Práctica 1: Representación de números enteros

Gabriel Budiño

Organización del computador I
DC - UBA

1er. cuatrimestre 2018

Número vs Numeral

Imaginen que le piden a una persona que escriba el número dos en un papel. ¿Qué esperan ver?

Número vs Numeral

Imaginen que le piden a una persona que escriba el número dos en un papel. ¿Qué esperan ver?
Dependiendo de a quién se lo pidan, pueden obtener distintos resultados:

a un japonés	a mí	a un romano hace mucho tiempo
二	2	Ⅱ

Número vs Numeral

Imaginen que le piden a una persona que escriba el número dos en un papel. ¿Qué esperan ver?
Dependiendo de a quién se lo pidan, pueden obtener distintos resultados:

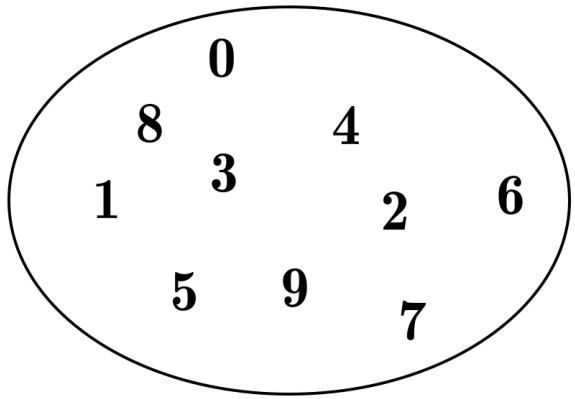
a un japonés	a mí	a un romano hace mucho tiempo
二	2	Ⅱ

Cada uno escribe algo distinto, pero piensa en lo mismo. Con lo que podemos decir que un **concepto** se puede representar de **varias** maneras.

Número vs Numeral

- **Número:** Abstracción que representa una cantidad o una magnitud.
- **Cifra:** Símbolo que representa un número.
- **Numeral:** Cadena de cifras que representa un número.

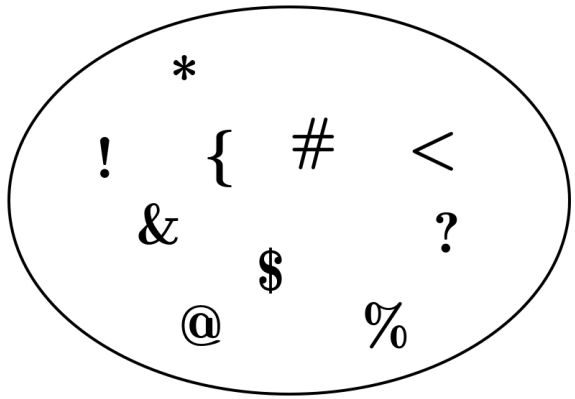
Sistema decimal



- Usa un conjunto de diez símbolos.
- Se dice un sistema posicional, pues cada posición en el numeral está asociada a una potencia de diez.
  Por ejemplo, $123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$

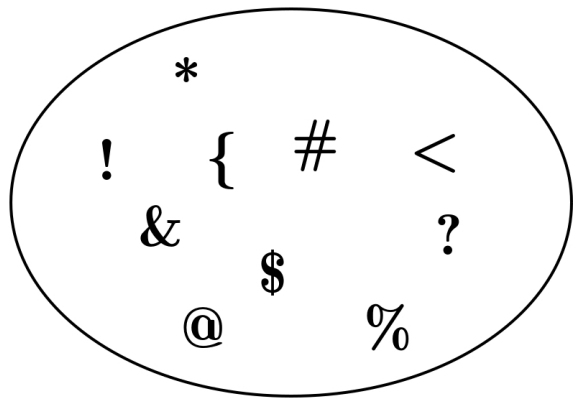
Sistema decimal alternativo

- ¿Podemos representar lo mismo si cambiamos los símbolos? Por ejemplo, a los siguientes:



Sistema decimal alternativo

- ¿Podemos representar lo mismo si cambiamos los símbolos? Por ejemplo, a los siguientes:



Sí, solo hay que asignarle a cada uno un número entre cero y nueve.

!	&	*	@	{	$	#	%	<	?
0	1	2	3	4	5	6	7	8	9

Base

- Como vimos, el sistema decimal usa un conjunto de diez símbolos para representar números.
- Cada posición del numeral está asociada a una potencia de diez.

¿Y si en lugar de diez símbolos usamos otra cantidad?

A esa cantidad la llamamos **base**.

