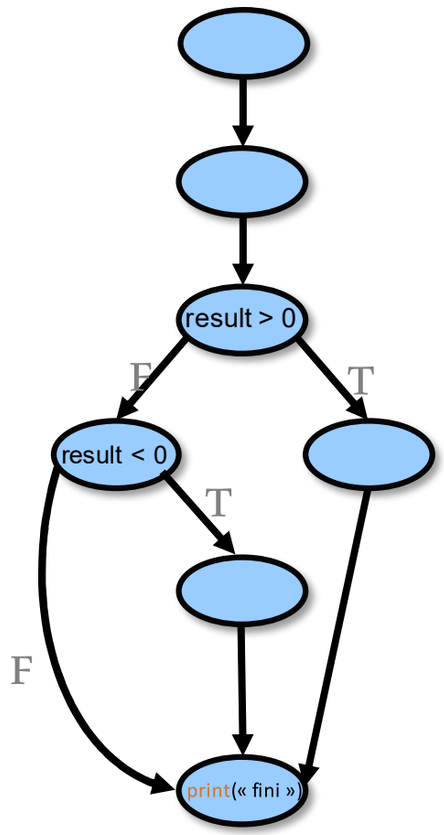
Tests en boîte blanche:
Graphe de flot de données

LOG3430 Méthodes de test et de validation du logiciel

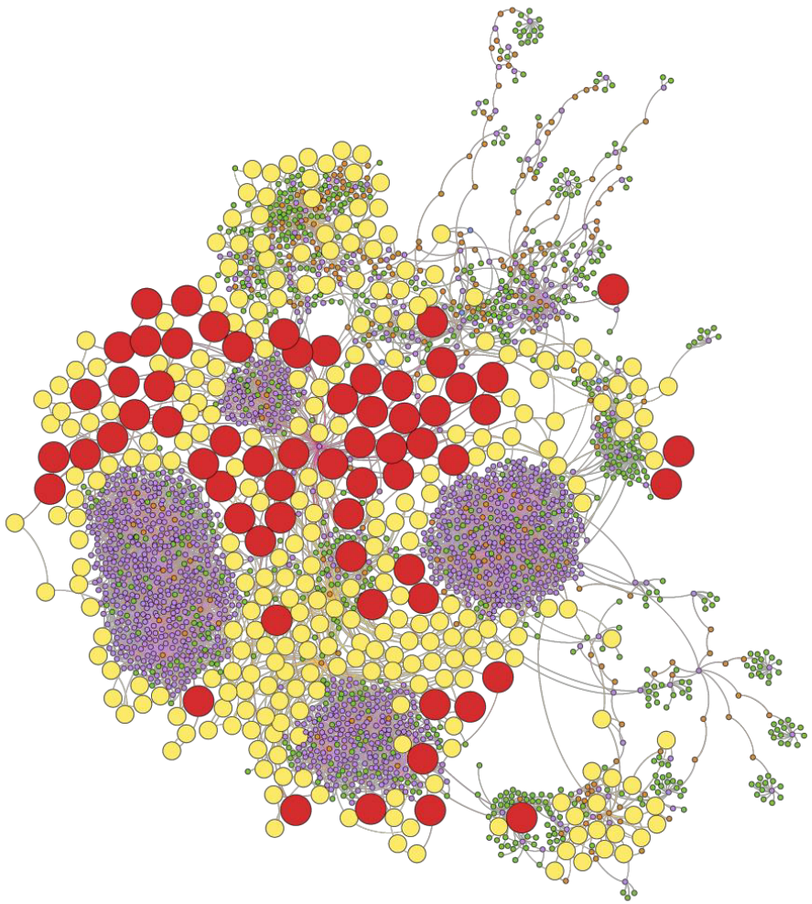
Les tests de flot de contrôle ne sont pas le seul moyen



Couverture de:

- Chemins
- Instructions (Nœuds)
- Branches
- Decisions

Les tests de flot de contrôle ne sont pas le seul moyen



Les tests de flot de données

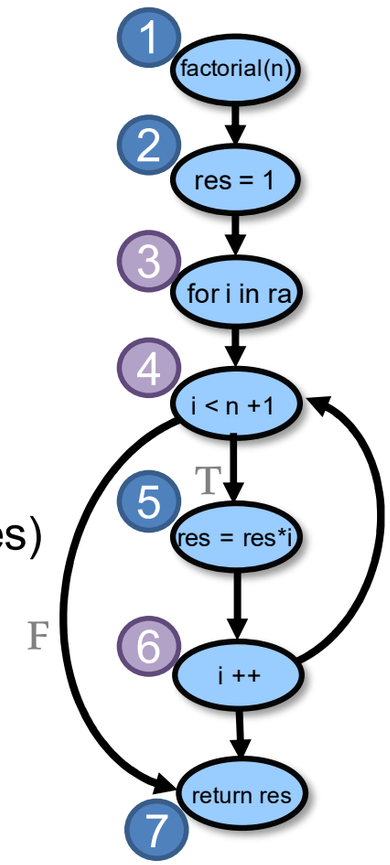
```
result = 10 + 5
```

```
print( result )
```

