Sistemas de numeración

Organización de Computadoras 2019

Universidad Nacional de Quilmes

1 Sistema Binario

Como sistemas de numeración hasta ahora conocemos el sistema decimal. Por ejemplo sabemos que 3548 representa el *tres mil quinientos cuarenta y ocho*, pero ¿cómo llegamos a esa conclusión? Probablemente sin darnos cuenta, lo que hicimos fue asignar **una potencia de 10** a cada posición y multiplicarla por el valor que tenemos en esa posición:

$$3 \times 10^3 + 5 \times 10^2 + 4 \times 10^1 + 8 \times 10^0$$

es decir:

$$3 \times 1000 + 5 \times 100 + 4 \times 10 + 8 \times 1$$

¿Por qué utilizamos potencias de 10? Pues porque el sistema decimal tiene 10 símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9.
Con estos símbolos se construyen cadenas que nos permiten representar todos los valores del conjunto de los números naturales, pero existen otros sistemas que tienen otros símbolos para representar los mismos números (ver figura 1).

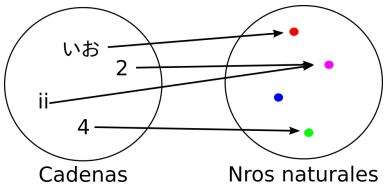


Figure 1: Conjuntos

Ahora, ¿y si quisiéramos operar en un sistema que tenga sólo 2 símbolos? Como tal es el caso del sistema binario, utilizamos **potencias de 2**, porque tiene base 2 y los símbolos que utiliza son 0 y 1.

Como todo sistema para poder operar con él vamos a necesitar conocer el conjunto de sus funcionalidades. En este caso podremos:

- Interpretar
- Representar
- Calcular su rango
- Realizar operaciones aritméticas

1.1 Interpretación

La interpretación nos sirve para determinar qué número representa una cadena. Por ejemplo se tiene la siguiente cadena 110101: debemos asignar una potencia de 2 a cada posición **empezando de derecha a izquierda y desde la potencia 0**, ubicando nuestra cadena y respetando las posiciones:

$$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

Finalmente sumamos todos los términos para obtener el número correspondiente en decimal.

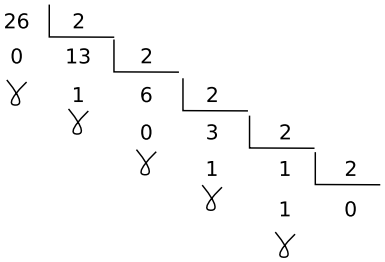
1.2 Representación

Es el proceso inverso a la interpretación, es decir, tenemos un número y queremos convertirlo en cadena. Para representar valores mediante cadenas se deben realizar sucesivas divisiones por la base correspondiente hasta obtener un cociente igual a 0 tomando cada resto como bits de la cadena. En el caso de cadenas binarias, será por 2, en caso de cadenas hexa será 16, y así sucesivamente.

Ejemplo:

Se necesita representar el número 26 en el sistema binario:

1. Se divide el valor 26 por 2 hasta encontrar un cociente 0
2. Se construye la cadena tomando solo los restos, empezando por el ultimo: 11010



Los dos mecanismos anteriores nos permiten relacionar los números naturales con las cadenas del sistema binario al momento de representar números o interpretar cadenas. Mediante la figura 2 podemos verlos gráficamente:

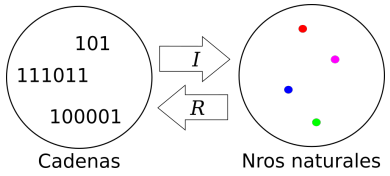


Figure 2: Mecanismos de interpretación y representación

1.3 Rango

Hasta ahora venimos hablando de los símbolos que utiliza cada sistema y cómo utilizarlos para interpretar y representar cadenas entre ellos, pero no hemos hablado de la cantidad de números que podemos llegar a representar. ¿Hay alguna restricción o todos los números son representables en todos los sistemas? Lo que nos lleva a preguntarnos si disponemos siempre de todos los recursos para operar con cada sistema, es decir, ¿la cantidad de bits que disponemos es infinita? La respuesta en no, por lo tanto un sistema de numeración está restringido a la cantidad de bits disponibles y en función de ello tendrá un mínimo y un máximo representable, y por ende el conjunto de números representables constituirá el rango del mismo (no el conjunto de cadenas, sino conjunto de números).

Consideremos por ejemplo un sistema binario restringido a 3 bits y que sólo contemple los números Naturales, lo llamamos *Sin Signo* y lo denotamos *(3)*. Calculemos su rango:

1. Determinamos el mínimo número representable: para ello interpretamos la cadena mas chica: 000 y esto nos da 0.
2. Determinamos el máximo número representable: para ello interpretamos la cadena mas grande: 111 y esto nos da 7.

Es decir, que nuestro rango son todos los números enteros comprendidos entre 0 y 7, esto se denota: [0, 7].

Si contamos la cantidad de elementos en el rango, da **8**. Dicho de otra manera, con **3** bits tenemos **8** números representables, es decir,

$$2^{3} = 8$$

Concluyendo, con n bits, es decir, BSS(n) tendremos como rango:

$$[0, 2^{n} - 1]$$

1.4 Operaciones Aritméticas

Como mencionamos previamente los sistemas de numeración deben proveer, además de los mecanismos de interpretación y representación, algoritmos para operar con esas cadenas: suma, resta, multiplicación y división.

