



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

# **Clase 8**

# **Cobertura en base a lógica**

## **IIC3745 – Testing**

Rodrigo Saffie

rasaffie@uc.cl

9 de septiembre de 2019

## 1. Clase pasada

- Cobertura en base a lógica
  - *Predicate Coverage*
  - *Clause Coverage*
  - *Combinatorial Coverage*
  - *General Active Clause Coverage*
  - *Correlated Active Clause Coverage*
  - *Restricted Active Clause Coverage*

## 2. Cobertura en base a lógica

- *Inactive Clause Coverage*
- Infactibilidad
- Definición de cláusulas activas

# ICC: Cobertura de cláusula inactiva

## *(Inactive Clause Coverage)*

- Los criterios de cobertura de cláusula activa aseguran que las cláusulas mayores tienen efecto sobre los predicados.
- La cobertura de cláusulas inactivas usa el enfoque opuesto: las cláusulas mayores no afectan a los predicados.
- Sirve para demostrar que una determinada acción **no puede** iniciar una acción.
  - Por ejemplo, si un avión vuela en modo seguro no es posible apagar los motores.

# ICC: Cobertura de cláusula inactiva

## *(Inactive Clause Coverage)*

Por cada  $p \in P$  y cada cláusula mayor  $c_i \in C_p$ , escoja las cláusulas menores  $c_j$  con  $i \neq j$  de modo que  $c_i$  no determina  $p$ . **TR** contiene cuatro requisitos por cada  $c_i$ :

1.  $c_i$  se evalúa como verdadero con  $p$  verdadero
2.  $c_i$  se evalúa como falso con  $p$  verdadero
3.  $c_i$  se evalúa como verdadero con  $p$  falso
4.  $c_i$  se evalúa como falso con  $p$  falso

Estos cuatro requisitos permiten demostrar que  $c_i$  no tiene incidencia alguna sobre  $p$ .

# ICC: Cobertura de cláusula inactiva

*(Inactive Clause Coverage)*

	a	b	c	$a \wedge (b \vee c)$
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

# ICC: Cobertura de cláusula inactiva

## *(Inactive Clause Coverage)*

- Al contrario de la cobertura de cláusulas activas, la noción de correlación no es relevante.
  - $c_i$  no determina a  $p$  de modo que no se pueden correlacionar
- La cobertura de predicados está siempre garantizada.

## **GICC:** Cobertura de cláusula inactiva general (*General Inactive Clause Coverage*)

Por cada  $p \in P$  y cada cláusula mayor  $c_i \in C_p$ , escoja las cláusulas menores  $c_j$  con  $i \neq j$  de modo que  $c_i$  no determina  $p$ . **TR** contiene cuatro requisitos por cada  $c_i$ :

1.  $c_i$  se evalúa como verdadero con  $p$  verdadero
2.  $c_i$  se evalúa como falso con  $p$  verdadero
3.  $c_i$  se evalúa como verdadero con  $p$  falso
4.  $c_i$  se evalúa como falso con  $p$  falso

Los valores de las cláusulas menores  $c_j$  no necesitan ser los mismos cuando  $c_i$  es verdadero y cuando  $c_i$  es falso.

