Examples: Floating Point Arithmetic

Since our computers are working in base $\beta = 2$, the storage looks like

$$\bar{x} = \pm 0.b_1 b_2 \ldots b_t \times 2^e, \quad \text{where} \quad m \le e \le M \text{ and } b_j \text{ is 0 or 1, but } b_1 = 1.$$

Recall that the binary fraction above is interpreted as

$$0.b_1 b_2, \ldots, b_k \equiv (0.b_1 b_2, \ldots, b_t)_2 = \frac{b_1}{2} + \frac{b_2}{4} + \cdots + \frac{b_t}{2^t},$$

so, e.g. $(0.1001)_2 = 1/2 + 1/16 = 9/16$. Now you can see why $1/3$ is not a float (in a (finite precision) floating point system with base $\beta = 2$):

$$\frac{1}{3} = (0.1)_3 = (0.333\ldots)_{10} = (0.010101\ldots)_2.$$

While it is probably good for one's intuition to do *a little* finite precision arithmetic by hand, it is probably not worth the effort to do it in binary. So, for most of our examples (including homework and tests), we will pretend that $\beta = 10$ simply to make the computations a little less onerous.

Take $x = 0.003245678$, $y = 212.3454$, and $z = -212.3456$. For this example, if $u \in \mathbb{R}$, let $\mathrm{fl}(u)$ be its floating point representation in base $\beta = 10$ with $t = 6$ digits, minimum exponent m and maximum exponent M (some call this $F(10, 6, m, M)$). Then $\mathrm{fl}(x) = 0.324568 \times 10^{-2} = 0.00324568$, $\mathrm{fl}(y) = 0.212345 \times 10^3 = 212.345$, and $\mathrm{fl}(z) = -212.346$. This should be clear. If it isn't, then get out a pen or pencil...

Now for some arithmetic: $x + y = 212.348645678$ and $\mathrm{fl}(x+y) \equiv \mathrm{fl}(\mathrm{fl}(x) + \mathrm{fl}(y)) = \mathrm{fl}(.00324568 + 212.345) = \mathrm{fl}(212.34824568) = 212.348$.

What about $\mathrm{fl}(x+y+z)$? Well, this is really a trick question, because $\mathrm{fl}(x+y+z) = \mathrm{fl}(\mathrm{fl}(x) + \mathrm{fl}(y) + \mathrm{fl}(z))$ is not even well-defined; here is why:

$$\mathrm{fl}((x+y)+z) = \mathrm{fl}((\mathrm{fl}(x) + \mathrm{fl}(y)) + \mathrm{fl}(z)) = \mathrm{fl}(212.348 - 212.346) = 0.00200000,$$

whereas

$$\mathrm{fl}(x+(y+z)) = \mathrm{fl}(\mathrm{fl}(x) + (\mathrm{fl}(y) + \mathrm{fl}(z))) = \mathrm{fl}(0.00324568 - .001) = 0.00224568.$$

So floating point addition is *not* associative! Unless there is an agreement on the order of operations (*there isn't!*), $\mathrm{fl}(x+y+z)$ does *not* specify an algorithm, it is an *expression.* A final note: In this example,

$$\frac{|(x+y) - \mathrm{fl}(x+y)|}{|x+y|} = \frac{|212.348645678 - 212.348|}{212.348645678} \approx 0.000003.$$

This is about 0.0003% error (not unreasonable in 6-digit arithmetic). Now look at

$$\frac{|(y+z) - \mathrm{fl}(y+z)|}{|y+z|} = \frac{|-0.0002 + 0.001|}{0.0002} = 4,$$

which is a whopping 400% error! Clearly, we will need to revisit this...