



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

24 de agosto de 2020

Encuestas

- Conocimientos generales
- Grupos proyectos semestral

1. Clase pasada

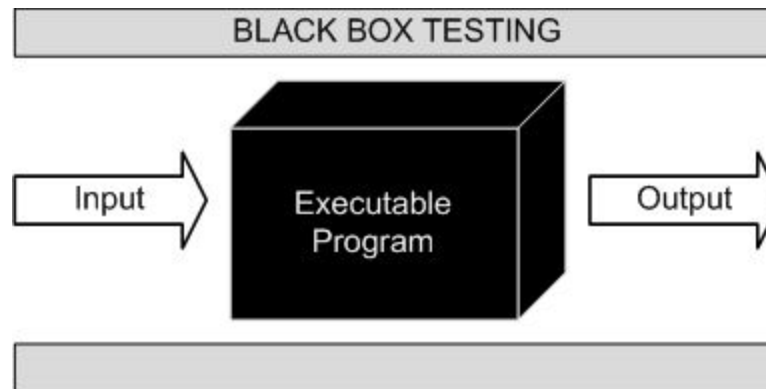
- Motivación
- Conceptos
- Tipos de *tests*
- Cobertura

2. Criterios de cobertura

- Cobertura basada en grafos

Black-box testing

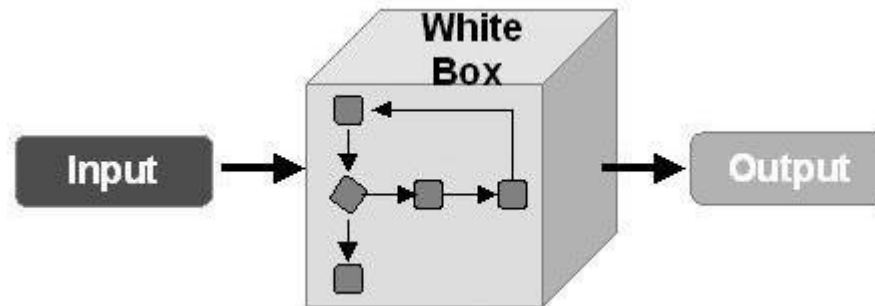
- No se conocen los detalles del *software*, solamente se considera *input* y *output*.



- Ejemplos
 - Pruebas de usuarios
 - Pruebas de seguridad

White-box testing

- Se conocen los detalles del software y se puede utilizar al diseñar los *tests*



Black-box vs White-box

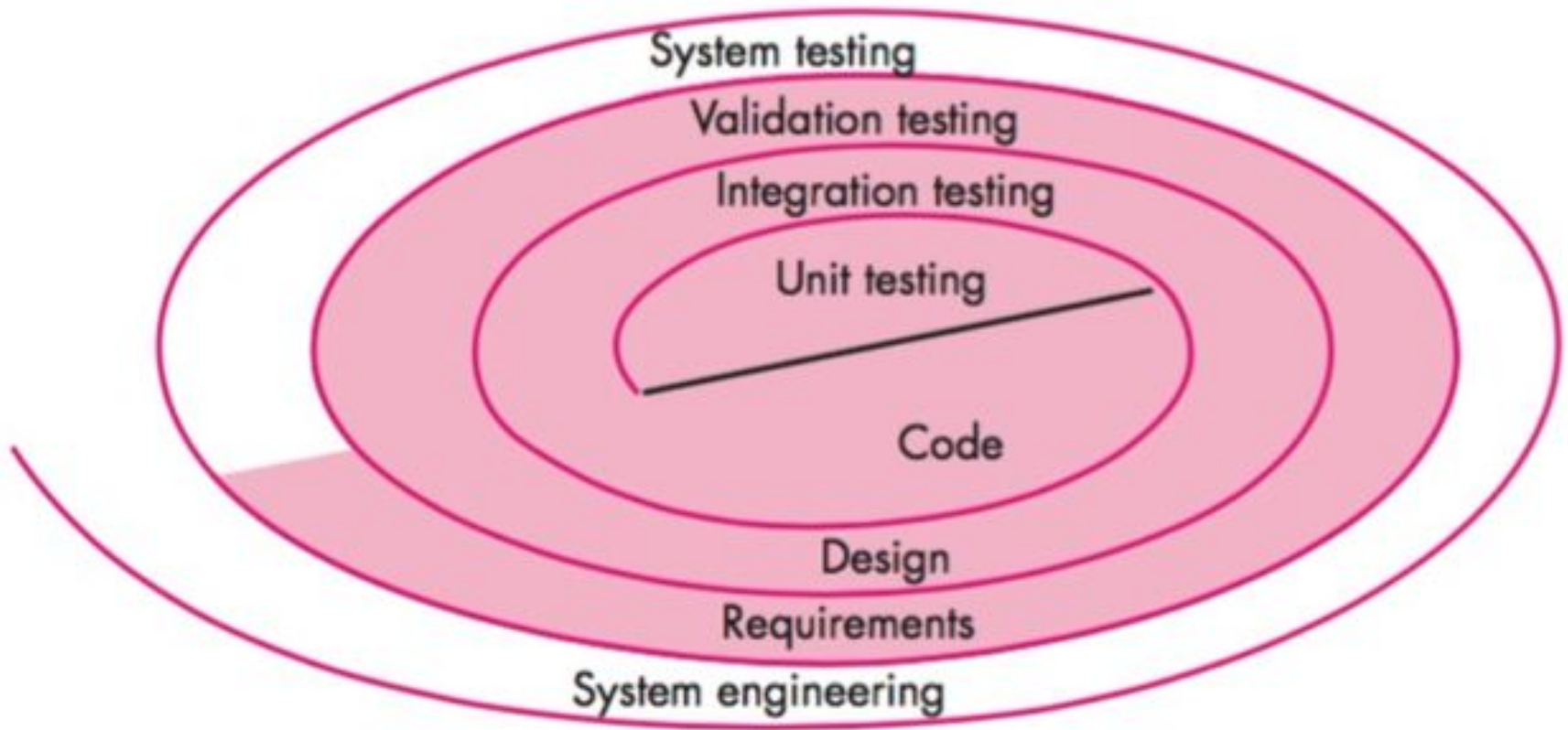
- *Black-box*

```
// Retorna la cantidad de ceros contenidos en un arreglo  
function countZeros(array) {
```