Graphe de flot de données

```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```

USE(res), USE(i), DEF(res)

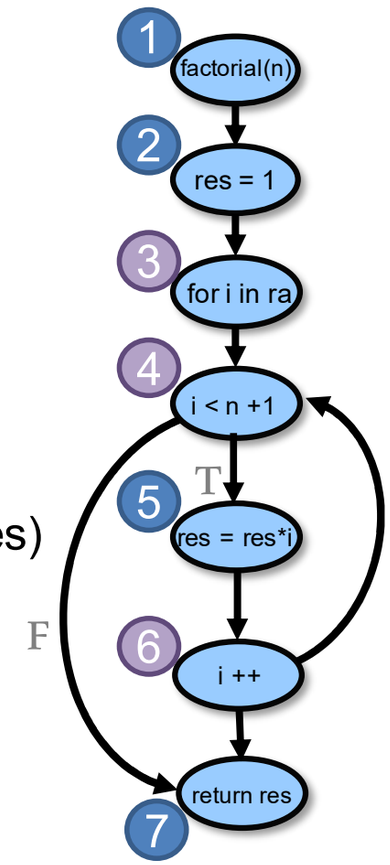


Nœud	DEF(i)	USE(i)
1	n	
2	res	
3	i	
4		i, n
5	res	res, i
6	i	i
7		res

Graphe de flot de données

```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```

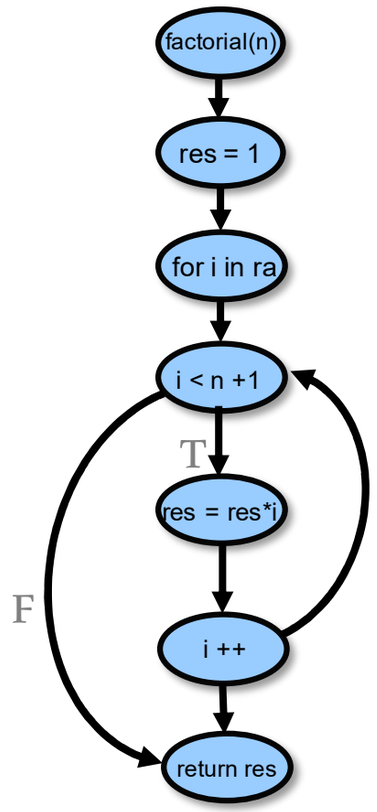
USE(res), USE(i), DEF(res)



Ligne	Nœud	DEF(i)	USE(i)
1	1	n	
2	2	res	
3a	3	i	
3b	4		i, n
4	5	res	res, i
3c	6	i	i
5	7		res

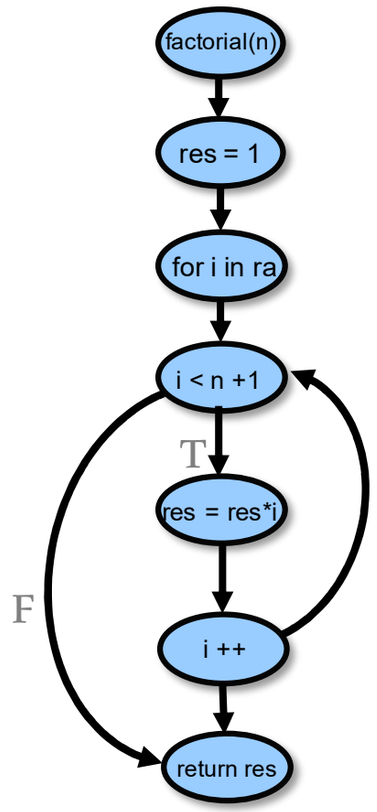
Affectation (DEF)

```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```



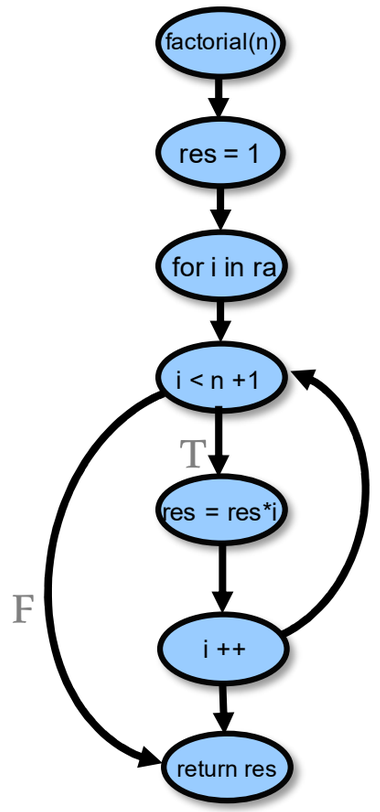
Affectation (DEF)

