Guía para el desarrollo del TP N°3
REPRESENTACIÓN DE DATOS

1) Representación de datos alfanuméricos.

Recordando lo visto anteriormente, indicaremos algunas diferencias entre EBCDIC y ASCII para representar caracteres.

- En EBCDIC y ASCII_8, las representaciones de cada carácter se realizan en un byte, dividido en una zona de 4 bits y una parte numérica de 4 bits.

```
 ________|________
|_|_|_|_|_|_|_|_|
 bits de|  bits
  zona     numéricos
```

- Cuando las letras se codifican en EBCDIC, se dividen en 3 grupos: de la A a la I tienen 1100 como zona, de la J a la R tienen 1101 y de la S a la Z tienen 1110.

- Cuando las letras se codifican en ASCII_8, se dividen en dos grupos: de la A a la O tienen 1010 como bits de zona y de la P a la Z tienen 1011.

- Cuando las letras se codifican en ASCII (de 7 bits), el octavo bit se utiliza como control de paridad, es decir que los bits de zona se reducen a 3; según se observa en la tabla, de la A a la O los bits son 100, de la P a la Z, 101.

- Para representar caracteres numéricos: los 3 sistemas utilizan la representacion binaria del dígito como la parte numérica del código; pero como bits de zona el EBCDIC utiliza: 1111, el ASCII_8: 0101 y el ASCII: 011.

Veremos algunos ejemplos en el EBCDIC y el ASCII:

CARÁCTER	HEX.	REP. EBCDIC	HEX.	REP. ASCII
D	C4	11000100	44	01000100
I	C9	11001001	49	01001001
M	D4	11010100	4D	01001101
T	E3	11100011	54	01010100
6	F6	11110110	36	00110110
9	F9	11111001	39	00111001

(según tablas)

2) Representación de datos numéricos:

- Formato de punto fijo.

En un sistema de representación de números naturales siempre tenemos una <u>base</u> y <u>cantidad de dígitos</u> **(b, d)**, Ej.: SN(10,3) -> sistema decimal de 3 dígitos.
La cantidad de <u>números representables</u> en un sistema es siempre $\mathbf{b^d}$.
Sabemos que el <u>Rango</u> de un sistema de representación *son los números que comprende*.

Los sistemas de representación de enteros (*"Formato de punto fijo"*) estan basados en los sistemas de naturales. SZ(SN, *técnica*). La técnica me dice a qué valor entero corresponde cada valor natural, con el cual se representará y almacenará en la computadora.

Hay 3 sistemas de representación de enteros:
- SVA = Signo valor absoluto ó Binario con signo;
- CB = Complemento a la base ó Complemento a dos;
- CD = Cero desplazado ó Característica del número.

SVA: en la representacion binario con signo, el bit mas significativo de la palabra se reserva para el signo (0: positivo, 1: negativo) y en los restantes bits, se representa el número en binario.

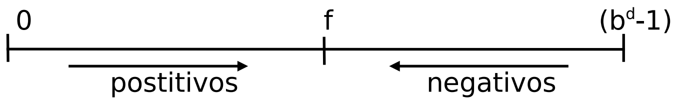
CB:

$n \in R(SN)$ y $e \in R(SZ)$ $\quad e = n - \delta(n) \times b^d \quad \delta(n) = 0$ si $n<f$

$f = b^d/2 \quad \delta(n) = 1$ si $n \geq f$

$n = e + \Phi(e) \times bd \quad \Phi(e) = 0$ si $e \geq 0$

$\Phi(e) = 1$ si $e<0$

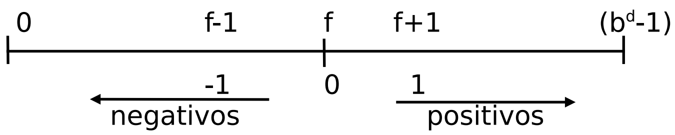


Una forma simple en complemento a 2, para representar un negativo es simplemente sumar 1 a la representacion en complemento a 1, luego de verificar que el número es representable en el sistema. La representación de complemento a 1 se explica más adelante.

