

Tarea I (10 puntos)

A continuación encontrará 4 preguntas, cada una compuesta de diferentes sub-preguntas. El valor de cada pregunta (y sub-pregunta) estará expresado entre paréntesis al inicio de las mismas.

En aquellas preguntas donde se le pida ejecutar un algoritmo o procesar una entrada, incluya los pasos relevantes de la ejecución del mismo con los cuales usted pudo alcanzar su conclusión. Sea lo más detallado y preciso posible en sus razonamientos y procedimientos.

La entrega se realizará por correo electrónico a rmonascal@gmail.com hasta las 11:59pm. VET del Domingo 05 de Junio de 2025.

1. (1.5 puntos) Considere el siguiente fragmento de código, escrito en TAC:

```
a := b + c
b := a / 2
b := d - c
a := a + d
c := b * a
```

Al final de esta instrucción, las variables vivas son: $\{b, d\}$.

Para cada instrucción, establezca el conjunto de variables vivas y la información de uso futuro antes y después de cada una de éstas.

2. (3.5 puntos) En clase, vimos como calcular el número de *Ershov* para expresiones de uno ($\otimes a$) o dos operandos ($a \otimes b$). Considere ahora expresiones de la siguiente forma: $\otimes(a, b, c)$ y código de cuatro direcciones FAC (*Four Address Code*), con instrucciones de la forma $\otimes R_r R_0 R_1 R_2$ que corresponde a $R_r := \otimes(R_0, R_1, R_2)$. Nótese que $TAC \subset FAC$, en el sentido que todas las instrucciones disponibles en TAC también están disponibles en FAC.

- a) (0.5 puntos) ¿Cómo se calcula $label(n)$, cuando n tiene tres hijos: c_{left} , c_{middle} y c_{right} ?
- b) (1 punto) Describa el proceso de generación de código usando este etiquetado extendido.
- c) (1 punto) Use el proceso diseñado en la pregunta anterior para generar código FAC de la siguiente expresión de alto nivel:

$(a \text{ ? } b : c) + d * e$

Puede suponer que tiene registros ilimitados.

- d) (1 puntos) Para la misma expresión de la pregunta anterior, calcule los costos de evaluación contigua usando el algoritmo de programación dinámica. Para esto, suponga que tiene a los sumo tres (3) registros disponibles.

3. (2 puntos) Considere la siguiente instrucción, escrita en el lenguaje de alto nivel a compilar:

`we[have] = to[go[deep] + er]`

- a) (0.5 puntos) Traduzca a TAC, suponiendo que los símbolos `we`, `to` y `go` corresponden a la dirección del primer elemento de arreglos de enteros.
- b) (1.5 puntos) Genere código de máquina para el fragmento empleando el algoritmo basado en Asignación de Registros según Descriptores de Uso y Asignación. Suponga que dispone de tres (3) registros para el cómputo de las expresiones. Muestre los descriptores en cada paso de generación.

4. (3 puntos) Sea un número entero positivo n . Se construye un nuevo número n' de la siguiente forma:

- Si n es par, entonces $n' = \frac{n}{2}$.
- Si n es impar, entonces $n' = 3 \times n + 1$

La conjetura de Collatz establece que si se vuelve a aplicar este proceso de forma repetida, tomando n' como entrada al mismo proceso, eventualmente se alcanza el número 1.

- a) (0.5 puntos) Escriba un programa en pseudo-código que, dado un n , devuelva la cantidad de pasos necesarios para alcanzar 1 siguiendo el procedimiento descrito anteriormente.
- b) (0.5 puntos) Traduzca su programa a TAC.
- c) (1 punto) Construya el grafo de flujo para el programa en TAC generado.
- d) (1 punto) Construya el grafo de interferencia y genere código con Asignación de Registros por Coloración de Grafos, suponiendo que se dispone de dos (2) registros.