Número de Orden: 6
Nombre y Apellido: Franco Muñiz

Libreta Universitaria: 330/20
Cantidad de Hojas Entregadas: 5

Ingeniería del Software II

Recuperatorio #1 – Análisis Estático de Programas

Ej1	Ej2	Ej3	Ej4	Nota
B	B	B	B	A

10

El exámen es a **libro abierto**. Agregar nombre y apellido y LU a cada hoja. Entregar los ejercicios en hojas **simple faz**. Cada ejercicio se evaluará como **Bien, Regular o Mal**. Para aprobar el examen es necesario tener al menos 2 ejercicios Bien y al menos 1 ejercicio Regular: Bien = **2,5p**, Regular = **1,25p**, Mal = **0p**.

Ejercicio 1

Se quiere hacer un análisis dataflow que pueda predecir el estado de una variable de tipo Contador que representa un contador de elementos.

Se extiende el lenguaje TIP para incluir la expresiones `crearContador()` que instancia un contador en cero, `incrementar(`*c*`)` que retorna el contador pero inrementado en 1 y `decrementar(c)` que deuvelve el mismo contador decrementado en 1 (requiere que c no este en cero). Notar que `incrementar(`*c*`)` y `decrementrar(`*c*`)` requieren que el contador este instanciado.

a. Se desea determinar si el contador ha sido instanciado, está en cero, vale exactamente uno o un valor estrictamente mayor que uno. Definir un reticulado L que modele *los potenciales estados del* contador. Definir las operaciones $\sqcup$ y $\sqsubseteq$.

b. Definir la semántica abstracta de las siguientes operaciones que modele de la forma *más precisa* posible, pero sobreaproximada de acuerdo al reticulado definido, el estado del contador luego de aplicar la operación. Definir las operaciones para cada elemento del reticulado (pensar que estas funciones se usarían en un may analysis).

c	`incrementar(c)`	`decrementar(c)`
...	...	... ...

Ejercicio 2

Para el siguiente programa:

```
1   g(k,h) {
2     return k+h;
3   }
4   main() {
5     var  x,y;
6     x = 0;
7     y = g(x,-x);
8     x = g(y,1)
9     output x;
10  }
```

a. Calcular el CFG de `f` y `main`.

b. Calcular el análisis interprocedural usando el dominio del **signo** sin contextos.

c. ¿Qué valor da in[9](x)? O sea ¿cuánto dió el análisis para x en la linea 9?

d. Calcular el análisis interprodecural pero usando la técnica de cloning y luego indicar in[9](x).

e. Recalcular el dataflow pero usando contexto funcional y luego indicar in[9](x).

Ejercicio 3

Dado el siguiente programa en Java:

```
class A { A f; };

int m(int x) {
  A a = new A();    // A1
  A b = new A();    // A2
  A e= a;
  if(i>0)  {
     e = b;
  }
  a = new A();    // A3
  e.f = a;
  A d = a.f;
}
```

a. Calcular las restricciones que genera el análisis de Andersen para este programa.

b. Encontrar la solución más exacta al sistema de restricciones.

c. Dibujar el points-to graph resultante.

Ejercicio 4

Dado el siguiente programa.

```
void main (int n, int m) {
 int i = m ;
 while(i>=n) {
     i= i-1;
 }
}
Pos: { i > 0 }
```

a. ¿Cuál es la weakest precondition de este programa?

b. ¿Cuál es el programa que surge de hacer un loop unroll de k=1?

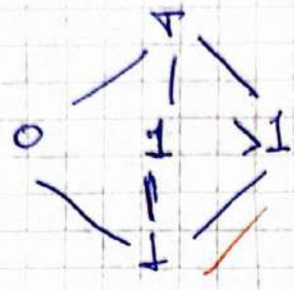
c. ¿Cuál es la weakest precondition del programa unrolleado del punto anterior?

Nota: No es necesario escribir el desarrollo del cálculo de la WP en los puntos anteriores.