CD:

$n \in R(SN)$ y $e \in R(SZ)$ $\quad e = n - f$

$f = b^d/2 \quad n = e + f$



Existe una representación de enteros alternativa, usada principalmente como representación intermedia en los cálculos internos o en grandes procesadores, llamada “Complemento a 1”, cuya forma establece:

todos los enteros positivos, se representan en formato binario correcto y los negativos, complementando cada uno de los dígitos de la representacion positiva.

Suponiendo una longitud de palabra de 16 bits, hallaremos las representaciones en binario con signo, complemento a 1, complemento a 2 y cero desplazado de los números indicados, mediante un procedimiento práctico:

ELEMENTOS DE INFORMÁTICA

Ejemplo: + 231 y - 231

```
B. c/ signo    |0|0|0|0|0|0|0|0|1|1|1|0|0|1|1|1|    +231

Comp. a uno    |0|0|0|0|0|0|0|0|1|1|1|0|0|1|1|1|    +231

Comp. a dos    |0|0|0|0|0|0|0|0|1|1|1|0|0|1|1|1|    +231

Cero despl.    |1|0|0|0|0|0|0|0|1|1|1|0|0|1|1|1|    +231


B. C/signo     |1|0|0|0|0|0|0|0|1|1|1|0|0|1|1|1|    -231

Comp. a uno    |1|1|1|1|1|1|1|1|0|0|0|1|1|0|0|0|    -231

Comp. a dos    |1|1|1|1|1|1|1|1|0|0|0|1|1|0|0|1|    -231

Cero despl.    |0|1|1|1|1|1|1|1|0|0|0|1|1|0|0|1|    -231


      +395 y -395

B. C/signo     |0|0|0|0|0|0|0|1|1|0|0|0|1|0|1|1|    +395

Comp. a uno    |0|0|0|0|0|0|0|1|1|0|0|0|1|0|1|1|    +395

Comp. a dos    |0|0|0|0|0|0|0|1|1|0|0|0|1|0|1|1|    +395

Cero despl.    |1|0|0|0|0|0|0|1|1|0|0|0|1|0|1|1|    +395


B.C/signo      |1|0|0|0|0|0|0|1|1|0|0|0|1|0|1|1|    -395

Comp. a uno    |1|1|1|1|1|1|1|0|0|1|1|1|0|1|0|0|    -395

Comp. a dos    |1|1|1|1|1|1|1|0|0|1|1|1|0|1|0|1|    -395

Cero despl.    |0|1|1|1|1|1|1|0|0|1|1|1|0|1|0|1|    -395
```

Ejemplos de sistemas de representación de enteros, o formatos de punto fijo, que utilizan los procesadores actuales son:

SZ(2,8,CB), SZ(2,16,CB)

- Formato de punto flotante

Con los números fraccionarios racionales e irracionales, más precisamente con los reales, que tienen tanto parte entera como fraccionaria, tenemos un rango de números posibles muy grande. Por ejemplo, en astronomía la masa del electrón 9 x 10^{-28} gramos o la del sol 2 x 10^{33} gramo, lo que supone una gama (rango) de números mayor a 10^{60}.

Como tanta cantidad de dígitos significativos no es necesaria porque dichas cifras no requieren tanta precisión, se necesita un sistema de representación en el que el rango de los números expresables no dependa de la cantidad de dígitos significativos.

Esto se logra con la notación científica empleada comúnmente en física, química e ingeniería. Para separar el rango y la precisión del número, esta notación lo expresa de esta forma: $n = f \times b^e$

Donde:

F = fracción o mantisa (con su signo), indica la precisión
E = exponente (número entero), indica el rango
B = base de la mantisa

Para las mantisas se utiliza su forma normalizada, a fin de evitar ambigüedades con ella. Para que una mantisa este normalizada el primer dígito a la derecha del punto decimal debe ser distinto de cero.

Resumiendo:

Un sistema de representación de punto flotante consta de un exponente y una mantisa, para una base de mantisa conocida. Por lo tanto se necesita valer de las representaciones de esos dos números enteros para representar un número real cualquiera. Generalmente en casi todas las computadoras, las mantisas están representadas con la técnica SVA y el exponente con CD.

