
Lenguajes de Programación y Procesadores de Lenguajes

(2º parcial)

16 de enero de 2018

1. Dado el siguiente programa C:

```
-----  
float G (int a, float b)  
{ float i=3.0;  
  ...  
  return b * i;          // <---- (1)  
}  
struct{int c1; float c2; int c3} x;  
int mem;  
}  
void main ()  
{ int i=2, A[10]; float k=5.0;  
  ...  
  printf("%f\n", G(i, k));      // <---- (2)  
}  
-----
```

- a) (0,75 ptos.) Suponiendo que la talla de enteros es 2, la de los reales es 4 y la talla del segmento de enlaces de control es 8, mostrad el contenido completo de la TDS en los puntos de control (1) y (2).
- b) (0,75 ptos.) ¿Qué es, y para qué sirve el Registro de Activación (RA) y cómo está organizado su contenido? Como ejemplo mostrad el RA de la función `G`.

2. Contestad brevemente a las siguientes cuestiones:

- a) (0,75 ptos.) Considerando que se ha completado la fase de declaración de los objetos, diseñad un ETDS para la comprobación de tipos y la generación de código intermedio para la regla:

$$S \rightarrow \text{id } [E] = E$$

- b) (0,75 ptos.) Describid brevemente en que consiste la optimización de código intermedio denominada “*eliminación de código inactivo*”, cómo se resuelve y como se aplicaría, por ejemplo, en el siguiente segmento de código intermedio:

(100)	$y = k * 4$	(102)	$y = k * 10$
(101)	$x = A[y]$	(103)	$x = B[y]$

- c) (0,5 ptos.) En el marco de la Generación y Optimización de Código, describid brevemente el problema de la asignación de registros y explicad en que consiste la solución propuesta por coloreado de grafos.

1.a)

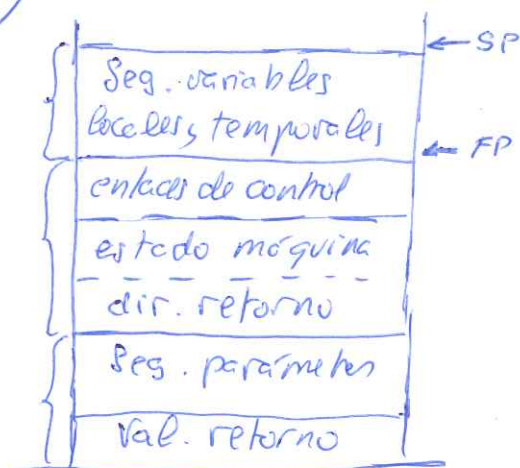
TDB		
0	0	0
1	1	3

TDS (punto 1)				
0	G	0	-	función $\Rightarrow ((tentero \times treal) \rightarrow treal)$
1	a	1	-10	tentero \boxtimes
2	b	1	-14	trecl \boxtimes
3	i	1	0	trecl \boxtimes

TDB		
0	0	3
1	4	6

TDS (punto 2)				
0	G	0	-	función $\Rightarrow ((tentero \times trecl) \rightarrow trecl)$
1	x	D	0	registro $\Rightarrow ((c1 \times tentero, x0) \times (c2, trecl, 2) \times (c3, tentero, 6))$
2	mem	0	8	tentero \boxtimes
3	main	0	-	función $\Rightarrow ((tentero) \rightarrow (tentero))$
4	i	1	0	tentero \boxtimes
5	A	1	22	tvector $\Rightarrow (10, tentero)$
6	K	1	22	trecl \boxtimes

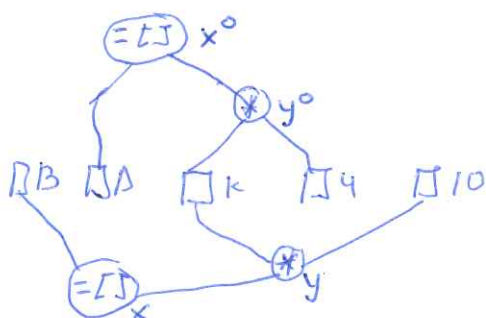
1.b)



Los Registros de Activación (RA) sirven para la adecuada gestión de memoria de las funciones. Contendrán el segmento de variables locales y temporales, el segmento de parámetros y el segmento de entres de control y el estado de la máquina antes de la llamada.

