



Testarea structurala White-Box

Rusu Cristian

Scopul: Crearea de cazuri de test pentru acoperirea starilor si tranzitiilor din structura programului testat, precum si crearea de cazuri de test pentru extinderea acoperirii daca aceasta nu este de 100%.

Tehnici de proiectare:

- Bazate pe specificatii (Bazate pe comportament sau Black-Box)
- Bazate pe structura (White- Bazate pe structura (White-Box)

Caracteristici:

- Necesita cunoasterea completa a structurii programului si accesul la codul sursa.
- Pune accentul pe acoperirea prin testare a cailor, ramificatiilor si fluxurilor programului.
- Se are in vedere gradul in care cazurile de test acopera sau executa codul sursa al programului.
- Tehnica utilizata cel mai des de programatori pentru testarea propriului cod.
- Se detecteaza erorile ce cauzeaza executarea unei alte cai a programului decat cea care trebuia sa se execute.

Fluxul de control:


Modulele de cod sunt convertite in grafuri, sunt analizate traseele din grafuri si sunt create cazuri de test in urma acestei analize. Exista diferite nivele de acoperire.

Elementele grafurilor:

Blocuri de Proces: o secventa de stari ale programului care se executa o secventa de stari ale programului care se executa secvential.

Punct de Decizie: un punct in modulul de cod in car Punct de Decizie: e fluxul de executie se poate schimba.

Punct de Jonctiune: un punct in modulul de cod in c Punct de Jonctiune: are fluxurile de executie se unesc.

Bloc de proces 

Puncte de decizie

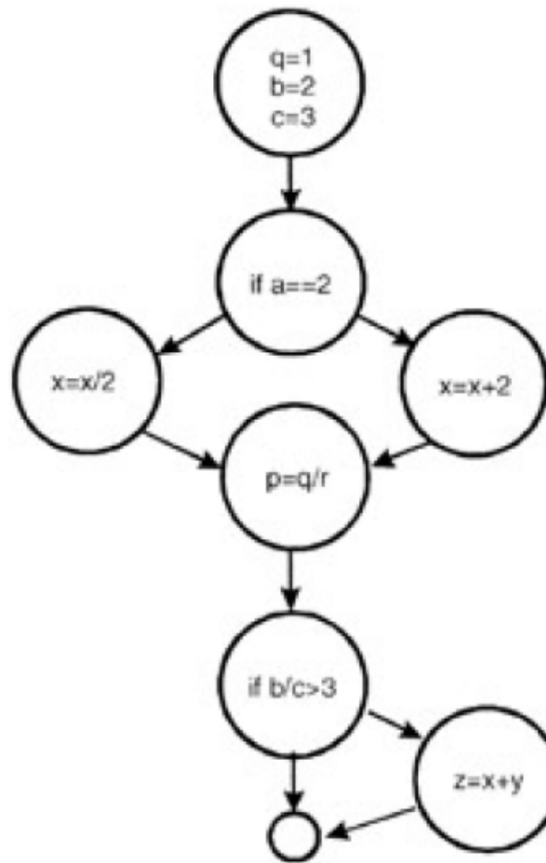


Punct de jonctiune



Exemplu de graf pentru un modul de cod

```
q=1;  
b=2;  
c=3;  
if (a==2) {x=x+2;}  
else {x=x/2;}  
p=q/r;  
if (b/c>3) {z=x+y;}  
z=x+y;
```



Tehnici:

- Acoperirea starilor (Statement coverage)
- Acoperirea arcurilor si deciziilor (Branch and Decision coverage)

Definitii

Testarea white - Testarea white-box: testarea bazata pe analiza structurii interne a componentei sau sistemului.

Acoperirea codului - o metoda de analiza care determina ce parti ale softului au fost executate (acoperite) de suita de teste si ce parti nu au fost acoperite, de exemplu acoperirea starilor, acoperirea deciziilor sau acoperirea conditiilor.

Acoperirea deciziilor: procentul iesirilor unei stari de decizie care au fost testate de o suita de teste. 100% acoperirea deciziilor implica 100% acoperirea traseelor si 100% acoperirea starilor

Acoperirea conditiilor : procentul iesirilor unei conditii care a fost acoperit de o suita de teste. Acoperirea conditiilor de 100% presupune ca fiecare conditie din fiecare stare de decizie sa fie testata atat pentru valoarea Adevarat cat si pentru Fals.

Acoperirea starilor - procentul starilor executabile care au fost antrenate de suita de teste.

