

	<b>Organigramme / Algorithme</b>	Dossier élève	1°SI
CI 10, II1 ; CI 11, II0 C24	<b>Algorithmique</b>	8 février 2009 (13:47)	

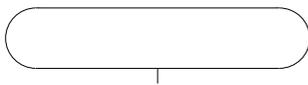
## 1. Introduction

Un organigramme (ou algorithme, lorsqu'il est plus particulièrement appliqué à l'algorithmique) est une représentation graphique normalisée<sup>1</sup>, utilisée pour analyser ou décoder un problème. Il est clair, mais peu adapté aux systèmes complexes.

Un algorithme considéré comme bon a les caractéristiques suivantes :

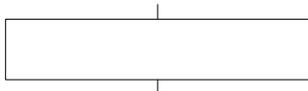
- il comporte des liaisons fléchées ;
- il est fermé ;
- il comporte un début et une fin<sup>2</sup>.

## 2. Symboles



### Début, fin, interruption

Le cas de l'interruption ne sera pas vu au sein de ce cours.



### Traitement

Opération ou groupe d'opérations sur des données, etc.



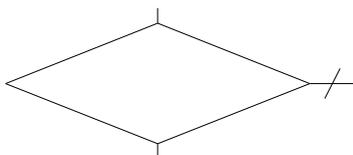
### Entrée / Sortie

Mise à disposition d'une information à traiter ou enregistrement d'une information traitée. Pour simplifier la première approche, nous considérerons dans un premier temps que cet élément est utilisé lors d'une entrée ou sortie du système étudié. Par exemple, un capteur fait partie du système, mais la grandeur captée en est extérieure.



### Sous-programme

Portion de programme considérée comme une simple opération.



### Embranchement (choix)

Test, question qui implique un choix.

Le sens général de lecture est de haut en bas, et de gauche à droite. Si ce n'est pas le cas, on flèche la liaison.

<sup>1</sup> NF Z 67-010 et ISO 5807.

<sup>2</sup> Ce point est discutable car les systèmes embarqués, par exemple, n'ont pas forcément de fin.

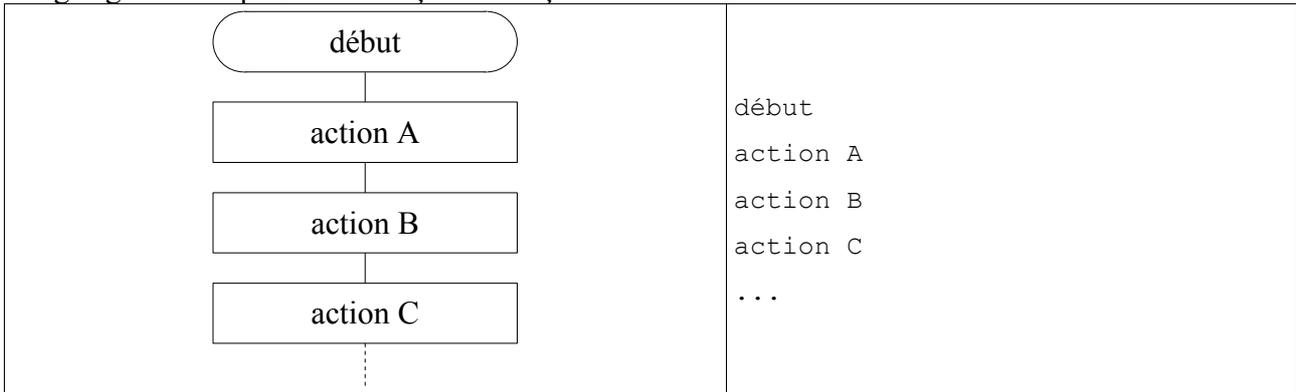
### 3. Relation entre algorithme et pseudo-code

Un algorithme est utilisé à dessein de programmer un système, généralement géré par microprocesseur ou microcontrôleur. C'est en effet une phase qui permet de décrire un comportement, et d'envisager les réactions d'un système. Il y a donc un lien direct entre un algorithme et un programme (écrit en Pascal, C, Basic, assembleur, etc.)

Nous n'allons pas étudier de langage de programmation spécifique, et nous nous référerons à un pseudo-langage simple, nommé « pseudo-code », composé de « pseudo-instructions ». Ce dernier permet en effet de comprendre les structures existantes, et il est simple de les adapter à un quelconque langage.

#### 3.1. Structure linéaire

Elle se caractérise par une suite d'actions à exécuter successivement, dans un ordre défini. La structure linéaire n'est pas une façon de programmer en soi ; il s'agit plutôt de parties d'algorithmes qui seront conçues de façon linéaire.

