

57GIIN – Lenguajes de Programacion y Procesadores de Lenguaje

Actividad 2 - Portafolio

Gagliardo Miguel Angel

10 de Mayo de 2024

Ejercicio 1

Escribe el **pseudocódigo** correspondiente, usando la notación que hemos estudiado en clases para la implementación de la generación de código de tres direcciones mediante **cuadrúplos** para la construcción **if-then-else**.

Tomando el siguiente pseudocódigo como ejemplo de partida:

```
read x;
if 0 < x then
  result = 1;
else
  result = 0;
write result;
end;
```

En este pseudocódigo, se verifica si x es mayor a 0. Si es así (**if-then**), se asigna 1 a result, si x no es mayor que 0 (**else**), se asigna 0 a result. Finalmente se escribe (guarda) aquello que se ha asignado a result y finaliza el programa.

```
(read, x, -, -)      // Lee el valor de x
(isbigger, t1, x, 0)  // Comprueba si x es mayor a 0
(if_false, t1, L1, -) // Si x NO es mayor a 0, salta a label L1
(assign, result, 1, -) // Si x ES mayor a 0, asigna 1 a result
(goto, L2, -, -)     // Saltar a L2
(label, L1, -, -)     // Etiqueta L1
(assign, result, 0, -) // Asigna 0 a result
(label, L2, -, -)     // Etiqueta L2 (fin del if-then-else)
(write, result, -, -) // Escribe el valor de result
(halt, -, -, -)      // Detiene la ejecución
```

Ejercicio 2

Escribe el **pseudocódigo** correspondiente, usando la notación que hemos estudiado en clases para la implementación de la generación de código de tres direcciones mediante **cuadрупlos** para la construcción **repeat-until**.

Tomando el siguiente pseudocódigo como ejemplo de partida:

```
contador = 0;
repeat
    read numero;
    contador = contador + 1;
until numero > 10;
write contador;
end;
```

Este pseudocódigo de ejemplo cuenta cuántos números (menores a 10) se ingresan. Inicia contador con un 0 y, dentro del bucle **repeat-until**, se lee el valor de la variable numero y se aumenta el valor de contador en 1. Se verifica si numero es mayor que 10, si es así, el bucle termina y se detiene la ejecución del programa, de otra manera vuelve a leer el numero hasta que el ingresado sea mayor a 10.

```
(assign, contador, 0, -)           // Inicializa contador en 0
(label, repeticion, -, -)          // Etiqueta repeticion para el bucle repeat-until
(read, numero, -, -)               // Lee un número desde la entrada
(add, contador, contador, 1)       // Incrementa el contador en 1
(isbigger, mayor_a_10, numero, 10) // Verifica si el número ingresado es mayor a 10
(if_true, mayor_a_10, fin, -)      // Si es mayor a 10, goto label fin
(goto, repeticion, -, -)           // Vuelve al inicio del bucle (label repeticion)
(label, fin, -, -)                 // Label fin
(write, contador, -, -)            // Escribe el valor de contador
(halt, -, -, -)                   // Detiene la ejecución del programa
```

Nota: Es verdad que el programa no es exacto (en rigor, debería haber un chequeo con un **if** antes del bucle repeat-until para que contador nunca se incremente en 1) pero el punto del ejercicio es mostrar la implementación de **repeat-until** mediante **cuadрупlos**, y no específicamente el correcto funcionamiento del programa.

Referencias

1. 57GIIN - Lenguajes de Programacion y Procesadores del Lenguaje - Materiales del Profesor - Tema 5 - Generación y Optimización de Código. Generación