2.a) $S \rightarrow id[E] = E$ Si ! $[ObtenerTDS(id.num, id.t, id.pos) \wedge (id.t = tvector(id.ne, id.te) \wedge (E.t = tentero) \wedge [E.t = id.te])] \{ MemEmar();$
 $Emite(E.pos \leftarrow E.pos * talla(id.te));$
 $Emite(id.pos[E.pos] \leftarrow E?.pos)$

2.b)



de transformación que elimina el código inactivo hecho de detectarlo, mediante el GDA y eliminarlo. Para el ejemplo el grafo es:

Donde el nodo $(=E) x°$ es un nodo raíz, no es segmento de otros nodos y su lista de variables está vacía. Al eliminarlo, el nodo $* y°$ es similar.

2.c) Dado que las operaciones con registros son mucho más eficientes, el objetivo de la asignación de registros es: asignar el mayor número de variables (temporales) al menor número de registros. teniendo en cuenta que: un conjunto de variables puede compartir un único registro o condición que a lo sumo solo una de ellas esté activa en cada punto del programa.

La solución es construir un grafo de interferencias: que representa las restricciones de dos o más variables con un tiempo de vida simultáneo: los nodos representan las variables y los arcos la imposibilidad de que los nodos (variables) enlazados puedan asignarse al mismo registro.

3. (3,5 ptos.) Diseñad un ETDS que genere código intermedio para el siguiente fragmento de una gramática:

$$I \rightarrow \text{do } \{ I1 \} \text{ for } (id ; E ; I2)$$

do-for es una instrucción repetitiva. Lo primero que hace la instrucción es inicializar la variable **id** a 0. Después de ello, mientras la expresión **E** sea cierta, se ejecutará el código de **I1**, después el código de **I2** y de nuevo se volverá a evaluar la expresión **E**. El bucle finalizará cuando **E** sea falso.

4. (1 pto.) Dado el siguiente fragmento de código intermedio de un bloque básico, aplicad las optimizaciones locales a partir de su GDA. A la salida del bloque solo estarán activas las variables: **a**, **b**.

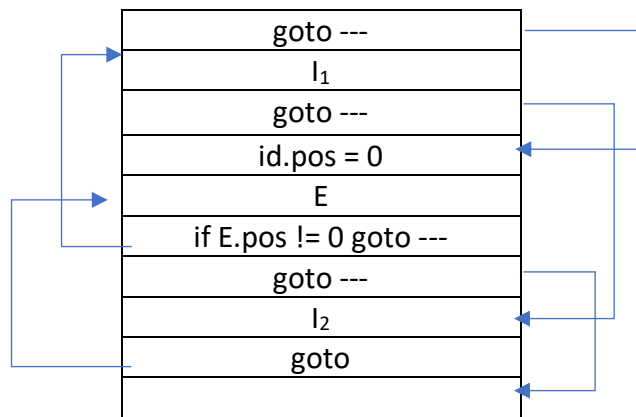
(100) $t_0 = N$	(104) $t_4 = 10$	(108) $a = b[t_3]$
(101) $t_1 = 2$	(105) $t_5 = t_1 * t_4$	(109) $b[t_7] = a$
(102) $t_2 = t_0 * t_1$	(106) $t_6 = N$	(110) if $a < t_5$ goto 150
(103) $t_3 = t_2 + 6$	(107) $t_7 = t_6 * 2$	

5. Dado el siguiente fragmento de código intermedio, y sabiendo que a la salida del bucle solo estará activa la variable: **v**:

(100) $a = 100$	(105) $f = a$
(101) $b = a * N$	(106) $v[e] = f$
(102) $c = b * 5$	(107) $b = b + 4$
(103) $d = c + 3$	(108) $h = N * 5$
(104) $e = d * 2$	(109) if $b < h$ goto 102

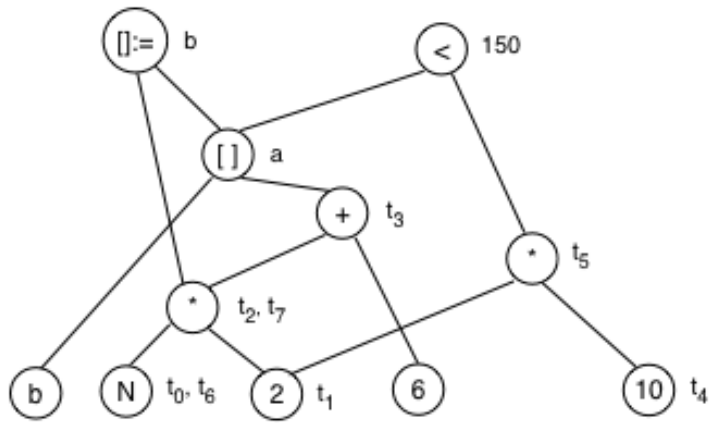
- a) (0,5 ptos.) Determinad los bloques básicos que forman el bucle. Extraed el código invariante e indicad las variables de inducción y sus ternas asociadas.
- b) (0,75 ptos.) Aplicad el algoritmo de reducción de intensidad
- c) (0,75 ptos.) Aplicad el algoritmo de eliminación de variables de inducción.

