1 Machine (Floating-Point) Numbers (Section 1.3)

Consider the decimal number `24.625`. We can decompose this into a sum of its components as

$$24.625 = 2 \cdot 10^1 + 4 \cdot 10^0 + 6 \cdot 10^{-1} + 2 \cdot 10^{-2} + 5 \cdot 10^{-5}.$$

We don't normally decompose numbers like this, but it's useful to know this regardless. To see why, let's consider binary numbers, or numbers in base 2. These numbers consist of 0's and 1's. For example, $(11000.101)_2$ is a binary number (base 2). So,

$$\begin{aligned}(11000.101)_2 &= 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= 16 + 8 + 0.5 + 0.125 \\ &= 24.625.\end{aligned}$$

Here, we're able to convert this binary number into its decimal representation (base 10). Note that it's possible to get a conversion error: if we cannot represent infinitely many digits in a different base, we end up with a slight conversion error. In other words, the conversion between number systems may result in round-off errors.

1.1 A Model Computer

Suppose we have a 32 bits model computer (Marc-32). We also have a floating-point standard. There are some components to this standard:

$$x^* = (-1)^s \cdot q \cdot 2^m,$$

where

- $(-1)^s$ is the sign bit, and $s \in \{0, 1\}$.
- $1 \le q < 2$ in binary, which looks like $(1.f)_2$, known as the mantissa.
- $m = e - 127$, where e is an integer in binary form. This is known as the exponent.

The storage of these components is divided up into the three parts:

`s`	`exp`	`frac`

More specifically, given 32 bits,

- 1 bit is for the sign s,
- 8 bits are for the exponent e,
- 23 bits are for the mantissa f.

We also have special conventions for the largest representable number in this standard. Note that

$$0 < e < (11111111)_2 = 255.$$

This also means that

- The largest representable number is $q_{\max} \cdot 2^{m_{\max}} = (2 - 2^{-23}) \cdot 2^{127}$.
- The smallest representable number is $q_{\min} \cdot 2^{m_{\min}} = 1 \cdot 2^{-126}$.

1.2 Finding the Closest Floating Point Number

Let

$$x = q \cdot 2^m$$

such that $1 \leq q < 2$ and $-126 \leq m \leq 127$. How do we find the closest floating-point number to the real number x? How do we find the binary representation of x?

1. Let $x = (1.a_1 a_2 \ldots a_{23} a_{24} a_{25} \ldots)_2 \cdot 2^m$, where each $a_i \in \{0, 1\}$ is a digit.
2. We begin by chopping off some of the digits to get

   $$x_{\text{-}} = (1.a_1 a_2 \ldots a_{23})_2 \cdot 2^m.$$

   This is because we want to find the nearby floating-point number, which we can obtain by discarding the excess bits $a_{24} a_{25}$.
3. Next, we round up.

   $$x_+ = ((1.a_1 a_2 \ldots a_{23})_2 + 2^{-23}) \cdot 2^m.$$
4. Select the closest of $x_+, x_{\text{-}}$, to x as the machine number x^*.

   ```
   |----------x-------------------------------------------|
   x_                                                     x+

   |-------------------------------------------x----------|
   x_                                                     x+
   ```

   In the first case (the top one), we pick $x_{\text{-}}$. For the second case (bottom one), we pick x_+.
5. We now want to determine the relative error (e.g., for the first case),

   $$\frac{|x - x_{\text{-}}|}{|x|} \leq \frac{1}{2} \frac{|x_+ - x_{\text{-}}|}{|x|} = \frac{1}{2} \frac{2^{m-23}}{q \cdot 2^m} = \frac{2^{m-24}}{q \cdot 2^m} \leq \frac{2^{m-24}}{2^m} = 2^{-24}.$$

   The second case is similar.