

Calidad, Validación y Verificación del Software

Técnicas Basadas en la
Estructura: Condiciones

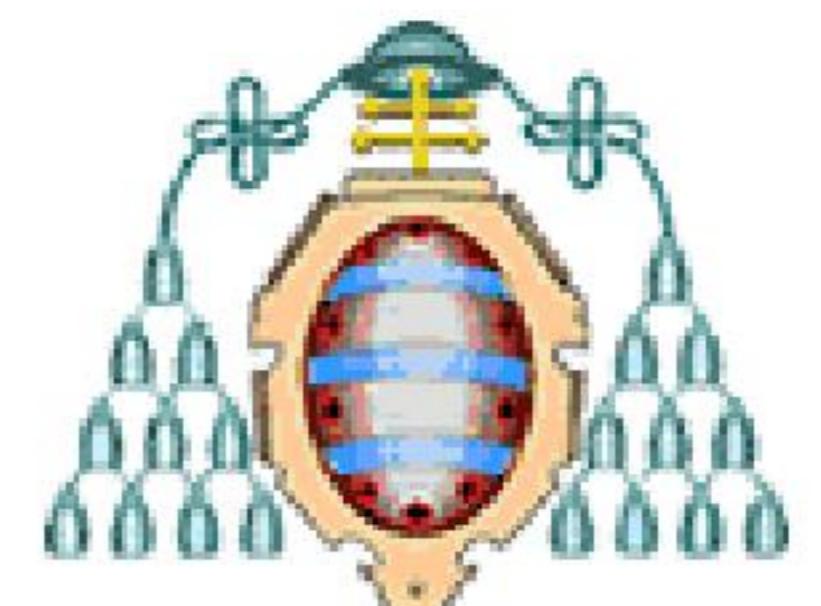
Grado en Ingeniería Informática del Software

Javier Tuya

Grupo de Investigación en Ingeniería del Software

<http://giis.uniovi.es>

Curso 2021-2022



Contenidos

- Técnicas Basadas en Condiciones
(parte de las técnicas basadas en la estructura):
 - Prueba de Decisiones
 - Prueba de Condiciones
 - Prueba de Condición/Decisión
 - Prueba de Múltiple Condición
 - Prueba de Condición/Decisión Modificada
- Pruebas Negativas
- Discusión Sobre Técnicas

Técnicas Basadas en Condiciones

- Utiliza la estructura de las decisiones lógicas que se definen explícita o implícitamente en la especificación
- Elementos básicos:
 - Condición: Expresión lógica primitiva sin operadores lógicos (ej. A>B)
 - Decisión: Expresión lógica compuesta por varias (o una) condiciones (ej. A>B AND C>D)
- Diferentes técnicas basadas en diferentes criterios:
 - Decisión
 - Condición
 - Condición/Decisión
 - Múltiple Condición
 - Condición/Decisión modificada (MCDC)

Técnicas Basadas en Condiciones

- Problema 1: La remuneración especial de las cuentas corrientes (que ofrece un tipo de interés superior al habitual) se aplica solamente cuando se ha mantenido la cuenta abierta al menos dos años a los clientes menores de 21 años y a los que han mantenido un saldo medio anual como mínimo de 50K euros durante el último año

Técnicas Basadas en Condiciones

- Problema 1: La remuneración especial de las cuentas corrientes (que ofrece un tipo de interés superior al habitual) se aplica solamente cuando se ha mantenido la cuenta abierta al menos dos años a los clientes menores de 21 años y a los que han mantenido un saldo medio anual como mínimo de 50K euros durante el último año
- Cuál es la decisión y las condiciones?
- Habilidad del tester: determinar las decisiones cuando no están explícitas

Prueba de Decisiones

■ Objetivo (Criterio de cobertura)

- Derivar situaciones a cubrir de forma que cada decisión tome los valores cierto y falso

■ $(e < 21 \text{ OR } s \geq 50) \text{ AND } a \geq 2$

Realizamos la tabla de verdad utilizando el criterio de cobertura de decisiones y luego ponemos los valores apropiados para el caso de prueba

Tabla de verdad			Salida	Entradas		
$e < 21$	$s \geq 50$	$a \geq 2$	Salida	e	s	a
T	T	T	T	20	50	2
T	T	F	F	20	50	1

■ Discusión: Qué ocurre si en la segunda condición ($s \geq 50$) se ha **implementado erróneamente otro límite diferente** como por ejemplo, 5 ?

Prueba de Condiciones

- Objetivo
 - Derivar situaciones a cubrir de forma que cada condición tome los valores cierto y falso
- $(e < 21 \text{ OR } s \geq 50) \text{ AND } a \geq 2$

Tabla de verdad				Entradas		
$e < 21$	$s \geq 50$	$a \geq 2$	Salida	e	s	a
F	F	T	F	21	49	2
T	T	F	F	20	50	1

- Discusión: Qué ocurre si en la expresión entre paréntesis se ha **implementado un AND en vez de un OR?** (la especificación parece definir una conjunción entre e y s)

Prueba de Decisión/Condición

- Objetivo
 - Derivar situaciones a cubrir de forma que cada condición y cada decisión tome los valores cierto y falso
- $(e < 21 \text{ OR } s \geq 50) \text{ AND } a \geq 2$

Tabla de verdad				Entradas		
$e < 21$	$s \geq 50$	$a \geq 2$	Salida	e	s	a
T	T	T	T	20	50	2
F	F	F	F	21	49	1

- Discusión: Qué ocurre si en el segundo operador lógico se ha implementado un OR en vez de un AND?

