



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 4

Cobertura basada en grafos

IIC3745 – Testing

Rodrigo Saffie

rasaffie@uc.cl

26 de agosto de 2019

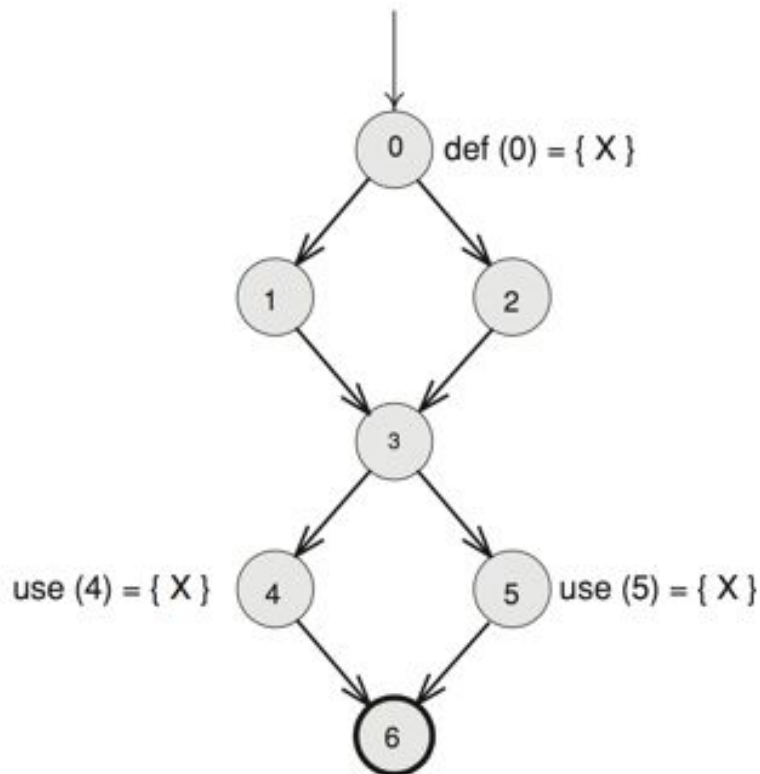
1. Clase pasada

- Actividad 1

2. Criterios de cobertura

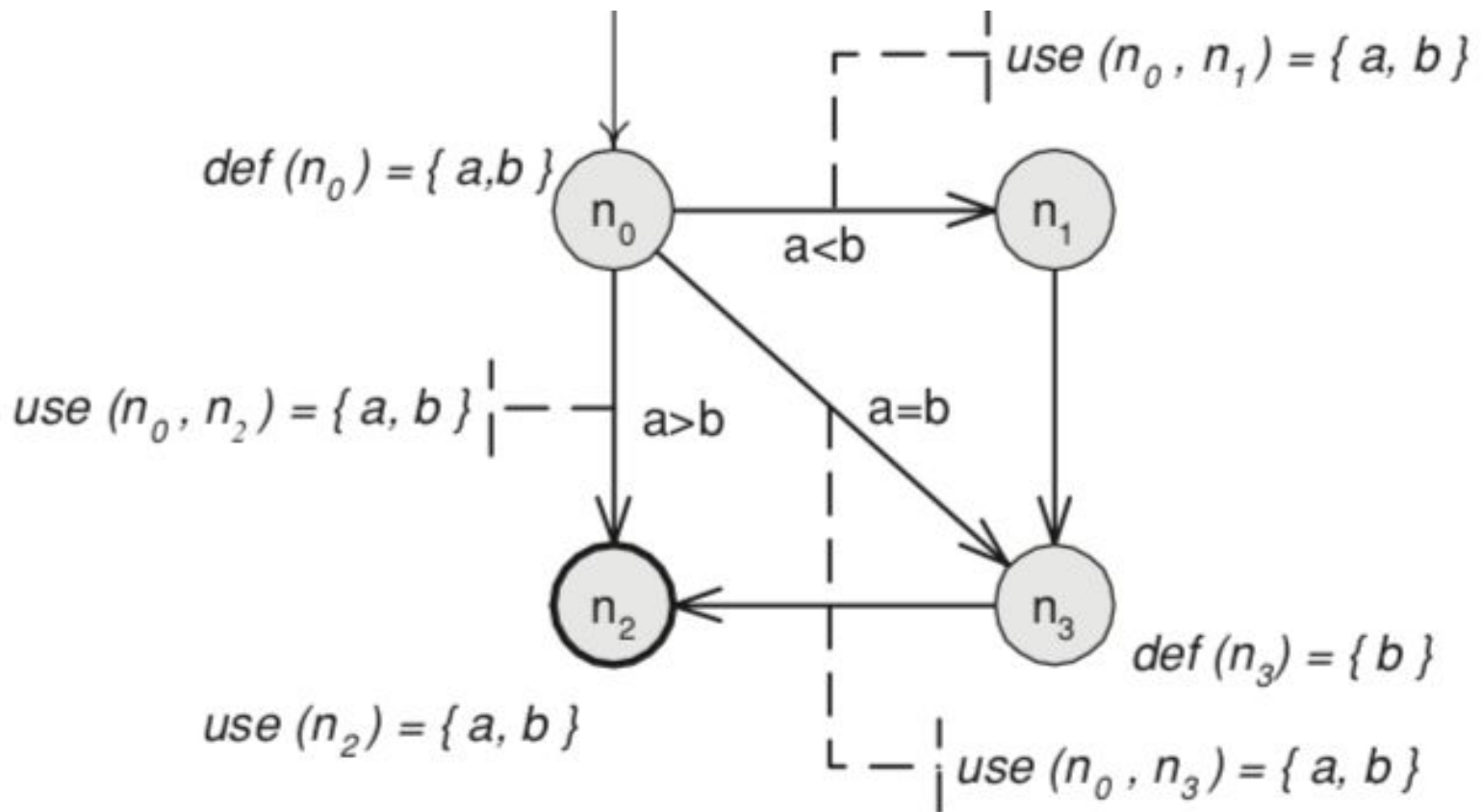
- Cobertura basada en grafos: flujo de información