Franco Muñiz 330/20 ord 6

hoja 1

Ej 1)

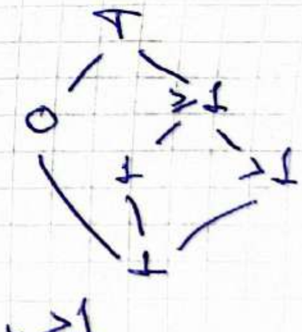


⊔	⊤	0	1	>1	⊥
⊤	⊤	⊤	⊤	⊤	⊤
0	⊤	0	⊤	⊤	0
1	⊤	⊤	1	⊤	1
>1	⊤	⊤	⊤	>1	>1
⊥	⊤	0	1	>1	⊥

Cada elem está incluido en el nivel superior. ok

c	inc(c)	dec(c)
⊤	⊤	⊤
0	1	⊥
1	>1	0
>1	>1	⊤
⊥	⊥	⊥

También se podría definir como: para poder definir mejor casos como dec(>1).

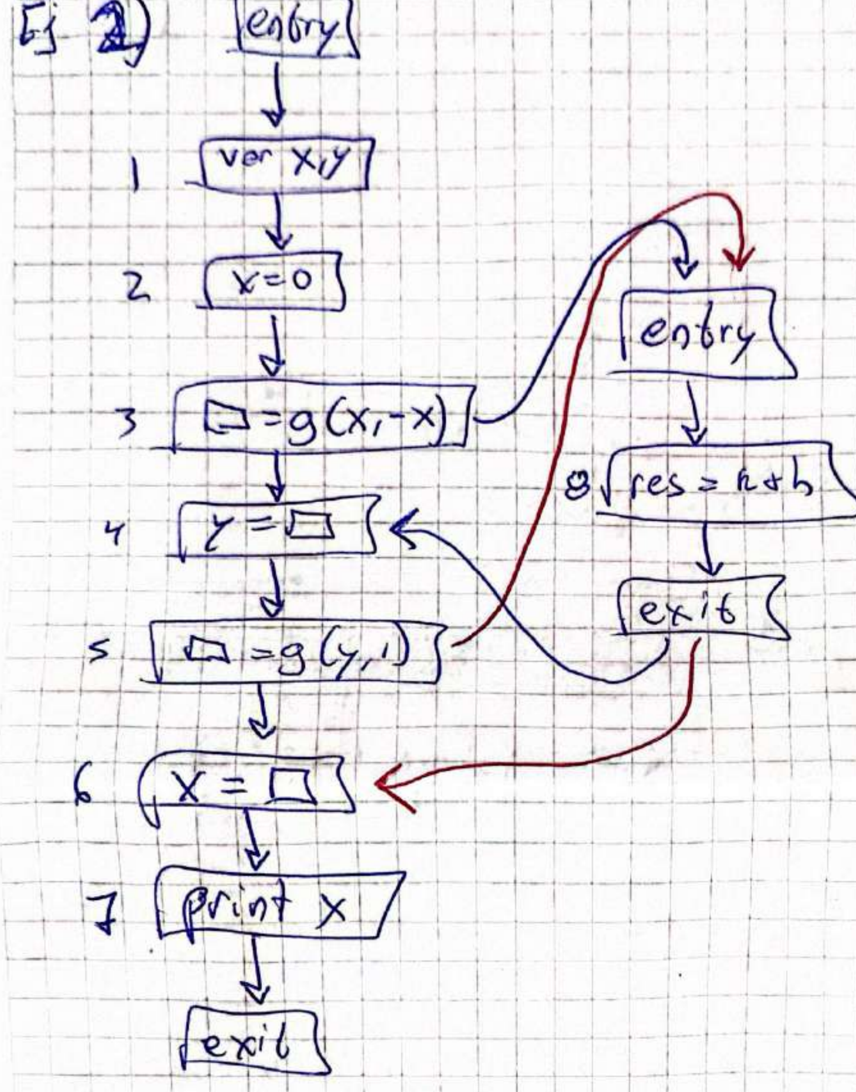


Y ≥1 sería el supremo de 1 y >1.

