

Préambule : le Codage

Introduction à l'algorithmique

1. Les Variables

2. Lecture et Ecriture

3. Les Tests

4. Encore de la Logique

Faut-il mettre un Et ? un OU ?

Au delà de la logique : le style

5. Les Boucles

6. Les Tableaux

7. Techniques Rusées

8. Tableaux Multidimensionnels

9. Fonctions Prédéfinies

10. Fichiers

11. Procédures et Fonctions

12. Notions Complémentaires

Liens

Souvent Posées Questions

Rappel : ce cours d'algorithmique et de programmation
est enseigné à l'Université Paris 7,
dans la spécialité PISE du Master MECI (ancien DESS AIGES)
par Christophe Darmangeat

PARTIE 4

ENCORE DE LA LOGIQUE

« La programmation peut être un plaisir ; de même que la cryptographie. Toutefois, il faut éviter de combiner les deux. » - Kreitzberg et Sneidermann

1. Faut-il mettre un ET ? Faut-il mettre un OU ?

Une remarque pour commencer : dans le cas de conditions composées, les parenthèses jouent un rôle fondamental.

```
Variables A, B, C, D, E en Booléen
Variable X en Entier
Début
Lire X
A ← X > 12
B ← X > 2
C ← X < 6
D ← (A ET B) OU C
E ← A ET (B OU C)
Ecrire D, E
Fin
```

Si $X = 3$, alors on remarque que D sera VRAI alors que E sera FAUX.

S'il n'y a dans une condition que des ET, ou que des OU, en revanche, les parenthèses ne changent strictement rien.

Dans une condition composée employant à la fois des opérateurs ET et des opérateurs OU, la présence de parenthèses possède une influence sur le résultat, tout comme dans le cas d'une expression numérique comportant des multiplications et des additions.

On en arrive à une autre propriété des ET et des OU, bien plus intéressante.

Spontanément, on pense souvent que ET et OU s'excluent mutuellement, au sens où un problème donné s'exprime soit avec un ET, soit avec un OU. Pourtant, ce n'est pas si évident.

Quand faut-il ouvrir la fenêtre de la salle ? Uniquement si les conditions l'imposent, à savoir :

```
Si il fait trop chaud ET il ne pleut pas Alors
    Ouvrir la fenêtre
Sinon
    Fermer la fenêtre
Finsi
```

Cette petite règle pourrait tout aussi bien être formulée comme suit :

```
Si il ne fait pas trop chaud OU il pleut Alors
    Fermer la fenêtre
Sinon
    Ouvrir la fenêtre
Finsi
```

Ces deux formulations sont strictement équivalentes. Ce qui nous amène à la conclusion suivante :

Toute structure de test requérant une condition composée faisant intervenir l'opérateur ET peut être exprimée de manière équivalente avec un opérateur OU, et réciproquement.

Ceci est moins surprenant qu'il n'y paraît au premier abord. Jetez pour vous en convaincre un œil sur les tables de vérité, et vous noterez la symétrie entre celle du ET et celle du OU. Dans les deux tables, il y a trois cas sur quatre qui mènent à un résultat, et un sur quatre qui mène au résultat inverse. Alors, rien d'étonnant à ce qu'une situation qui s'exprime avec une des tables (un des opérateurs logiques) puisse tout aussi bien être exprimée avec l'autre table (l'autre opérateur logique). Toute l'astuce consiste à savoir effectuer correctement ce passage.

Bien sûr, on ne peut pas se contenter de remplacer purement et simplement les ET par des OU ; ce serait un peu facile. La règle d'équivalence est la suivante (on peut la vérifier sur l'exemple de la fenêtre) :

Si A ET B Alors

Instructions 1

Si non

Instructions 2

Finsi

équivalent à :

Si NON A OU NON B Alors

Instructions 2

Si non

Instructions 1

Finsi

Cette règle porte le nom de **transformation de Morgan**, du nom du mathématicien anglais qui l'a formulée.

[Exercice 4.1](#)

[Exercice 4.2](#)

[Exercice 4.3](#)

[Exercice 4.4](#)

[Exercice 4.5](#)



[Retour Haut de Page](#)

2. Au-delà de la logique : le style

Ce titre un peu provocateur (mais néanmoins justifié) a pour but d'attirer maintenant votre attention sur un fait fondamental en algorithmique, fait que plusieurs remarques précédentes ont déjà dû vous faire soupçonner : il n'y a jamais une seule manière juste de traiter les structures alternatives. Et plus généralement, il n'y a jamais une seule manière juste de traiter un problème. Entre les différentes possibilités, qui ne sont parfois pas meilleures les unes que les autres, le choix est une affaire de **style**.

C'est pour cela qu'avec l'habitude, on reconnaît le style d'un programmeur aussi sûrement que s'il s'agissait de style littéraire.

Reprenons nos opérateurs de comparaison maintenant familiers, le ET et le OU. En fait, on s'aperçoit que l'on pourrait tout à fait s'en passer ! Par exemple, pour reprendre l'exemple de la fenêtre de la salle :

Si il fait trop chaud ET il ne pleut pas Alors

Ouvrir la fenêtre

Si non

Fermer la fenêtre

Finsi

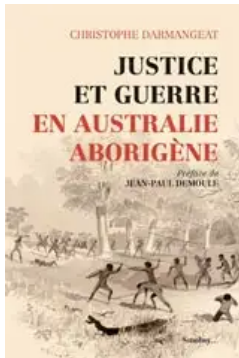
Possède un parfait équivalent algorithmique sous la forme de :

Très loin de l'informatique, pas tout près de l'économie :



Mon blog, [la Hutte des Classes](#)
À propos d'anthropologie sociale, de préhistoire et de marxisme.

Et mes livres...



```
Si il fait trop chaud Alors  
  Si il ne pleut pas Alors  
    Ouvrir la fenêtre  
  Sinon  
    Fermer la fenêtre  
  Finsi  
Sinon  
  Fermer la fenêtre  
Finsi
```

Dans cette dernière formulation, nous n'avons plus recours à une condition composée (mais au prix d'un test imbriqué supplémentaire)

Et comme tout ce qui s'exprime par un ET peut aussi être exprimé par un OU, nous en concluons que le OU peut également être remplacé par un test imbriqué supplémentaire. On peut ainsi poser cette règle stylistique générale :

Dans une structure alternative complexe, les conditions composées, l'imbrication des structures de tests et l'emploi des variables booléennes ouvrent la possibilité de choix stylistiques différents. L'alourdissement des conditions allège les structures de tests et le nombre des booléens nécessaires ; l'emploi de booléens supplémentaires permet d'alléger les conditions et les structures de tests, et ainsi de suite.

[Exercice 4.6](#)

[Exercice 4.7](#)

[Exercice 4.8](#)

Si vous avez compris ce qui précède, et que l'exercice de la date ne vous pose plus aucun problème, alors vous savez tout ce qu'il y a à savoir sur les tests pour affronter n'importe quelle situation. Non, ce n'est pas de la démagogie !

Malheureusement, nous ne sommes pas tout à fait au bout de nos peines ; il reste une dernière structure logique à examiner, et pas des moindres...



[Retour Haut de Page](#)