```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```



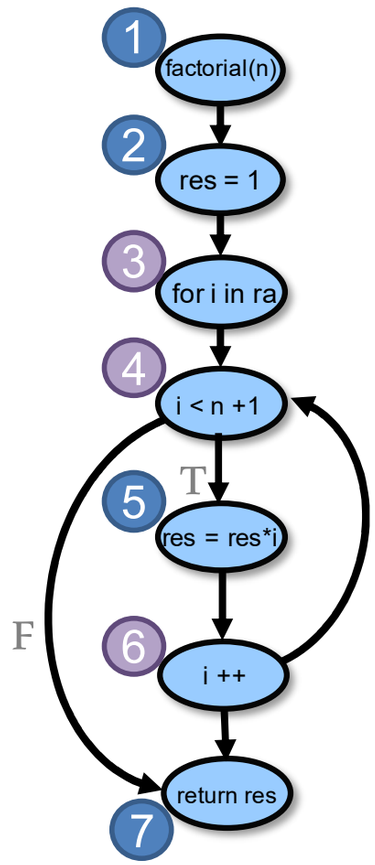
Utilisation (USE)

```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```



C-USE et P-USE

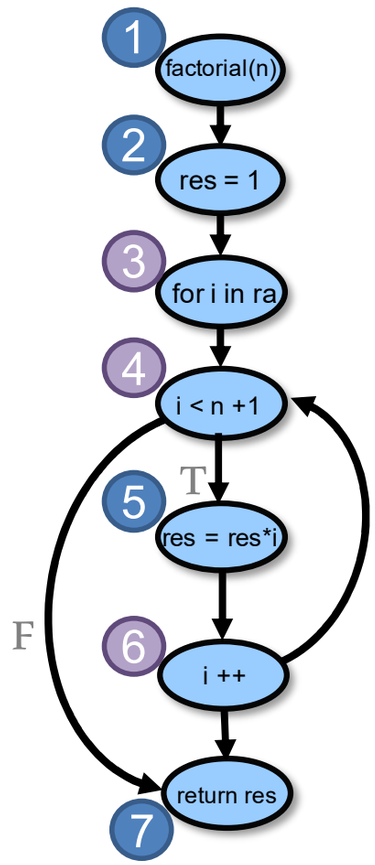
```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```



Nœud	DEF(i)	USE(i)
1	n	
2	res	
3	i	
4		i, n
5	res	res, i
6	i	i
7		res

C-USE et P-USE

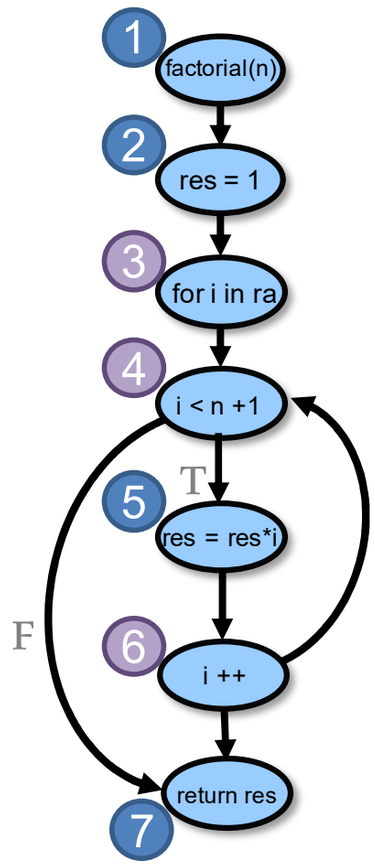
```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```



Nœud	DEF(i)	USE(i)	P-USE
1	n		
2	res		
3	i		
4		i, n	i,n
5	res	res, i	
6	i	i	
7		res	

C-USE et P-USE

