



## ***Test du Logiciel (Certification ISTQB)***

# **Chapitre II: Le test Structurel**

*Support de cours de:*  
**Wafa Boussellaa**

*Filières : LCE2\_IOT ; LCS2*

***Année : 2023-2024***

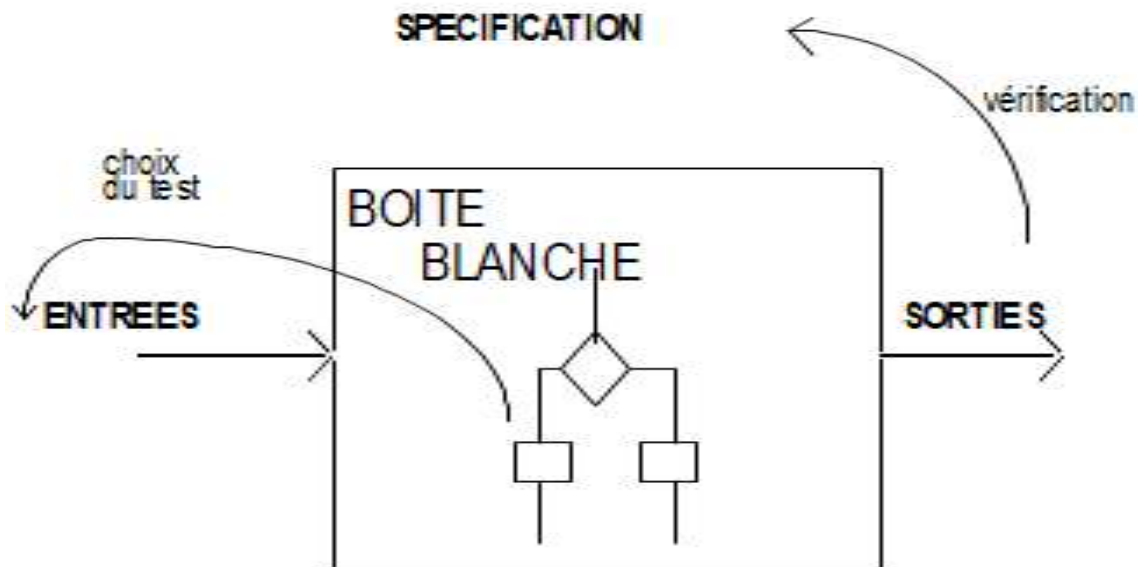
# [ Chapitre II. Le Test Structurel ]

---

1. Définition
2. Les Types de test
3. Le Test Structurel Dynamique

# [ Définition ]

Le test structurel ou « test à boîte blanche » s'appuie sur l'analyse du code source de l'application (ou d'un modèle de celui-ci).



# [Test structurel Statique/dynamique]

- **Analyse dynamique**

Nécessite l'exécution du code

**Principe** : A partir du code source et la spécification, produire des **DT** qui exécuteront un ensemble de comportements, et comparer les résultats avec ceux attendus.

**Méthode** : Techniques de couverture du graphe de contrôle

- Couverture du flot de contrôle
- Couverture du flot de données

- **Analyse statique**

Ne nécessite pas l'exécution du code

**Méthode** : Parmi les techniques les plus utilisée :

- Estimation de la complexité

# Test structurel dynamique : Couverture du graphe de contrôle

## ➤ Objectif :

Produire des DT qui exécutent un ensemble de comportements du programme. Ce test utilise la spécification, le code source et le code exécutable du programme.