## **RICC:** Cobertura de cláusula inactiva restrictiva (*Restrictive Inactive Clause Coverage*)

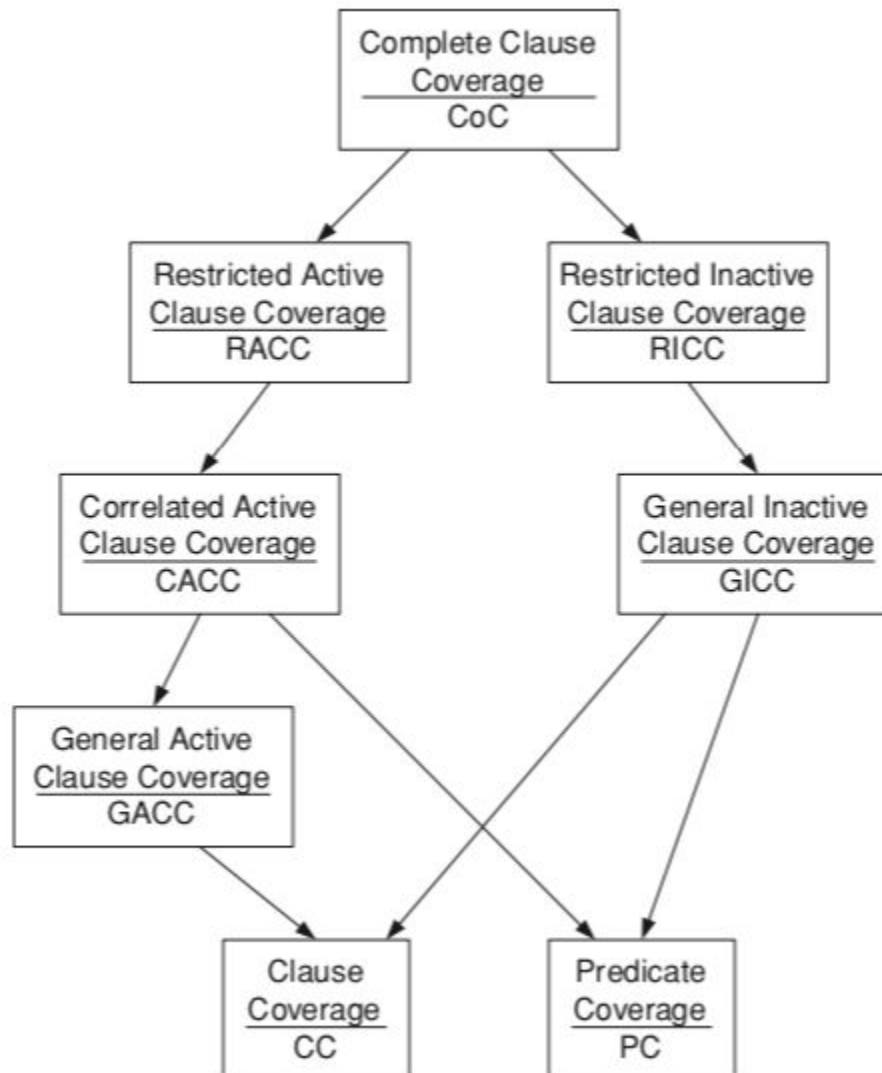
Por cada  $\mathbf{p} \in \mathbf{P}$  y cada cláusula mayor  $\mathbf{c}_i \in \mathbf{C}_p$ , escoja las cláusulas menores  $\mathbf{c}_j$  con  $i \neq j$  de modo que  $\mathbf{c}_i$  no determina  $\mathbf{p}$ . **TR** contiene cuatro requisitos por cada  $\mathbf{c}_i$ :

1.  $\mathbf{c}_i$  se evalúa como verdadero con  $\mathbf{p}$  verdadero
2.  $\mathbf{c}_i$  se evalúa como falso con  $\mathbf{p}$  verdadero
3.  $\mathbf{c}_i$  se evalúa como verdadero con  $\mathbf{p}$  falso
4.  $\mathbf{c}_i$  se evalúa como falso con  $\mathbf{p}$  falso

Los valores de las cláusulas menores  $\mathbf{c}_j$  deben ser los mismos cuando  $\mathbf{c}_i$  es verdadero y cuando  $\mathbf{c}_i$  es falso.



# Subsumición cobertura lógica



# Infactibilidad

- En la práctica existen varias complicaciones para aplicar estos criterios.
- Generalmente aparecen combinaciones de valores imposibles dado que las cláusulas están relacionadas.

```
while (i < n && a[i] != 0) {do something to a[i]}
```

- Por esta razón se busca satisfacer únicamente los requisitos de pruebas factibles.
- Además, se priorizan criterios con la mayor cantidad de opciones posibles (*CACC* sobre *RACC*).

