

Criterio di copertura MC/DC

La copertura delle condizioni/decisioni modificate (MC/DC) è una buona risposta.

Il criterio MC/DC esamina **le combinazioni** di condizioni, come fa la copertura del percorso.

Invece di testare *tutte* le possibili combinazioni, MC/DC identifica le combinazioni *importanti* che devono essere testate.

Se le condizioni hanno solo esiti **binari** (ovvero, *vero* o *falso*), il numero di test necessari per raggiungere una copertura MC/DC del 100% è $N + 1$, dove N è il numero di condizioni nella decisione.

Nota che $(N+1) \ll 2^N$



Criterio di copertura MC/DC: un esempio

Le condizioni a, b, c vengono valutate come valori booleani:

`se(a && (b || c))`

Quanti test vengono eseguiti per MC/DC? E per la copertura del percorso?

Se applichiamo il criterio MC/DC saranno sufficienti solo $3+1 = 4$ casi di test, invece di 8 (criterio di copertura del percorso)

Per ogni condizione (ad esempio 'a') dovremmo cercare:

- Un caso in cui la condizione 'a' è vera (T1)
- Un caso in cui la condizione 'a' è falsa (T2)
- T1 e T2 devono avere esiti diversi (uno vero e uno falso)
- Le variabili 'b' e 'c' in T1 devono avere lo stesso valore in T2

T1 e T2 sono chiamate **coppie indipendenti**, perché la variabile 'a' influenza in modo indipendente il risultato (decisione).



Criterio di copertura MC/DC: un esempio

- Per ogni condizione (ad esempio 'a') dovremmo cercare:
- Un caso in cui la condizione 'a' è vera (T1)
 - Un caso in cui la condizione 'a' è falsa (T2)
 - T1 e T2 devono avere esiti diversi (uno vero e uno falso)
 - Le variabili 'b' e 'c' in T1 devono avere lo stesso valore in T2

Test case	isLetter (a)	last == s (b)	last == r (c)	decision
T1	true	true	true	true
T2	true	true	false	true
T3	true	false	true	true
T4	true	false	false	false
T5	false	true	true	false
T6	false	true	false	false
T7	false	false	true	false
T8	false	false	false	false

T1 e T5 sono una coppia di test (una coppia di indipendenza) in cui isLetter è l'unico parametro diverso e il risultato (decisione) cambia.



Criterio di copertura MC/DC: un esempio

Per ogni condizione (ad esempio 'a') dovremmo cercare:

- Un caso in cui la condizione 'a' è vera (T1)
- Un caso in cui la condizione 'a' è falsa (T2)
- T1 e T2 devono avere esiti diversi (uno vero e uno falso)
- Le variabili 'b' e 'c' in T1 devono avere lo stesso valore in T2

Test case	isLetter	last == s	last == r	decision
T1	true	true	true	true
T2	true	true	false	true
T3	true	false	true	true
T4	true	false	false	false
T5	false	true	true	false
T6	false	true	false	false
T7	false	false	true	false
T8	false	false	false	false



Criterio di copertura MC/DC: un esempio

Abbiamo trovato le seguenti coppie:

\ddot{y} èLettera: {1, 5}, {2, 6}, {3, 7} \ddot{y} ultima

$==$ s: {2,4}

\ddot{y} ultimo $==$ r: {3,4}

Quale coppia scegliere per ciascuna variabile?

Per le ultime due condizioni è semplice perché abbiamo solo una coppia, quindi abbiamo sicuramente bisogno dei test: T2, T3, T4

Quale coppia scegliere per isLetter?



Criterio di copertura MC/DC: un esempio

Abbiamo trovato le seguenti coppie:

isLetter: {1, 5}, {2, 6}, {3, 7} is ultima

== s: {2,4}

is ultimo == r: {3,4}

Per le ultime due condizioni è semplice perché abbiamo solo una coppia, quindi abbiamo sicuramente bisogno dei test: **T2, T3, T4**

Quale coppia scegliere per isLetter?