## ➤ Principe :

- Chaque programme est représenté par **un graphe de contrôle (G)** ;
- ✓ **Un graphe de contrôle comporte :**
  - ✓ un seul sommet **entrée** ;
  - ✓ un seul sommet **sortie** ;
  - ✓ Les **nœuds** du graphe = les blocs d'instructions ;
  - ✓ Les **arcs** ou **branches** = la possibilité de transfert de l'exécution d'un nœud à un autre ;

# Test structurel dynamique : Couverture du graphe de contrôle

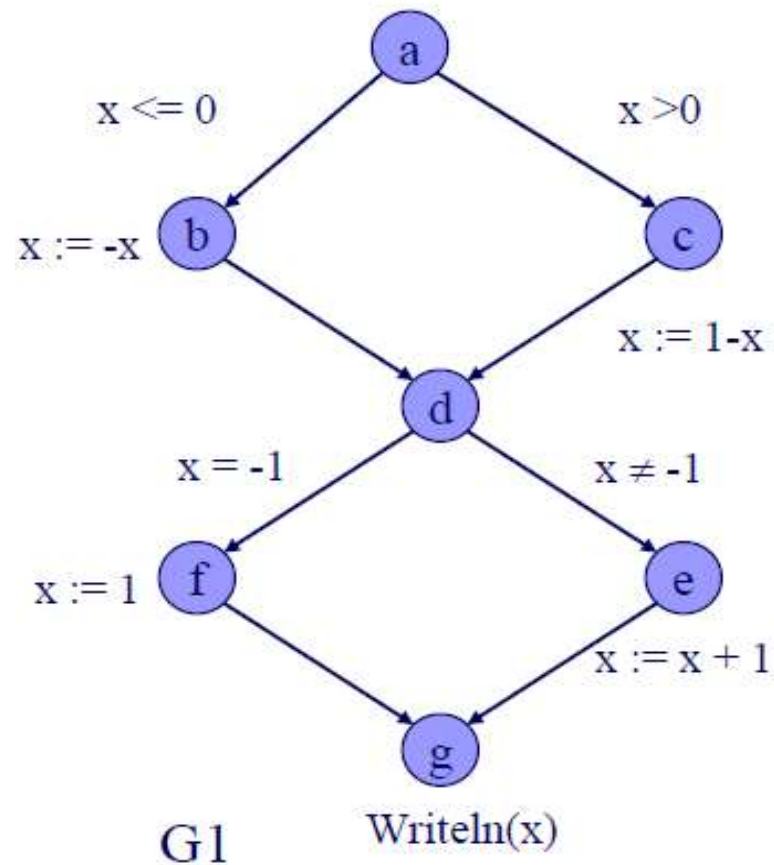
## Exemple1 :

Soit le programme **P1** suivant :

```
if  $x \leq 0$  then  $x := -x$   
else  $x := 1 - x$  ;  
if  $x = -1$  then  $x := 1$   
else  $x = x + 1$  ;
```

writeln(x) ;

Ce programme admet le graphe  
de contrôle **G1**



# Test structurel dynamique : Couverture du graphe de contrôle

## ➤ Chemins dans le graphe de contrôle

Le graphe G1 est un graphe de contrôle qui admet **une entrée (le noeud a), une sortie (le noeud g).**

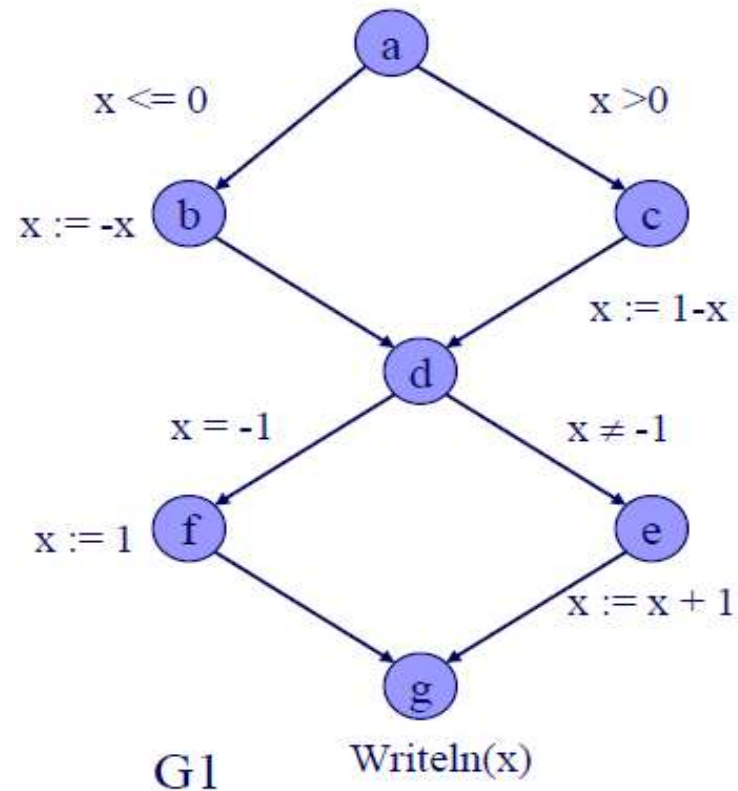
- ➔ Le chemin [a, c, d, e, g]  
est un chemin de contrôle,
- ➔ Le chemin [b, d, f, g]  
n'est pas un chemin de contrôle.
- ➔ Le graphe G1 comprend 4 chemins  
de contrôle :