- *White-box*

```
// Retorna la cantidad de ceros contenidos en un arreglo  
function countZeros(array) {  
  var count = 0;  
  for (var i = 1; i < array.length; i++) {  
    if (array[i] === 0) {  
      count++;  
    }  
  }  
  return count;  
}
```

Niveles tradicionales de *Testing*



Tests de humo

- Subconjunto de pruebas enfocadas en garantizar las funcionalidades más importantes
- Se ejecutan de manera rápida y barata antes de cada implementación



Tests de mutación

- Se introducen pequeñas variaciones en el código con el objetivo de cuantificar cuántos *tests* fallan
- Evalúan la calidad de los *tests* existentes

```
# Original
if x > y:
    z = x - y
else:
    z = 2 * x
```

```
# Mutación 1
if x >= y:
    z = x - y
else:
    z = 2 * x
```

```
# Mutación 2
if x > y:
    z = x + y
else:
    z = 2 * x
```

```
# Mutación 3
if x > y:
    z = x - y
else:
    z = 2 * y
```

Tests de regresión

- *Tests* para garantizar que luego de modificar un software las funcionalidades originales siguen respetando las especificaciones
- Pueden ser un subconjunto de las pruebas o la batería completa
- Pueden ser útiles al momento de versionar código según el criterio [SemVer](#)

Cobertura (*coverage*)

- Una métrica que mide la proporción de código fuente que son probadas por una batería de *tests*.
- Existen distintos criterios de cobertura de pruebas.

Criterios de cobertura de pruebas

- Las pruebas son caras y consumen esfuerzo
- Los criterios de cobertura sirven para decidir qué entradas de prueba usar
- Los criterios hacen que las pruebas sean más eficientes y efectivas:
 - Es más probable encontrar problemas
 - Generan mayor confianza en la calidad del código
 - Se responde al por qué de cada prueba

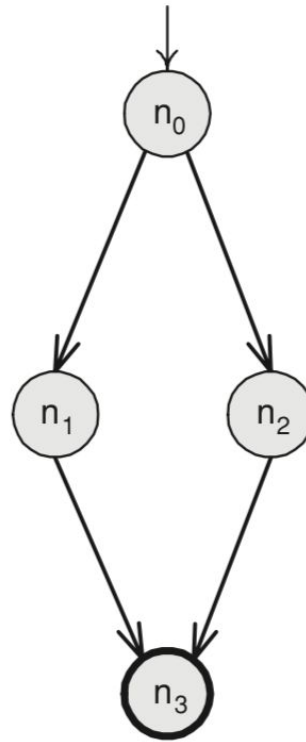
Criterios de cobertura de pruebas

- Basados en grafos
- Expresiones lógicas
- Dominio de parámetros de entrada
- Estructuras sintácticas

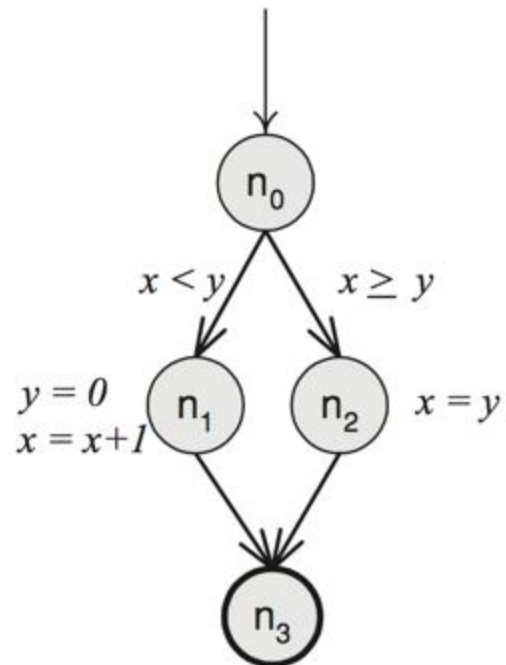
Grafo

- N : Set de nodos
- N_0 : Set de nodos iniciales, subconjunto de N
- N_f : Set de nodos finales, subconjunto de N
- E : Set de aristas, subconjunto de $N \times N$