Prueba de Múltiple Condición

- Objetivo
 - Derivar situaciones a cubrir de forma que se ejercite cada una de las combinaciones lógicas de las condiciones.
- $(e < 21 \text{ OR } s \geq 50) \text{ AND } a \geq 2$
 - Hay tres condiciones, 8 combinaciones.
 - Permite detectar los tres defectos discutidos anteriormente?
- Discusión: Es lo mismo que tablas de decisión?
- Problema: crecimiento exponencial:
 - Para 4 condiciones: 16 combinaciones.
 - Para 5 condiciones: 32 combinaciones
 - En general para n condiciones son 2^n combinaciones
- Cómo se puede reducir?

Prueba de Condición/Decisión Modificada (MCDC)

- Objetivo
 - Derivar situaciones a cubrir de forma que cada condición **afecta de forma independiente** al resultado de la decisión (por parejas, variar cada condición manteniendo inalteradas el resto)
- $(e < 21 \text{ OR } s \geq 50) \text{ AND } a \geq 2$, aplicando además VL

	Tabla de verdad				Entradas		
	$e < 21$	$s \geq 50$	$a \geq 2$	Salida	e	s	a
(1)	T	F	T	T	20	49	2
(2)	F	F	T	F	21	49	2
(3)	F	T	T	T	21	50	2
(4)	T	F	F	F	20	49	1

- Discusión: Permite detectar los tres defectos discutidos anteriormente?

(e<21 OR s≥50) AND a≥2

(e<21 OR s≥50) AND a≥2

Variantes de MCDC

- Habitualmente, de n condiciones resultan $n+1$ combinaciones
- En ocasiones si las condiciones son complejas y existen variables que están en varias condiciones puede ser difícil de conseguir. Dos variantes principales
 - **Unique-Cause MCDC:** Ejemplo anterior: par 1, 4 (ver tabla) mantiene valores en las dos primeras condiciones, la salida cambia al cambiar la tercera condición.
 - **Masking MCDC:** Relaja este requisito, solo se requiere que mantengan el valor las subexpresiones.
Ej. (1 con 4, 4', 4'' cumplen masking:
en todas $e < 12$ OR $s \geq 50$ es True)

	$e < 21$	$s \geq 50$	$a \geq 2$	
(1)	T	F	T	T
(4)	T	F	F	F
(4')	F	T	F	F
(4'')	T	T	F	F

Masking MCDC

■ Ejemplo:

- $(e < 21 \text{ AND } x) \text{ OR } (e \geq 21 \text{ AND } y)$
 - Condiciones 1 y 3 dependientes entre ellas
 - No es posible cambiar la condición 1 y mantener la 3

Tabla de verdad (1^a condición)

e<21	x	e≥21	y	Salida
F	T	T	F	F
T	T	F	F	T

■ Usar cuando no es posible mantener Unique-Cause

- Por defecto al hablar de MCDC nos referimos Unique-Cause
- Salvo que sea imposible como en ejemplo anterior
- En ocasiones puede originar pares imposibles de satisfacer

Pruebas Negativas

- Intentar ver si hace lo que no debería hacer
 - Ejercitar la funcionalidad que trata con fallos/rechazo de entradas, excepciones, etc. (concepto típico de pruebas negativas)
 - Además: Incluir pruebas/valores/condiciones para determinar valores que NO influyen en la salida
- En Problema 1:
 - Las condiciones dependen del saldo medio anual
 - No dependen de otros saldos (a final de año, saldo actual...)
 - Qué ocurre si se implementa incorrectamente y se confunde el valor usado para determinar el saldo?
 - Se podría aplicar el principio de MCDC (sin cambio en salida)
- Si los datos de entrada están en una BD:
 - Cuando $s \geq 50$ incluir un dato de saldo a fin de año < 50 y viceversa

Dónde están la Test Condition y los Test Coverage Items?

- $(e < 21 \text{ OR } s \geq 50) \text{ AND } a \geq 2$

	e < 21	s \geq 50	a \geq 2
(1)	T	F	T
(2)	F	F	T
(3)	F	T	T
(4)	T	F	F

Test Condition
(Condición de prueba):
La expresión lógica
en su conjunto

Test Coverage Items
(Situaciones a cubrir):
Cada una de las
combinaciones
lógicas que se probarán,
dependiendo de la
técnica
utilizada

Analogía con clases de equivalencia:

Si cada condición expresa una condición de entrada con dos clases de equivalencia, la tabla de verdad expresa combinaciones que tienen en cuenta tanto las clases de equivalencia como la relación entre ellas expresada por la decisión

Resumen/Discusión

- Aplicar valores límite en las condiciones cuando sea posible (se ha hecho en varios de los ejemplos anteriores)
- Grandes grupos de técnicas de prueba
 - Basadas la **especificación** (p.e. basadas en partición en clases de equivalencia)
 - Basadas en la **estructura** (p.e. las basadas en condiciones)
 - La especificación se estructura como decisiones/condiciones
 - También se pueden aplicar a las expresiones en el código fuente
 - Precaución: Si nos basamos en la implementación podemos estar derivando casos de prueba a partir de condiciones que son incorrectas
- Tradicionalmente se han denominado como
 - Basadas en la estructura: técnicas de caja blanca (**no correcto**)
 - Basadas en la especificación: técnicas de caja negra
 - Conceptos ortogonales: tipo de técnica vs. caja blanca/negra
- Independientemente de la técnica, lo habitual será partir de la especificación (salvo algunas de las que hace el desarrollador)

Resumen/Discusión

- La prueba se realiza ejecutando casos de prueba (test cases)
 - Para ello hay que determinar qué es lo que hay que probar (test conditions) y qué situaciones se probarán (test coverage ítems) apoyándose en el uso de técnicas de prueba
 - La cobertura es el porcentaje de situaciones (test coverage ítems) cubiertas por los casos de prueba