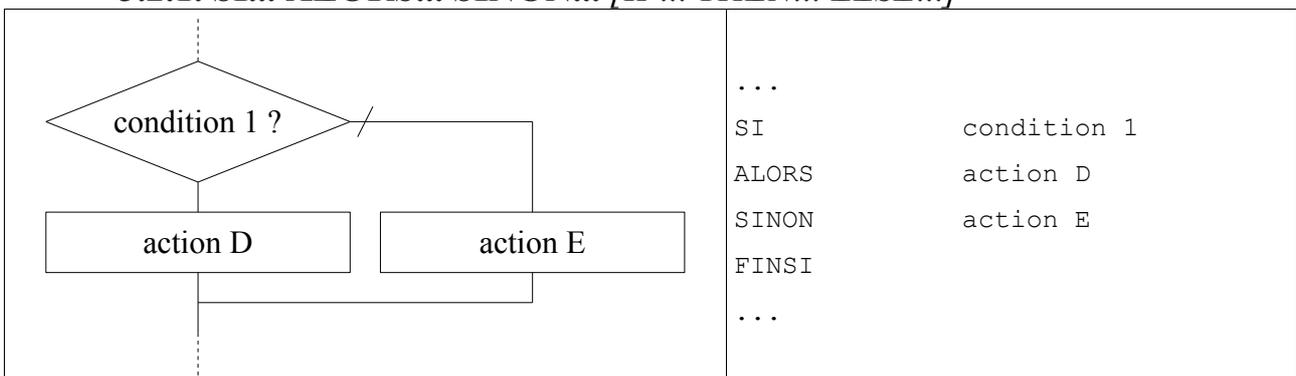


#### 3.2. Structure alternative

Elle propose un choix ; une condition va entraîner telle ou telle réaction du système. Les deux issues s'excluent mutuellement.

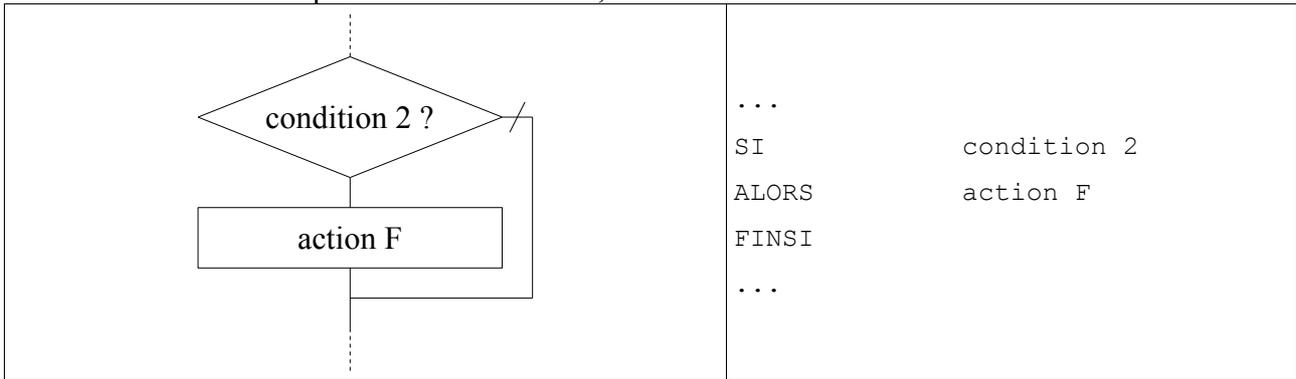
Cette structure peut intégrer des portions de structures linéaires.

##### 3.2.1. SI... ALORS... SINON... [IF... THEN... ELSE...]



### 3.2.2. SI... ALORS... [IF... THEN...]

Il s'agit de la même structure que précédemment, mais la partie « SINON... » est omise ; il en résulte un choix non pas entre deux actions, mais entre l'exécution d'une action ou non.