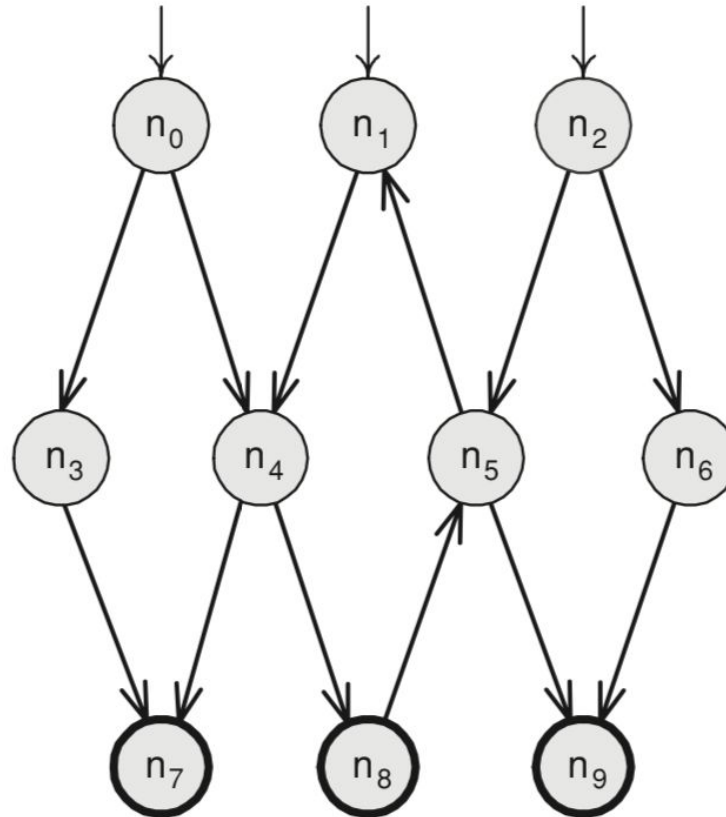
Grafo: Ejemplos


$$N = \{ n_0, n_1, n_2, n_3 \}$$
$$N_0 = \{ n_0 \}$$
$$E = \{ (n_0, n_1), (n_0, n_2), (n_1, n_3), (n_2, n_3) \}$$

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



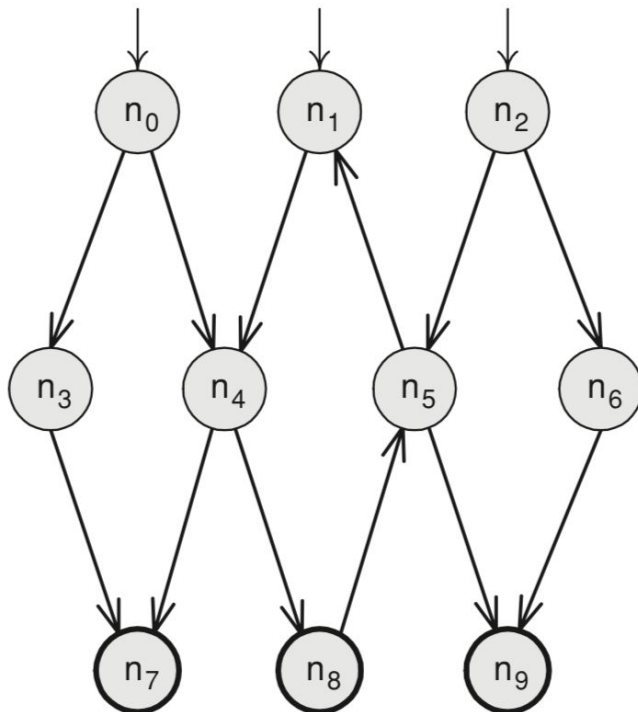
Grafo: Ejemplos


$$N = \{ n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9 \}$$
$$N_0 = \{ n_0, n_1, n_2 \}$$
$$|E| = 12$$

Grafo: Definiciones

Camino:

- secuencia de nodos donde cada par de nodos adyacentes (n_i, n_{i+1}) está contenido en el set **E** de aristas.