Flujo de Información (*Data Flow*)



- Definición (**def**): lugar donde se asigna un valor a una variable en memoria
- Uso (**use**): lugar donde el valor de una variable es accedido
- **def(n), def(e)**: conjunto de variables que son definidas en el nodo **n** o arista **e**
- **use(n), use(e)**: conjunto de variables que son usadas en el nodo **n** o arista **e**

Flujo de Información (*Data Flow*)



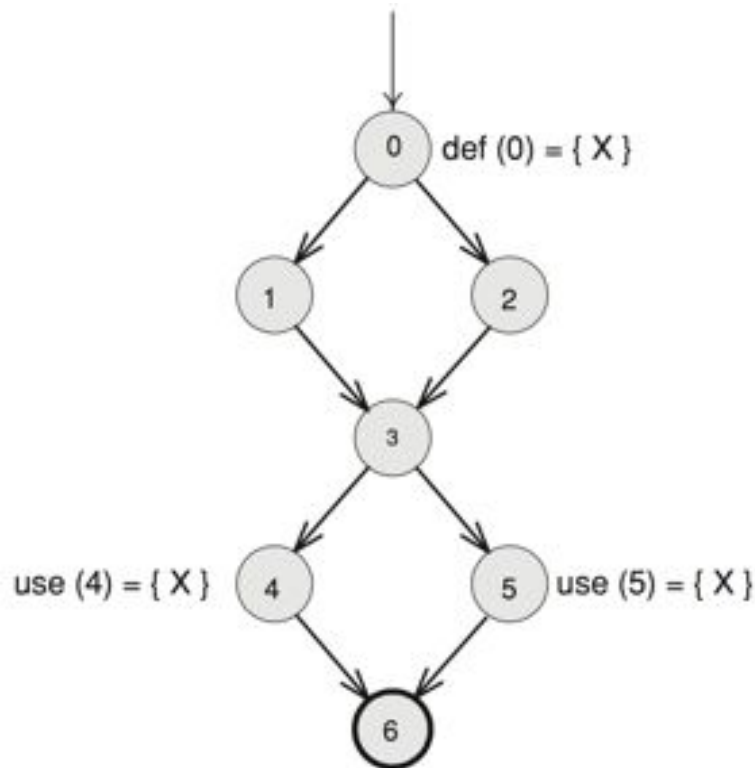
Pares DU y Caminos DU

- **Par-DU**: par de lugares (l_1, l_2) tal que una variable v es definida en l_1 y usada en l_2
- **def-clear**: se dice que un camino p de l_1 a l_2 es *def-clear* respecto a la variable v si no se asigna un valor a v en ningún nodo ni arista entre l_1 y l_2
- Alcance (**reach**): si existe un camino *def-clear* entre l_1 y l_2 con respecto a v se dice que la definición de v en l_1 alcanza a l_2
- **Camino-DU**: un subcamino simple *def-clear* respecto a v que comienza con una definición de v y termina con un uso de v

$du(n_i, n_j, v)$ = conjunto de caminos **du** con respecto a v desde n_i hasta n_j
 $du(n_i, v)$ = conjunto de caminos **du** con respecto a v que parten en n_i

Recorriendo caminos DU

- **recorre-du**: un camino de prueba p *recorre-du* el subcamino q con respecto a v si p recorre q y el subcamino q es *def-clear* respecto a v .



