

## Ingeniería del Software II

### Práctica #1 - IP – Análisis Estático Interprocedural

#### Ejercicio 1

Para el siguiente programa:

```
1  int double(int x) {  
2    int result = 2*x;  
3    return result;  
4  }  
5  void main() {  
6    int x,y;  
7    x = 5;  
8    y = double(x);  
9    x = 0;  
10   y = double(x);  
11   print x, y  
12 }
```

1. Calcular el CFG de double y main.
2. Calcular el análisis interprocedural usando el dominio del signo sin contextos.
3. Qué valor calcula el análisis para el print?
4. Recalcular el dataflow luego de aplicar clonning a la función doble.
5. Recalcular el dataflow usando cadenas de llamadas con k=1.
6. Recalcular el dataflow usando functional context.

#### Ejercicio 2

En el programa anterior reemplazar double por:

```
1  int double(int x) {  
2    return double2(x);  
3  }  
4  int double2(int x) {  
5    int result = 2*x;  
6    return result;  
7  }
```

Cuál es el efecto en los items d) e) y f) del ejercicio anterior?

#### Ejercicio 3

Para el siguiente programa:

```
1  inc(a) {  
2    return a+1;  
3  }  
4  main() {  
5    y = input  
6    x = 0;  
7    while(y>0) {
```

```
8      inc(x);  
9      y--;  
10     }  
11     print x;  
12 }
```

1. Calcular el CFG de inc y main.
2. Calcular el análisis interprocedural usando el dominio del signo sin contextos.
3. Qué valor calcula el análisis para el print?
4. Recalcular el dataflow luego de aplicar clonning a la función inc.
5. Recalcular el dataflow usando cadenas de llamadas con k=1.
6. Recalcular el dataflow usando functional context.