### 3.3. Structure itérative

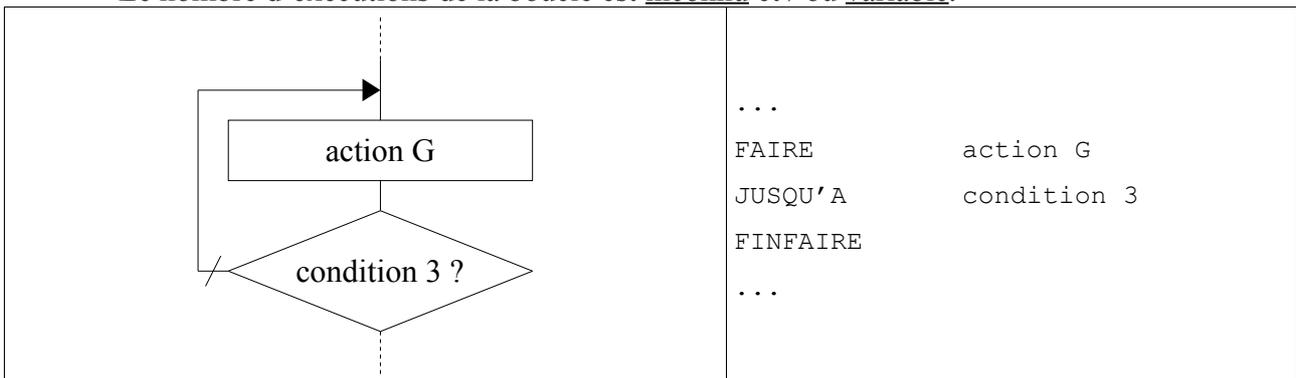
La structure itérative, aussi appelée structure répétitive, permet par exemple de répéter une action jusqu'à ce qu'une condition soit validée.

Elle intègre *de facto* une structure alternative, et éventuellement des structures linéaires.

#### 3.3.1. FAIRE... JUSQU'A... [DO... UNTIL...]

L'action est exécutée au moins une fois.

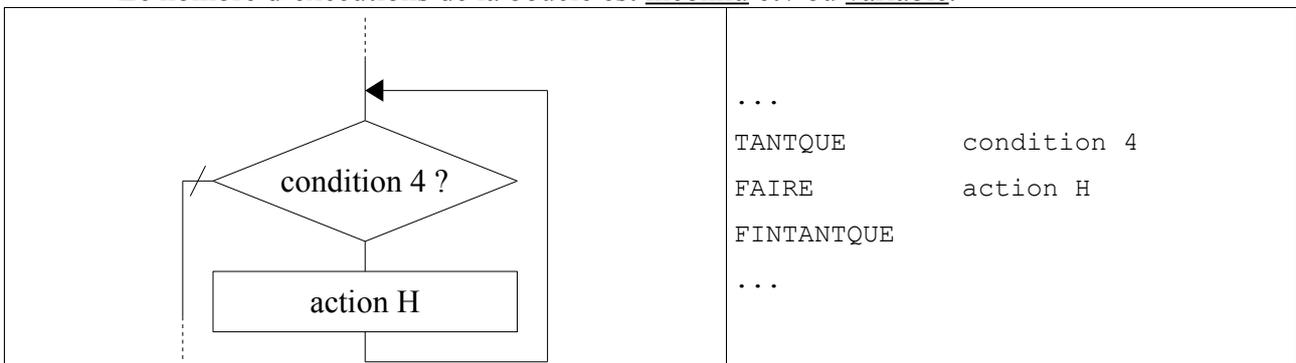
Le nombre d'exécutions de la boucle est inconnu et / ou variable.



#### 3.3.2. TANTQUE... FAIRE... [WHILE... DO...]

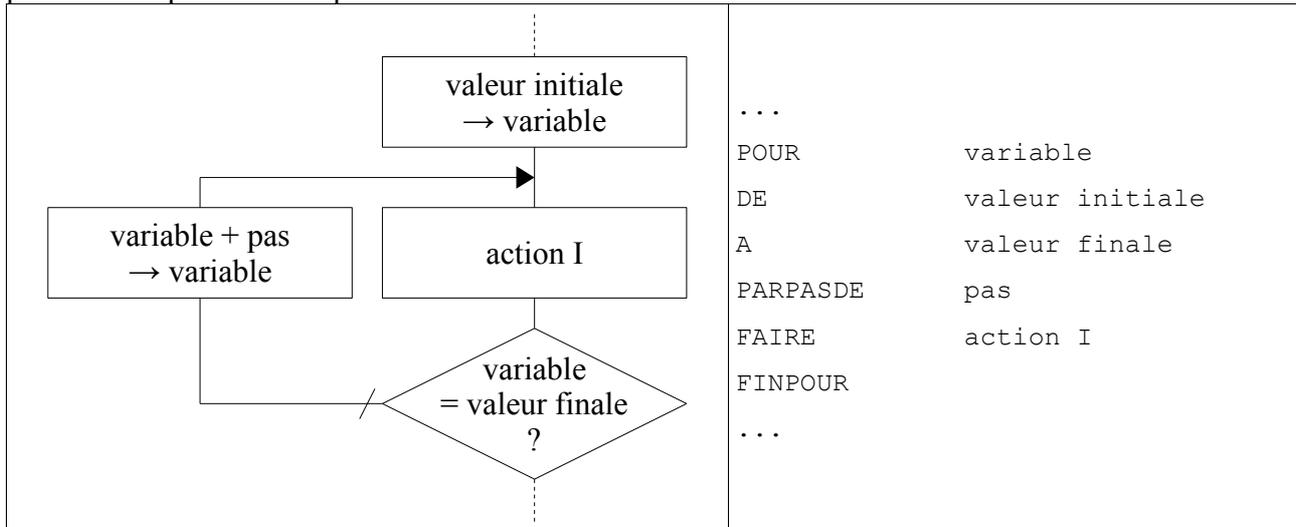
L'action n'est pas forcément exécutée.