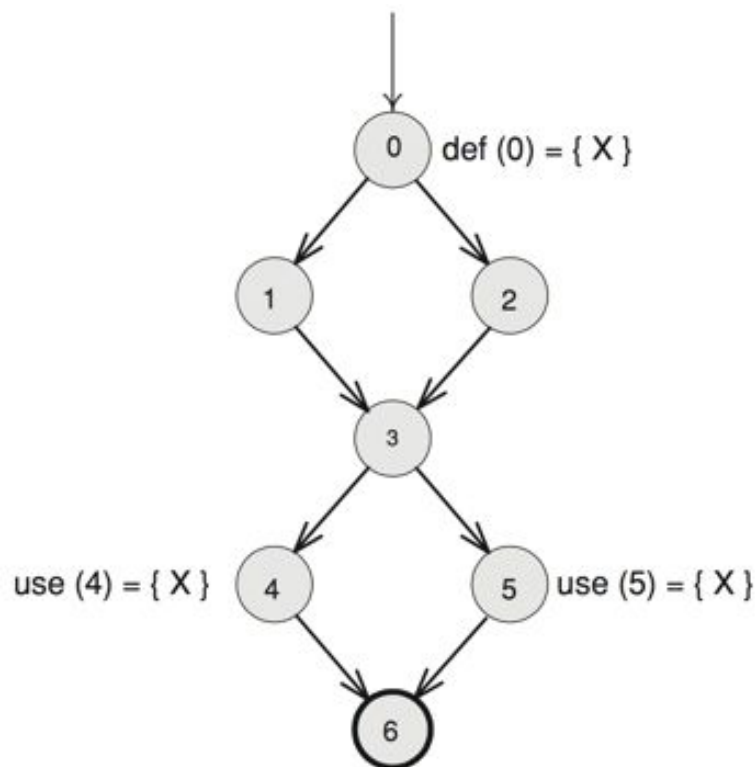
$$q = [n_0, n_1, n_3]$$

$$p = [n_0, n_1, n_3, n_4, n_6]$$

Criterios de cobertura

- **All-defs coverage (ADC)**: Para cada conjunto de *caminos-du* $S = du(n, v)$, TR contiene al menos un camino d en S .
 - Cada **def** debe alcanzar al menos un **use**
- **All-uses coverage (AUC)**: Para cada conjunto de *caminos-du* $S = du(n_i, n_j, v)$, TR contiene al menos un camino d en S .
 - Cada **def** debe alcanzar todos sus **use**
- **All-du-paths coverage (ADUPC)**: Para cada conjunto de *caminos-du* $S = du(n_i, n_j, v)$, TR contiene todos los caminos d en S .
 - Cada **def** debe alcanzar todos sus **use** a través de todos los *caminos-du*.

Criterios de cobertura: ejemplo



ADC:

- $p_1 = [n_0, n_1, n_3, n_4]$

AUC:

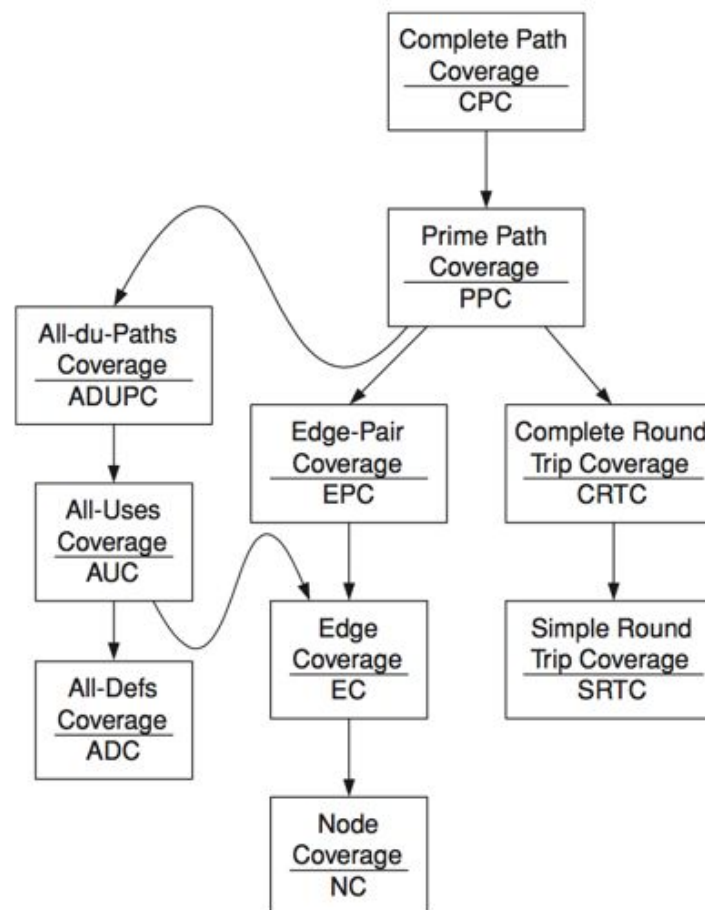
- $p_1 = [n_0, n_1, n_3, n_4]$
- $p_2 = [n_0, n_1, n_3, n_5]$