Tomando como ejemplo la serie 360/370 de IBM, la cual utiliza una palabra de 32 bits para su formato de punto flotante, en la cual se distribuyen: signo de la mantisa, exponente de la mantisa, como un entero de punto fijo (para el signo del exponente) y el valor absoluto de la mantisa, tomando el punto fraccionario a la izquierda del mismo.

Las palabras de punto flotante de 32 bits tienen el siguiente formato:

```
 ______________________|___________________________________________________________
|S|1 -->     7|8 -->          M A  N T I S  A                        31|
 ______________________|___________________________________________________________
 signo   exponente                (como SZ(16,6,SVA)
 de la   (como SZ(2,7,CD)           en este caso + 1 bit por el
mantisa   o característica          signo que esta afuera
```

El bit del signo ocupa la posición del extremo izquierdo y da el signo del número.
Llamaremos factor de escala al resultado de elevar 16 a la potencia E, o sea $16^E = 16^{C-64}$; ya que 16 es la base del sistema hexadecimal que se usará para representar en forma normalizada la fracción contenida en la mantisa:

0,|________________| x 16^E
M A N T I S A

Es decir, que la magnitud real de un número representado en una palabra de punto flotante es igual al factor de escala multiplicado por la fracción contenida en los 24 bits correspondientes a la mantisa.

El punto de la raíz se supone que esta a la izquierda del último bit de la mantisa. Esto es, que si los 32 bits contienen:

signo	característica	mantisa		
0	1000001	11100000	00000000	00000000

32 bits

- El bit de signo es 0, o sea, el número representado es positivo.
- La característica, que es el natural almacenado que representa un entero, tiene un valor binario 1000001 es decir $(65)_{10}$, por lo tanto el exponente E = C - 64 es E = 65 - 64 = 1; esto sale de la técnica CD (cero desplazado) que se utiliza para representar al exponente. El factor de escala entonces es 16^1.

- La parte fraccionaria tiene un valor 0,111 binario, o sea:

$(0,111)_2=1x2^{-1} + 1x2^{-2} + 1x2^{-3} = (0,875)_{10}$

- El número representado es: $0,875x16^1 = 14$

Otro ejemplo:

Supongamos que quisiéramos conocer el valor decimal de la siguiente representación hexadecimal de punto flotante:

43	54	CB	9D

4 bytes

43	54 CB 9D
caract.	mantisa

Analicemos el byte correspondiente a la característica:

$(43)_{16}$ = (0 1000011 $)_2$
signo+ característica

- El bit de signo es 0, o sea el número representado es positivo.
- La característica tiene un valor binario 1000011 o sea $(67)_{10}$, por lo tanto el exponente del factor de escala es:

E = C - 64 = 67 - 64 = 3 ; E = 3

y el factor de escala es 16^3.
- La parte fraccionaria tiene un valor $(0,54CB9D)_{16}$ es decir el número hexadecimal es $(0,54CB9Dx10^3)_{16}$

- Convirtiendo a decimal:
$(0,54CB9D)_{16} = 5x16^{-1} + 4x16^{-2} + 12x16^{-3} + 11x16^{-4} + 9x16^{-5} + 13x16^{-6} = (0,331231892)_{10}$

El número representado será: $0,331231892x16^3 = 1356,726$
Normalizando: $0,1356726x10^4$

ELEMENTOS DE INFORMÁTICA

Haremos la operación inversa, es decir, representar un número decimal en formato de punto flotante hexadecimal:
Si el número es: $0{,}1356726 \times 10^4 = 1356{,}726$

- Su representación hexadecimal es:

$$(1356{,}726)_{10} = (54C{,}B9D)_{16}$$

```
1356 |_16_
  12   84,75|_16_
         4     5,25 |_16_
                 5     0
```

$0{,}726 \times 16 = \underline{11}{,}616$
$0{,}616 \times 16 = \underline{\ 9}{,}856$
$0{,}856 \times 16 = \underline{13}{,}696$