3.-

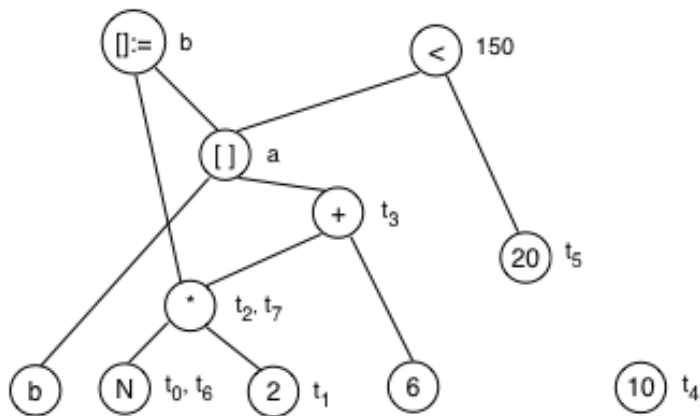


l -> do {	l.inicia = CreaLans(SIGINST); emite('goto' ---); l.cuerpo = SIGINST;
l₁ }	l.incr = CreaLans(SIGINST); emite('goto' ---); CompletaLans(l.inicia, SIGINST);
for (id ;	id.pos = BuscaPos(id.nom) ; emite(id.pos ':= 0') ; l.expr = SIGINST ;
E ;	emite('if' E.pos '!= 0 goto' l.cuerpo); l.fin = CreaLans(SIGINST) ; emite('goto' ---); CompletaLans(l.incr, SIGINST);
l₂)	emite('goto', l.expr); CompletaLans(l.fin, SIGINST) ;

4.-



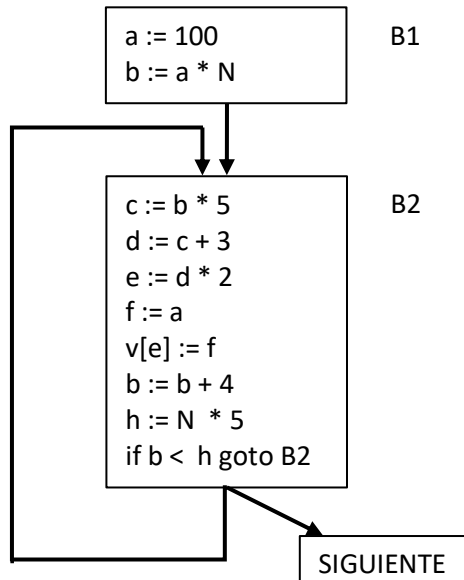
Tras el plegado de constantes:



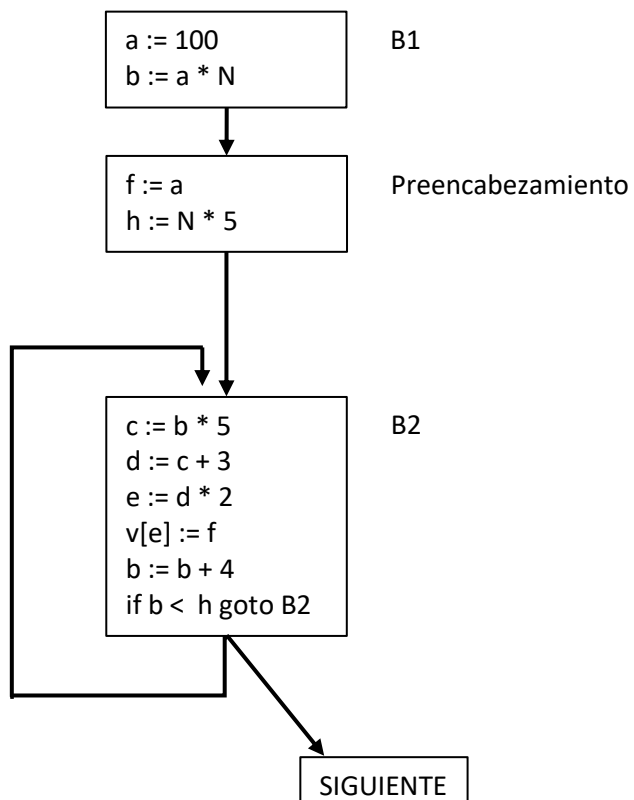
```

t7 := N * 2
t3 := t7 + 6
a := b[t3]
b[t7] := a
if a < 20 goto 150
  
