- 对于单精度正规数，表示的数为：$(-1)^s \times 1.f \times 2^{e-127}$。
- 对于双精度正规数，表示的数为：$(-1)^s \times 1.f \times 2^{e-1023}$。

例 1.1 单精度所能表示的十进制数范围。

单精度格式位模式	值
$0 < e < 255$	$(-1)^s \times 1.f \times 2^{e-127}$ (正规数)
$e = 0, f \neq 0$	$(-1)^s \times 0.f \times 2^{-126}$ (次正规数)
$e = 0, f = 0$	$(-1)^s \times 0.0$ (有符号的零)
$s = 0, e = 255, f = 0$	+inf (正无穷大)
$s = 1, e = 255, f = 0$	+inf (负无穷大)
$e = 255, f \neq 0$	NaN (非数、非确定值)

单精度所能表示的

- 最大正规数为 $7F7FFFFF_{16} = 3.40282347 \times 10^{38}$,
- 最小正规数为 $00800000_{16} = 1.17549435 \times 10^{-38}$,
- 最大次正规数为 $007FFFFF_{16} = 1.17549421 \times 10^{-38}$,
- 最小次正规数为 $00000001_{16} = 1.40129846 \times 10^{-45}$。

例 1.2 双精度所能表示的十进制数范围。

双精度格式位模式	值
$0 < e < 2047$	$(-1)^s \times 1.f \times 2^{e-1023}$ (正规数)
$e = 0, f \neq 0$	$(-1)^s \times 0.f \times 2^{-1022}$ (次正规数)
$e = 0, f = 0$	$(-1)^s \times 0.0$ (有符号的零)
$s = 0, e = 2047, f = 0$	+inf (正无穷大)
$s = 1, e = 2047, f = 0$	+inf (负无穷大)
$e = 2047, f \neq 0$	NaN (非数、非确定值)

双精度所能表示的

- 最大正规数为 $7FEFFFFF\ FFFFFFFF_{16} = 1.7976931348623157 \times 10^{308}$，
- 最小正规数为 $00100000\ 00000000_{16} = 2.2250738585072014 \times 10^{-308}$，
- 最大次正规数为 $000FFFFF\ FFFFFFFF_{16} = 2.2250738585072009 \times 10^{-308}$，
- 最小次正规数为 $00000000\ 00000001_{16} = 4.9406564584124654 \times 10^{-324}$。

例 1.3 把二进制数 $(1001.0101)_2$ 转换成十进制数。

$$\begin{aligned}(1001.0101)_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-2} \\ &= 9.3125_{10}\end{aligned}$$

例 1.4 把十进制数 13.125_{10} 转换成二进制数。

整数部分：$13_{10} = 1101_2$

小数部分：

- $0.125 \times 2 = 0.25$，整数位是 $0 \rightarrow .0$;
- $0.25 \times 2 = 0.5$，整数位是 $0 \rightarrow .00$;
- $0.5 \times 2 = 1$，整数位是 $1 \rightarrow .001$;

所以 $13.125_{10} = 1101.001_2$

一个十进制数能否用二进制浮点数精确表示，关键在于小数部分。

例 1.5 十进制数 0.1_{10} 能否用二进制数精确表示？

- $0.1 \times 2 = 0.2$，整数位是 $0 \rightarrow .0$;
- $0.2 \times 2 = 0.4$，整数位是 $0 \rightarrow .00$;
- $0.4 \times 2 = 0.8$，整数位是 $0 \rightarrow .000$;
- $0.8 \times 2 = 1.6$，整数位是 $1 \rightarrow .0001$;
- $0.6 \times 2 = 1.2$，整数位是 $1 \rightarrow .00011$;
- $0.2 \times 2 = 0.4$，整数位是 $0 \rightarrow .000110$;
- $0.4 \times 2 = 0.8$，整数位是 $0 \rightarrow .0001100$;
- $0.8 \times 2 = 1.6$，整数位是 $1 \rightarrow .00011001$;
- $0.6 \times 2 = 1.2$，整数位是 $1 \rightarrow .000110011$;

- ⋯ ⋯

得到一个无限循环的二进制小数，显然用有限位字长是无法表示的，因此 0.1_{10} 无法用 IEEE 754 浮点数精确表示。

同理可知 $0.2, 0.4, 0.6, 0.8, 0.3, 0.7, 0.9$ 是无法用 IEEE 754 浮点数精确表示的，故 0.1 至 0.9 的9个小数中，只有 0.5 可以用 IEEE 754 浮点数精确表示。

例 1.6 能用二进制数精确表示的十进制数。易知

$$
\begin{aligned}
&0.1_2 = 2_{10}^{-1} = 0.5\\
&0.01_2 = 2_{10}^{-2} = 0.25\\
&0.001_2 = 2_{10}^{-3} = 0.125\\
&0.0001_2 = 2_{10}^{-4} = 0.0625\\
&0.00001_2 = 2_{10}^{-5} = 0.03125\\
&0.000001_2 = 2_{10}^{-6} = 0.0156255\\
&0.0000001_2 = 2_{10}^{-7} = 0.00781255\\
&0.00000001_2 = 2_{10}^{-8} = 0.00390625\\
&\cdots\ \cdots
\end{aligned}
$$

由此可知，一个十进制小数要能用浮点数精确表示，最后一位必须是 5。当然这是必要条件，并非充分条件。

例 1.7 N 位二进制小数能精确表示的十进制小数总共有多少个?