ADUPC:

- $p_1 = [n_0, n_1, n_3, n_4]$
- $p_2 = [n_0, n_1, n_3, n_5]$
- $p_3 = [n_0, n_2, n_3, n_4]$
- $p_4 = [n_0, n_2, n_3, n_5]$

Subsumir

“Incluir algo como componente en una clasificación más abarcadora”



Cobertura de grafos aplicada

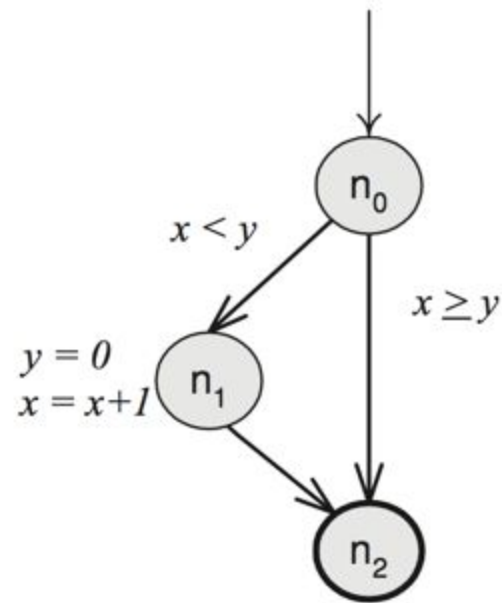
- Código fuente
- Elementos de diseño
- Especificación
- Casos de uso

Cobertura de código fuente

- Principal aplicación de los criterios de cobertura en grafos, que son representados como diagramas de control de flujo (**CFG**: *control flow graph*).
 - **Nodos**: Instrucciones o secuencias de instrucciones (bloques básicos)
 - **Aristas**: Opciones de flujo (ramificaciones)

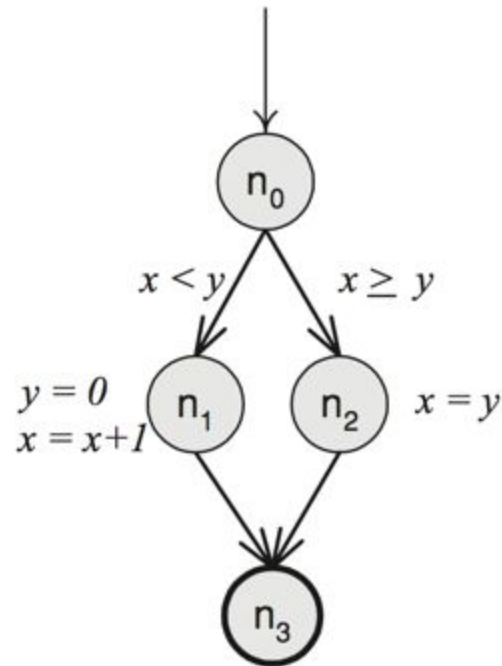
CFG: If

```
if ( x < y )  
{  
  y = 0;  
  x = x + 1;  
}
```



