

# Machine arithmetics

Representation of real numbers

# Representation of reals: floating-point numbers

Floating-point numbers are represented by a *mantissa*,  $m$ , and an *exponent*,  $p$

$$m \times 2^p ,$$

with

$$\frac{1}{2} \leq |m| < 1$$

# Representation of reals: floating-point numbers

$$m \times 2^p ,$$

*significand* or *matissa* has  $t$  binary digits:

$$m = \pm \mu_1 \mu_2 \cdots \mu_t$$

$$m = \pm (\mu_1 \times 2^{-1} + \mu_2 \times 2^{-2} + \cdots + \mu_t \times 2^{-t})$$

# Representation of reals: floating-point numbers

$$m \times 2^p$$

The *exponent*,  $p$ , is a fixed-width integer,

$$p = \gamma_0 \gamma_1 \gamma_2 \cdots \gamma_L$$

Here  $\gamma_0$  is the sign bit.

# Floating-point numbers

A *binary word* representation of a floating-point number is a string of bits,

$$\pm \underbrace{\gamma_0, \gamma_1, \dots, \gamma_L}_{\text{exponent}} \underbrace{\mu_1, \mu_2, \dots, \mu_t}_{\text{mantissa}},$$

## IEEE 754 standard

- ▶ *single precision*, 32 bits — 23 bit mantissa, 8 bit exponent
- ▶ *double precision*, 64 bits — 52 bit mantissa, 11 bit exponent
- ▶ *extended precision*

# Special floating-point numbers

Signed *infinities*: `inf`, defined as

$$x < \text{inf} \quad \forall \text{ finite } x$$

and the corresponding `-inf`.

*Not-a-number*: `nan`, defined via

$$x = \text{nan} \quad \text{iff} \quad x \neq x$$

Is meant to represent results of invalid operations, e.g.  $1/0$ .

Also, *signed zeros*, *denormal numbers* etc. Full details: D. Goldberg, *What Every Computer Scientist Should Know About Floating-Point Arithmetic*, ACM Computing Surveys **23**, No 1, March 1991.

# Not all real numbers can be represented

$$\pm (\mu_1 \times 2^{-1} + \mu_2 \times 2^{-2} + \cdots + \mu_t \times 2^{-t}) \times 2^p$$

The distance between two consecutive numbers with exponent  $p$  is

$$2^{p-t}$$

The granularity depends on the *bit-width* of the mantissa and the *value* of the exponent.

# Not all real numbers can be represented

$$\pm (\mu_1 \times 2^{-1} + \mu_2 \times 2^{-2} + \cdots + \mu_t \times 2^{-t}) \times 2^p$$

The distance between two consecutive numbers with exponent  $p$  is

$$2^{p-t}$$

The granularity depends on the *bit-width* of the mantissa and the *value* of the exponent.

$\Rightarrow$  Any computation incurs *rounding errors*.



## Rounding errors: machine epsilon

$$\pm (\mu_1 \times 2^{-1} + \mu_2 \times 2^{-2} + \cdots + \mu_t \times 2^{-t}) \times 2^p$$

The absolute value of a rounding error  $2^{p-t}$  depends on  $p$ .

The relative rounding error,

$$\sim \frac{2^{p-t}}{2^p} = 2^{-t},$$

only depends on the bit width of mantissa.

## Rounding errors: machine epsilon

$$\pm (\mu_1 \times 2^{-1} + \mu_2 \times 2^{-2} + \cdots + \mu_t \times 2^{-t}) \times 2^p$$

The absolute value of a rounding error  $2^{p-t}$  depends on  $p$ .

The relative rounding error,

$$\sim \frac{2^{p-t}}{2^p} = 2^{-t},$$

only depends on the bit width of mantissa.

*Machine epsilon* is the minimal number  $\epsilon > 0$ , for which

$$1 + \epsilon \neq 1$$

## Machine zero; Underflow

$$\pm (\mu_1 \times 2^{-1} + \mu_2 \times 2^{-2} + \cdots + \mu_t \times 2^{-t}) \times 2^p$$

The exponent,  $p$ , is a signed integer of  $L$  bits. The maximum value of the exponent is

$$p_{\max} = 2^{L-1}$$

Therefore, nonzero numbers less than

$$X_0 = 2^{-p_{\max}} \times 2^{-1} = 2^{-p_{\max}-1}$$

cannot be represented as floating-point values.

# Machine zero; Underflow

*Machine zero* is the minimum nonzero value  $X_0$ , such that

$$X_0/2 = 0$$

Notice that normally

$$X_0 \ll \epsilon .$$

Here we gloss over *denormal numbers*. See, e.g., Goldberg, *ibid*.

## Machine infinity; Overflow

The maximum value of exponent is  $p_{\max} = 2^{L-1}$ . Therefore,

$$x > X_{\infty} = 2^{p_{\max}}$$

cannot be represented in floating point.

Notice that  $X_{\infty}$  differs from the special IEEE 754 value `inf`.

*Machine infinity* is the minimal nonzero value  $X_{\infty}$ , such that

$$X_{\infty} \times 2 \quad \text{overflows}$$