

# Ingeniería del Software II

## Práctica #1 - PT – Análisis de Punteros

### Parte 1 – Análisis sensibles e insensibles a flujo

#### Ejercicio 1

Para el siguiente programa:

```
1 A a = new A(); // call site #1
2 B b = new B(); // call site #2
3 a.f = b;
4 c = a;
5 c.f = a;
6 c = b
```

1. Calcular el CFG.
2. Calcular el PTG al final del programa utilizando el análisis de flujo de datos presentado en clase.
3. Calcular el PTG utilizando el enfoque de Andersen.
4. Ejecutar de nuevo los puntos 2 y 3 sobre el mismo programa pero intercambiando las líneas 4 y 6. Que diferencias se ven en cada caso? Que afirmación confirma este pequeño experimento?
5. Calcular el PTG utilizando el enfoque de Steengard

#### Ejercicio 2

Dado el siguiente programa (asumiendo que B,C son subclases de A)

1	<b>class</b> A {	15	}
2	<b>void</b> main() {	16	<b>class</b> B <b>extends</b> A {
3	A a = <b>new</b> A();	17	m2(A k) {
4	B b = <b>new</b> B();	18	B l = <b>new</b> B();
5	A c = b.m1(a);	19	l.f = k;
6	A d = a.m2(b);	20	<b>return</b> l;
7	}	21	}
8	<b>void</b> m1(A p) {	22	}
9	A e = m2(p);	23	
10	}	24	<b>class</b> C <b>extends</b> A {
11		25	m2(A k) {
12	<b>void</b> m2(A p) {	26	k;
13	<b>return</b> p;	27	}
14	}	28	}

1. Construya el grado de llamadas mediante el análisis de jerarquía de clases.
2. Ahora usando Rapid Type Analysis.

#### Ejercicio 3

Dado el siguiente programa (suponiendo que la subclase B es de A):

```

1  void main() {
2  A a = new A();
3  B b = new B();
4  A c = m1(a);
5  A d = m2(b);
6  }
7  void m1(A p) {
8      A e = m2(p);
9  }
10
11 m2(A k) {
12     B l = new B();
13     l.f = k;
14     return l;
15 }

```

1. Calcular el PTG para el final del método m0:
2. Sin usar contexto para las allocaciones en memoria.
3. Usando una cadena de llamada de tamaño = 2

#### Ejercicio 4

Dado el siguiente programa :

```

1  void main() {
2  x = new h1;
3  z = new h2;
4  y = f(x);
5  w = f(z);
6  }
7  int f(object v) {
8      u = v;
9      return u;
10 }

```

1. Calcular el PTG para el final del método main usando Andersen sin contexto.
2. Calcular el PTG luego de hacer cloning.
3. Calcular PTG usando contexto de llamada

#### Ejercicio 5

Dado el siguiente programa :

```

1  void main() {
2  x = new h1;
3  z = new h2;
4  y = f(x);
5  w = f(z);
6
7  int f(object v) {
8      if (v == null)
9          v = f(v)
10     return v;
11 }
12 }

```

1. Calcular el PTG para el final del método main usando Andersen sin contexto.
2. Se puede hacer cloning?.
3. Calcular PTG usando contexto de llamada con k = 2
4. Calcular PTG usando contexto funcional