Bases

- en base 2 usamos los símbolos 0 y 1 y escribimos los naturales: 0, 1, 10, 11, 100, 101 ...
- en base 3 usamos los símbolos 0, 1 y 2 y escribimos los naturales: 0, 1, 2, 10, 11, 12, 20 ...
- ... y así ...

Bases más comunes

Base	Símbolos usados
2 (binario)	0, 1
8 (octal)	0, 1, 2, 3, 4, 5, 6, 7
10 (decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 (hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

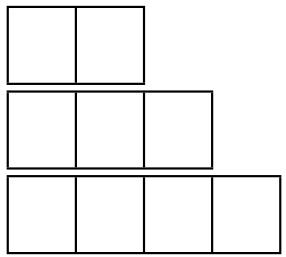
Ejercitemos

- **Cambiando de base:** Escribir en base 2, 8 y 16 los números diez y quinientos doce
- **Sumando:** Sumar en base 3, 212 y 101
- **Multiplicando:** Multiplicar en base 3, 12 y 21

Tiras de símbolos

Si sólo pudiésemos escribir numerales de **longitud fija**:

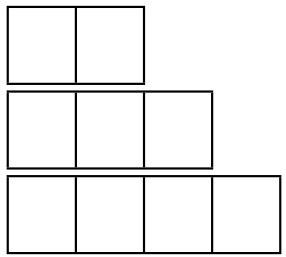
- ¿Cuántos números podemos representar?
- ¿De qué depende?



Tiras de símbolos

Si sólo pudiésemos escribir numerales de **longitud fija**:

- ¿Cuántos números podemos representar?
- ¿De qué depende?



- A esto se lo llama precisión fija.
- Cuando operamos con precisión fija puede haber overflow. Esto ocurre cuando necesitamos una tira de mayor longitud para representar al resultado.

Recordando

Hasta ahora vimos como:

- Leer naturales
- Escribir naturales
- Operar con naturales

¿Y los enteros?

Codificando enteros con numerales binarios

- Sin signo
  - como venimos usando
  - solo sirve para enteros positivos
- Exceso a m
  - se representa a n como m + n
  - desplaza m lugares las representaciones sin signo
- Signo + Magnitud
  - el primer bit representa el signo (1 $\equiv$ -, 0 $\equiv$ +)
  - los bits restantes representan la magnitud del número
- Complemento a 2
  - el primer bit indica el signo (1 $\equiv$ -, 0 $\equiv$ +)
  - los positivos se representan igual que antes
  - los negativos n se representan como 2^k + n, siendo k = #bits

Codificando

- En base 2, con numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3			
-2			
-8			

Codificando

- En base 2, con numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	Overflow
-2			
-8			

Codificando

- En base 2, con numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	Overflow
-2	1010	1110	1101
-8			

Codificando

- En base 2, con numerales de 4 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0011	0011	Overflow
-2	1010	1110	1101
-8	Overflow	1000	0111

Codificando con más bits

- En base 2, con numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3			
-2			
-8			

Codificando con más bits

- En base 2, con numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3			0001 0010
-2			
-8	1000 1000		

Codificando con más bits

- En base 2, con numerales de 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 1010	1111 1110	0000 1101
-8	1000 1000	1111 1000	0000 0111

Codificando con más bits

- Similitudes entre 4 y 8 bits

	Signo + Magnitud	Complemento a 2	Exceso a 15
3	0000 0011	0000 0011	0001 0010
-2	1000 0010	1111 1110	0000 1101
-8	1000 1000	1111 1000	0000 0111

- Signo + Magnitud: Se extiende con 0's, pero el bit más significativo se mantiene indicando el signo.
- Complemento a 2: Se extiende con el valor del bit más significativo.
- Exceso a m: Se extiende siempre con 0's.

Codificando enteros con numerales binarios

Sin signo

Solo sirve para enteros no negativos.

numeral $\rightarrow$ número que representa

1111	$\rightarrow$	15_{10}	0111	$\rightarrow$	7_{10}
1110	$\rightarrow$	14_{10}	0110	$\rightarrow$	6_{10}
1101	$\rightarrow$	13_{10}	0101	$\rightarrow$	5_{10}
1100	$\rightarrow$	12_{10}	0100	$\rightarrow$	4_{10}
1011	$\rightarrow$	11_{10}	0011	$\rightarrow$	3_{10}
1010	$\rightarrow$	10_{10}	0010	$\rightarrow$	2_{10}
1001	$\rightarrow$	9_{10}	0001	$\rightarrow$	1_{10}
1000	$\rightarrow$	8_{10}	0000	$\rightarrow$	0_{10}

