Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2343 — Arquitectura de Computadores (2020-2)

Ayudantía 1

Representación de Números Enteros y Racionales

1 Representación Números Enteros

Ejercicio 1: T1 2018/2

a) Indique la base β en la cual la siguiente ecuación es correcta: $7_\beta + 8_\beta = 13_\beta$

Siguiendo la fórmula posicional en base decimal, tenemos que:

$13_\beta = \beta^1 * 1_{10} + \beta^0 * 3_{10} = \beta + 3_{10}$

Ahora para la parte izquierda de la igualdad:

$\beta^0 * 7_{10} + \beta^0 * 8_{10} = 7_{10} + 8_{10} = 15_{10}$

Si igualamos ambas expresiones, tenemos que:

$\beta + 3_{10} = 15_{10}$

$\beta = 15_{10} - 3_{10}$

$\beta = 12_{10}$

b) ¿Para qué números $\alpha \in \mathbb{R}$, existe β, tal que $\alpha = 10\beta$? Indique una expresión analítica que caracterice β en función de α.

Sabemos que: $\alpha = 10\beta$

Ahora si nuevamente utilizamos la fórmula posicional en base decimal:

$\alpha = \beta^1 * 1_{10} + \beta^0 * 0_{10} = \beta + 0$

Por ende: $\alpha = \beta$

Ejercicio 2: **EX** 2020/1

a) Transforme el número 10645_7 a base 14.Incluye tu procedimiento.

NOTA: todos los números debiesen ir con un subíndice 14, representando la base en la que se encuentran. Por temas de legibilidad se omiten.

$10645_7 = 5 * 7^0 + 4 * 7^1 + 6 * 7^2 + 0 * 7^3 + 1 * 7^4$

$10645_7 = 5 + 4 * 7 + 6 * 7 * 7 + 7 * 7 * 7 * 7$

Iremos viendo término a término:

$4 * 7 = 7 + 7 + 7 + 7$

Aquí tomaremos un par de 7's y seguimos la lógica de sumar $7 + 7$ tal como si estuviesemos en base decimal. Sin embargo, nos podemos dar cuenta que el resultado es 14_{10} lo cual en base 14 no podemos representar de esa forma (solo contamos con los dígitos $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D$. Al obtener 14 al hacer la suma decimal, debo restar 14 de ese resultado, obteniendo un 0 para la posición 0 y se genera un "carry" o "reserva" para la posición 1 de una unidad (restamos 14 una vez).

De esta forma resulta: $7 + 7 = 10_{14}$

De esta manera, $7 + 7 + 7 + 7 = 10 + 10 = 20_{14}$

Ahora, seguimos con el $7 * 7$:

$7 * 7 = 7 + 7 + 7 + 7 + 7 + 7 + 7$

Si agrupamos de a dos y seguimos el razonamiento anterior, tenemos:

$7 * 7 = 10 + 10 + 10 + 7 = 37_{14}$

Añadiendo todo lo anterior en la expresión inicial:

$10645_7 = 5 + 20 + 6 * 37 + 37 * 37$

$10645_7 = 25 + (6 + 37) * 37$

Ahora sumamos $6 + 37$, donde primero sumamos 6 con 7 obteniendo 13 en decimal, que corresponde a la letra D en base 14. Y luego sumamos el 3 con 0, y obtenemos de resultado final de la suma $3D$.

$10645_7 = 25 + (3D) * 37$

Para simplificar la multiplicación dividimos el factor 37 en $37 = 30 + 7$. De esta forma, primero buscamos multiplicar $3D * 30$ y luego sumarselo a $3D * 7$.

Para multiplicar $3D * 30$ seguimos la misma idea de la multiplicación que nos enseñaron en el colegio, pero teniendo en consideración que la cantidad máxima que podemos representar utilizando solo una cifra en base 14 corresponde al número decimal 13 o D. De esta forma cualquier multiplicación que sobrepase este número, debo restarle repetitivamente 14 hasta lograr una cantidad que no supere el

número 13. Recordar que la cantidad de veces que le resto 14 corresponde al carry o "reserva" que debo considerar en la siguiente multiplicación.

Bajo esta lógica, tenemos que:

$3D * 30 = BB0$

$3D * 7 = 1D7$

Así reemplazando obtenemos:

$10645_7 = 25 + BB0 + 1D7$

$10645_7 = BD5 + 1D7$

$10645_7 = DCC$

Ejercicio Propuesto: I1 2016/2

a) Demuestre que el complemento a 2 del complemento a 2 de un número x es igual a x, *i.e.*, $x = C_2(C_2(x))$. **HINT:** $C_2(x + y) = C_2(x) + C_2(y)$**.**

$x + C_2(x) = 0$ $/C_2()$

$C_2(x + C_2(x)) = 0$ /**utilizamos el hint**

$C_2(x) + C_2(C_2(x)) = 0$ /**utilizo igualdad** (1)

$C_2(x) + C_2(C_2(x)) = x + C_2(x)$

$C_2(C_2(x)) = x$

2 Representación Números Racionales

Ejercicio 1: I1 2014/1

a) Escriba en formato float el número -48. Indique cómo se compone y qué significa cada una de las partes de la secuencia de bits.

