



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

# **Clase 7**

# **Cobertura en base a lógica**

## **IIC3745 – Testing**

Rodrigo Saffie

rasaffie@uc.cl

4 de septiembre de 2019

## 1. Clase pasada

- Cobertura aplicada en grafos
  - Especificación del diseño: máquinas de estado
  - Casos de uso

## 2. Cobertura en base a lógica

# Cobertura de expresiones lógicas

- Exigido por la Administración Federal de Aviación (FAA) de USA para software crítico
- Las expresiones lógicas generan ramificaciones en el flujo de un *software*:
  - Si estas son simples la cobertura de aristas es suficiente
  - Si estas son complejas se necesita un análisis más detallado para detectar defectos

# Cobertura de expresiones lógicas

- Están presentes en distintos artefactos de *software*
  - Código
    - *if, elsif, else*
    - *while, for*
    - *switch*
    - *excepciones*
  - Máquinas de estados
    - Pre-condiciones
  - Requisitos
    - Formales o implícitos

# Cláusulas y predicados

- **Cláusula** (*clause*): unidad atómica que se evalúa a un booleano
  - Puede ser de la forma:
    - Variable booleana
    - Expresión con operador relacional
      - $>$ ,  $<$ ,  $=$ ,  $\geq$ ,  $\leq$ ,  $\neq$
    - Llamada a función booleana
- **Predicado** (*predicate*): combinación de cláusulas a través de operadores lógicos
  - negación:  $\neg$
  - conjunción:  $\wedge$
  - disyunción:  $\vee$
  - implicancia:  $\rightarrow$
  - exclusión:  $\oplus$
  - equivalencia:  $\leftrightarrow$

# Ejemplo

$$(a < b) \vee f(x) \wedge D$$

3 cláusulas:

- $(a < b)$  - expresión relacional
- $f(x)$  - función booleana
- $D$  - variable booleana

# Definiciones

- $P$  es el conjunto de predicados
- $p$  es un predicado tal que  $p \in P$
- $C$  es el conjunto de cláusulas en  $P$
- $C_p$  es el conjunto de cláusulas en el predicado  $p$
- $c, c_i, c_j$  son cláusulas contenidas en  $C$

# Cobertura de predicados y cláusulas

Cobertura de predicados (**PC**): Por cada  $p \in P$ ,  $TR$  contiene dos requisitos:  $p$  se evalúa como verdadero y  $p$  se evalúa como falso.

- Equivalente a la cobertura de aristas basada en grafos.

Cobertura de cláusulas (**CC**): Por cada  $c \in C$ ,  $TR$  contiene dos requisitos:  $c$  se evalúa como verdadero y  $c$  se evalúa como falso.



## Ejemplo *PC*

$$p = ((a > b) \vee C) \wedge f(x)$$

2 pruebas son necesarias:

- *p* = true

- *a* = 5
- *b* = 4
- *C* = true
- *p*(*x*) = true

- *p* = false

- *a* = 5
- *b* = 6
- *C* = false
- *p*(*x*) = false

## Ejemplo **CC**

$$p = ((a > b) \vee C) \wedge f(x)$$

$TR = \{(a > b) = true; (a > b) = false; C = true; C = false; f(x) = true; f(x) = false\}$

2 pruebas son necesarias:

- $a = 5$
- $b = 4$
- $C = true$
- $p(x) = true$
- $a = 5$
- $b = 6$
- $C = false$
- $p(x) = false$

# Problemas con **PC** y **CC**

- **PC** no depende de todos los átomos, en especial cuando hay evaluación con corto circuito.
- **PC** no asegura **CC** ni viceversa

	a	b	$a \vee b$
1	T	T	T
2	T	F	T
3	F	T	T
4	F	F	F

{2 , 3} satisface **CC** pero no **PC**

{2 , 4} satisface **PC** pero no **CC**

# CoC: Cobertura combinatoria

## *(Combinatorial Coverage)*

Por cada  $p \in P$ ,  $TR$  contiene requisitos para las cláusulas en  $C_p$  de modo de evaluar cada combinación posible de valores de verdad

- Se consideran todas las combinaciones posibles

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

# CoC: Cobertura combinatoria

## *(Combinatorial Coverage)*

Por cada  $p \in P$ ,  $TR$  contiene requisitos para las cláusulas en  $C_p$  de modo de evaluar cada combinación posible de valores de verdad

- Esto es simple, completo pero muy caro...
  - $2^n$  casos de prueba, con  $n$  número de cláusulas
  - Impracticable para predicados complejos
- Posible mejora: probar cada **cláusula activa**

# Cláusula activa

No todas las cláusulas impactan al valor del predicado

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

# Cláusula activa

No todas las cláusulas impactan al valor del predicado

**Determinación:** Una cláusula  $c_i$  en un predicado  $p$  determina a  $p$  si y sólo si los valores de las demás cláusulas  $c_j$  son tales que al cambiar  $c_i$  se cambia el valor de  $p$ .