Codificando enteros con numerales binarios

Sin signo

Solo sirve para enteros no negativos.

numeral $\rightarrow$ número que representa

1111	$\rightarrow$	15_{10}	0111	$\rightarrow$	7_{10}
1110	$\rightarrow$	14_{10}	0110	$\rightarrow$	6_{10}
1101	$\rightarrow$	13_{10}	0101	$\rightarrow$	5_{10}
1100	$\rightarrow$	12_{10}	0100	$\rightarrow$	4_{10}
1011	$\rightarrow$	11_{10}	0011	$\rightarrow$	3_{10}
1010	$\rightarrow$	10_{10}	0010	$\rightarrow$	2_{10}
1001	$\rightarrow$	9_{10}	0001	$\rightarrow$	1_{10}
1000	$\rightarrow$	8_{10}	0000	$\rightarrow$	0_{10}

Rango: $[0, 2^k - 1]$

Codificando enteros con numerales binarios

Exceso a m

El número n se representa como m + n (m = 5 en el ejemplo)

cuentas → numeral → número que representa

$5+(10)=15$	$\rightarrow$	1111	$\rightarrow$	10_{10}	$5+(2)=7$	$\rightarrow$	0111	$\rightarrow$	2_{10}
$5+(9)=14$	$\rightarrow$	1110	$\rightarrow$	9_{10}	$5+(1)=6$	$\rightarrow$	0110	$\rightarrow$	1_{10}
$5+(8)=13$	$\rightarrow$	1101	$\rightarrow$	8_{10}	$5+(0)=5$	$\rightarrow$	0101	$\rightarrow$	0_{10}
$5+(7)=12$	$\rightarrow$	1100	$\rightarrow$	7_{10}	$5+(-1)=4$	$\rightarrow$	0100	$\rightarrow$	-1_{10}
$5+(6)=11$	$\rightarrow$	1011	$\rightarrow$	6_{10}	$5+(-2)=3$	$\rightarrow$	0011	$\rightarrow$	-2_{10}
$5+(5)=10$	$\rightarrow$	1010	$\rightarrow$	5_{10}	$5+(-3)=2$	$\rightarrow$	0010	$\rightarrow$	-3_{10}
$5+(4)=9$	$\rightarrow$	1001	$\rightarrow$	4_{10}	$5+(-4)=1$	$\rightarrow$	0001	$\rightarrow$	-4_{10}
$5+(3)=8$	$\rightarrow$	1000	$\rightarrow$	3_{10}	$5+(-5)=0$	$\rightarrow$	0000	$\rightarrow$	-5_{10}

Codificando enteros con numerales binarios

Exceso a m

El número n se representa como m + n (m = 5 en el ejemplo)

cuentas → numeral → número que representa

$5+(10)=15$	$\rightarrow$	1111	$\rightarrow$	10_{10}	$5+(2)=7$	$\rightarrow$	0111	$\rightarrow$	2_{10}
$5+(9)=14$	$\rightarrow$	1110	$\rightarrow$	9_{10}	$5+(1)=6$	$\rightarrow$	0110	$\rightarrow$	1_{10}
$5+(8)=13$	$\rightarrow$	1101	$\rightarrow$	8_{10}	$5+(0)=5$	$\rightarrow$	0101	$\rightarrow$	0_{10}
$5+(7)=12$	$\rightarrow$	1100	$\rightarrow$	7_{10}	$5+(-1)=4$	$\rightarrow$	0100	$\rightarrow$	-1_{10}
$5+(6)=11$	$\rightarrow$	1011	$\rightarrow$	6_{10}	$5+(-2)=3$	$\rightarrow$	0011	$\rightarrow$	-2_{10}
$5+(5)=10$	$\rightarrow$	1010	$\rightarrow$	5_{10}	$5+(-3)=2$	$\rightarrow$	0010	$\rightarrow$	-3_{10}
$5+(4)=9$	$\rightarrow$	1001	$\rightarrow$	4_{10}	$5+(-4)=1$	$\rightarrow$	0001	$\rightarrow$	-4_{10}
$5+(3)=8$	$\rightarrow$	1000	$\rightarrow$	3_{10}	$5+(-5)=0$	$\rightarrow$	0000	$\rightarrow$	-5_{10}

Rango: $[-m,\ 2^k - 1 - m]$

Codificando enteros con numerales binarios

Signo + Magnitud

Se usa el primer bit para el signo y los restantes para el módulo.

