



Cours ASD : La Récursivité

Définition:

On appelle récursive toute fonction ou procédure qui s'appelle elle-même.

Algorithme Fact

Entrée : un entier positif N

Sortie : factorielle de N

si N = 0 retourner 1

sinon retourner N x Fact(N-1)

Comment ça marche ?

Appel à fact(4)

. 4*fact(3) = ?

. Appel à fact(3)

.. 3*fact(2) = ?

.. Appel à fact(2)

... 2*fact(1) = ?

... Appel à fact(1)

.... 1*fact(0) = ?

.... Appel à fact(0)

.... Retour de la valeur 1

.... 1*1

... Retour de la valeur 1

... 2*1

.. Retour de la valeur 2

.. 3*2

. Retour de la valeur 6

. 4*6

Retour de la valeur 24

Notion de pile d'exécution

La Pile d'exécution (call stack) du programme en cours est un emplacement mémoire destiné à mémoriser les paramètres, les variables locales ainsi que l'adresse de retour de chaque fonction en cours d'exécution.

les règles de conception :

Première règle:

Tout algorithme récursif doit distinguer plusieurs cas, dont l'un au moins ne doit pas comporter d'appel récursif. Souvent ce sont les cas les plus simples.

Sans cela, on risque de tourner en rond et de faire des exécutions qui ne se finissent pas.



Conditions d'arrêt:

Ces cas non récurifs sont appelés cas de base et les Conditions que doivent satisfaire les données dans ces cas de base sont appelées conditions d'arrêt ou de terminaison .

Seconde règle:

On doit conduire le programme vers les cas de base :

tout appel récursif doit se faire avec des données plus proches des conditions de terminaison!

Cette règle utilise le théorème suivant :

Théorème:

Il n'existe pas de suite infinie strictement décroissante d'entiers positifs ou nuls.

Ainsi, l'arrêt de l'exécution d'un programme récursif est garanti quand les deux règles sont appliquées!

Exemple:

```
procédure récursive(paramètres) :  
si TEST D'ARRÊT:  
instructions du point d'arrêt  
sinon  
instructions récursive(paramètres changés); // appel récursif  
instructions
```

Les différents types de récursivité:

1. **récursivité terminale**, la fonction se termine avec l'unique appel récursif.

$f(X) = \text{si cond alors } a(X) \text{ sinon } f(b(X)) \text{ fsi}$

2. **récursivité multiple** , si l'un des cas traités se résout avec plusieurs appels récursifs.

$f(n) = \text{si } n=0 \text{ ou } n=1 \text{ alors } 1 \text{ sinon } f(n-1) + f(n-2) \text{ fsi}$

3. **récursivité croisée ou mutuelle** , deux algorithmes sont mutuellement récursifs si l'un fait appel à l'autre et vice-versa. f et g récursives
 $f(g(X))$ et $g(f(Y))$

4. **récursivité imbriquée**, si la fonction contient comme paramètre un appel à elle-même
 $f(f(X))$