```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```



Nœud	DEF(i)	USE(i)	P-USE	C-USE
1	n			
2	res			
3	i			
4		i, n	i,n	
5	res	res, i		res, i
6	i	i		i
7		res		res

KILL node

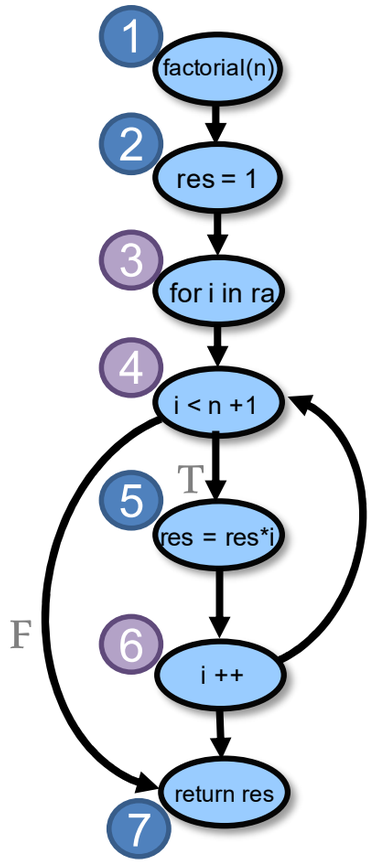
```
public int multByPointFive (int number) {
  double answer = 0;

  answer = number * 0.5;


  return answer;
}
```

Kill node pour answer

Les chemins DEF-USE

```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```

DC-PATH(n, 1, 4)



Definition-Clear DC-PATH(v, n, m) où

v est la variable d'intérêt

n est le noeud DEF(v, n) initial et

m est le noeud USE(v, m) final du chemin.

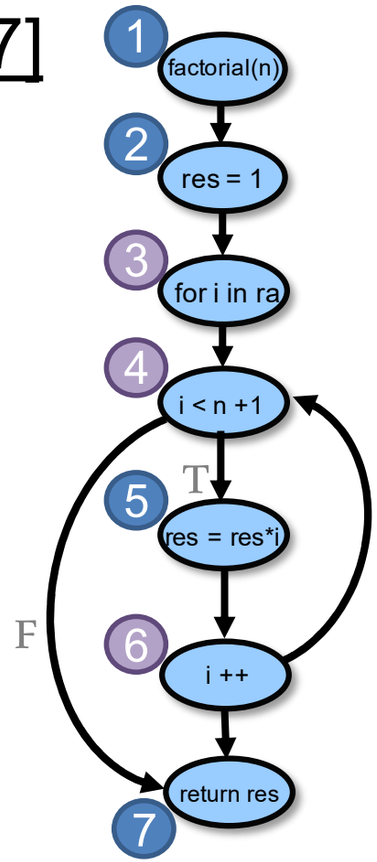
- Aucun autre noeud n dans le chemin est un noeud de définition pour la variable v.

Les chemins DEF-USE