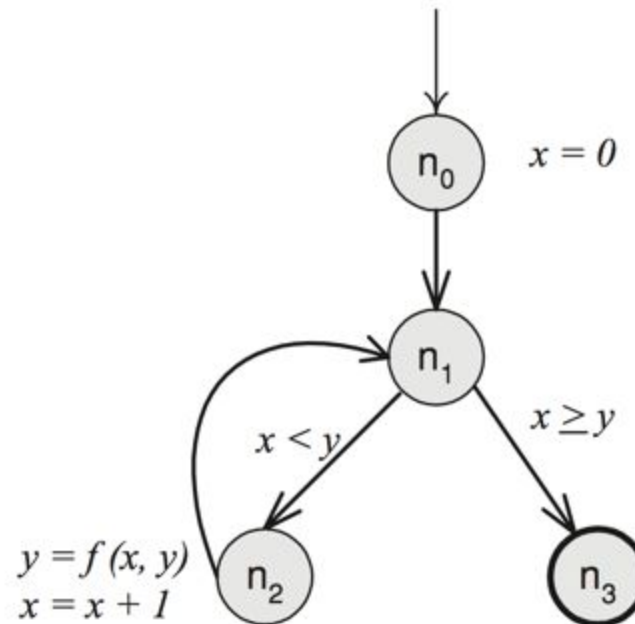
CFG: If-else

```
if (x < y)
{
  y = 0;
  x = x + 1;
}
else
{
  x = y;
}
```



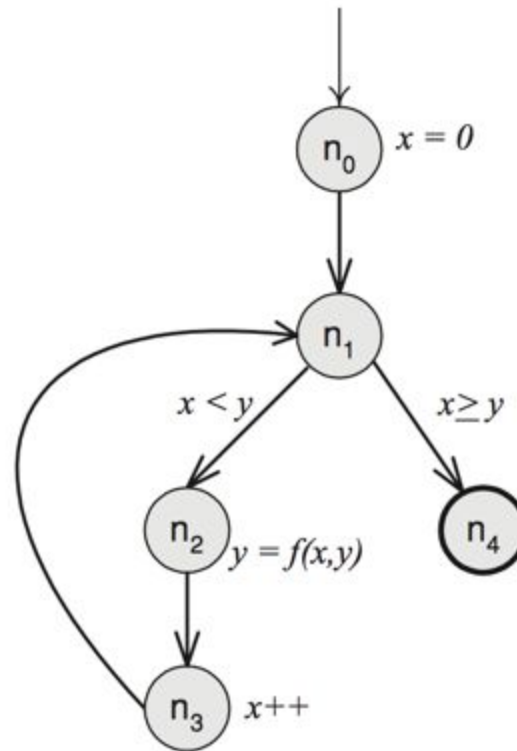
CFG: While

```
x = 0;  
while (x < y)  
{  
  y = f(x, y);  
  x = x + 1;  
}
```



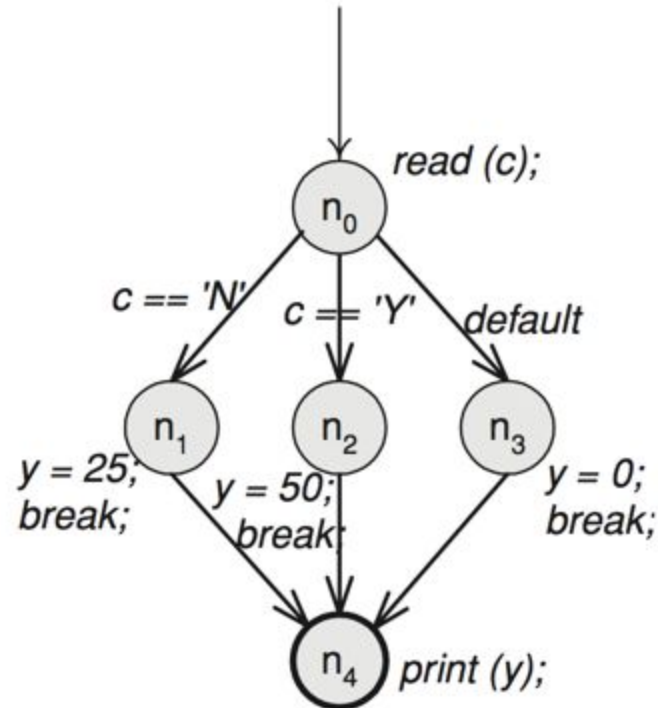
CFG: For

```
for (x = 0; x < y; x++)  
{  
  y = f(x,y);  
}
```



CFG: Flujo de información

```
read (c);  
switch (c)  
{  
  case 'N':  
    y = 25;  
    break;  
  case 'Y':  
    y = 50;  
    break;  
  default:  
    y = 0;  
    break;  
}  
print (y);
```



Cobertura de grafos aplicada

- Código fuente
- Elementos de diseño
- Especificación
- Casos de uso



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 4

Cobertura basada en grafos

IIC3745 – Testing

Rodrigo Saffie

rasaffie@uc.cl

26 de agosto de 2019