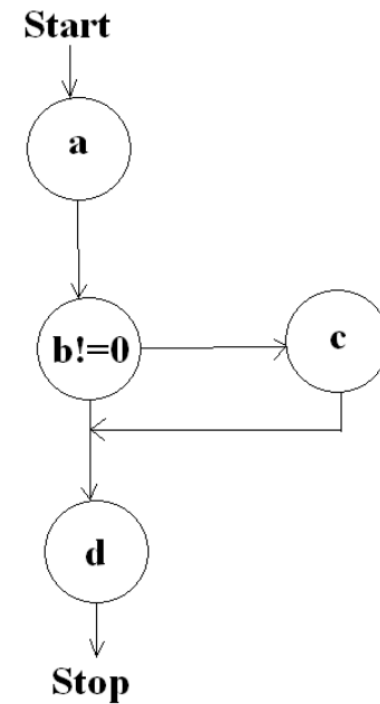
Acoperirea starilor

Acoperirea starilor = $\text{Nr. starilor executate} / \text{Nr. total al starilor executabile}$

Exemplu:

Daca $b \neq 0$ va rezulta
o acoperire a starilor
de 100%

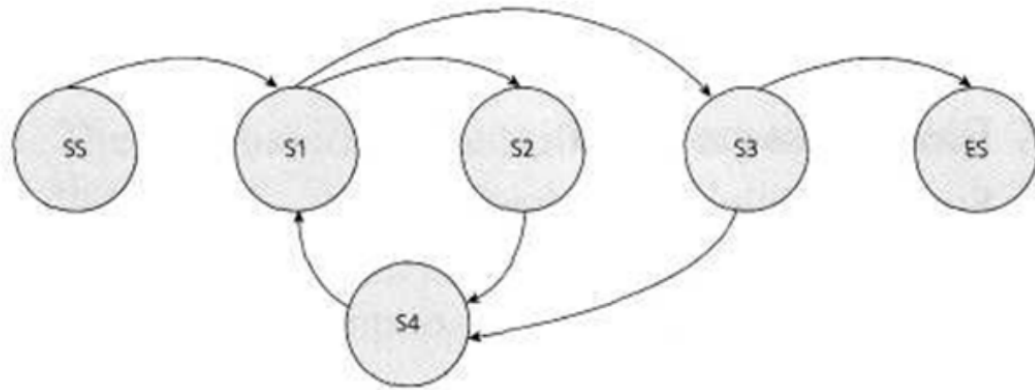
```
a;  
if (b) {  
    c;  
}  
d;
```



Exercitiu

Fiind data diagrama de stari alaturata, ce caz de test reprezinta seria minima de tranzitii valide care sa acopere fiecare stare?

- a. **SS-S1-S2-S4-S1-S3-ES**
- b. SS-S1-S2-S3-S4-ES
- c. SS-S1-S2-S4-S1-S3-S4-S1-S3-ES
- d. SS-S1-S4-S2-S1-S3-ES



Acoperirea arcurilor si deciziilor

Arc = un transfer conditionat dintr-o stare a Arc grafului in orice alta stare a grafului.

Acoperirea arcurilor = arcurile executate/Nr. total de arcuri

Acoperirea deciziilor = iesirile deciziilor executate/Nr. total al deciziilor

Pentru componentele cu un singur punct de intrare

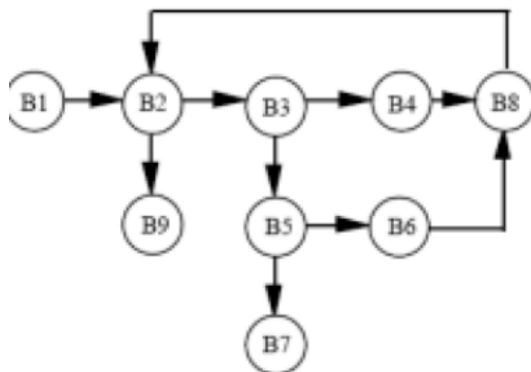
100% Acoperirea Arcurilor este echivalent cu

100% Acoperirea Deciziilor.

Exemplu

```
int binsearch (char *word, struct key tab[], int n) {  
    int cond;  
    int low, high, mid;
```

```
B1      low = 0;  
        high = n - 1;  
-----  
B2      while (low <= high) {  
-----  
B3          mid = (low+high) / 2  
            if ((cond = strcmp(word, tab[mid].word)) < 0)  
-----  
B4                high = mid - 1;  
-----  
B5          else if (cond > 0)  
-----  
B6                low = mid + 1;  
-----  
B7          else  
            return mid;  
-----  
B8      }  
-----  
B9      return -1;  
}
```



Deciziile=B2,B3, B5, fiecare cu cate 2 iesiri=3*2=6

Arcurile=(Cate sageti?)=10

Q1. Care este acoperirea deciziilor si a arcurilor pentru secventa (B1->B2->B9)?

Q2. Dar pentru (B1->B2->B3->B4->B8->B2->B3->B5->B6->B8->B2->B3->B5->B7)?

R: 1. 1/6, 2/10 2. 5/6, 9/10