$\beta_1 = [a, b, d, f, g]$

$\beta_2 = [a, b, d, e, g]$

$\beta_3 = [a, c, d, f, g]$

$\beta_4 = [a, c, d, e, g]$



# Test structurel dynamique : Couverture du graphe de contrôle

## ➤ Expression des chemins de contrôle

- Le graphe G1 peut être exprimé sous la forme algébrique suivante :

$$G1 = abdfg + abdeg + acdfg + acdeg$$

Le signe « + » désigne le « ou » logique entre chemins.

- Simplification de l'expression de chemins :

$$G1 = a ( bdf + bde + cdf + cde ) g$$

$$G1 = a ( df (b+c) + (b+c)de ) g$$

$$G1 = a ( (b+c) (de+df) ) g$$

$$G1 = a ( b + c ) d ( e + f ) g$$



# Test structurel dynamique : Couverture du graphe de contrôle

## ➤ Calcul de l'expression des chemins de contrôle

On associe une opération d'addition ou de multiplication à toutes les structures primitives apparaissant dans le graphe de flot de contrôle.

Nous adoptons les règles suivantes :

- **Pour une forme séquentielle :**

le produit **ab** exprime le fait que

Le nœud **a** est suivi du nœud **b**



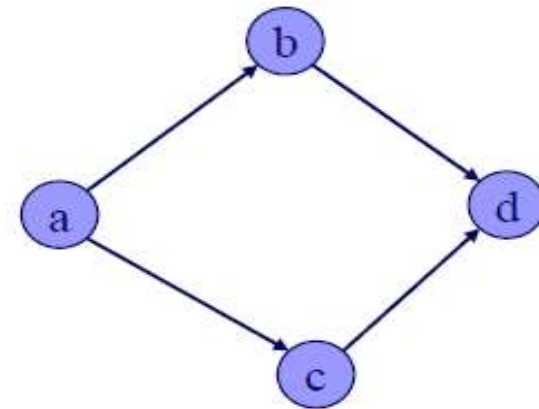
Forme séquentielle :  $ab$

- **Pour une forme alternative :**

L'expression **a(b+c)d** exprime le fait que

le nœud **a** est suivi du nœud **b** ou

du nœud **c**, tous deux suivis du nœud **d**.



Forme alternative :  
 $a(b + c)d$

# Test structurel dynamique : Couverture du graphe de contrôle

- Calcul de l'expression des chemins de contrôle

- Pour la forme itérative :

L'ensemble des chemins prend la forme :

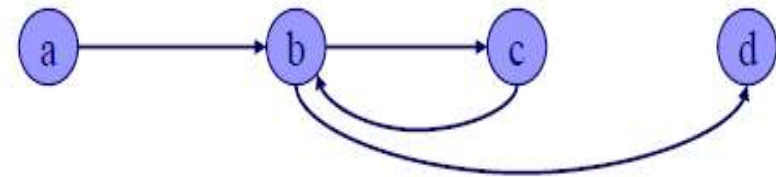
$$G = abd + abcbcd + abcbcbcd + \dots$$

En utilisant la notation :