```
def factorial (n):
    res = 1
    for i in range(1, n+1):
        res = res * i
    return res:
```

DU-PATH(res, 2, 7) [2,3,4,5,6,7]

DC-PATH(res, 2, 7) [2,3,4,7]



Definition-USE DU-PATH(v, n, m) où

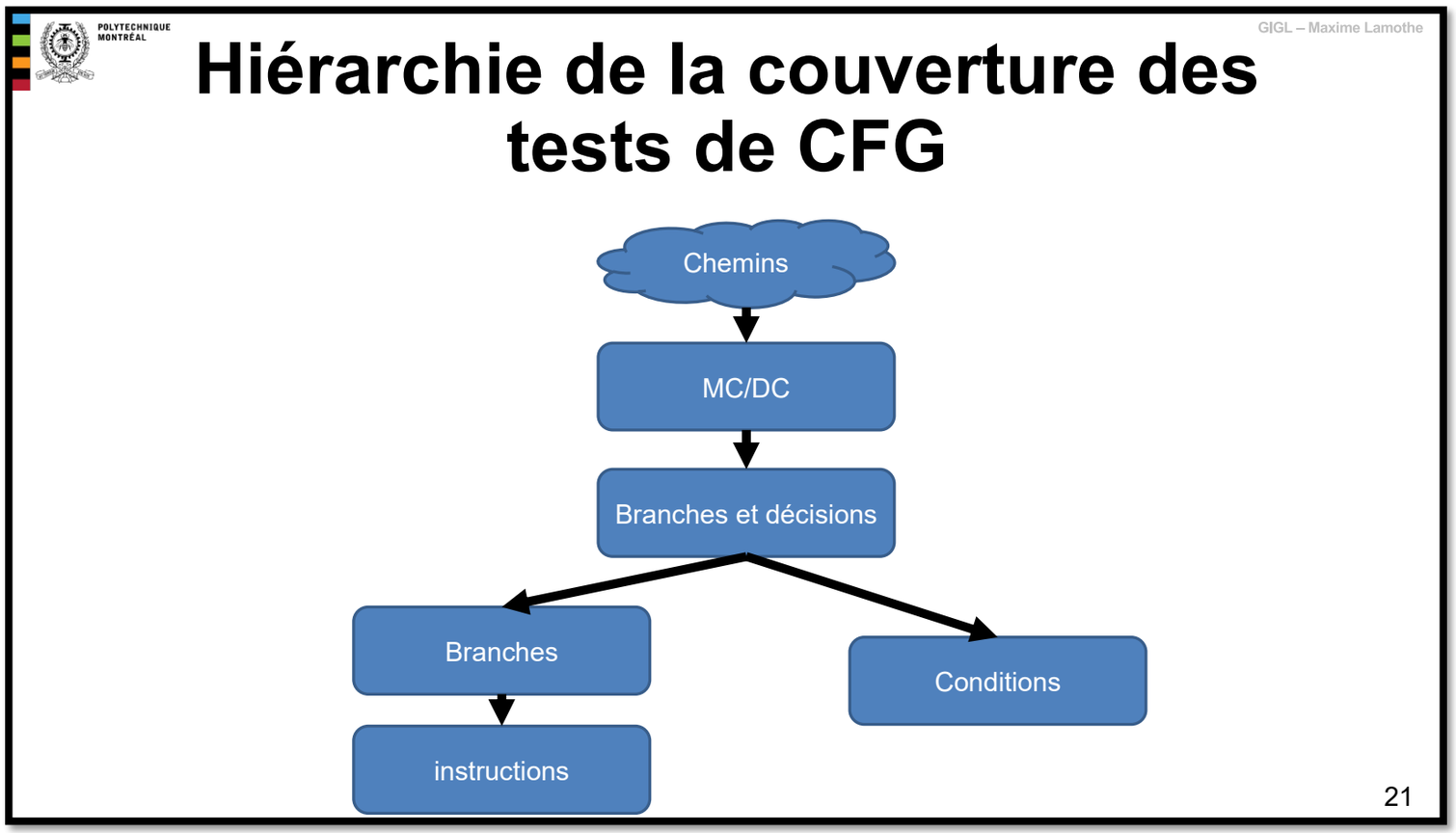
v est la variable d'intérêt

n est le noeud DEF(v, n) initial et

m est le noeud USE(v, m) final du chemin.

- Il *peut* avoir des re-definitions de la variable v dans le chemin.

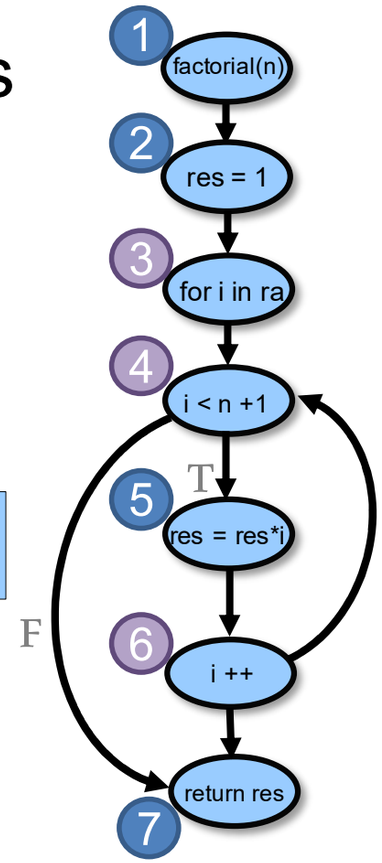
Les mesures de couvertures



All-Defs

Un jeu de tests T satisfait le critère « all-Defs » pour le programme P ssi, pour chaque variable v∈V, il y a au moins un chemin *definition-clear* pour chaque noeud de définition de v.

couverture de tous les noeuds DEF en utilisant des def-clear



All-Defs

couverture de tous les noeuds DEF en utilisant des def-clear

n:

DC-PATH(n, 1, 4): {1,2,3,4}

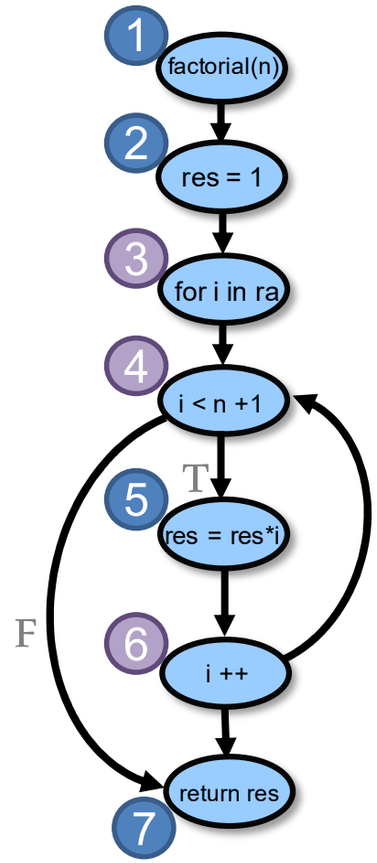
res:

DC-PATH(res, 2, 5): {2,3,4,5}
DC-PATH(res, 5, 7): {5,6,4,7}