```

5.-

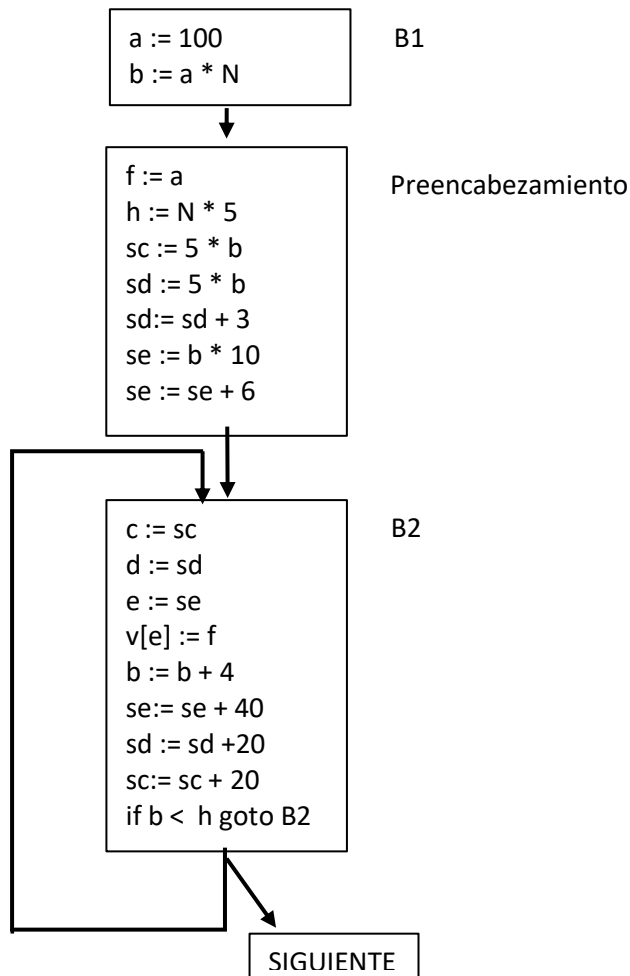


Extracción código invariante

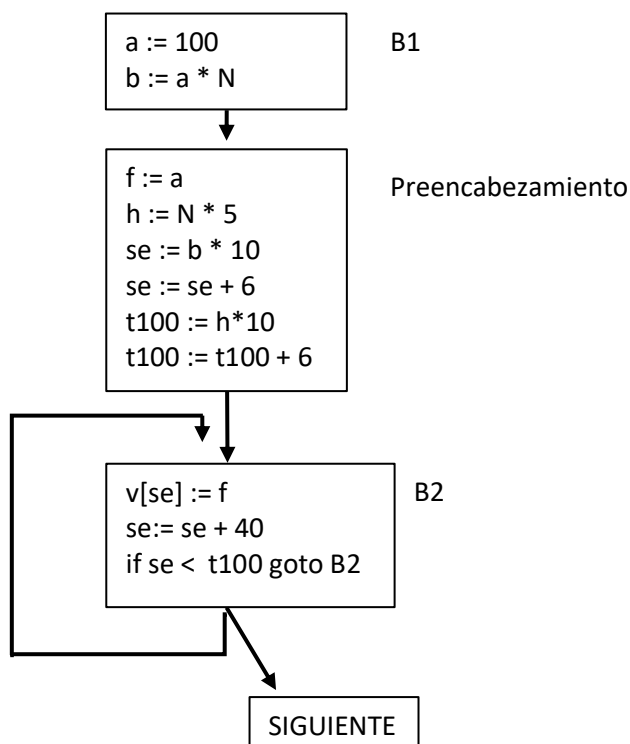


$b(b,1,0)$, $c(b,5,0)$, $d(b,5,3)$, $e(b,10,6)$

Reducción de intensidad



Eliminación variables de inducción:



Se pueden aplicar optimaciones locales al preencabezamiento construyendo el GDA