$cbcb = (cb)^2$  et aussi  $cbcbcb = (cb)^3$ , etc

Nous obtenons :

$$G = ab(1 + cb + (cb)^2 + (cb)^3 + \dots) d$$



Forme itérative :  $ab (cb)^* d$

Ceci implique :  $G = ab(cb)^*d$  ;

Si la structure while est exécutée au plus 4 fois, l'expression devient :

$$G = ab(cb)^*d = ab(\underline{cb})^4 d \text{ (on emploie un exposant souligné)}$$

# Test structurel dynamique : Couverture du graphe de contrôle

## Chemins de contrôle avec boucles

### Exemple 2 :

Soit le programme **P2** suivant :

$i := 1$  ;

$found := false$  ;

while (not found) do

begin

if ( $a[i] = E$ ) then

begin

$found := true$  ;

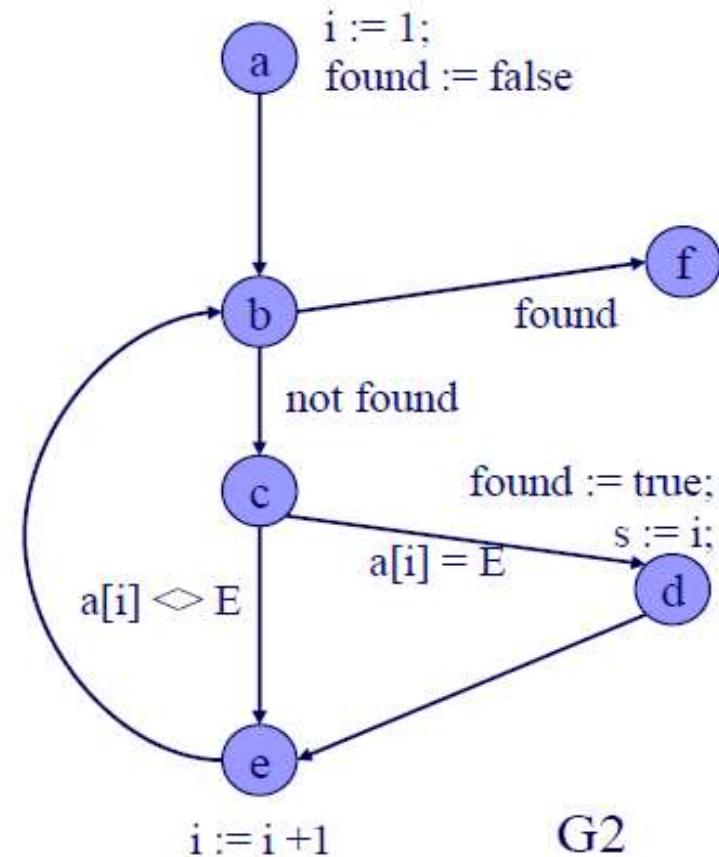
$s := i$  ;

end ;

$i := i+1$  ;

end ;

$G2 = ab [c (1 + d) eb]^* f$



# Test structurel dynamique : Couverture du graphe de contrôle

- Exemple 3 :

Soit le programme P3 suivant :

```
if  $n \leq 0$  then
```

```
     $n := 1 - n$ 
```

```
end ;
```

```
if  $2 \text{ div } n$  then
```

```
     $n := n/2$  ;
```

```
else
```

```
     $n := 3 * n + 1$  ;
```

```
end ;
```

