

## Examen II – Lenguajes y Autómatas.

29 – Noviembre – 2018

Nombre: Jesús Alexis Jimenez Reyna

Matricula: 15480465

**1. Es la fase cuyo objetivo consiste en modificar el código objeto generado por el generador de código, para mejorar su rendimiento.**

**R:** La optimización de código

**3. Cuáles son los tipos de optimización.**

Las optimizaciones se pueden dividir en dos grandes grupos, en función de que se puedan aplicar únicamente en una maquina concreta, o en cualquiera.

- Optimizaciones dependientes de la maquina:

Minimización del uso de registro en maquinas en las que no se dispongan de un conjunto de registros muy grandes.

Uso de instrucciones especiales de la maquina

- Optimizaciones independientes de la maquina:

Mejora la eficiencia sin depender de la maquina concreta utilizada

**4. Ejemplos de optimización dependientes de la máquina.**

```
w = 0;  
x = x + y;  
y = 0;  
if( x > z)  
{  
    y = x;  
    x++;  
}  
else  
{  
    y = z;  
    z++;  
}  
w = x + z;
```

Código fuente

```
w = 0;  
x = x + y;  
y = 0;  
if( x > z)
```

```
y = x;  
x++;
```

```
y = z;  
z++;
```

```
w = x + z;
```

Bloques básicos

## 5. Ejemplos de optimización independientes de la máquina.

En esta optimización, el compilador toma en el código intermedio y transforma una parte del código que no implique un registro de la CPU y/o ubicaciones de memoria absoluta. Por ejemplo:

```
do
{
    item = 10;

    value = value + item;
} while(value<100);
```

Este código implica repetir la asignación de elemento identificador, que si ponemos esta forma:

```
Item = 10;

do
{
    value = value + item;
} while(value<100);
```

## 6. Explica con tus palabras como funciona el reordenamiento de código.

permite reducir el tamaño o la complicación del código objeto. Esto pasa, por ejemplo, cuando hay que calcular varias veces el mismo resultado intermedio: generándolo una sola vez antes de utilizarlo, se puede obtener una versión optimizada.

## 7. Explica un ejemplo de optimización en tiempo de compilación utilizando cuádruplas.

```
int i;
float f;
i = 2+3;      (+, 2, 3, t1)      (=, 5, , i)
              (=, t1, , i)
i = 4;        (=, 4, , i)        (=, 4, , i)
f = i+2.5;     (CIF, i, , t2)     (=, 6.5, , f)
              (+, t2, 2.5, t3)
              (=, t3, , f)
```

La ejecución se aplica principalmente a las operaciones aritméticas (+-\*/ ) y a las conversiones de tipo.

## 8. Explica un ejemplo de eliminación de redundancias.

```
int a,b,c,d;
a = a+b*c;    (*, b, c, t1)      (*, b, c, t1)
              (+, a, t1, t2)      (+, a, t1, t2)
              (=, t2, , a)        (=, t2, , a)
```

```

d = a+b*c;      (*,b,c,t3)
                 (+,a,t3,t4)      (+,a,t1,t4)
                 (=,t4,,d)        (=,t4,,d)

b = a+b*c;      (*,b,c,t5)
                 (+,a,t5,t6)
                 (=,t6,,b)        (=,t4,,b)

```

Una solución: el programador podría reescribir su programa así:

```

int a,b,c,d,e;
e = b*c;      (*,b,c,t1)
              (=,t1,,e)

a = a+e;      (+,a,e,t2)
              (=,t2,,a)

d = a+e;      (+,a,e,t3)
              (=,t3,,d)

b = d;        (=,d,,b)

```

9. Explica un ejemplo de reordenamiento de operaciones.

Ejemplo	
Problema	Simplifica $3 + 5 \cdot 2$ .
	$3 + 5 \cdot 2$ El orden de operaciones te dice que hagas la multiplicación antes que la suma.  $3 + 10$ Ahora suma.
<i>Respuesta</i>	$3 + 5 \cdot 2 = 13$

10. Quien es el Crush famoso de su profesor.