Le nombre d'exécutions de la boucle est inconnu et / ou variable.



### 3.3.3. POUR... FAIRE... [FOR... DO...]

Le nombre d'exécutions de la boucle est connu. Il peut être variable dans certains cas particuliers qui ne seront pas vus ici.



## 4. Exercices

### 4.1. Choix d'une résistance

Soit un système dont la matière d'œuvre est un flot de résistances de valeur 1 kΩ.

La tolérance des résistances est de +/- 10 %.

La partie du système étudiée doit : détecter la présence d'une résistance (le composant), mesurer sa valeur exacte en ohms, et vérifier que cette valeur est bien comprise dans la plage des +/- 10 %.

- Quelles sont les valeurs minimale et maximale admises par le système ?
- Proposer l'algorithme de gestion du système, et le pseudo-code correspondant.
- Quelle est la principale structure utilisée ?

### 4.2. Montée en température d'une enceinte

Une enceinte close doit être portée à une température de 65 °C. Nous n'allons voir que la partie de l'algorithme qui réalise cette montée en température, et non le système régulation intégral ; une fois cette température atteinte, le traitement s'arrête.

- Proposer l'algorithme de gestion du système, et le pseudo-code correspondant.
- Quelle est la principale structure utilisée ?

### 4.3. Remplissage d'une caisse

On veut compter le passage de bouteilles sur un tapis. Lorsque six bouteilles sont passées, l'arrivée de nouvelles bouteilles est stoppée (un système annexe placera les six bouteilles dans une caisse, mais cette gestion ne sera pas étudiée ici).

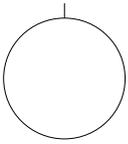
- Proposer l'algorithme de gestion du système, et le pseudo-code correspondant.
- Quelle est la principale structure utilisée ?

### 4.4. Gestion d'un ascenseur

Une personne entre dans un ascenseur, et appuie sur le bouton-poussoir d'un étage. Proposer l'algorithme simplifié de gestion de l'ensemble : début lors de la demande d'un étage et fin lorsque l'étage désiré est atteint.

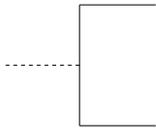
– *Partie enseignant* –

## 5. Symboles annexes



### Renvoi

Permet d'améliorer la lisibilité d'un algorithme. Il y a forcément un nombre pair de renvois au sein d'un organigramme.



### Commentaire

Permet d'expliciter un comportement grâce à un commentaire.

## 6. Correction des exercices

### 6.1. Choix d'une résistance

```
début
détecter la présence d'une résistance
mesurer la valeur ohmique de la résistance → R
SI      R > 1100 Ω
ALORS  jeter la résistance
SINON  SI      R < 900 Ω
        ALORS  jeter la résistance
        SINON  conserver la résistance
        FINSI
FINSI
fin
```

(Structure alternative)

### 6.2. Montée en température d'une enceinte

*première solution :*

```
début
TANTQUE T < 65 °C
FAIRE   chauffer
FINTANTQUE
fin
```

*deuxième solution (plus exacte) :*

```
début
contrôler la température → T
TANTQUE T < 65 °C
FAIRE   chauffer
        contrôler la température → T
FINTANTQUE
fin
```

*Note : Une structure à base de FAIRE... JUSQU'A... est aussi utilisable (et est même plus simple), mais cela implique une mise en marche du système de chauffe, même si la température initiale est supérieure ou égale à 65 °C.*

(Structures alternatives)

### 6.3. Remplissage d'une caisse

*première solution :*

```
début
POUR      Nb
DE        1
A         6
PARPASDE  1
FAIRE     détecter bouteille
FINPOUR
stopper l'arrivée des bouteilles
fin
```

*deuxième solution (plus courante) :*

```
début
POUR      Nb
DE        6
A         1
PARPASDE -1
FAIRE     détecter bouteille
FINPOUR
Stopper l'arrivée des bouteilles
fin
```

*Note : On n'incrémente pas (ou ne décrémente pas) la variable Nb ; cette opération est implicitement réalisée par la structure elle-même.*

(Structures itératives)