Examen II – Lenguajes y Autómatas.
29 – Noviembre – 2018

Nombre: Aaron Uises Nuñez Avila
Matricula: 15480456

1. **Es la fase cuyo objetivo consiste en modificar el código objeto generado.**

   Optimización de código

2. **por el generador de código, para mejorar su rendimiento.**

   Compilador

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

   - ✘ Optimizaciones dependientes de la máquina
   - ✘ Optimizaciones independientes de la máquina

4. **Ejemplos de optimización dependientes de la máquina.**
   - ✘ Minimización del uso de registros en máquinas en las que no se disponga de un conjunto de registros muy grande. Puede llegar a generarse código que utilice sólo un registro.
   - ✘ Uso de instrucciones especiales de la máquina, que supongan una optimización respecto al uso de construcciones más generales, presentes en todos los lenguajes de máquina.
   - ✘ Reordenación de código: algunas arquitecturas son más eficientes cuando las operaciones se ejecutan en un orden determinado. Modificando el código para sacar provecho de ese orden se puede optimizar el programa objeto.
5. **Ejemplos de optimización independientes de la máquina.**
   - ✘ Ejecución parcial del código por parte del compilador, en lugar de retrasar su ejecución al programa objeto.
   - ✘ Eliminación de código que resulta redundante, porque previamente se ha ejecutado un código equivalente.
   - ✘ Cambio de orden de algunas instrucciones, que puede dar lugar a un código más eficiente.
   - ✘ Es frecuente que los bucles sean poco eficientes, porque se ejecuten en su cuerpo instrucciones que podrían estar fuera de él, o porque la reiteración inherente al bucle multiplique la ineficiencia causada por el uso de operaciones costosas, cuando podrían utilizarse otras menos costosas y equivalentes.

6. **Explica con tus palabras como funciona el reordena miento de código.**

   Es oredenar nueva mente el codigo lo cual reduce el tamaño al momento de compilarlo ademas cuando se compila el codigo para pruebas este proporciona una version optimizada

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

Se va a aplicar la optimización de ejecución en tiempo de compilación al siguiente bloque de programa,

escrito en el lenguaje C:

```
{    int i;
     float f;

     i=2+3;
     i=4;
     f=i+2.5;    }
```

La siguiente secuencia de cuádruplas es equivalente al bloque anterior. En ellas se utiliza el operador CIF, que significa convertir entero (integer) en real (float).

```
(+,      2,     3,   t1)
(=,     t1,      ,    i)
(=,      4,      ,    i)
(CIF,    i,      ,   t2)
(+,     t2,   2.5,   t3)
(=,     t3,      ,    f)
```

Cuádruplas	T	Tratamiento
(=, 4, , i)	{(t1,5), (i,4)}	**Caso 4**: • Se elimina de T el par (i, 5). • 4 es un valor constante, se añade (i, 4) a T.
(CIF, i, , t2)	{(t1,5), (i,4), (t2,4.0)}	**Caso 1:** • Se ***sustituye*** en la cuádrupla i por 4. ***(CIF, 4, , t2)*** **Caso 3:** CIF 4 se evalúa sin errores, su resultado es 4.0. • Se ***elimina*** la cuádrupla. • No hay ningún par para t2 en T. • Se añade a T el par (t2, 4.0) en.
(+,t2, 2.5, t3)	{(t1,5), (i,4), (t2,4.0), (t3,6.5)}	**Caso 1:** • Se ***sustituye*** en la cuádrupla t3 por 4.0. ***(+, 4.0, 2.5, t3)*** **Caso 3:** 4.0+2.5 se evalúa sin errores, su resultado es 6.5. • ***Se elimina*** la cuádrupla. • No hay ningún par para t3 en T. Se añade a T el par (t3, 6.5) en.
(=,t3, , f)	{(t1,5), (i,4), (t2,4.0), (t3,6.5), (f,6.5)}	**Caso 1:** • Se ***sustituye*** en la cuádrupla t3 por 6.5. ***(=, 6.5, , f)*** **Caso 4**: • No hay ningún par para f en T. • 6.5 es un valor constante, se añade (f, 6.5) a T.

Cuádrupla original	Resultado
(+, 2, 3, t1)	Eliminada
(=, t1, , i)	(=, 5, , i)
(=, 4, , i)	(=, 4, , i)
(CIF, i, , t2)	Eliminada
(+, t2, 2.5, t1)	Eliminada
(=, t3, , f)	(=, 6.5, , f)

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

   Esta optimización también se relaciona, casi siempre, con el código generado para las expresiones aritméticas.
   int a,b,c,d;
   a = a+b*c; (*, b, c, t1)
   (+, a, t1, t2)
   (=, t2, , a)
   d = a+b*c; (*, b, c, t3)
   (+, a,t3, t4)
   (=, t4, , d)
   b = a+b*c; (*, b, c, t5)
   (+, a,t5, t6)
   (=, t6, , b)

- La primera almacena el valor de b*c en una nueva variable temporal.
- La segunda almacena en una nueva variable temporal el valor de la suma con la variable a de la variable temporal creada en la primera cuádrupla.
- La tercera asigna el resultado, contenido en la variable temporal creada en la segunda cuádrupla, a la variable correspondiente, según el código fuente.

9. **Explica un ejemplo de reordenamiento de operaciones.**

   cuando hay que calcular varias veces el mismo resultado intermedio: generándolo una sola vez antes de utilizarlo, se puede obtener una versión optimizada.

10. **Quien es el Crush famoso de su profesor.**