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

IIC2343 – Arquitectura de Computadores (II/2020)

Solución Tarea 1

1. Representaciones Numéricas

a) (T) Asumiendo que se usan registros del mismo tamaño y que procesar floats es siempre más lento que procesar enteros, ¿cuál podría ser una ventaja de representar enteros como floats en un computador? Justifique su respuesta detalladamente. **(1.0 pto.)**

Solución: Una posible ventaja es que con `float` es posible representar números de mayor valor absoluto con la misma cantidad de bits (exponente en float puede moverse más posiciones).

Asignación de puntaje:

- 0,5 ptos. por indicar alguna ventaja (correcta).
- 0,5 ptos. por justificar la ventaja razonablemente.

b) (T) Un programa lee desde la memoria principal de un computador, cuatro palabras contiguas a partir de la dirección 0x36. Al interpretar estos datos como un número entero, el programa obtiene el valor decimal 230. ¿Cuál sería el valor del número leído, si se lee el dato cambiando el endiannes original y luego reinterpreta como `float`? Justifique su respuesta mostrando los pasos que lo llevaron a ella. **(1.0 pto.)**

Solución: Si se asume little endian inicialmente, 230 se representa como 0xE6000000 y al pasar a big endian queda como 0x000000E6. Finalmente, al pasarlo a float obtenemos 0 00000000 00000000000000011100110, que al interpretarlo como decimal queda $3.22298646795 \times 10^{-43}$. En el otro sentido (asumiendo big endian inicialmente) el procedimiento es análogo.

Asignación de puntaje:

- 0,5 ptos. por hacer el cambio de endianness correctamente.
- 0,5 ptos. por transformar a float e interpretar el valor.

c) (T) En una representación posicional de número enteros de **K** cifras con base **N**, ¿cuántos números son iguales a su complemento a **N**? Justifique su respuesta formal y detalladamente. **(2.0 ptos.)**

Solución: El aspecto principal es notar que dependiendo de la paridad de la base, se generará una cantidad distinta de elementos independiente del valor de K: si N es par la cantidad de elementos generados son una cantidad par, y si N es impar la cantidad es impar. Luego, dado que todas las representaciones deben incluir solo una versión del número cero, las bases pares siempre tendrán un elemento negativo (o positivo) más, codificado como $\frac{N}{2}$ en la cifra más significativa, seguido de ceros. Al aplicar complemento a N, dado que la precisión está acotada a K cifras, el resultado siempre será el mismo número (al igual que el cero), por lo que las bases pares siempre tendrán 2 elementos iguales a su complemento. En el caso de las bases impares, esta situación no se da, ya que negativos y positivos tienen la misma cantidad de elementos, de lo que se desprende que solo el 0 será igual a su complemento.

Asignación de puntaje:

- 1 pto. por identificar la diferencia entre bases pares e impares.
- 0,5 ptos. por justificar para el caso de las bases pares.
- 0,5 ptos. por justificar para el caso de las bases impares.

d) (P) Considere una lista E de números naturales menores o iguales a 100, donde E_i es el elemento ubicado en el índice i de esta. Escriba un programa en Python que reciba como entrada una lista E y retorne una lista S, donde el elemento en el índice i de esta corresponda al sucesor de 2^{E_i} en el tipo de dato IEEE754 de doble precisión (tipo `float` en Python). Su solución debe consistir en una única función que realice los cálculos necesarios y que cumpla con la siguiente definición: `def sucesor(E: [int]) -> [float])`. **(2.0 ptos.)**

Nota: se define como sucesor de un número x en representación IEEE754 de doble precisión, como el número entero mayor más cercano a x, que puede ser representado en IEEE754 de doble precisión.

Solución: La respuesta debe considerar e identificar que a partir de cierta base el sucesor se encuentra a una distancia mayor a 1, debido a que la cifra menos significativa del significante, al ser elevada al exponente, es mayor a 1. El algoritmo ideal debiese tomar la codificación de 2^{E_i}, sumarle 1 al significante y verificar el resultado de esta operación: si el número queda igual el sucesor es $2^{E_i} + 1$, si el número se modifica, el sucesor es el resultado entregado por esta operación.

Asignación de puntaje por casos:

- 2.0 ptos. por usar el algoritmo ideal descrito anteriormente, o uno análogo.
- 2,0 ptos. por usar fuerza bruta para encontrar el resultado u otro algoritmo, siempre y cuando se justifique el por qué el sucesor está a una distancia mayor a uno, y que los resultados sean correctos
- 1,5 ptos. por usar fuerza bruta sin justificar.
- 1,0 ptos. por usar algún algoritmo razonable, pero con resultados incorrectos en algunos pocos casos de borde (inicio y fin del rango por ejemplo).
- 0 ptos. en cualquier otro caso. Esto incluye soluciones donde la lógica no está clara y/o donde hay errores en casos no de borde.
- 0,5 ptos. de descuento por no respetar el formato indicado en el enunciado.

2. Programabilidad

De acuerdo a Wikipedia, una máquina de Turing es un "dispositivo que manipula símbolos en una cinta de acuerdo con una tabla de reglas"[1]. Es posible demostrar que una máquina de Turing se puede adaptar para simular la lógica de cualquier algoritmo ejecutable por un computador. En este ejercicio, deberá (de)mostrar la implicación inversa, es decir, que un computador puede ejecutar lo mismo que una máquina de Turing.

a) (T) Diagrame los componentes de un computador que simule el funcionamiento de una máquina de Turing. Haga explícitas las relaciones entre los elementos de la máquina de Turing y del computador. **(3.0 ptos.)**

Solución: La soluciones de las partes a) y b) están muy relacionadas, ya que casi cualquier diseño propuesto en la a) puede ser correcto, si las instrucciones de la parte b) cubren lo necesario de la máquina de Turing y además pueden ser ejecutadas en la máquina propuesta en a).