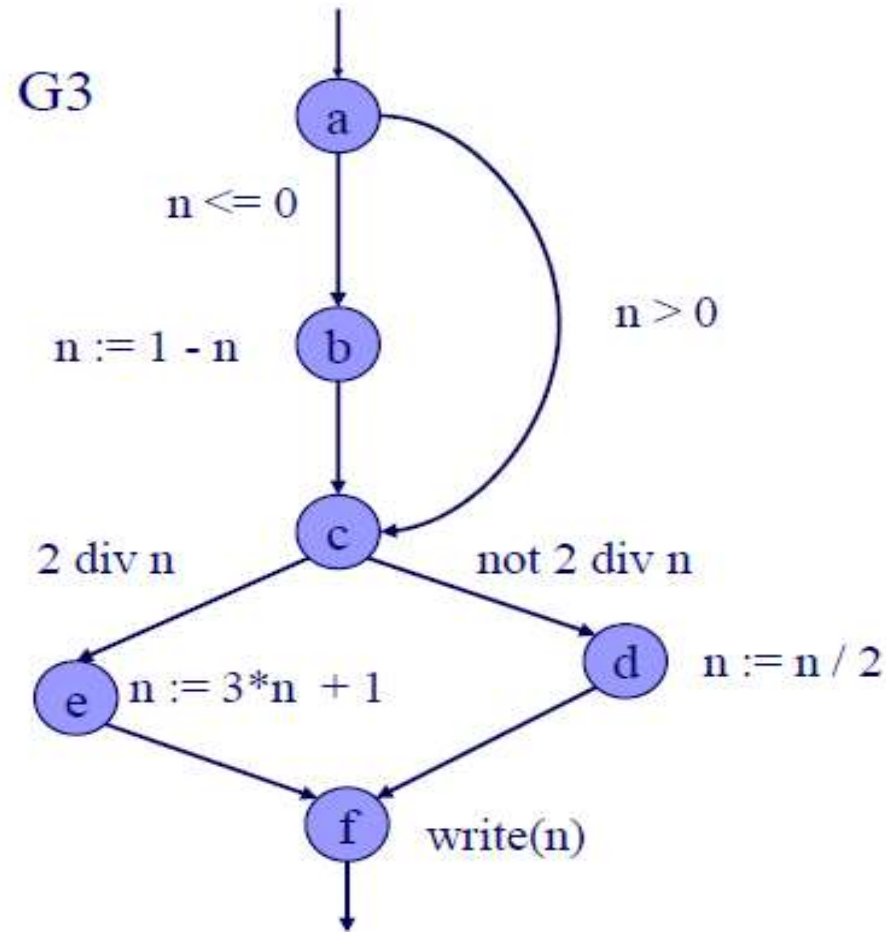
```
write( $n$ ) ;
```

1. Etablir le graphe de flot de contrôle de ce programme;
2. Fournir l'expression des chemins ;

# Test structurel dynamique : Couverture du graphe de contrôle

Réponse :

$G3 = a (1 + b) c (e + d) f$



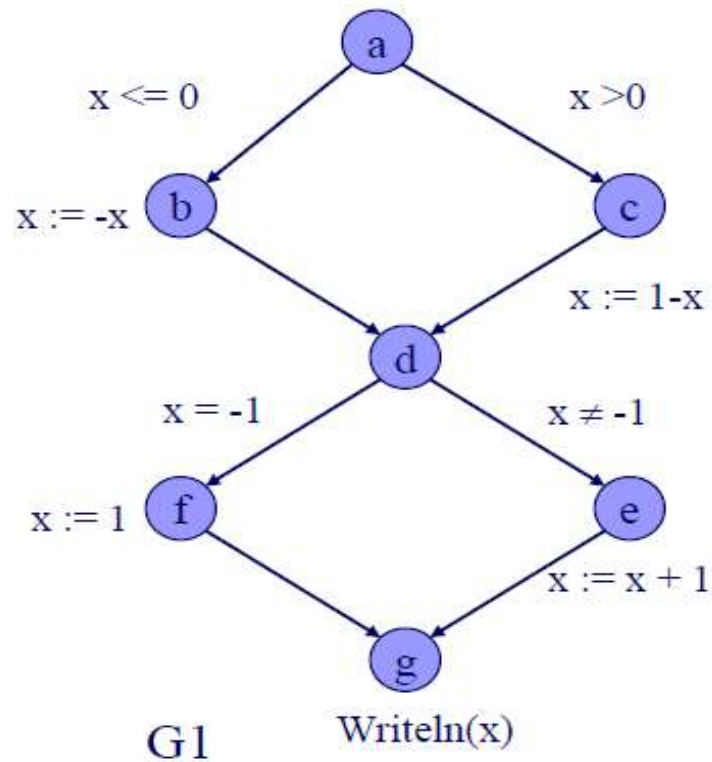
# Test structurel dynamique : Couverture du graphe de contrôle