Se scegliamo {1, 5} avremo alla fine 5 test (T1, T2, T3, T4, T5)!

Mentre con {2, 6}, {3, 7} non fa differenza, in entrambi i casi abbiamo solo 4 test (come previsto), il che è meglio di 8 (copertura del percorso)!:

- {2, 6} -> T2, T3, T4, **T6**

- {3, 7} -> T2, T3, T4, **T7**



Criterio di copertura MC/DC: esercizi

Esercitati con MC/DC!

Esercizio 1 (10 minuti):

- Considera l'espressione $(A \ \& \ B) \ | \ C$

Quale suite di test serve per ottenere una copertura MC/DC del 100%?

Caso di prova A		B	C	$(A \ \& \ B) \ \ C$
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	T
6	F	T	F	F
7	F	F	T	T
8	F	F	F	F



Criterio di copertura MC/DC: esercizi

Soluzione dell'esercizio 1:

- Considera l'espressione $(A \text{ e } B) \mid C$ Qual
è la suite di test per ottenere una copertura MC/DC del 100%?

$\{2, 3, 4, 6\}$

$\{2, 4, 5, 6\}$



Criterio di copertura MC/DC: esercizi

Esercizio 2 (10 minuti):

- Consideriamo l'espressione $A \ \&\& \ (A \ || \ B)$

Quale suite di test serve per ottenere una copertura MC/DC del 100%?

Test	A	B	Decison
1	F	F	F
2	F	T	F
3	T	F	T
4	T	T	T



Criterio di copertura MC/DC: esercizi

Soluzione dell'esercizio 2:

- Consideriamo l'espressione $A \ \&\& \ (A \ || \ B)$

Quale suite di test serve per ottenere una copertura MC/DC del 100%?

A: $\{(1, 3), (2, 4)\}$

B: $\{ \text{(vuoto)} \}$

1. Non esiste una coppia di indipendenza per B. Pertanto, non è possibile ottenere Copertura MC/DC per questa espressione.
2. Poiché non esiste una coppia di indipendenza per B, questo parametro non influenzare il risultato. Dovresti consigliare allo sviluppatore di ristrutturare l'espressione senza usare B, rendendo il codice più facile da gestire.
3. Questo esempio mostra che i tester del software possono contribuire al codice qualità non solo individuando bug ma anche suggerendo modifiche che si traducono in una migliore manutenibilità.



Criterio di copertura MC/DC: esercizi

Domanda 3:

- Considera l'espressione $(A \ \&\& \ B) \ || \ (A \ \&\& \ C)$

Cosa noti?



Criterio di copertura MC/DC: esercizi

Domanda 3:

- Considera l'espressione $(A \ \&\& \ B) \ || \ (A \ \&\& \ C)$

Cosa noti?

È impossibile cambiare la prima A senza cambiare la seconda A!

In questi casi, permettiamo ad A di variare, ma fissiamo tutte le altre variabili (questo è chiamato **MC/DC mascherato**).



Criterio di copertura MC/DC: esercizi

Domanda 4:

- Considera l'espressione $(A \ \&\& \ B) \ || \ (A \ \&\& \ \text{non } B)$

Cosa noti?



Criterio di copertura MC/DC: esercizi

Domanda 4:

- Considera l'espressione $(A \ \&\& \ B) \ || \ (A \ \&\& \ \text{non } B)$

Cosa noti?

È impossibile ottenere MC/DC in tale espressione

Non ci sono coppie che mostrano l'indipendenza di B

Rivedi l'espressione semplicemente A



Criterio di adeguatezza del confine dei loop

Considerato che è impossibile effettuare test esaustivi, come gestire:

- Un ciclo di lunga durata (che si ripete per molte iterazioni)
- Un ciclo illimitato (che viene eseguito un numero sconosciuto di volte)

Criterio di adeguatezza del confine del loop:

una suite di test soddisfa questo criterio se e solo se per ogni ciclo \exists Esiste un caso di test che esercita il ciclo **zero** volte.