numeral $\rightarrow$ número que representa

1111	$\rightarrow$	-7_{10}	0111	$\rightarrow$	7_{10}
1110	$\rightarrow$	-6_{10}	0110	$\rightarrow$	6_{10}
1101	$\rightarrow$	-5_{10}	0101	$\rightarrow$	5_{10}
1100	$\rightarrow$	-4_{10}	0100	$\rightarrow$	4_{10}
1011	$\rightarrow$	-3_{10}	0011	$\rightarrow$	3_{10}
1010	$\rightarrow$	-2_{10}	0010	$\rightarrow$	2_{10}
1001	$\rightarrow$	-1_{10}	0001	$\rightarrow$	1_{10}
1000	$\rightarrow$	-0_{10}	0000	$\rightarrow$	0_{10}

Codificando enteros con numerales binarios

Signo + Magnitud

Se usa el primer bit para el signo y los restantes para el módulo.

numeral $\rightarrow$ número que representa

1111	$\rightarrow$	-7_{10}	0111	$\rightarrow$	7_{10}
1110	$\rightarrow$	-6_{10}	0110	$\rightarrow$	6_{10}
1101	$\rightarrow$	-5_{10}	0101	$\rightarrow$	5_{10}
1100	$\rightarrow$	-4_{10}	0100	$\rightarrow$	4_{10}
1011	$\rightarrow$	-3_{10}	0011	$\rightarrow$	3_{10}
1010	$\rightarrow$	-2_{10}	0010	$\rightarrow$	2_{10}
1001	$\rightarrow$	-1_{10}	0001	$\rightarrow$	1_{10}
1000	$\rightarrow$	-0_{10}	0000	$\rightarrow$	0_{10}

Rango: $[-(2^{k-1}-1),\ 2^{k-1}-1]$

Codificando enteros con numerales binarios

Complemento a 2

Los numerales que representan positivos son iguales a los anteriores
Dado un n negativo, se lo representa escribiendo $2^k + n$ en notación sin signo.

cuentas	→	numeral	→	número que representa			
$2^4 + (-1) = 15$	→	1111	→	-1_{10}	0111	→	7_{10}
$2^4 + (-2) = 14$	→	1110	→	-2_{10}	0110	→	6_{10}
$2^4 + (-3) = 13$	→	1101	→	-3_{10}	0101	→	5_{10}
$2^4 + (-4) = 12$	→	1100	→	-4_{10}	0100	→	4_{10}
$2^4 + (-5) = 11$	→	1011	→	-5_{10}	0011	→	3_{10}
$2^4 + (-6) = 10$	→	1010	→	-6_{10}	0010	→	2_{10}
$2^4 + (-7) = 9$	→	1001	→	-7_{10}	0001	→	1_{10}
$2^4 + (-8) = 8$	→	1000	→	-8_{10}	0000	→	0_{10}

Codificando enteros con numerales binarios

Complemento a 2

Los numerales que representan positivos son iguales a los anteriores
Dado un n negativo, se lo representa escribiendo $2^k + n$ en notación sin signo.

cuentas	→	numeral	→	número que representa			
$2^4 + (-1) = 15$	→	1111	→	-1_{10}	0111	→	7_{10}
$2^4 + (-2) = 14$	→	1110	→	-2_{10}	0110	→	6_{10}
$2^4 + (-3) = 13$	→	1101	→	-3_{10}	0101	→	5_{10}
$2^4 + (-4) = 12$	→	1100	→	-4_{10}	0100	→	4_{10}
$2^4 + (-5) = 11$	→	1011	→	-5_{10}	0011	→	3_{10}
$2^4 + (-6) = 10$	→	1010	→	-6_{10}	0010	→	2_{10}
$2^4 + (-7) = 9$	→	1001	→	-7_{10}	0001	→	1_{10}
$2^4 + (-8) = 8$	→	1000	→	-8_{10}	0000	→	0_{10}

Rango: $[-2^{k-1}, 2^{k-1} - 1]$

¿Qué vimos hoy?

- Diferencia entre número y numeral.
- Cómo interpretar números en distintas bases.
- Cómo expresar un número en distintas bases.
- Precisión fija y overflow.
- Distintas formas de representación de enteros.

¿Qué sigue?

- **Bibliografía:**
  - William Stallings: Computer Organization and Architecture, capítulos 9 y 10
  - Linda Null & Julia Lobur: The Essentials of Computer Organization and Architecture, capítulo 2
- **La práctica:**
  - Con lo visto hoy pueden realizar toda la guía 1 de ejercicios.
- **La próxima clase:**
  - En instantes... clase de Lógica Digital.