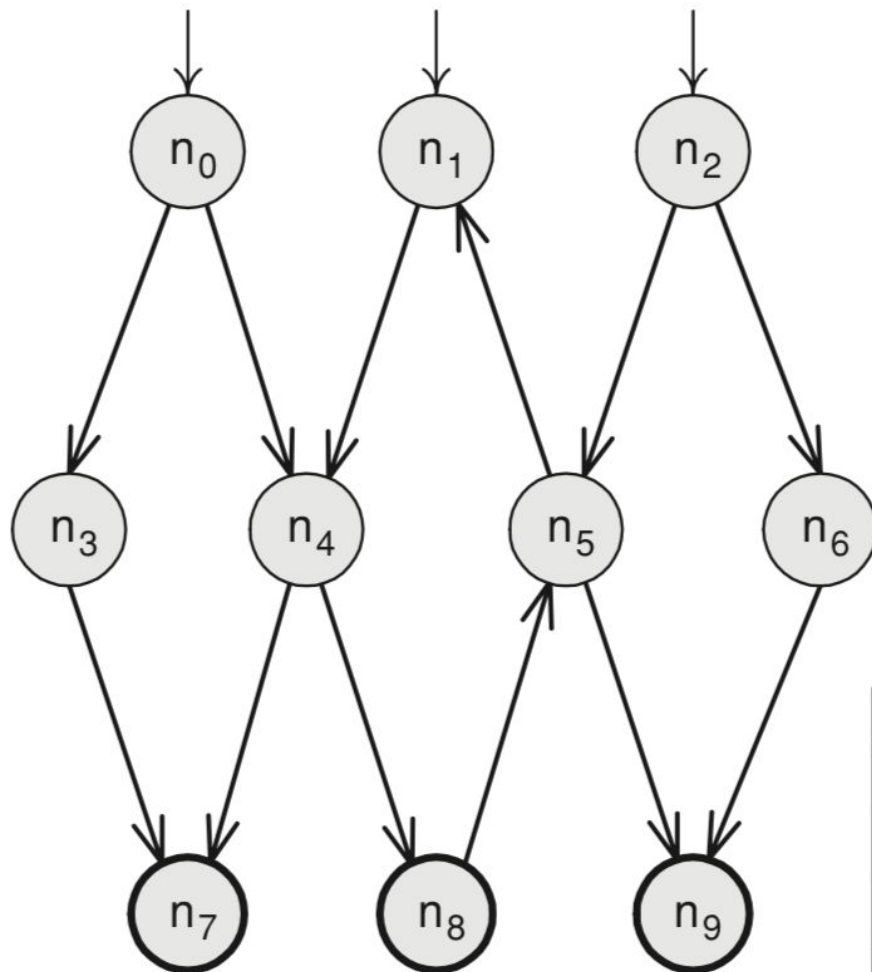
Path Examples	
1	n_0, n_3, n_7
2	n_1, n_4, n_8, n_5, n_1
3	n_2, n_6, n_9

Invalid Path Examples	
1	n_0, n_7
2	n_3, n_4
3	n_2, n_6, n_8

Grafo: Definiciones

- *Camino*: secuencia de nodos donde cada par de nodos adyacentes (n_i, n_{i+1}) está contenido en el set E de aristas.
- *Sub-camino*: subsecuencia de nodos de un camino p .
 $[n_0, n_3]$ es subcamino de $[n_0, n_3, n_7]$
- *Largo de camino*: cantidad de aristas contenidas en el camino.

Alcance Sintáctico y Semántico



¿Es n_2 alcanzable desde n_0 ?

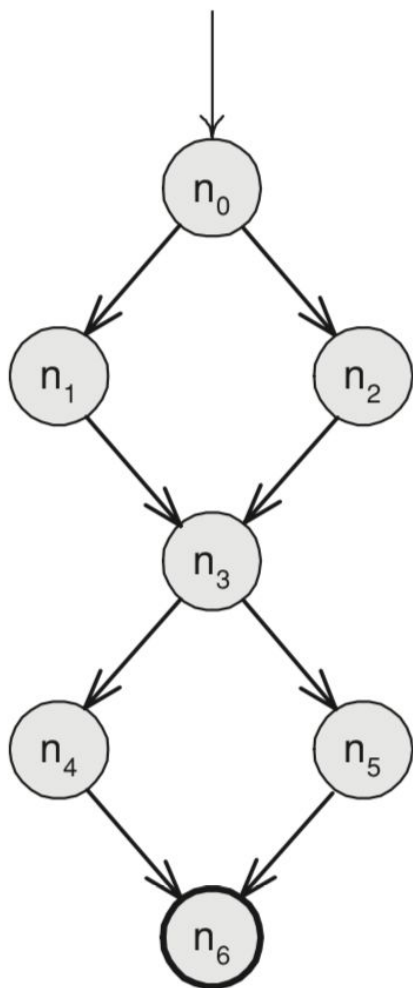
¿Es n_7 alcanzable desde n_1 ?

Sintáctico: Depende de la estructura del grafo.

Semántico: Depende la semántica del software.

