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

IIC2343 – Arquitectura de Computadores (II/2018)

Tarea 1 - Pauta

Fecha de entrega: lunes 20 de agosto de 2018 a las 23:59 horas

Rúbrica:

- **1 punto:** Respuesta correcta con una justificación sin errores.
- **0.5 puntos:** La respuesta está correcta pero tiene errores de conceptos.
- **0 puntos:** Errores constantes, falta de justificación o en blanco.

Pregunta 1

a) Indique la base β en la cual la siguiente ecuación es correcta: $7_\beta + 8_\beta = 13_\beta$.
Respuesta: Para responder esta pregunta se debe considerar que: $13_\beta = \beta^1 \cdot 1 + \beta^0 \cdot 3 = 7_{10} + 8_{10} = 15_{10}$. O sea,

$$\beta + 3 = 15$$
$$\beta = 12$$

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.
Respuesta: Esto ocurre para los números en los que $\alpha = \beta$. Sabemos que:

$$\alpha = 10_\beta$$

Si lo pasamos a base decimal:

$$\alpha = \beta^1 \cdot 1 + \beta^0 \cdot 0$$
$$\alpha = \beta$$

c) Extienda el análisis del ítem anterior, considerando esta vez $\alpha = 10^\gamma_\beta$. Indique las expresiones analíticas relevantes para caracterizar β y γ.
Respuesta: Esto ocurre para los números en los que $\alpha = \beta^\gamma$. Sabemos que:

$$\alpha = 10^\gamma_\beta$$

Si lo pasamos a base decimal:

$$\alpha = (\beta^1 \cdot 1 + \beta^0 \cdot 0)^\gamma$$
$$\alpha = \beta^\gamma$$

Pregunta 2

Considere una representación posicional de base β, en la cual se desean representar números enteros.

a) Considere el conjunto numérico $K_t = \{k \in \mathbb{N} \mid k \leq t\}$. Considere además una representación posicional binaria de n **bits** para estos números. Si se desea ahora representar el conjunto $\hat{K}_t = \{k \in \mathbb{Z} \mid |k| \leq t\}$, ¿cuántos **bits** necesita como mínimo la nueva representación posicional? Considere que la nueva representación debe ser algebraicamente consistente.

Respuesta: Aquí lo importante es notar que queremos incluir la representación de números negativos. Si la representación de números naturales ocupa un total de n bits, entonces:

$$|K_t| \leq 2^n$$

Es decir, el conjunto numérico K_t puede tener, a lo más, 2^n elementos (dado que es la cantidad de números que podemos representar en este contexto). A través del mismo análisis, podemos decir que:

$$|\hat{K}_t| \leq 2^{\hat{n}}$$

Siendo $\hat{n}$ la cantidad de bits necesarias para representar al conjunto $\hat{K}_t$. Ahora, como queremos considerar la inclusión de la representación negativa de todos los números del conjunto original, se deduce que:

$$2^{\hat{n}} = 2^n + (2^n - 1)$$

$$2^{\hat{n}} = 2^{n+1} - 1$$

Vemos que puede tener, como mucho, **casi** el doble de elementos del conjunto K_t (la unidad que se resta corresponde a la representación del 0, que mantenemos igual por consistencia). De esta forma:

$$\log_2 2^{\hat{n}} = \log_2(2^{n+1} - 1)$$

$$\log_2 2^{\hat{n}} = \lceil \log_2(2^{n+1} - 1) \rceil$$

Notar que se usa la función **techo**, dado que el número de bits debe ser una cantidad natural. Luego, desarrollando:

$$\hat{n} = \left\lceil \log_2(2^{n+1}) + \log_2(1 - \frac{1}{2^{n+1}}) \right\rceil$$

$$\hat{n} = \left\lceil (n+1) + \log_2(1 - \frac{1}{2^{n+1}}) \right\rceil$$

Dado que $-1 \leq \log_2(1 - \frac{1}{2^{n+1}}) \leq 0$:

$$\hat{n} = n + 1$$

Es decir, necesitamos de un bit adicional en la nueva representación del conjunto $\hat{K}_t$ para incluir los inversos aditivos de los elementos en K_t. Finalmente, basta con hacer uso del complemento a 2 dentro de nuestra representación para poder mantener la consistencia algebraica.

b) Al utilizar complemento a 2 para representaciones posicionales binarias, se genera una representación no equilibrada, donde existen más elementos negativos que positivos. Modifique el algoritmo de transformación, tal que ahora existan más números positivos que negativos.
Respuesta: Basta con realizar un simple cambio al algoritmo de complemento a 2 visto en clases:

- Anteponer un 1 al número (y no un 0, como en el algoritmo original).
- Negar todos los bits.
- Sumar 1.

Por construcción, se puede ver entonces que se sigue respetando el resultado nulo ante la suma entre un número y su inverso aditivo obtenido a partir de este algoritmo.

No es válido simplemente decir que los positivos ahora son negativos y viceversa para luego usar el algoritmo de siempre. Se pedía explícitamente hacer un cambio en este y no simplemente cambiar la asignación. Por otra parte, se puede corroborar que la asociación de números positivos y sus inversos aditivos **cambia**, no es exactamente la inversión del caso original.

c) Caracterice el conjunto de bases de representaciones posicionales, tales que al aplicar su correspondiente complemento, se genere una representación equilibrada, con la misma cantidad de elementos positivos y negativos. Demuestre que este conjunto existe y es infinito.
Respuesta: La clave es notar que las bases requeridas son **impares**. La particularidad de estas es que la cantidad de números que pueden representar es también impar. Es decir, la potencia de un número impar también lo es.

Demostración: Utilizando la representación de número impar $2k+1$, tenemos que:

$$(2k+1)^N = \sum_{i=0}^{N} \binom{N-i}{i} (2k)^{N-i} 1^i = \binom{N}{0}(2k)^N 1^0 + \binom{N}{1}(2k)^{N-1}1^1 + ... + \binom{0}{N}(2k)^0 1^N$$

Es fácil ver que todos los términos, salvo el último (que es igual a 1), son factores de 2 (esto, debido a las potencias de $2k$). Entonces, podemos abreviar esta expresión como:

$$(2k+1)^N = 2k' + 1$$

la que corresponde a un número impar. ■

Habiendo verificado este hecho, vemos que al existir una cantidad impar de elementos, basta con asignar la unidad "sobrante" al cero, de forma que nos quede una cantidad que se puede dividir de forma equitativa para números positivos y negativos, operable a partir del complemento **de la base**. La única salvedad a tener con estas representaciones es que el dígito más significativo no será siempre coherente con el signo del número (por ejemplo, números positivos y negativos con el mismo bit más significativo). No obstante, este detalle no influye la consistencia de las operaciones aritméticas, cumpliendo siempre $X + C_\beta(X) = 0$ (con β la base impar utilizada).

En resumen, si tomamos C_β como el conjunto de representaciones posicionales de base β y C el conjunto pedido, entonces:

$$C = \{C_\beta \mid \exists k \in \mathbb{N}.\ 2 \cdot k + 1 = \beta\}$$

Nota: La demostración utilizada en este ejercicio no era necesaria.