
Lenguajes de Programación y Procesadores de Lenguajes

(2º parcial)

14 de enero de 2015

1. Dado el siguiente programa C:

```
-----
float prod( float a, float b)
{ return a*b }

float sum(int a, int b)
{ float t
  if (a <= 5) {t= prod(a,b); return t;}      // <--- TDS
  else {                                         // <--- PILA
    return suma(a-1, b-1);}                  // <--- PC1
  }

int main ()
{ int a; int b;
  a = 7;                                         // <--- PC2
  b = 9;                                         // <--- TDS
  printf( "Resultado %f\n", sum(a, b)); }
-----
```

- a) (0,75 ptos.) Suponiendo que la talla de enteros es 2 y la de los reales es 4, mostrad el contenido completo de la TDS en el punto de control TDS.
- b) (0,5 ptos.) Indicad el desplazamiento relativo de las variables *a* y *b* que aparecen en las instrucciones de los puntos de control: PC1 y PC2.
- c) (0,75 ptos.) Mostrad el contenido (en términos de Registros de Activación) del estado de la pila de ejecución cada vez que se pasa por las instrucciones del punto de control PILA.
2. (1,5 ptos.) Contestad brevemente a las siguientes cuestiones:
- a) ¿Cómo se realiza el proceso de asignación de registros mediante coloreado de grafos?
- b) Describid las acciones necesarias para cargar un registro de activación en un lenguaje de alto nivel basado en bloques.
- c) Indicad dos formas de realizar la selección de instrucciones en la generación de código y describelas brevemente.
3. (3.5 ptos.) Diseñad un ETDS que genere código intermedio para el siguiente fragmento de una gramática:

$$S \rightarrow \text{repeat-if } E \text{ then } S \text{ else } S \text{ until } E$$

repeat-if es una instrucción repetitiva en la que, dependiendo del valor de la expresión E^1 , se ejecutará S^1 en caso de que sea TRUE y S^2 en caso de FALSE. Este proceso se repetirá hasta que la expresión E^2 sea TRUE.

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**, **x**, **y**, **z**.

(100) $t_1 := 10$	(106) $t_5 := c + x$	(112) $z := x + y$
(101) $t_2 := t_1 * 2$	(107) $x := t_5$	(113) $t_8 := 1$
(102) $c := t_2$	(108) $t_6 := d * y$	(114) $a := z + t_9$
(103) $t_3 := 5$	(109) $b := c + x$	(115) $t_9 := 1$
(104) $t_4 := t_3 + 2$	(110) $t_7 := d * y$	(116) $z := x * y$
(105) $d := t_4$	(111) $y := t_7$	(117) $a := b + 1$

5. Dado el siguiente fragmento de código intermedio:

(100) $k := 0$	(105) $t_4 := 5$	(110) $t_9 := t_3 + t_8$
(101) $m := 100$	(106) $t_5 := t_4 * 2$	(111) $A[t_2] := t_9$
(102) $t_1 := k * 2$	(107) $t_6 := t_1 + t_5$	(112) <i>if</i> $k > m$ <i>goto</i> 115
(103) $t_2 := t_1 * 2$	(108) $t_7 := t_6 * 2$	(113) $k := k + 1$
(104) $t_3 := A[t_2]$	(109) $t_8 := A[t_7]$	(114) <i>goto</i> 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.