Índice

Representación de datos numéricos

Hemos visto muchos ejemplos de sistemas de numeración: en base 6, en base 10 o decimal, en base 2 o binario, en base 16 o hexadecimal, y en base 8 u octal; y sabemos convertir la representación de un número en cada una de estas bases, a los sistemas en las demás bases. Sin embargo, aún nos falta considerar la representación numérica de muchos e importantes casos:

- Hemos utilizado estos sistemas para representar únicamente números enteros. Nos falta ver de qué manera representar números racionales, es decir aquellos que presentan una parte fraccionaria (los "decimales").
- Además estos enteros han sido siempre **no negativos**, es decir, sabemos representar únicamente el 0 y los naturales. Nos falta considerar los negativos.
- Por otra parte, no nos hemos planteado el problema de la **cantidad de dígitos**. Nuestros sistemas de numeración podían usar infinitos dígitos para representar números arbitrariamente grandes. Si bien esto es matemáticamente correcto, las computadoras son objetos físicos que tienen unas ciertas limitaciones, y pronto encontraremos que no es posible representar números de infinita cantidad de dígitos con ellas.

En esta parte de la unidad mostraremos sistemas de representación utilizados en computación que permiten tratar estos problemas.

Datos enteros

Veremos tres sistemas de representación de datos numéricos enteros, llamados **signo-magnitud**, **complemento a 2** y **exceso a** 2^{n-1}.

Datos fraccionarios

Para representar fraccionarios consideraremos los sistemas de **punto fijo** y **punto flotante**.

Rango de representación

Cada sistema de representación de datos numéricos tiene su propio **rango de representación** (que podemos abreviar **RR**), o intervalo de números representables. Nungún número fuera de este rango puede ser representado en dicho sistema. Conocer este intervalo es importante para saber con qué limitaciones puede enfrentarse un programa que utilice alguno de esos sistemas.

El rango de los números representados bajo un sistema está dado por sus **límites inferior y superior**, que definen qué zona de la recta numérica puede ser representada. Como ocurre con todo intervalo numérico, el rango de representación puede ser escrito como $[a, b]$, donde a y b son sus límites inferior y superior, respectivamente.

Por la forma en que están diseñados, algunos sistemas de representación sólo pueden representar números muy pequeños, o sólo positivos, o tanto negativos como positivos. En general, el RR **será más grande cuantos más dígitos binarios**, o bits, tenga el sistema. Sin embargo, el RR depende también de la forma como el sistema **utilice** esos dígitos binarios, ya que un sistema puede ser más o menos **eficiente** que otro en el uso de esos dígitos, aunque la cantidad de dígitos sea la misma en ambos sistemas.

Por lo tanto, decimos que el rango de representación depende a la vez de la **cantidad de dígitos** y de la **forma de funcionamiento** del sistema de representación.

Representación sin signo

Consideremos primero qué ocurre cuando queremos representar números enteros **no negativos** (es decir, **positivos o cero**) sobre una cantidad fija de bits. Simplemente usando el sistema binario de numeración, tal como lo conocemos, pero limitándonos a una cantidad fija de bits, podemos representar los números desde cero hasta un cierto límite superior.

Si la cantidad de dígitos binarios en este sistema es k, ¿cuál es el número más grande que podremos representar? Seguramente es aquel donde todos los k dígitos valen **1**. La Expresión General que hemos visto nos dice que si un número n está escrito en base 2, **con k dígitos**, entonces

$$n = x_{k-1} \times 2^{k-1} + ... + x_1 \times 2^1 + x_0 \times 2^0$$

y si queremos escribir el más grande de todos, serán todos los x_i iguales a 1.

Pregunta

¿Por qué si el número n tiene k dígitos binarios, el índice del más significativo es $k - 1$ y no k?

Esta suma vale entonces

$$x_{k-1} \times 2^{k-1} + ... + x_1 \times 2^1 + x_0 \times 2^0 =$$

$$
\begin{aligned}
&= 1 \times 2^{k-1} + ... + 1 \times 2^1 + 1 \times 2^0 = \\
&= 2^{k-1} + ... + 2^1 + 2^0 = \\
&= 2^k - 1
\end{aligned}
$$