## ➤ Chemins exécutables / non exécutables :

➤ Prenons le programme P1, donné précédemment, et son graphe G1.

```
if x ≤ 0 then x := -x  
else x := 1 - x ;  
if x = -1 then x := 1  
else x = x + 1 ;  
writeln(x) ;
```

**G1 = a ( b + c ) d ( e + f ) g**



# Test structurel dynamique : Couverture du graphe de contrôle

- Chemins exécutables / non exécutables :

- Soit  $DT1 = \{x=2\}$

Nous disons que cette donnée sensibilise le chemin  $\beta3 = [a, c, d, f, g]$  car pour cette valeur de  $x$  l'algorithme a suivi ce flot de contrôle en visant consécutivement ces noeuds

$\{x=2\}$  sensibilise  $\beta3 = [a, c, d, f, g]$

$\{x=0\}$  sensibilise  $\beta2 = [a, b, d, e, g]$

$\{x=1\}$  sensibilise  $\beta4 = [a, c, d, e, g]$

- $\beta3, \beta2, \beta4$  est un chemin **exécutable** ;

- $\beta1$  (  $\beta1 = [a, b, d, f, g]$  ) est un chemin **non exécutable** : aucune DT capable de sensibiliser ce chemin ;

**Remarque :** L' Existence de chemins non exécutables : signe de mauvais codage ;

# Test structurel dynamique : Couverture du graphe de contrôle

- **Calcul du nombre de chemins de contrôle G :**
  - Se déduit directement de l'expression des chemins de G.
  - Remplacer chaque nœud, dans l'expression des chemins, par l'unité (1) et exécuter les opérations d'une façon arithmétique :

## Exemples :

- Graphe G1 :

$$G1 = (a(b+c)d(e+f)g = 1.(1+1).1.(1+1).1 = 2.2 = 4$$

- Graphe G4 :

$$G4 = ab[c(1+d)eb]*f = 1.1[1.(1+1).1.1]^3.1 = 2^3 = 1+2+2^2+2^3 = 15$$

**Nb chemins de contrôle = Nb chemins exécutables + Nb chemins non exécutables**



# Test structurel dynamique : Couverture du graphe de contrôle

- **Problème des chemins non exécutables**
- Parfois le Nb chemins non exécutables peut être important :

## Exemple :

Soit le programme **P5** et son graphe de contrôle **G5** :

```
begin  
s:=0;  
for i:=1 to 1000 do  
    s:=s+a[i];  
end
```

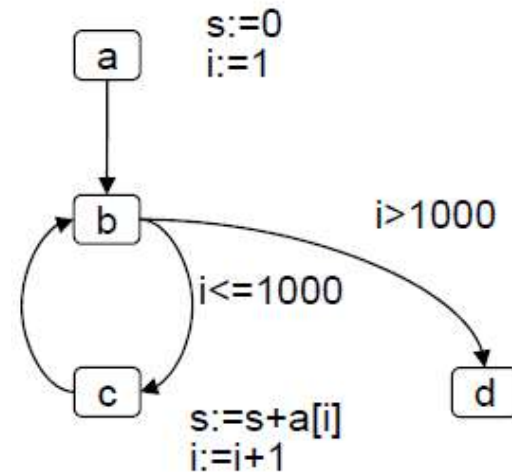
## Expression des chemins de G5 :

$G = a.b.(cb)^{1000}.d$

## Nombre de chemins :

$G = 1.1.(1.1)^{1000}.1 = 1 + 1^1 + 1^2 + \dots + 1^{1000} = 1001$

Parmi ces 1001 chemins **un seul chemin est exécutable** et a comme expression :  **$a.b.(cb)^{1000}.d$**



# Test structurel dynamique : Couverture du graphe de contrôle

## Satisfaction d'un test structurel avec couverture

Soit  $T$  un test structurel qui nécessite la couverture d'un ensemble de chemins  $\{\delta_1, \dots, \delta_k\}$  du graphe de contrôle.

