Índice

Representación de datos numéricos

Hemos visto 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 varios casos importantes:

- Hemos utilizado estos sistemas para representar únicamente números **enteros**. Nos falta ver de qué manera representar números racionales, es decir aquellos que tienen 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**. Idealmente, un sistema de numeración puede 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 con ellas no es posible representar números de infinita cantidad de dígitos.

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 usamos el sistema binario de numeración, tal como lo conocemos, pero limitándonos a una cantidad fija de bits o dígitos binarios. ¿Cuál será el rango de representación?

El **cero** siempre puede representarse (es decir, el límite inferior del rango de representación será 0). Pero ¿cuál será el límite superior? Es decir, si la cantidad de dígitos binarios en este sistema es k, ¿cuál es el número más grande que podremos representar?

Podemos estudiarlo de dos maneras.

1. Usando combinatoria

Contemos cuántos números diferentes podemos escribir con k dígitos binarios. Imaginemos un número binario cualquiera con k dígitos. El dígito de más a la derecha tiene únicamente dos posibilidades (0 o 1). Por cada una de éstas hay

nuevamente dos posibilidades para el siguiente hacia la izquierda (lo que da las cuatro posibilidades 00, 01, 10, 11). Por cada una de éstas, hay dos posibilidades para el siguiente (dando las ocho posibilidades 000, 001, 010, 011, 100, 101, 110, 111), etc., y así hasta la posición k. No hay más posibilidades. Como hemos multiplicado 2 por sí mismo k veces, la cantidad de números que se pueden escribir es 2^k. Luego, el número más grande posible es $2^k - 1$. (**Pregunta**: ¿Por qué $2^k - 1$ y no 2^k?).

2. Usando álgebra

El número más grande que podemos representar en un sistema sin signo a k dígitos es, seguramente, 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, deberán ser 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

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

Usando ambos argumentos hemos llegado a que el número más grande que podemos representar con k dígitos binarios es $2^k - 1$. Por lo tanto, **el rango de representación de un sistema sin signo a k dígitos es** $[0, 2^k - 1]$. Todos los números representables en esta clase de sistemas son **positivos o cero**.

Representación con signo

En la vida diaria manejamos continuamente números negativos, y los distinguimos de los positivos simplemente agregando un signo "menos". Representar esos datos en la memoria de la computadora no es tan directo, porque, como hemos visto, la memoria **solamente puede alojar ceros y unos**. Es decir, ¡no podemos simplemente guardar un signo "menos"! Lo único que podemos hacer es almacenar secuencias de ceros y unos.

Esto no era un problema cuando los números eran no negativos. Para poder representar, ahora, tanto números **positivos como negativos**, necesitamos cambiar la forma de representación. Esto quiere decir que una secuencia particular de dígitos binarios, que en un sistema sin signo tiene un cierto significado, ahora tendrá un significado diferente. Algunas secuencias, que antes representaban números positivos, ahora representarán negativos.

Veremos los **sistemas de representación con signo** llamados **Signo-magnitud (S-M)**, **Complemento a 2 (C2)** y **En exceso a** $2^n - 1$**.

Es importante tener en cuenta que **solamente se puede operar entre datos representados con el mismo sistema de representación**, y que el **resultado** de toda operación **vuelve a estar representado en el mismo sistema**.

Sistema de Signo-magnitud (S-M)

El sistema de **Signo-magnitud** no es el más utilizado en la práctica, pero es el más simple de comprender. Se trata