# Infactibilidad

$$(a > b \wedge b > c) \vee c > a$$

- No es factible que:
  - $a > b = \text{true}$
  - $b > c = \text{true}$
  - $c > a = \text{true}$
- Los requisitos de pruebas que no son factibles deben ser **identificados** e **ignorados**.

# Definición cláusulas activas

- En predicados simples es fácil encontrar valores para cláusulas menores.
- Para encontrar los valores de cláusulas menores que definen una cláusula mayor se debe resolver:

$$\mathbf{p_c} = \mathbf{p_{c=true}} \oplus \mathbf{p_{c=false}}$$

- Luego de simplificar  $\mathbf{p_c}$  describe exactamente los valores necesarios para que  $\mathbf{c}$  determine a  $\mathbf{p}$ .
- Asimismo,  $\neg \mathbf{p}$  describe los valores necesarios para que  $\mathbf{c}$  no determine a  $\mathbf{p}$ .

# Evaluación disyunción exclusiva

$$p \oplus q$$

$$= (p \vee q) \wedge \neg(p \wedge q)$$

$$= (p \wedge \neg q) \vee (\neg p \wedge q)$$

# Ejemplos

$$p = a \vee b$$

$$\begin{aligned} p_a &= p_{a=true} \oplus p_{a=false} \\ &= (true \vee b) \oplus (false \vee b) \\ &= true \oplus b \\ &= \neg b \end{aligned}$$

$$p = a \wedge b$$

$$\begin{aligned} p_a &= p_{a=true} \oplus p_{a=false} \\ &= (true \wedge b) \oplus (false \wedge b) \\ &= b \oplus false \\ &= b \end{aligned}$$

$$p = a \vee (b \wedge c)$$

$$\begin{aligned} p_a &= p_{a=true} \oplus p_{a=false} \\ &= (true \vee (b \wedge c)) \oplus (false \vee (b \wedge c)) \\ &= true \oplus (b \wedge c) \\ &= \neg(b \wedge c) \\ &= \neg b \vee \neg c \end{aligned}$$

# Variables repetidas

$$(a \wedge b) \vee (b \wedge c) \vee (a \wedge c)$$

- Si bien hay 6 cláusulas, solamente son 3 únicas
- Existen 8 pruebas posibles (no 64)
- Conviene probar predicados simples
  - Se evitan casos de pruebas redundantes

# Variables repetidas

$$p = a \wedge b \vee a \wedge \neg b$$

$$p_a = p_{a=true} \oplus p_{a=false}$$

$$= true \wedge b \vee true \wedge \neg b \oplus false \wedge b \vee false \wedge \neg b$$

$$= b \vee \neg b \oplus false$$

$$= true \oplus false$$

$$= true$$

$$p_b = p_{b=true} \oplus p_{b=false}$$

$$= a \wedge true \vee a \wedge \neg true \oplus a \wedge false \vee a \wedge \neg false$$

$$= a \vee false \oplus false \vee a$$

$$= false$$



# Variables repetidas

$$p = a \wedge b \vee a \wedge \neg b$$

- ***a*** siempre determina a ***p***
- ***b*** nunca determina a ***p***

$$p = a$$

- Error conceptual que se debe detectar al momento de diseñar pruebas

# Aplicación en artefactos de *software*

- Código fuente
- Especificación de requisitos
- Máquinas de estados
- Forma Normal Disyuntiva (***DNF***)

# Código fuente

```
if (a && b)
    S1;
else
    S2;
```



```
if (a)
{
    if (b)
        S1;
    else
        S2;
}
else
    S2;
```

# Código fuente

```
if ((a && b) || c)
    S1;
else
    S2;
```



```
if (a)
    if (b)
        S1;
    else
        if (c)
            S1;
        else
            S2;
else
    if (c)
        S1;
    else
        S2;
```



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

# **Clase 8**

# **Cobertura en base a lógica**

## **IIC3745 – Testing**

Rodrigo Saffie

rasaffie@uc.cl

9 de septiembre de 2019