i:

DC-PATH(i, 3, 5): {3,4,5}
DC-PATH(i, 6, 4}: {6,4}

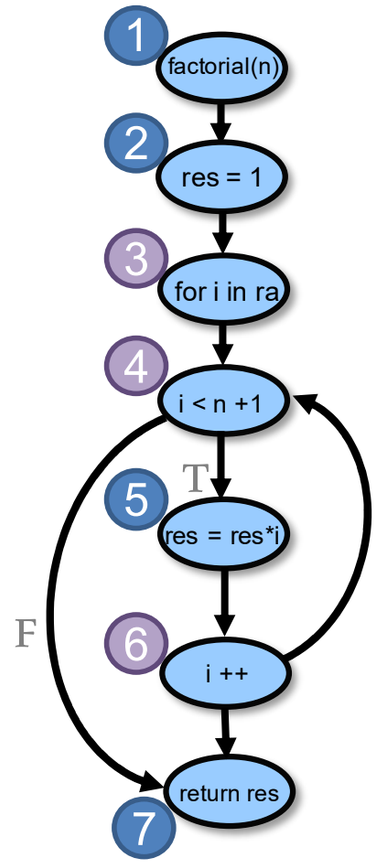
Nœud	DEF(i)	P-USE	C-USE
1	n		
2	res		
3	i		
4		i, n	
5	res		res, i
6	i		i
7			res



All-Uses

Un jeu de tests T satisfait le critère « all-Uses » pour le programme P ssi, pour chaque variable v∈V, il y a au moins un chemin *definition-clear* à partir de chaque noeud de définition de v et vers chaque noeud d'utilisation de v.

couverture de tous les nœuds DEF et tous les nœuds USE
en utilisant des def-clear



All-Uses

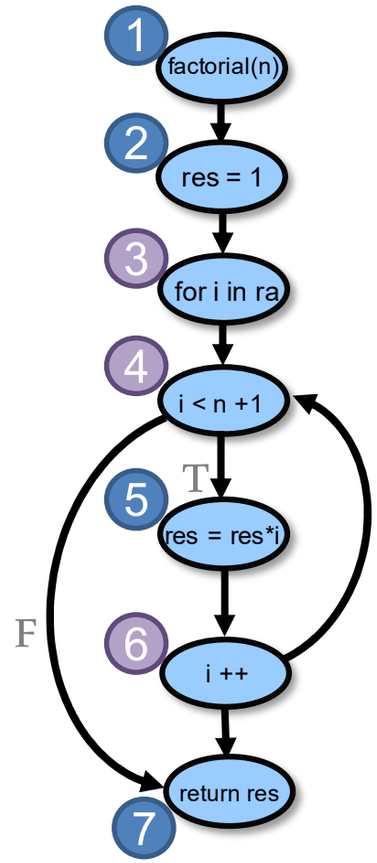
couverture de tous les nœuds DEF et tous les nœuds USE
en utilisant des def-clear

n:
DC-PATH(n, 1, 4): {1,2,3,4}

res:
DC-PATH(res, 2, 5): {2,3,4,5}
DC-PATH(res, 5, 7): {5,6,4,7}

i:
DC-PATH(i, 3, 5}: {3,4,5}
DC-PATH(i, 6, 4}: {6,4}
DC-PATH(i, 3, 6}: {3,4,5,6}

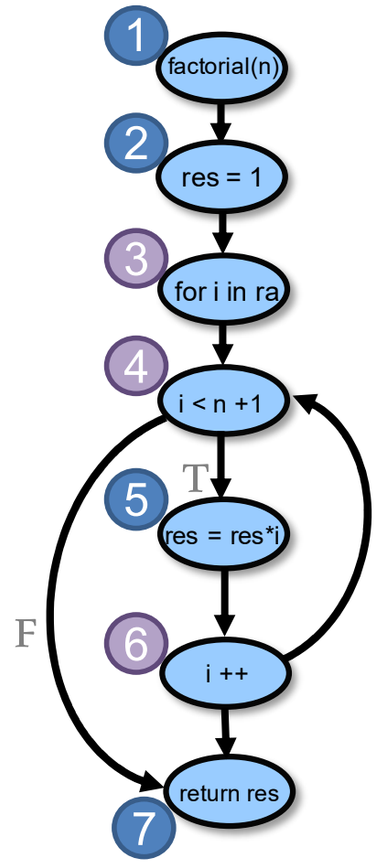
Nœud	DEF(i)	P-USE	C-USE
1	n		
2	res		
3	i		
4		i,n	
5	res		res, i
6	i		i
7			res