**On note :**  $T = \{\delta_1, \dots, \delta_k\}$

Soit  $DT$  une donnée de test qui sensibilise le chemin de contrôle  $C$ .

**Définition:**  $DT$  satisfait  $T$  ssi  $C$  couvre tous les chemins de  $T$ .

**Exemple :** considérons le graphe de contrôle  $G_5$

Soient  $\delta_1 = cdebcde$  et  $\delta_2 = ce$  et  $T_1 = \{\delta_1, \delta_2\}$ .

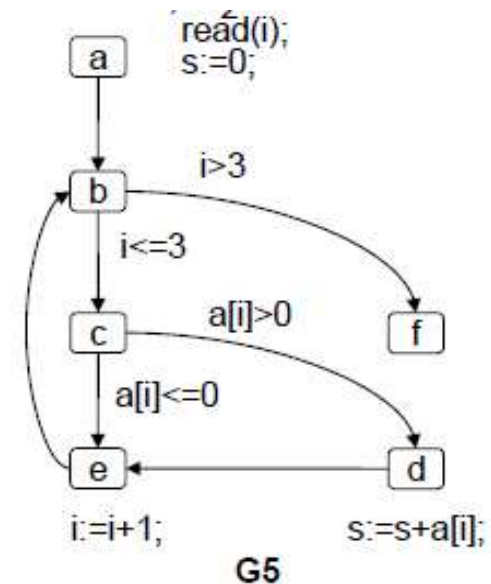
$DT_1 = \{a[1] = -2, a[2] = 3, a[3] = 17, i = 1\}$  satisfait-il  $T_1$  ?

$DT_1$  sensibilise  $M_1 = abcebcdebcdeb f$

$M_1 = abcebcdebcdeb f$  couvre  $\delta_1 = cdebcde$

$M_1 = abcebcdebcdeb f$  couvre  $\delta_2 = ce$

Donc  **$DT_1$  satisfait  $T_1$**



# Test structurel dynamique : Couverture du graphe de contrôle

**Deux catégories de critères de couverture :**

➤ **Critère de couverture «tous-les-noeuds»**

**But** : sensibiliser tous les chemins de contrôle qui nous permettent de visiter tous les **noeuds** du graphe de contrôle.

**Taux de couverture**: TER1 (Test Effectiveness Ratio1 ou C1)

$$\text{TER1} = |\{\text{nombre noeuds couverts}\}| / |\{\text{nombre noeuds}\}|$$

➤ **Critère de couverture «tous-les-arcs»**

Si on cherche à couvrir tous les noeuds sans couvrir tous les arcs, on risque de ne pas détecter certains défauts sur les arcs non couverts...

**But** : sensibiliser tous les chemins de contrôle qui nous permettent de visiter tous les **arcs** du graphe de contrôle.

$$\text{TER2} = |\{\text{nombre arcs couverts}\}| / |\{\text{nombre arcs}\}|$$

# Test structurel dynamique : Couverture du graphe de contrôle

**Application :** soit le programme P suivant :

```
begin
  read x ;
  read y ;
  z := 0 ;
  signe := 1 ;
  if x < 0
  then
    signe := -1 ;
    x := - x ;
  end if ;
  if y < 0
  then
    signe := - signe ;
    y := - y ;
  end if ;
  while x >= y
  do
    x := x - y ;
    z := z + 1 ;
  end while
  z := signe * z ;
end ;
```

- 1) Présenter Le graphe de contrôle du programme P
- 2) Donner des Dts satisfaisant les critères :
  - Tous les nœuds
  - Tous les arcs