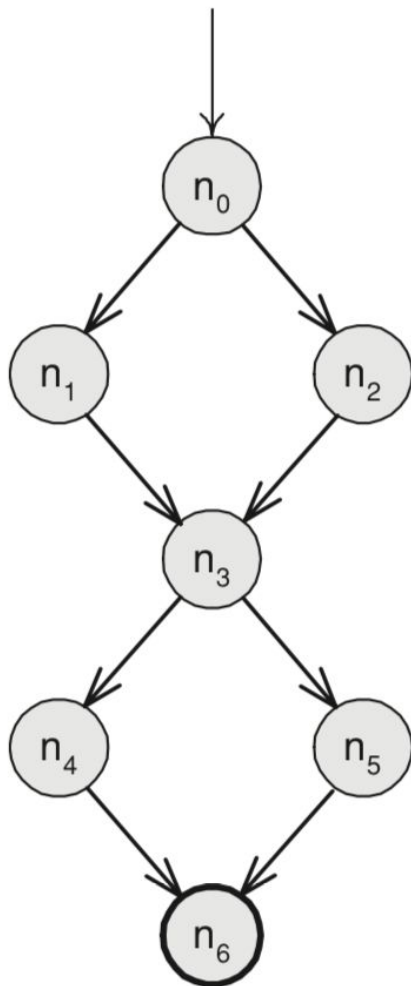
Reachability Examples	
1	$reach(n_0) = N - \{n_2, n_6\}$
2	$reach(n_0, n_1, n_2) = N$
3	$reach(n_4) = \{n_1, n_4, n_5, n_7, n_8, n_9\}$
4	$reach([n_6, n_9]) = \{n_9\}$

Grafo Single Entry Single Exit (SESE)



- $|\mathbf{N}_0| = 1$
- $|\mathbf{N}_f| = 1$
- $\text{reach}(\mathbf{n}_0) = \mathbf{G}$
- $\text{reach}(\mathbf{n}_f) = [\mathbf{n}_f]$

Camino de prueba (test path)



- Es un camino p que comienza en algún nodo de N_0 y termina en algún nodo de N_f .
- Corresponde a la ejecución de uno o varios casos de pruebas.