Franco Muñiz 330/20 ord 6

hoja 2

[j] 2)



b)

nodo	in	out
1	$[x \to \bot, y \to \bot]$	$[x \to \top, y \to \top]$
2	$[x \to \top, y \to \top]$	$[x \to 0, y \to \top]$
3	$[x \to 0, y \to \top]$	$[x \to 0, y \to \top]$
8(1)	$\begin{bmatrix} n \to 0 \\ h \to 0 \end{bmatrix} \sqcup \begin{bmatrix} n \to \bot \\ h \to \bot \end{bmatrix}$	$\begin{bmatrix} n \to 0 & res \to 0 \\ h \to 0 & \end{bmatrix}$
4(1)	$[x \to 0, y \to \top]$	$[x \to 0, y \to 0]$
6(1)	$[x \to \bot, y \to \bot]$	$[x \to 0, y \to \bot]$
5	$[x \to 0, y \to 0]$	$[x \to 0, y \to 0]$
8(2)	$\begin{bmatrix} n \to 0 \\ h \to 0 \end{bmatrix} \sqcup \begin{bmatrix} n \to 0 \\ h \to \top \end{bmatrix}$	$\begin{bmatrix} n \to 0 & res \to \top \\ h \to \top & \end{bmatrix}$
4(2)	$[x \to 0, y \to 0]$	$[x \to 0, y \to \top]$
6(2)	$[x \to 0, y \to \top]$	$[x \to \top, y \to \top]$
7(1)	$[x \to 0, y \to \bot]$	$[x \to 0, y \to \bot]$
7(2)	$[x \to \top, y \to \top]$	$[x \to \top, y \to \top]$

$in[7](x) = [x \to \top, y \to \top]$

Franco Muñiz 330/20 ord 6

hoja 3

(d) Contexto funcional

nodo	in	out
1	$[x \to \bot, y \to \bot]$	$[x \to \top, y \to \top]$
2	$[x \to \top, y \to \top]$	$[x \to 0, y \to \top]$
3	$[x \to 0, y \to \top]$	$[x \to 0, y \to \top]$
8(1)	$\begin{bmatrix} n \to 0 \\ h \to 0 \end{bmatrix} \xrightarrow{*1} \begin{bmatrix} n \to 0 \\ h \to 0 \end{bmatrix}$	$\begin{bmatrix} n \to 0 \\ h \to 0 \end{bmatrix} \xrightarrow{*2} \begin{bmatrix} n \to 0 \\ h \to 0 \\ res \to 0 \end{bmatrix}$
4	$[x \to 0, y \to \top]$	$[x \to 0, y \to 0]$
5	$[x \to 0, y \to 0]$	$[x \to 0, y \to 0]$
8(2)	$\left\{ *1, \underbrace{\begin{bmatrix} n \to 0 \\ h \to + \end{bmatrix} \to \begin{bmatrix} n\,0 \\ h\,+ \end{bmatrix}}_{*3} \right.$	$*2, *3 \to \begin{bmatrix} n \to 0 \\ h \to + \end{bmatrix} res \to +$
6	$[x \to 0, y \to 0]$	$[x \to +, y \to 0]$
7	$[x \to +, y \to 0]$	$[x \to +, y \to 0]$

$in[9](x) = [x \to +, y \to 0]$

(c) Cloning

nodo	in	out
1	$[x \to \bot, y \to \bot]$	$[x \to \top, y \to \top]$
2	$[x \to \top, y \to \top]$	$[x \to 0, y \to \top]$
3	$[x \to 0, y \to \top]$	$[x \to 0, y \to \top]$
8	$\begin{bmatrix} n \to 0 \\ h \to 0 \end{bmatrix}$	$\begin{bmatrix} n \to 0 \\ h \to 0 \end{bmatrix} res \to 0$
4	$[x \to 0, y \to \top]$	$[x \to 0, y \to 0]$
5	$[x \to 0, y \to 0]$	$[x \to 0, y \to 0]$
9 (2da copia f)	$[n \to 0, h \to +]$	$[n \to 0, h \to +, res \to +]$
6	$[x \to 0, y \to 0]$	$[x \to +, y \to 0]$
7	$[x \to +, y \to 0]$	$[x \to +, y \to 0]$