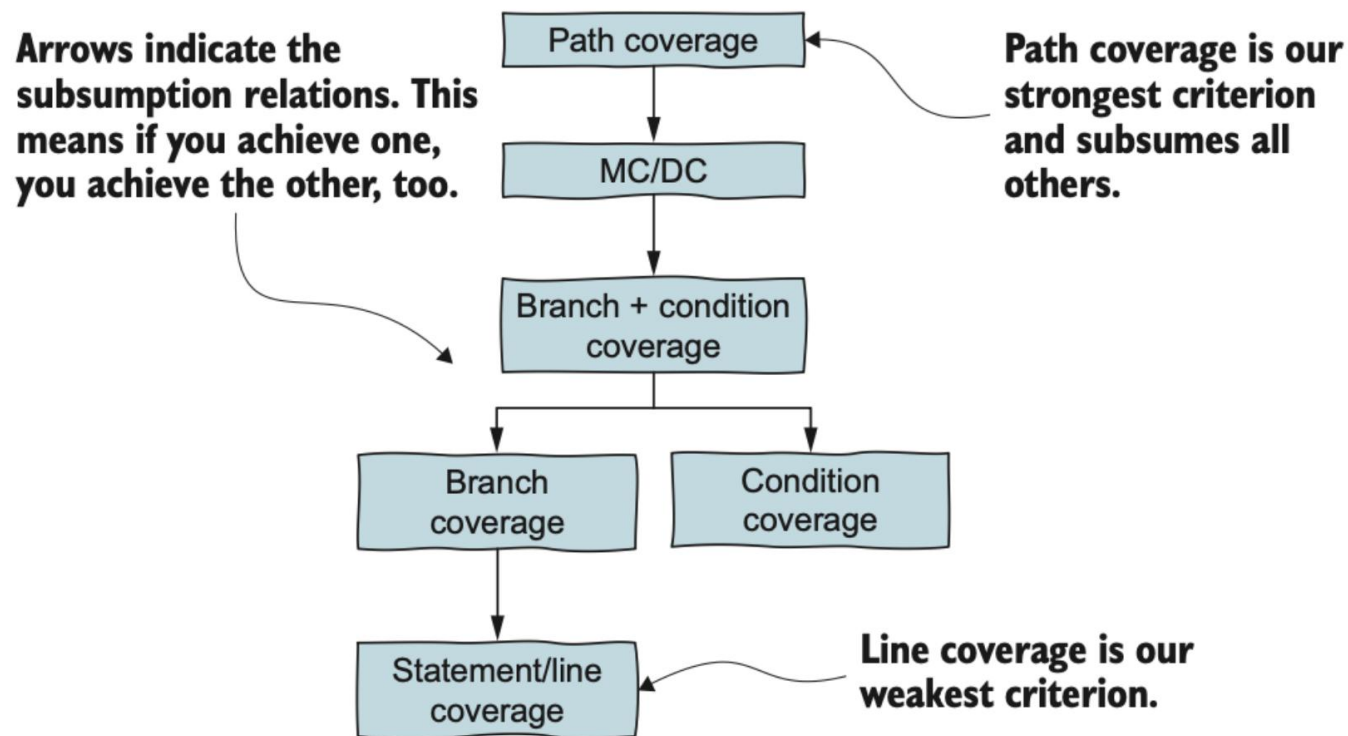
\exists Esiste un caso di prova che esercita il ciclo **una volta**.

\exists Esiste un caso di prova che esercita il ciclo **più** volte.

NB: È possibile creare anche più di un test per il caso “più volte”.



