ALGORITHMIQUE ET PROGRAMMATION INFORMATIQUE

Chapitre 5 : La logique propositionnelle

I. Définition et utilité :

La logique est utile au programmeur en ce sens que :

- La logique est une façon de formaliser notre raisonnement ;
- Il n'y a pas une logique mais DES logiques ;
- La **logique propositionnelle** est un modèle mathématique qui nous permet de raisonner sur la nature vraie ou fausse des expressions logiques.

II. Proposition

C'est une **expression** qui peut prendre la valeur **VRAI** ou **FAUX**

Exemples:

- 2 et 2 font 4
- 1 et 1 font 10
- Il pleut
- x > y
- 2 + 2 = 4
- 1 + 1 = 0
- Le soleil brille
- Il a les yeux rouges
- un carré est un polygone
- Malo est gentil

III. Éléments de logique propositionnelle

1. Les constantes

Les constantes sont V et F : 0 (Faux, F ou $\perp$) et 1 (Vrai, V, ou $\top$)

2. Variable propositionnelle

C'est une proposition considérée comme indécomposable les variables forment un ensemble dénombrable, et on les désignera ici par des lettres romaines : p, q,

3. Connecteurs logiques

On dispose d'un connecteur unaire ¬, et de quatre connecteurs binaires : ∨, ∧, ⇒ et ⇔.

- négation ***non,*** ¬
- conjonction ***et,*** ∧
- implication⇒
- disjonction ***ou,*** ∨

4. Formule

a. Définition

C'est une expression logique composée de variables propositionnelles et de connecteurs logiques. Les formules propositionnelles sont définies à l'aide de constantes, variables et connecteurs.

On note également que :

- toute constante est une formule ;
- toute variable est une formule ;
- si *e* est une formule et si ⊗ est un connecteur unaire, (⊗*e*) est une formule ;
- si *e* et *f* sont des formules et si ⊗ est un connecteur binaire, (*e* ⊗ *f*) est une formule.
- Le nombre de connecteurs n'est pas limité dans une formule.

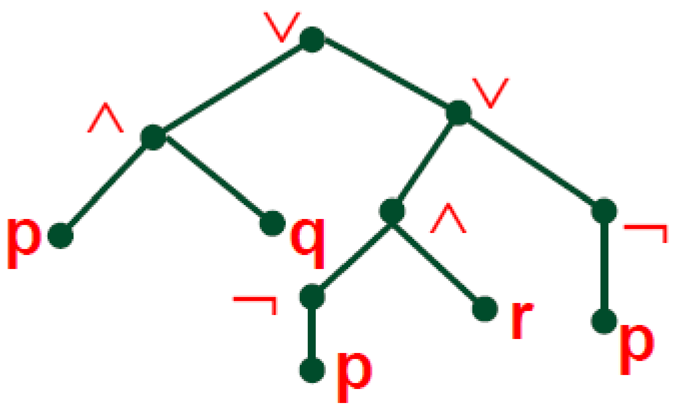
Exemple : p et q variables propositionnelles

L'écriture ((¬p ∨q) ∧¬q) ∨ (p ∨¬q) est une formule

b. Représentations d'une formule

(p ∧q) ∨ ((¬p ∧r) ∨¬p))

Par un arbre syntaxique :



En utilisant la notation préfixée (polonaise) :

∨∧p q ∨∧¬p r ¬p

En utilisant la notation post fixée:

p q ∧p ¬r ∧p ¬∨∨

IV. Tables de vérité

C'est une représentation des valeurs de vérité associées à une expression logique.