Asignación de puntaje:

- 2.0 ptos. por incluir componentes análogos a los siguientes: cinta, cabezal, estado actual, reglas. Cada uno de estos que falte descontará 0,4 ptos. En caso de no ser análogos, debe quedar claro cómo se maneja esa funcionalidad en el computador propuesto.
- 1 pto. por conectar de manera razonable los componentes anteriores, de manera que la máquina puede funcionar (aunque lo haga mal). Errores menores en conexiones descuentan 0,3 ptos en total como máximo. Si las conexiones no tienen sentido, no hay puntaje.

b) (T) Describa las instrucciones para la máquina del ítem anterior. Indique opcodes y señales de control. **(2.0 ptos.)**

Solución: La soluciones de las partes a) y b) están muy relacionadas, ya que casi cualquier diseño propuesto en la a) puede ser correcto, si las instrucciones de la parte b) cubren lo necesario de la máquina de Turing y además pueden ser ejecutadas en la máquina propuesta en a). Las funcionalidades que tienen que estar sí o sí soportadas con las siguientes:

- Leer el símbolo actual desde la cinta.
- Leer el estado actual de la máquina.
- Escribir en la posición actual de la cinta.
- En base al estado y símbolo actual, mover el cabezal o dejarlo quieto.
- Actualizar el estado actual de la máquina.

Cada una de estas funcionalidades puede estar implementada por una o más instrucciones de la máquina propuesta. En caso de necesitarse más de una instrucción, debe quedar claro como componer la funcionalidad mediante estas instrucciones. Cada una de las instrucciones de la máquina propuesta debe tener una palabra de control y/o un opcode. No es necesario que la máquina tenga un assembly.

Asignación de puntaje:

- 2.0 ptos. por incluir todas las funcionalidades con sus instrucciones asociadas. Por cada uno de estas funcionalidades que falte se descontarán 0,4 ptos.

c) (T) ¿Son exactamente equivalentes la máquina de Turing y el computador descrito anteriormente? Justifique su respuesta. **(1.0 pto.)**

Solución: La respuesta depende fuertemente de las partes a) y b), pero algo básico que deben identificar todas es la extensión no infinita de cinta.

[1] `https://es.wikipedia.org/wiki/M%C3%A1quina_de_Turing`

3. Assembly

Escriba los siguientes programas en el assembly del computador básico. Utilice el emulador del computador básico disponible en el sitio del curso para ejecutar y probar sus soluciones. Para cada pregunta, se indicarán los nombres de los labels donde se encontrarán los datos y donde se deben entregar los resultados. Estos labels **no pueden ser usados en sus respuestas finales**, ya que serán de uso reservado para el corrector automático. Fuera de estos labels, no hay limitante para utilizar otros, tanto para datos como para subrutinas. Con el fin de evitar errores de formato, se subirán al sitio del curso archivos de muestra que podrán utilizar con el emulador.

a) (P) Dado un grafo no dirigido y dos de sus nodos, escriba un programa que encuentre la distancia (número de aristas) que separan a estos dos nodos en el grafo. Para construir su solución, debe asumir lo siguiente:

- El grafo será codificado como una matriz de adyacencia, almacenada a partir de la dirección indicada por el label `M` en orden de filas. Cada entrada en esta matriz utilizará 1 byte.
- La cantidad de nodos del grafo se almacenará en la dirección indicada por el label `N`. La cantidad de nodos del grafo siempre estará en el rango $[2, 12]$.
- Los índices de los nodos comienzan en 0.
- El índice del primer nodo se almacenará en la dirección indicada por el label `n1` y el índice del segundo se almacenará en la dirección indicada por el label `n2`.
- La distancia entre los nodos debe ser almacenada en la dirección indicada por el label `res`.

(3.0 ptos.)

Solución: El puntaje de este ejercicio se contabilizó completamente en base al uso correcto del assembly, es decir, no hubo descuento por corrección automática. Para obtener el puntaje completo, la respuesta entregada debe cumplir lo siguiente:

- No deben existir errores de sintaxis (0,9 ptos.).
- La respuesta debe cumplir las exigencias especificadas en el enunciado (0,9 ptos.).
- El algoritmo utilizado debe tener sentido, aunque no esté implementado correctamente (1,2 ptos.).

b) (P) Dada un arreglo de número enteros, donde cada uno se encuentra en el rango $[-60, 60]$, encuentre el índice de aquel que corresponde a la mediana de este conjunto. En el caso que el tamaño del arreglo sea par, puede elegir cualquiera de los dos elementos centrales como respuesta. Para construir su solución, debe asumir lo siguiente:

- El arreglo estará codificado de manera continua en memoria, a partir de la dirección indicada por el label `arr`. Cada entrada del arreglo utilizará 1 byte.
- La cantidad de elementos en el arreglo se almacenará en la dirección indicada por el label `N`. La cantidad de elementos del arreglo siempre estará en el rango $[0, 50]$.
- Los índices de los elementos del arreglo comienzan en 0.
- El índice de la mediana debe ser almacenado en la dirección indicada por el label `res`.

(3.0 ptos.)

Solución: El puntaje de este ejercicio se contabilizó completamente en base al uso correcto del assembly, es decir, no hubo descuento por corrección automática. Para obtener el puntaje completo, la respuesta entregada debe cumplir lo siguiente:

- No deben existir errores de sintaxis (0,9 ptos.).
- La respuesta debe cumplir las exigencias especificadas en el enunciado (0,9 ptos.).
- El algoritmo utilizado debe tener sentido, aunque no esté implementado correctamente (1,2 ptos.).