$c_i$  es la cláusula mayor

$c_j$  son las cláusulas menores

# ACC: Cobertura de cláusula activa (Active 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  $c_i$  determina  $p$ . **TR** contiene dos requisitos por cada  $c_i$ :  $c_i$  se evalúa como verdadero y  $c_i$  se evalúa como falso.

$$p = a \vee b$$

	a	b
$c_i = a$	<b>T</b>	f
	<b>F</b>	f
$c_i = b$	f	<b>T</b>
	<del>f</del>	<del><b>F</b></del>



# ACC: Ambigüedad

¿Es obligatorio que las cláusulas menores tengan los mismos valores cuando se evalúa el átomo mayor?

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

$$c_i = a$$

$$t_1: \{a=true, b=false, c=true\}$$

$$t_2: \{a=false, b=false, c=false\}$$

## **GACC:** Cobertura de cláusula activa general (*General active 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  $c_i$  determina  $p$ . **TR** contiene dos requisitos por cada  $c_i$ :  $c_i$  se evalúa como verdadero y  $c_i$  se evalúa como falso. Los valores escogidos para los átomos menores  $c_j$  **no necesitan ser el mismo** cuando  $c_i$  es verdadero y cuando  $c_i$  es falso.

# **GACC:** Cobertura de cláusula activa general (*General active clause coverage*)

$$p = a \leftrightarrow b$$

$$t_1: \{a=true, b=true\} \rightarrow p = true$$

$$t_2: \{a=false, b=false\} \rightarrow p = true$$

- Se satisface GACC pero ***p*** nunca fue ***false***

**GACC no implica PC**

- No es un buen criterio para realizar pruebas

## **RACC:** Cobertura de cláusula activa restrictiva (*Restrictive active 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  $c_i$  determina  $p$ . **TR** contiene dos requisitos por cada  $c_i$ :  $c_i$  se evalúa como verdadero y  $c_i$  se evalúa como falso. Los valores escogidos para los átomos menores  $c_j$  **deben ser el mismo** cuando  $c_i$  es verdadero y cuando  $c_i$  es falso.

# **RACC:** Cobertura de cláusula activa restrictiva (*Restrictive active clause coverage*)

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

$$c_i = a$$

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

- 3 opciones para *set* de pruebas

**CACC:** Cobertura de cláusula activa correlacionada  
(*Correlated active 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  $c_i$  determina  $p$ . **TR** contiene dos requisitos por cada  $c_i$ :  $c_i$  se evalúa como verdadero y  $c_i$  se evalúa como falso. Los valores escogidos para las cláusulas menores  $c_j$  **deben causar** que  $p$  sea verdadero para un valor de  $c_i$  y falso para el otro valor de  $c_i$ .

**CACC:** Cobertura de cláusula activa correlacionada  
(*Correlated active clause coverage*)

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

$$c_i = a$$

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

- 9 opciones para set de pruebas

# CACC vs RACC

- Considere un sistema con 2 variables:
  - modo de operación (“*Operacional*” o “*En espera*”)
  - válvula con posiciones “*abierta*” o “*cerrada*”
- Suponga las siguientes restricciones:
  - Si el modo de operación es “*Operacional*” la válvula debe estar abierta. En cualquier otro modo estará cerrada.
  - El modo no puede “*Operacional*” o “*En espera*” al mismo tiempo.



# CACC vs RACC

- El escenario cuenta con 3 cláusulas:
  - $a$ : válvula está cerrada
  - $b$ : modo es “Operacional”
  - $c$ : modo es “En espera”
- Considere una acción que sólo se ejecuta con la válvula cerrada, en modos *Operacional* o *En espera*.
  - $p = a \wedge (b \vee c)$
  - $r_1: \neg a \leftrightarrow b$
  - $r_2: \neg(b \wedge c)$

# CACC vs RACC

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

- Solamente 3, 4 y 6 son viables, pero **a** no es activo para 4.
- Con (3,6) se satisface CACC
- RACC es inalcanzable



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

# **Clase 7**

# **Cobertura en base a lógica**

## **IIC3745 – Testing**

Rodrigo Saffie

rasaffie@uc.cl

4 de septiembre de 2019