Soient ***p et q*** des variables propositionnelles on a les tables de vérités suivants :

Négation

p	¬ p
V	F
F	V

Conjonction

p	q	p∧q
V	V	V
V	F	F
F	V	F
F	F	F

Disjonction

p	q	p∨q
V	V	V
V	F	V
F	V	V
F	F	F

Implication

p	q	p→q
V	V	V
V	F	F
F	V	V
F	F	V

V. Equivalences classiques

- Commutativité
  - p ∧ q équivalent à q ∧ p
  - p ∨ q équivalent à q ∨ p
- Associativité
  - p ∧ (q ∧ r) équivalent à (p ∧q) ∧ r
  - p ∨ (q ∨ r) équivalent à (p ∨q) ∨ r
- Distributivité
  - p ∧ (q ∨r) équivalent à (p ∧q) ∨ (p ∧ r)
  - p ∨ (q ∧r) équivalent à (p ∨q) ∧ (p ∨ r)

VI. Lois de Morgan

- ¬ (p ∧ q) équivalent à (¬p) ∨ (¬q)
- ¬ (p ∨ q) équivalent à (¬p) ∧ (¬q)

p	q	p ∧q	¬(p ∧q)	¬p	¬q	¬p ∨ ¬q

VII. Vocabulaire

Une formule logique *e* est

- une **tautologie** si pour tout contexte μ on a [μ]*e* = 1 c'est-à-dire vraies pour toute assignation de valeurs de vérité aux variables.

Exemple: **p** ∨ ¬**p**

p	¬ p	p ∨ ¬ p

- une **contradiction** si pour tout contexte μ on a [μ]*e* = 0 c'est-à-dire fausses pour toute assignation de valeurs de vérité aux variables.

Exemple: **p** ∧ ¬**p**

p	¬ p	p ∧ ¬ p

- satisfiable s'il existe au moins un contexte μ pour lequel [μ]*e* = 1.

VIII. Les formules équivalentes:

Ce sont des formules qui ont la même valeur de vérité pour toute assignation de la même valeur de vérité aux variables.

Exemples:

p ⇒q est équivalent à ¬p ∨q

p ⇒q est équivalent à ¬q ⇒¬p

p	q	p ⇒ q	¬ p	q	¬ p ∨ q

IX. Quelques usages de la logique

1. Pour vérifier l'équivalence de deux formules

"être mineur (p) ou majeur (¬p) non imposable (q) " équivaut à *"être mineur (p) ou non imposable (q) "*

p	q	¬p∧q	p∨(¬ p∧q)	p∨q

2. Applications à l'algorithmique

- **<u>Interpréter</u>** (et bien comprendre!) l'arrêt des itérations à la sortie d'une boucle.

tant que<cond> **faire**

À l'entrée : <cond> *est vrai*

À la sortie : **non**(<cond>) *est vrai* c'est-à-dire <cond> *est faux*

donc si cond= p **et** q

à la sortie : **non**(p **et** q) c'est-à-dire **non** p **ou non** q

<u>Exemple</u>: avec **<cond>** égal à: val ≠STOP **et** nbVal< MAX

non(<cond>) égal à: val =STOP **ou** nbVal≥MAX

- **<u>Simplifier</u> une écriture par substitution d'une formule équivalente**

  si (Age = "Mineur" ou (non (Age = "Mineur") et non Fisc = Imposable"))) alors...

Equivalent à :

si (Age = "Mineur" ou non (Fisc = "Imposable")) alors...

- **<u>Vérifier</u> la validité d'une condition**

  si Valeur< 10 et Valeur >100 alors…

Nous avons un cas improbable ici car *Valeur* ne peut pas être inférieur à 10 et en même temps supérieur à 100.

- **<u>Ecrire la négation</u> d'une condition**

  si on veut P et Q et R : répéter …. tant que **non** P **ou non** Q **ou non** R **ou**…

TD4

Exercice 1

Si Pierre vient, on joue aux cartes ;

Si Pierre et Jean viennent, il y a des disputes ;

Si on ne joue pas aux cartes, il n'y a pas de dispute ;

Pierre ne vient pas.

Exercice 2

Si Didier est l'auteur de ce bruit, il est stupide ou dépourvu de principes.

Didier n'est ni stupide ni dépourvu de principes.

Didier n'est pas l'auteur de ce bruit.