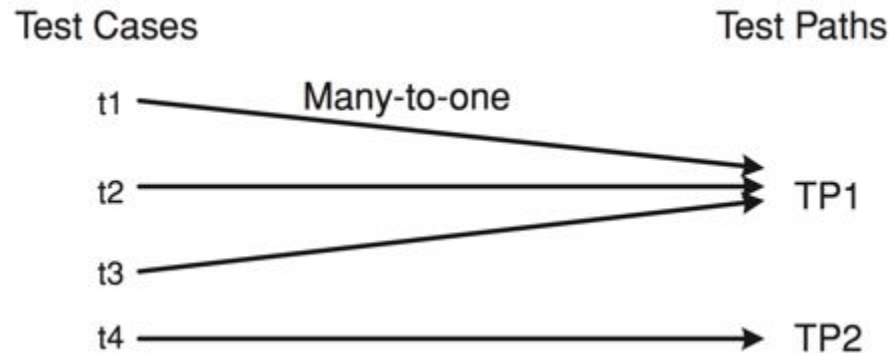
$$t_1 = [n_0, n_1, n_3, n_4, n_6]$$

$$t_2 = [n_0, n_1, n_3, n_5, n_6]$$

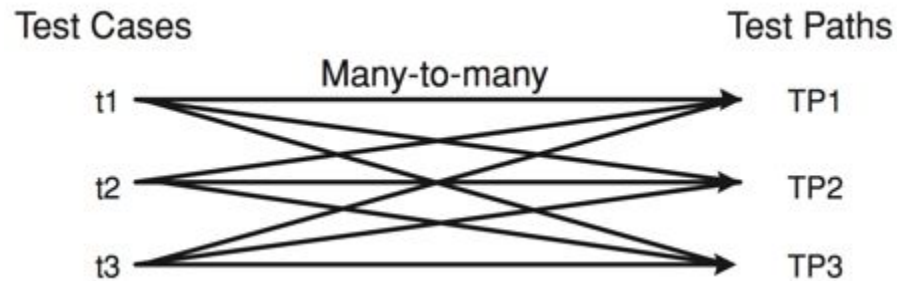
$$t_3 = [n_0, n_2, n_3, n_4, n_6]$$

$$t_4 = [n_0, n_2, n_3, n_5, n_6]$$

Casos y Caminos de prueba

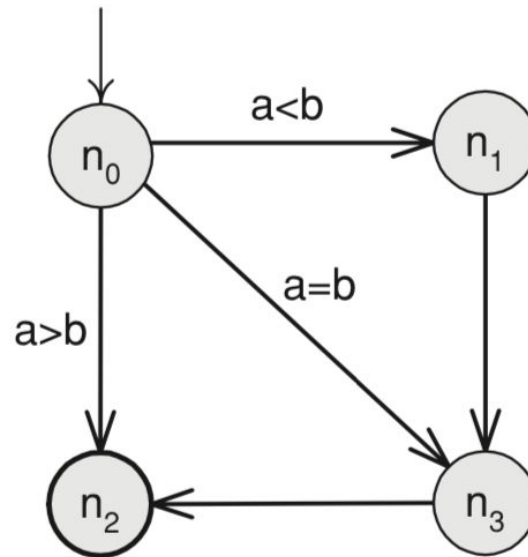


Software Determinístico



Software No Determinístico

Casos y Caminos de prueba

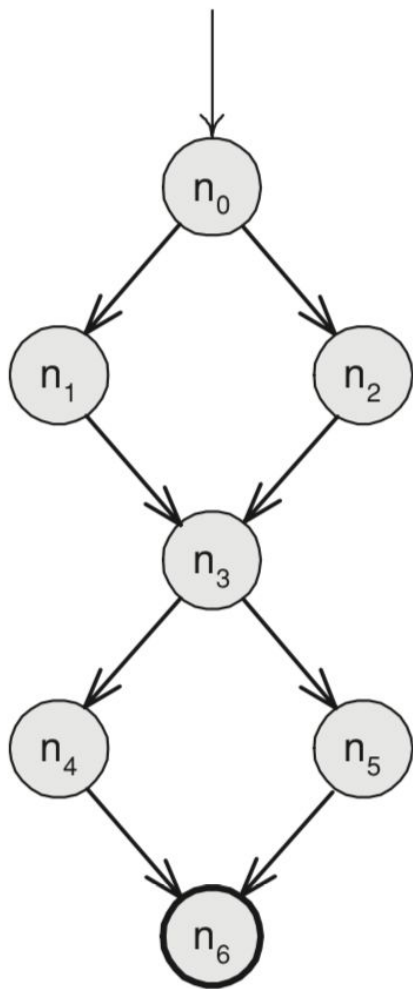


Test case $t_1 : (a=0, b=1)$ $\xrightarrow{\text{Map to}}$ [Test path $p_1 : n_0, n_1, n_3, n_2$]

Test case $t_2 : (a=1, b=1)$ $\xrightarrow{\quad}$ [Test path $p_2 : n_0, n_3, n_2$]

Test case $t_3 : (a=2, b=1)$ $\xrightarrow{\quad}$ [Test path $p_3 : n_0, n_2$]

Visitar y Recorrer



- $p = [n_0, n_1, n_3, n_4, n_6]$
- p “visita” el nodo n si $n \in p$
- p “visita” la arista e si $e \in p$
- p “recorre” el sub-camino q si $q \in p$

Criterios y Requerimientos de prueba

Criterio de prueba (C):

- reglas que definen requerimientos de prueba.

Ej: “Visitar todos los nodos”

Requerimiento de prueba (TR):

- describe propiedades de un camino de prueba.

Ej: “Visitar n_0 ”

Criterios y Requerimientos de prueba

“Dado un conjunto de requerimientos de prueba **TR** para un criterio de cobertura **C** , un conjunto de pruebas **T** satisface **C** si y sólo si para cada requerimiento **tr** en **TR** hay al menos un camino de prueba **p** que satisface **tr** .”

Criterios de cobertura en grafos

Los criterios de cobertura en grafos se dividen en 2 tipos:

- Estructural
- Flujo de información

Estructural

- En general hacen referencia a visitar nodos o aristas según un criterio específico.
- Se utilizará la siguiente notación:

$$\mathbf{TR} = \{\text{visita } \mathbf{n}_0, \text{ visita } \mathbf{n}_1\}$$
$$\mathbf{TR} = \{\mathbf{n}_0, \mathbf{n}_1\}$$

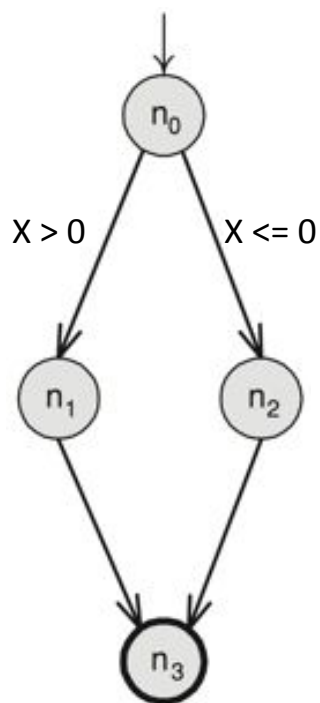
NC: Cobertura de nodos

(Node Coverage o Statement Coverage)

- Criterio:

TR contiene todos los nodos alcanzables de **G**.

$$\mathbf{TR} = \{n_0, n_1, n_2, n_3\}$$



$$\mathbf{p}_1 = [n_0, n_1, n_3]$$

$$\mathbf{p}_2 = [n_0, n_2, n_3]$$

$\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2\}$ tal que:

- $\text{path}(\mathbf{t}_1) = \mathbf{p}_1$
- $\text{path}(\mathbf{t}_2) = \mathbf{p}_2$

$$\mathbf{t}_1 = \{x=1\}$$

$$\mathbf{t}_2 = \{x=0\}$$

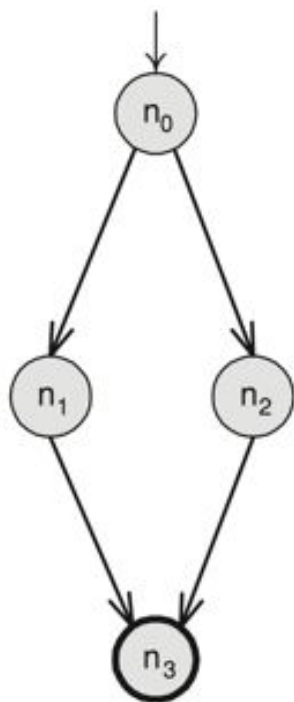
EC: Cobertura de aristas

(*Edge Coverage o Branch Coverage*)