Criteri di sussunzione



Criterio *più debole* : più economico e veloce, lasciare il codice scoperto

Criterio *più forte* : costo più elevato, copertura migliore del codice



Copertura filiale vs condizione

Esempio:

$Se(A||B)$

$T1 = \{vero, falso\}$

$T2 = \{falso, vero\}$

Quale criterio stiamo soddisfacendo?



Copertura filiale vs condizione

Esempio:

$Se(A||B)$

$T1 = \{vero, falso\}$

$T2 = \{falso, vero\}$

Quale criterio stiamo soddisfacendo?

Condizione, ma non copertura della filiale!



Copertura del codice (CC): riassunto

- Non guardare ciecamente il numero di copertura (non prendere il 100% come obiettivo, ma prova per capire i numeri)
- Non manipolare la metrica
- Test strutturali e test basati sulle specifiche di aumento/complemento CC:
 - Identificare la parte del codice non ancora coperta dalla suite di test
 - Identificare le partizioni mancanti
- A volte va bene lasciare una parte del codice scoperta
- CC non è un *obiettivo di qualità* in quanto non indica una buona suite di test (una copertura del 100% non è un indicatore della qualità di una suite di test)
- Se la copertura è *bassa*, il sistema NON è stato testato correttamente (ma il contrario non è vero)



Criteri di copertura del codice

Regola pratica:

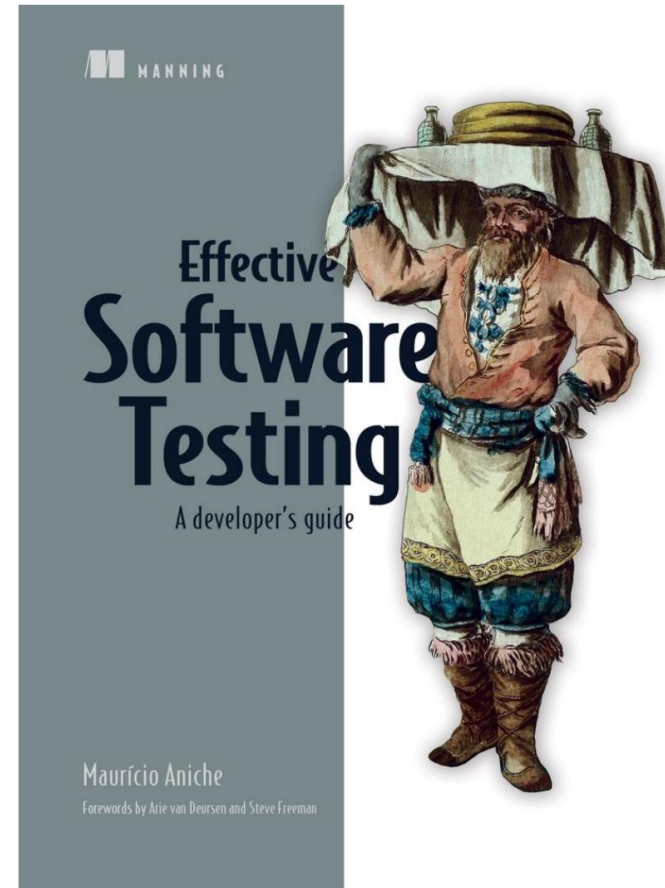
- Iniziare con la copertura delle filiali;
- Se ci sono espressioni più complesse: valutare se è necessario b+c o MC/DC.

Se le espressioni sono troppo complicate, prova a suddividerle in parti più piccole.



Libro di riferimento:

Test software efficaci. Guida
per sviluppatori. Mauricio Aniche.
Ed. Manning. **(Capitolo 3)**



L



Riferimenti

- Documentazione sulla copertura del codice JetBrains IntelliJ IDEA:
<https://www.jetbrains.com/help/idea/code-coverage.html>

L





Azzurra Ragone

Dipartimento di Informatica - Piano VI - Stanza 616 Email:
azzurra.ragone@uniba.it