9 = nodo 7

$in[9](x) = [x \to +, y \to 0]$

Franco Muñiz 330/20 ord 6

hoja 4

Ej 3)

a) $\{A1\} \subseteq L(a)$

$\{A2\} \subseteq L(b)$

$L(a) \subseteq L(e)$

$L(b) \subseteq L(e)$

$\{A3\} \subseteq L(a)$

$L(a) \subseteq \cap \{E(k,f) \mid k \in L(e)\}$

$\cup \{E(k,f) \mid k \in L(a)\} \subseteq L(d)$

b) $L(a) = \{A1, A3\}$

$L(b) = \{A2\}$

$L(e) = \{A1, A2, A3\}$

$L(d) = \{A1, A3\}$

$E(A1, f, A1)$

$E(A1, f, A3)$

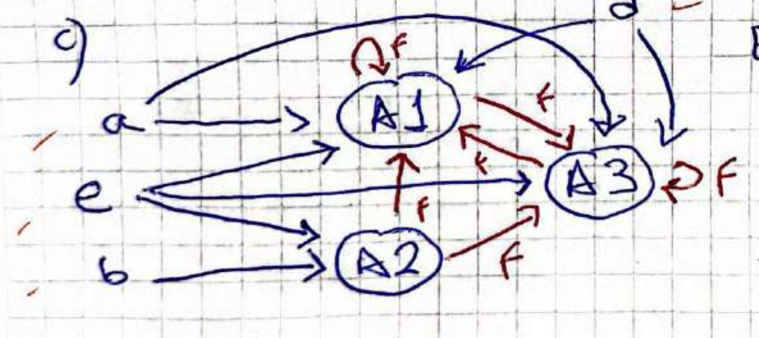
$E(A2, f, A1)$

$E(A2, f, A3)$

$E(A3, f, A3)$

$E(A3, f, A1)$

c)



Andersen repite hasta estabilizar.

En la segunda pasada, e ~~...~~ pasa a apuntar también a A3 (porque $A3 \subseteq L(a)$), y luego e.f = A suma ejes de A3 a A3 y de A3 a A1.

Franco Murio 330/20 ord6

hoja 5

Ej 4) (a) La wp es $\{m \geq 1 \wedge n > 1\}$. ✓

~~es necesario para que~~

Sabemos $m \leq 0$ falla trivialmente, ya que el while solo decrementa. El i máximo está dado por m.

Luego $m \geq 1$, pero no alcanza.

Si $m = 1$, aún falla con $n \leq 1$, ya que el (último) ciclo sería $1 \geq 1 = true$, luego i pasa a ser 0.
(en el mejor caso $n = 1$).
(si es más bajo, i sigue bajando)

Luego necesitamos $n > 1$ tal que la última comparación verdadera sea $2 \geq 2$, sin importar el m (i) inicial.

(b)

```
void main (int n, int m){
    int i = m;
    ~~while(i≥n)~~
    if (i ≥ n){
        i = i - 1;
        if (i ≥ n){
            assume false;
        }
    }
}
```

✓

Hay una más débil

$(n > 1 \wedge m > 0) \vee m \geq n+1$

(c)

En el programa unrolleado, i es o bien m, o bien $m - 1$.

Si $m > 1$, siempre se cumple la postcondición, por lo dicho arriba.

Si $m < 1$, nunca se cumple.

Si $m = 1$ se necesita no decrementar, luego $n > 1$ necesariamente.

La condición luego es $\{(m = 1 \wedge n > 1) \vee (m > 1)\}$