- se normaliza el número hexadecimal:

$$54C{,}B9D = 0{,}54CB9D \times (10)^{\overset{E}{3}}_{16}$$

$$C = E + 64 \quad \therefore \quad C = 3 + 64 = 67$$

- la representación hexadecimal en 32 bits:

```
 característica  mantisa normalizada
 ______________|_____________________
| 8 bits       |    24 bits          |   , será:
 ______________|_____________________
```

$C = (67)_{10} = (43)_{16}$ M = 54CB9D

43	54	CB	9D

32 bits

Tomando como ejemplo los equipos PDP-11 de Digital, los que utilizan también una palabra de 32 bits para su formato de punto flotante, en la cual se distribuyen: signo de la mantisa, exponente de la mantisa, como un entero de punto fijo (para el signo del exponente) y el valor absoluto de la mantisa, tomando el punto fraccionario a la izquierda del mismo.

Las palabras de punto flotante de 32 bits tienen el siguiente formato:

```
     ____________________|_____________________________________________
     |S|1 -->       8|9 -->          M A  N T I S  A                31|
     ____________________|_____________________________________________
 signo   exponente                      (como SZ(2,24,SVA)
 de la   (como SZ(2,8,CD)                 (la mantisa se explica abajo)
mantisa   o característica
```

El bit del signo ocupa la posición del extremo izquierdo y da el signo del número o mantisa, por lo cual el sistema de enteros utilizado SZ(2,24,SVA) no incluye el bit de signo como estudiamos, pero, si contamos tenemos sólo 23 bits almacenados. ¿Dónde esta el bit faltante?, como toda mantisa normalizada el dígito a derecha del punto decimal es distinto de cero, y como la base es binaria, entonces siempre será un 1. Por ello se elimina y no se lo almacena, pero debemos agregarlo para realizar las conversiones.

En este caso el factor de escala será 2 (base del sistema) a la potencia E, o sea $2^E = 2^{C-128}$; por lo tanto:

```
 0,|________________| x 2^E
       M A N T I S A
```

Ejemplos en la práctica.

<u>Algo para recordar</u>:

Si debemos hacer un cambio de base con un número cuya magnitud es muy grande o muy pequeña, o sea que esta expresado en notación científica, utilizamos la siguiente expresión:

$$M \times b^e = B^x \quad \rightarrow \quad x = (\ln M + e \ln b) / \ln B$$

Donde M es la mantisa del nro. a convertir,
b es la base del nro.
e es el exponente
B es la base a la que se quiera pasar
x es la incognita, que será el exponente a la que se elevará la B
Luego se debe operar para hallar una mantisa que quedará multiplicada por la base B y un exponente entero. En la práctica se demuestra con un ejemplo.

- Representación de números con EBCDIC y BCD.

En EBCDIC, cada dígito de un número esta representado por su código de caracteres excepto que la zona del último dígito se usa para el signo, con la forma:
1111 para enteros sin signo (o positivos)
1100 para enteros positivos
1101 para enteros negativos

Por ejemplo el número -274 en EBCDIC, se representa:

```
      2           7         (-)   4
 ___________|___________|___________
|1111  0010|1111  0111|1101  0100|
 ___________|___________|___________
```

+ 274

```
      2           7         (+)   4
 ___________|___________|___________
|1111  0010|1111  0111|1100  0100|
 ___________|___________|___________
```

o

```
      2           7           4
 ___________|___________|___________
|1111  0010|1111  0111|1111  0100|
 ___________|___________|___________
```

- El sistema BCD utiliza un código binario de 4 bits para cada dígito decimal y el último campo de bits para el signo.

```
    2        7        4
 ________|________|________|________
| 0010   | 0111   | 0100   | 1111   |
 ________|________|________|________

    2        7        4       (+)
 ________|________|________|________
| 0010   | 0111   | 0100   | 1100   |
 ________|________|________|________

    2        7        4       (-)
 ________|________|________|________
| 0010   | 0111   | 0100   | 1101   |
 ________|________|________|________
```