- Criterio:

TR contiene todos los caminos de largo 1 alcanzables de **G**

$$\mathbf{TR} = \{(n_0, n_1) ; (n_1, n_3) ; (n_0, n_2) ; (n_2, n_3)\}$$



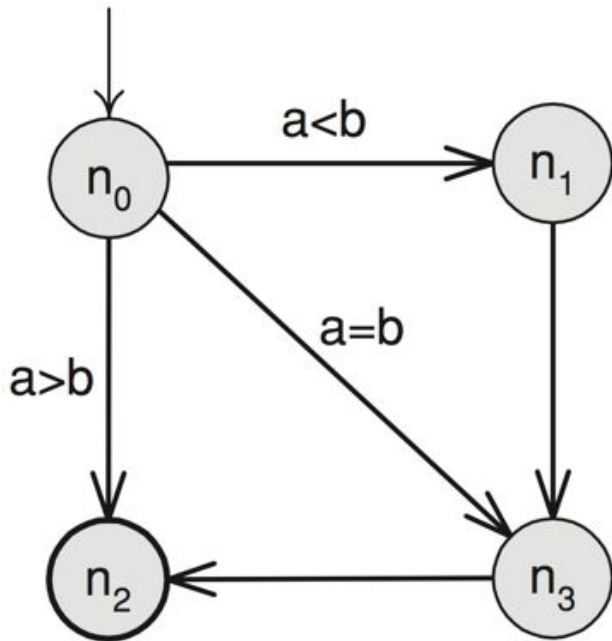
$$\mathbf{p}_1 = [n_0, n_1, n_3]$$

$$\mathbf{p}_2 = [n_0, n_2, n_3]$$

$\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2\}$ tal que:

- $\text{path}(\mathbf{t}_1) = \mathbf{p}_1$
- $\text{path}(\mathbf{t}_2) = \mathbf{p}_2$

NC vs EC



NC:

$$p_1 = [n_0, n_1, n_3, n_2]$$

EC:

$$p_1 = [n_0, n_2]$$

$$p_2 = [n_0, n_3, n_2]$$

$$p_3 = [n_0, n_1, n_3, n_2]$$

EPC: Cobertura de pares de aristas

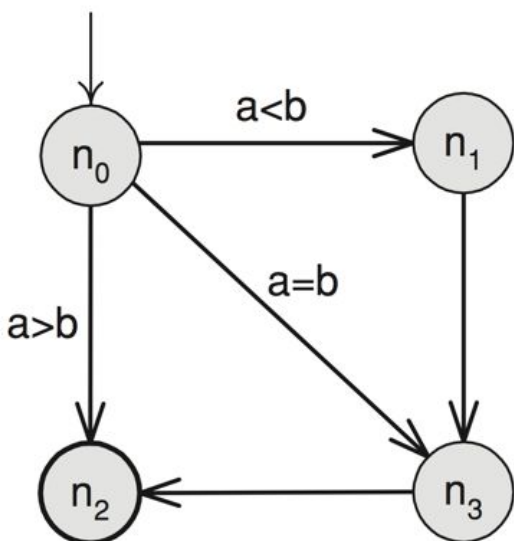
(*Edge Pair Coverage*)

- Criterio:

TR contiene todos los caminos de largo hasta 2 alcanzables de **G**

$$\mathbf{TR} = \{(n_0, n_1, n_3) ; (n_0, n_3, n_2) ; (n_1, n_3, n_2) ; (n_0, n_2) ; (n_0, n_1) ; (n_0, n_3) ; (n_1, n_3) ; (n_3, n_2)\}$$

$$\begin{aligned} p_1 &= [n_0, n_1, n_3, n_2] \\ p_2 &= [n_0, n_3, n_2] \\ p_2 &= [n_0, n_2] \end{aligned}$$



Camino Simple y Camino Primo

Camino Simple:

- Un camino desde n_i a n_j es simple si cada nodo no aparece más de una vez, con excepción del primer y último nodo que pueden ser el mismo.
- Son caminos sin ciclos internos, aunque pueden ser un ciclo en sí.
- Cualquier camino puede ser creado componiendo caminos simples

Camino Primo:

- Un camino desde n_i a n_j es primo si es un camino simple y no existe otro camino simple que lo contenga.
- Son caminos simples de largo máximo.