Tal como hemos aprendido a sumar *"en papel"* para el sistema decimal pensaremos la suma como un algoritmo por columnas: se suman las columnas empezando por la de menor peso (a la derecha) y acarreando a la siguiente columna si corresponde. Por ejemplo, si en decimal debemos sumar dos dígitos cuya suma no alcanza el límite de la base (que es 10), como ser el caso de 9 + 3, en la base colocamos el 2 y *"nos llevamos 1"*, a este concepto lo llamamos acarreo, de la misma forma ocurre con el sistema binario.

$$\begin{array}{cc} 1 & \\ + & 9 \\ & 3 \\ \hline 1 & 2 \end{array}$$

Suponer la suma **01 + 11**. El resultado esperado es **1 + 3 = 4**, y su representación en BSS es **100**.

$$\begin{array}{cc} 0 & 1 \\ + \ 1 & 1 \\ \hline \end{array}$$

Veamos el cómputo de la primer columna: al sumar los bits 1 con 1, que representan el valor 1, se espera obtener el valor 2, cuya cadena es **10**, pero como excede un bit (el de la columna), en esta se deja el bit menos significativo y *"se arrastra"* el 1 a la siguiente columna.

$$\begin{array}{cc} 1 & \\ 0 & 1 \\ + \ 1 & 1 \\ \hline & 0 \end{array}$$

A continuación se computa la segunda columna, teniendo que sumar nuevamente los bits 1 (el arraste) con 1 (del segundo operando). También se obtiene la cadena **10** y nuevamente se deja el **0** en la columna actual y se *"se arrastra"* el **1** a la columna siguiente:

$$\begin{array}{ccc} 1 & 1 & \\ & 0 & 1 \\ + & 1 & 1 \\ \hline 1 & 0 & 0 \end{array}$$

¿Cómo comprobar que la suma anterior es correcta? Interpretando los operandos y el resultado.

2 Sistema Hexadecimal

Este es un sistema nuevo. En este caso, tal como su nombre lo indica, el sistema opera con 16 símbolos. Como no alcanza con los conocidos del sistema decimal que van del 0 al 9, se agregan también letras del alfabeto latino, quedando los 16 símbolos de la siguiente manera: *0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.*

Este sistema de numeración también contiene un conjunto de funcionalidades para poder operar con él y de la misma manera que en el sistema binario podremos:

1. Interpretar
2. Representar
3. Calcular su rango
4. Agrupación de bits

2.1 Interpretación

Traspolando el mismo razonamiento que hicimos para el sistema binario, en este caso asignamos una potencia de 16 a cada posición comenzando de derecha a izquierda (desde la potencia 0) y ubicamos nuestra cadena respetando las posiciones. Veamos varios ejemplos:

- Queremos interpretar la cadena 128, entonces:

$$1 \times 16^2 + 2 \times 16^1 + 8 \times 16^0$$

  El resto es sólo resolver la suma.

- Ahora bien, qué pasa si la cadena que necesitamos interpretar contiene una letra: 2A, entonces, siguiendo la misma lógica quedaría:

$$2 \times 16^1 + A \times 16^0$$

  Pero, ¿cómo hacemos para resolver una multiplicación con una letra en el 2do término? Para ello el sistema cuenta con una tabla que nos permite conocer qué número representa cada letra en el sistema decimal:

A	10
B	11
C	12
D	13
E	14
F	15

  Entonces, continuando con el ejemplo anterior, para poder resolverlo vamos a necesitar consultar dicha tabla. Traduciendo la A nos queda:

$$2 \times 16^1 + 10 \times 16^0$$

- Por último, también podremos tener una combinación de ambos: A3F

$$A \times 16^2 + 3 \times 16^1 + F \times 16^0$$

.

Traduciendo la A y la F nos queda:

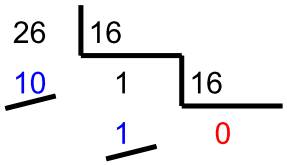
$$10 \times 16^2 + 3 \times 16^1 + 15 \times 16^0$$

.

2.2 Representación

Siguiendo la lógica del sistema binario, para representar valores mediante cadenas se deben realizar sucesivas divisiones por la base, que en este caso es 16, hasta obtener un cociente igual a 0 tomando cada resto como bits de la cadena.
Ejemplo:
Se necesita representar el número 26 en hexadecimal:

1. Se divide el valor 26 por 16 hasta encontrar un cociente 0
2. Se construye la cadena tomando solo los restos, empezando por el último



Uno de los restos es 10, entonces debemos traducirlo a la letra correspondiente aplicando la tabla de interpretación de hexadecimal. El valor **10** es equivalente a la letra `A`, quedando entonces `1A`. Esto quiere decir que el valor 26 en decimal se corresponde con la cadena `1A` en hexadecimal.

2.3 Rango

De la misma manera que en el sistema binario debemos calcular el mínimo número representable interpretando la cadena más chica y la más grande. Siendo el rango todos los números comprendidos entre ambos. Supongamos el sistema hexadecimal de 2 dígitos:

El mínimo valor representable es el resultado de interpretar la cadena `00`, es decir:

$$0 \times 16^1 + 0 \times 16^0 = 0$$

El máximo valor representable es el resultado de interpretar la cadena `FF`

$$15 \times 16^1 + 15 \times 16^0 = 255$$

(aplicando la tabla de interpretación de hexadecimal)
Por lo tanto el rango de este sistema es:

$$[0, 255]$$

2.4 Agrupación de bits

Este es un método que nos permite convertir de manera directa cadenas en binario a cadenas en hexadecimal.
Ejemplo, se tiene la siguiente cadena de 16 bits: **1001011010100101**. Debemos:

1. Separar los bits agrupando en grupos de a 4: **1001 0110 1010 0101**
2. Interpretar cada cuarteto por separado: **9 6 10 5**
3. De ser necesario, como en este caso, expresamos el valor en la letra correspondiente para representar la cadena en hexadecimal: **96A5**