Primero recordemos que el estándar IEEE754 de 32 bits incluye:

1 bit de signo
8 bits de esponente (con desfase de 127)
23 bits de significante o mantisa normalizado

En primer lugar, dado que el número -48 es negativo, el bit de signo es 1.

Ahora necesitamos pasar el 48 a binario, lo cual corresponde a: $48_{10} = 110000b$

Debemos normalizar este exponente (dejarlo de la forma $1, ...$), lo que en este caso equivale a correr la coma 5 espacios hacia la izquierda. De esta forma:

$48 = 110000b = 1,10000 * 2^5$

De aquí obtenemos que el significante es: 10000000000000000000

Ahora, para el exponente, nos podemos dar cuenta al normalizar el significante, que el exponente en decimal corresponde a 5. A este número debemos agregarle el desfase de 127, por ende:

$exponente = 5 + 127 = 132.$

Al pasar el 132 a binario se obtiene: $132_{10} = 10000100b$.

Ahora si juntamos las tres partes en una sola secuencia de bits, obtenemos que la representación en IEEE754 para el -48 es:11000010010000000000000000000000.

Ejercicio 2: I2 2020/1

a) Explica clara y detalladamente: ¿Por qué, por la convención IEEE754, al sumar $2^{31} + 1$ obtenemos 2^{31}? Incluye los valores representados en notación científica pero en binario.

Al escribir el 2^{31} en binario se obtiene: $10000000000000000000000000000000b$

De aquí podemos ver que:
El significante son puros 0.
El bit de signo corresponde a 0 (es un número positivo).
El exponente es 31, pero al sumarle el desfase de 127 se obtiene un exponente de 158. Este en binario corresponde a $158 = 10011110b$

Así si en notación científica resulta: $1,0_2 * 2^{10011110b}$.
Y bajo la convención IEEE754 se representa de la forma: 01001111000000000000000000000000

Ahora, el número 1 puede expresarse de la forma: $1,0 * 2^0$, que en notación científica en binario lo único que cambia es el exponente. Al tener un desfase de 127, que en binario es $01111111b$, tenemos que la representación científica en binario es: $1,0_2 * 2^{01111111b}$.

Para poder hacer la suma, debemos igualar los exponentes. Dado que $127 < 158$, igualamos al exponente mayor (158). Esto es equivalente a mover la coma del número 1, 31 espacios hacia al izquierda, lo que resulta en el número:

$0,00000000000000000000000000000001_2 * 2^{10011110b}$

Ahora como tienen el mismo exponente puedo sumar los significantes:

$0,00000000000000000000000000000001 + 1,0 = 1,00000000000000000000000000000001$

Ahora, al traspasarlo al formato IEEE754, tenemos que:

El bit de signo corresponde a 0. (Sumamos dos números positivos).

El exponente corresponde a 158 o en binario $10011110b$ que corresponde a 31 en decimal (si resto los 127 de desfase.
Para determinar el significante, me quedo con los 23 bits después de la coma (que se posiciona después del primer 1). En este caso, dado que el segundo 1 recién está en la posición 31, tenemos que el significante son puros 0.

Dada esta descripción nos podemos dar cuenta que el número que representamos es:
01001111000000000000000000000000.
El cual es la representación de 2^{31}.

La idea detrás del ejercicio es mostrar como al sumar números tan diferentes en magnitud (uno muy grande como 2^{31} en comparación al 1) y dada la restricción de precisión que tiene la convención de IEEE754, tenemos que la cifra que "marca la diferencia" se encuentra fuera del rango que incide en el significante (solo tomamos los primeros 23 bits que le siguen a la coma) por lo que a pesar de estar sumando una unidad, se obtiene el valor inicial.

Ejercicio Propuesto: I1 2014/1

a) Considere el estándar IEEE754 para la representación de números de punto flotante de precisión simple (32 bits). ¿Cuántos números se pueden representar con este estándar,sin considerar $\pm\infty$ ni NaN?

Consideremos que existen 2^{32} combinaciones posibles.
De las cuales:

2 corresponden a $+\infty$ y $-\infty$.

$2^{23} - 1$ corresponden a NaN, ya que para caer en esta categoría, algún bit del significante es $\neq 0$ y le restamos el caso en que todos son 0. Esto debo multiplicarlo por 2, dada la posibilidad del bit de signo (puede ser 0 o 1).

Finalmente, el cero tiene dos representaciones $+0y - 0$, por lo que restamos una.

Así se tiene que la cantidad de números que se pueden representar son:

$2^{32} - 2 - 2 * (2^{23} - 1) - 1 = 4278190079$