All-P-uses / Some-C-uses

Un jeu de tests T satisfait le critère « all-P-Uses / some-C-Uses » pour le programme P ssi, pour chaque variable v∈V, il y a au moins un chemin *definition-clear* :

- Partant de chaque noeud de définition de v à chaque utilisation-prédicat de v, ou
- Si une définition de v n'est pas utilisée dans un prédicat, il y a au moins un chemin *definition clear* vers une utilisation de calcul.

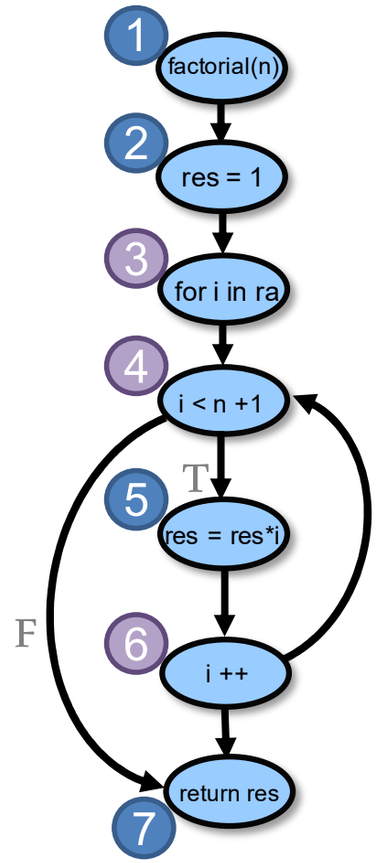


Critère : couverture de tous les noeuds DEF(v, n) et tous les noeuds P-USE(v ,n)
(et quelques C-USE)

All-C-uses / Some-P-uses

Un jeu de tests T satisfait le critère « all-C-Uses / some-P-Uses » pour le programme P ssi, pour chaque variable v∈V, il y a au moins un chemin *definition-clear* :

- Partant de chaque noeud de définition de v à chaque utilisation-calcul de v, ou
- Si une définition de v n'est pas utilisée dans un calcul, il y a au moins un chemin *definition clear* vers une utilisation de prédicat.