PPC: Cobertura de camino primo

(*Prime path coverage*)

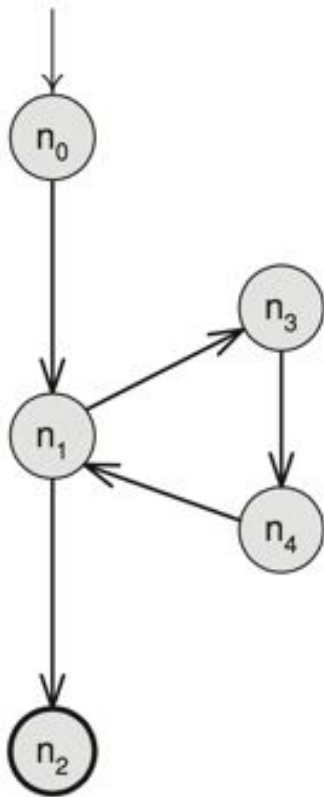
- Criterio:

TR contiene todos los caminos primos de **G**.

$$\text{TR} = \{(n_0, n_1, n_2) ; (n_0, n_1, n_3, n_4) ; (n_1, n_3, n_4, n_1) ; (n_3, n_4, n_1, n_3) ; (n_4, n_1, n_3, n_4) ; (n_3, n_4, n_1, n_2)\}$$

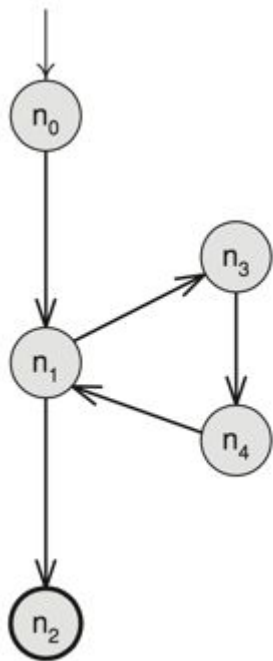
$$p_1 = [n_0, n_1, n_2]$$

$$p_2 = [n_0, n_1, n_3, n_4, n_1, n_3, n_4, n_1, n_2]$$



Round-Trip

- Un camino primo que comienza y termina en el mismo nodo.



$$\begin{aligned} p_1 &= [n_1, n_3, n_4, n_1] \\ p_2 &= [n_3, n_4, n_1, n_3] \\ p_3 &= [n_4, n_1, n_3, n_4] \end{aligned}$$

Criterios con *Round-Trips*

- **SRTC: *Simple Round Trip Coverage***
 - TR contiene al menos un round-trip para cada nodo n alcanzable en G que comienza y termina un camino de *round-trip*.
- **CRTC: *Complete Round Trip Coverage***
 - TR contiene todos los *round-trips* para cada nodo n alcanzable de G .

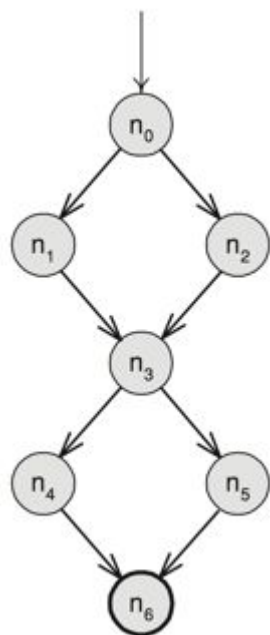
CPC: Cobertura de camino completo

(Complete path coverage)

- Criterio:

TR contiene todos los caminos en **G**.

$$\text{TR} = \{(n_0, n_1, n_3, n_4, n_6) ; (n_0, n_1, n_3, n_5, n_6) ; (n_0, n_2, n_3, n_4, n_6) ; (n_0, n_2, n_3, n_5, n_6)\}$$



$$p_1 = [n_0, n_1, n_3, n_4, n_6]$$

$$p_2 = [n_0, n_1, n_3, n_5, n_6]$$

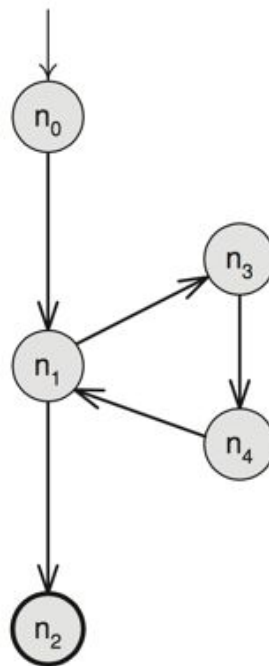
$$p_3 = [n_0, n_2, n_3, n_4, n_6]$$

$$p_4 = [n_0, n_2, n_3, n_5, n_6]$$

CPC: Cobertura de camino completo

(*Complete path coverage*)

- Criterio:
TR contiene todos los caminos en **G**.



¿Qué pasa con este grafo?

SPC: Cobertura de camino específico

(Specific path coverage)

- Criterio:

TR contiene un conjunto **S** de caminos de prueba, donde **S** es entregado como parámetro.

Criterios de cobertura en grafos

- Estructural:
 - **NC**: Cobertura de nodos
 - **EC**: Cobertura de aristas
 - **EPC**: Cobertura de pares de aristas
 - **PPC**: Cobertura de camino primo
 - **SRTC**: Cobertura simple con *round-trips*
 - **CRTC**: Cobertura completa con *round-trips*
 - **CPC**: Cobertura de camino completo
 - **SPC**: Cobertura de camino específico
- Flujo de información



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

24 de agosto de 2020