- 1 位二进制小数能精确表示的有 $2^0 = 1$ 个 $(0.1_2 = 0.5_{10})$;
- 2 位二进制小数能精确表示的有 $2^1 = 2$ 个 $(0.01_2 = 0.25_{10}, 0.11_2 = 0.75_{10})$;
- 3 位二进制小数能精确表示的有 $2^2 = 4$ 个
- ⋯ ⋯
- N 位二进制小数能精确表示的有 2^{N-1} 个

所以 N 位二进制小数能精确表示的十进制小数总共有 $2^N - 1$ 个。

1.4.2 IEEE 中的浮点数运算

- IEEE 754 标准也定义了浮点数的运算规则：

- 浮点运算的准确度要求：加、减、乘、除、平方根、余数、将浮点格式的数舍入为整数值、在不同浮点格式之间转换、在浮点和整数格式之间转换以及比较。

  求余和比较运算必须精确无误。其他的每种运算必须向其目标提供精确的结果，除非没有此类结果，或者该结果不满足目标格式，此时运算必须按照下面介绍的舍入模式对精确结果进行最低限度的修改，并将经过修改的结果提供给运算的目标。

- 在十进制字符串和两种基本浮点格式之一的二进制浮点数之间进行转换的准确度、单一性和一致性要求。

  对于在指定范围内的操作数，这些转换必须生成精确的结果 (如果可能的话)，或者按照规定的舍入模式，对此类精确结果进行最低限度的修改。对于不在指定范围内的操作数，这些转换生成的结果与精确结果之间的差值不得超过取决于舍入模式的指定误差。

- 五种类型的 IEEE 浮点异常，以及用于向用户指示发生这些类型异常的条件。

  五种类型的浮点异常是：**无效运算**、**被零除**、**上溢**、**下溢和不精确**。

- 四种舍入模式：向**最接近的可表示的值** (就近舍入)；当有两个最接近的可表示的值时首选 “偶数” 值；向**负无穷大方向** (向下)；向**正无穷大方向** (向上) 以及向 0 **方向** (截断)。

注：不同编译器可能有不同的处理方式。

- 下溢

  当运算结果非常小时，就会发生下溢。下表是下溢阈值。

目标的精度	下溢阈值	
单精度	最小正规数	$1.17549435 \times 10^{-38}$
	最大次正规数	$1.17549421 \times 10^{-38}$
双精度	最小正规数	$2.2250738585072014 \times 10^{-308}$
	最大次正规数	$2.2250738585072009 \times 10^{-308}$

  IEEE 算法处理下溢的方式是**渐进下溢**：当生成的正确结果的数量级低于最小正正规数时，就会生成次正规数，而不是返回零。

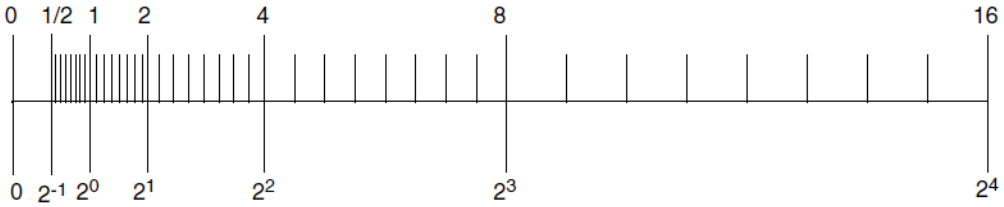
- 相对表示误差：用 $\tilde{x}$ 表示 x 时，其相对表示误差为

$$\frac{|\tilde{x} - x|}{|x|}.$$

最大相对表示误差 (机器精度)：设尾数有 p 位数字，基为 β，则浮点运算中的最大相对表示误差为：$0.5\times\beta^{1-p}$，即 1 与下一个浮点数 $1+\beta^{1-p}$ 之间的距离的一半。

精度	最大相对表示误差
单精度	$2^{-23}\approx 1.192093\times 10^{-7}$
双精度	$2^{-52}\approx 2.220446\times 10^{-16}$

例 1.8 假定要使用只有三个精度位的二进制算法。那么，在任意两个 2 的幂之间，只有 $2^3-1=7$ 个可表示数字，如下图所示。



数轴显示了数字之间的差距是随着指数增加而加倍增加的。

在 IEEE 单精度格式中，两个最小正次正规数之间的差大约是 10^{-45}，而两个最大有限数之间的数量级差大约是 10^{31}！

- 舍入误差：当两个数 a,b 的运算结果 $a\odot b$ 不能精确表示时，则按照舍入模式选取一个近似值 $fl(a\odot b)$ 来表示，精确值与这个近似值之间的差 $a\odot b-fl(a\odot b)$ 就称为舍入误差。通常记

$$fl(a\odot b)=(a\odot b)(1+\delta),$$

其中 $|\delta|\le$ 最大相对表示误差 (机器精度)，运算 $\odot$ 可以为加、减、乘、除、平方根。

- 当运算结果处于两个浮点数的正中间时，选取最低有效位为 0 的浮点数作为近似值 (“偶数”)。
- 精确是偶然的，误差是必然的。做数值算法，惟一能做的就是误差不积累。