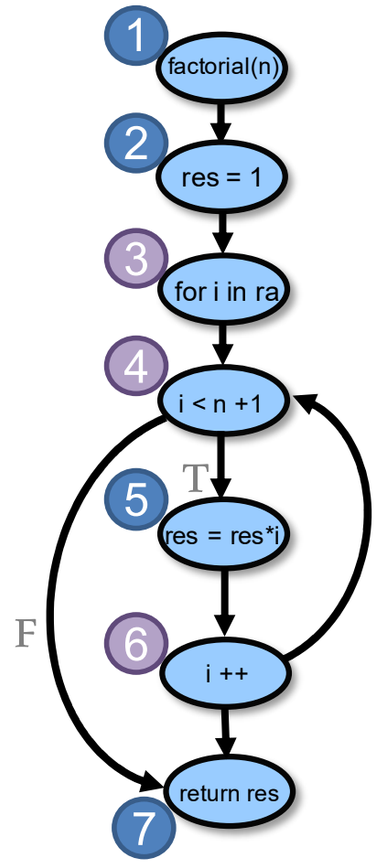


Critère : couverture de tous les noeuds DEF(v, n) et tous les noeuds C-USE(v ,n)
(et quelques P-USE)

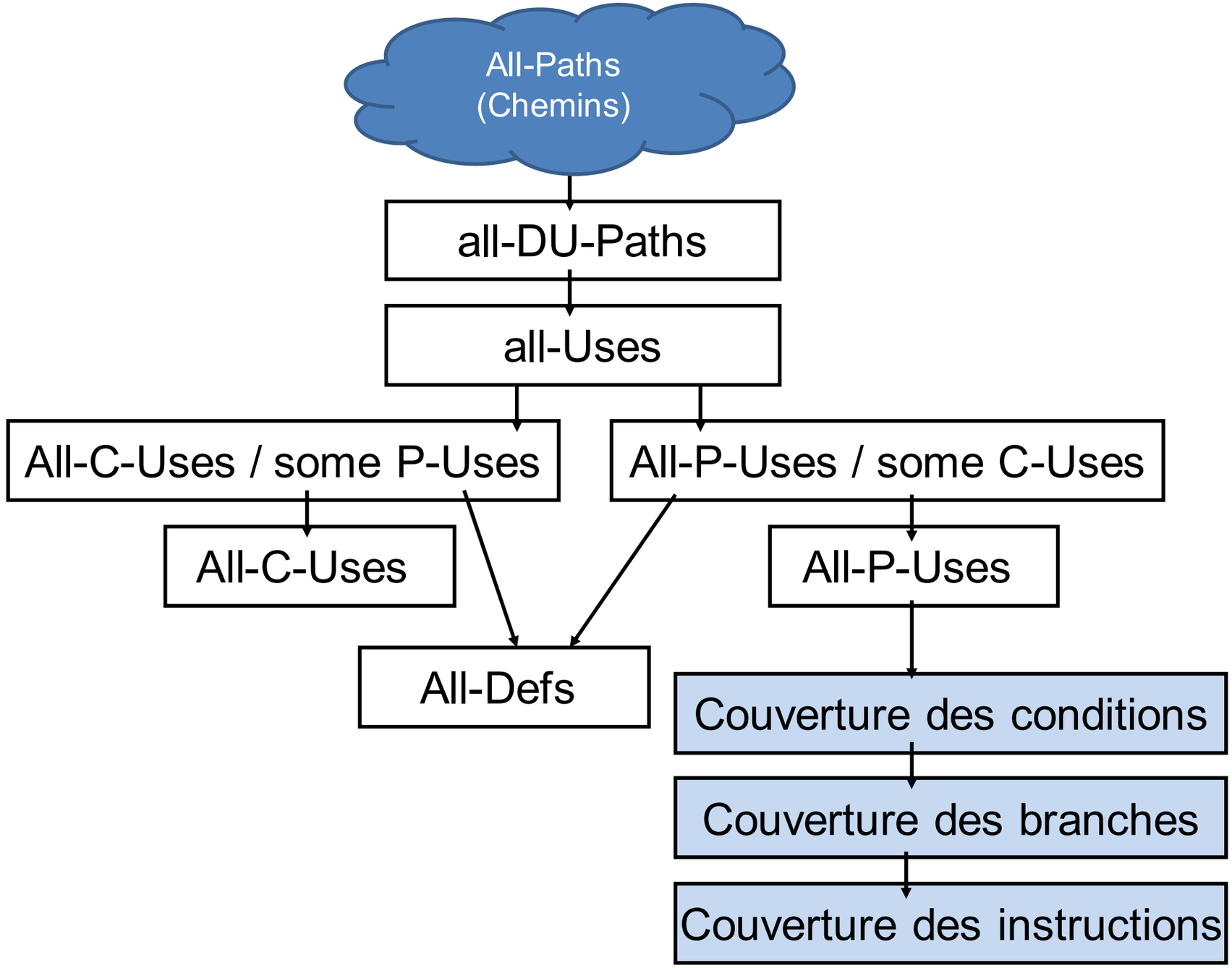
All-DU-Paths

Un jeu de tests T satisfait le critère « all-DU-Paths » pour le programme P ssi, pour chaque variable v∈V, on passe par **tous** les chemins *definition-clear* :

- Partant de chaque noeud de définition de v à chaque noeud d'utilisation de v atteignable, et
- Que ces chemins soit des traverses simples de boucles (i.e. une itération) ou qu'ils ne passent pas dans la boucle du tout (i.e. zéro itération).



Critère : couverture de toutes les paires possibles de *definition-use* (DU).


All-Paths
(Chemins)
all-DU-Paths
all-Uses
All-C-Uses / some P-Uses
All-P-Uses / some C-Uses
All-C-Uses
All-P-Uses
All-Defs
Couverture des conditions
Couverture des branches
Couverture des instructions

Vous êtes maintenant prêt à faire le quiz 2 sur Moodle