La logique propositionnelle

Le matériel

- Synthèse du professeur
- Site Web: https://www.prodafor.com/informatique

> Section Algèbre de Boole

Marc Filion

Logique

- C’est vers le milieu du 19ième siècle que le mathématicien et logicien anglais George Boole publie son traité d’algèbre intitulé: “Investigation des lois du raisonnement sur lesquelles reposent la théorie mathématique de la logique”.

- Il faut par contre attendre près d’un siècle avant que l’on trouve des applications pratiques à cette théorie.

- Aujourd’hui, la logique est à la base de l’électronique et de l’informatique.

- En informatique, elle fait partie intégrante de tout langage de programmation. Elle est au cœur des base de données et elle joue un rôle important en intelligence artificielle.

- La logique est une branche des mathématiques visant à représenter formellement les structures du raisonnement.

- Les éléments de base en logique sont les ***propositions***. Une proposition est toute expression à laquelle on peut, sans ambiguïté, associer la valeur de vérité « **vraie** » ou « **faux** ».

- En électronique, le courant électrique passe ou ne passe pas, ce que l'on peut noter également par 1 ou 0. En fait c'est sur ce concept fondamental, pour le moins assez simple, que sont construits tous les ordinateurs électroniques.

Propositions:

Une proposition est une assertion (phrase éventuellement) qui ne peut qu'être vraie ou fausse. [Donc deux valeurs de vérités Vrai (v ou 1) et Faux (f ou 0)].

- *Exemple :*
  - 'La terre est carrée' est une assertion *Fausse*
  - 'Trois-Rivières est au Québec' est une assertion *Vraie*

- **Exercice**

Parmi les énoncés qui suivent, lesquels sont des propositions?

  - «3 est un nombre pain»
  - «Encore bravo!»
  - «$x + 6 > y$»
  - «La carotte est un légume»

- <u>Variable propositionnelle</u> :

  Une variable propositionnelle est une variable qui ne peut prendre que les valeurs vrai ou faux. Nous utiliserons les lettres minuscules *p, q, r…, pour représenter une telle variable*.

- *<u>Fonction propositionnelle</u>:*

  Un énoncé qui devient une proposition si on connaît la valeur de son inconnue.

  <u>Exemple</u>: $P(x) = x < 10$

- *<u>Forme propositionnelle</u>:*

  Une expression formée de variables propositionnelles reliées par des opérateurs logiques.

  <u>Exemple</u>: $(p \vee \neg q) \rightarrow \neg p$

Opérateurs Logiques :

- Il y a plusieurs façons de combiner des propositions simples de sorte à produire des formes propositionnelles.

- Des opérateurs (connecteurs) comme et ($\wedge$), ou ($\vee$), non($\neg$, ~), si … alors (implique, $\rightarrow$), si et seulement si (équivalent à, $\leftrightarrow$) permettent de construire à partir d'une proposition donnée une forme propositionnelle.

- L'opérateur « non » opère sur une proposition.

- Les autres opérateurs opèrent sur plusieurs propositions (deux au moins).

- Pour connaître la valeur de vérité d'une forme propositionnelle, nous avons le concept de la table de vérité.

Le connecteur de négation « non » :

Si p est une proposition alors non p notée $\neg p$ est une proposition. La proposition $\neg p$ est définie par la table de vérité suivante :

p	$\neg p$
V	F
F	V

- **Exemple**

La valeur de vérité de la proposition $\neg(4 < 5)$ est F.

Le connecteur de conjonction « et » :

Si p et q sont des propositions alors p et q notée p∧q est une proposition. En C++, la conjonction s'écrit « && ». La proposition p∧q est définie par la table de vérité suivante :

p	q	$p \wedge q$
F	F	
F	V	
V	F	
V	V	

Exemples

- La valeur de vérité de $(3 > 1) \wedge (12\ est\ pair)$ est V, car les deux propositions sont V.
- La valeur de vérité de $(3 > 1) \wedge (4\ est\ un\ nombre\ premier)$ est F, car l'une des 2 propositions est F.

Le connecteur de disjonction « ou » :

Si p et q sont des propositions alors p ou q notée p∨q est une proposition. La disjonction en C++ s'écrit «| |».La proposition p∨q est définie par la table de vérité suivante :

p	q	$p \vee q$
F	F	
F	V	
V	F	
V	V	

Exemples

- La valeur de vérité de $(3 > 1) \vee (4\ est\ un\ nombre\ premier)$ est V, car au moins l'une des deux propositions est V.
- La valeur de vérité de $(5\ mod\ 6 = 0) \vee (4\ est\ un\ nombre\ premier)$ est F, car aucune des deux propositions n'est V.

Le connecteur de disjonction exclusive « xor »:

Si p et q sont des propositions alors p ou exclusif q notée $p \oplus q$ est une proposition. La proposition $p \oplus q$ est définie par la table de vérité suivante :

p	q	$p \oplus q$
F	F	
F	V	
V	F	
V	V	

Remarque: Un ou l'autre, mais pas les deux.

Tables de vérités et formes propositionnelles:

- Construire la table de vérité de cette forme propositionnelle : $(p \vee \neg q)$ où p et q sont des variables propositionnelles.

- Construire la table de vérité de cette forme propositionnelle : $(p \wedge q) \vee (\neg q \wedge r)$ où p, q et r sont des variables propositionnelles.

p	q	r	$(p \wedge q)$	$\neg q$	$(\neg q \wedge r)$	$(p \wedge q) \vee (\neg q \wedge r)$
F	F	F				
F	F	V				
F	V	F				
F	V	V				
V	F	F				
V	F	V				
V	V	F				
V	V	V				

Remarque :

Le nombre de combinaisons dépend du nombre de variables propositionnelles. Ainsi si nous avons 2 variables propositionnelles nous aurons alors une table de vérité de 4 combinaisons. Si nous avons 3 variables propositionnelles alors nous aurons 8 combinaisons. En fait si le nombre de variables propositionnelles est x alors le nombre de combinaisons est 2 * 2 * 2 … * 2 x fois (autrement dit 2^x).

Exercices

$$p \wedge \neg (p \vee r) \oplus (p \wedge q)$$

Le connecteur biconditionnelle $\leftrightarrow$:

Si p et q sont deux propositions alors la condition si et seulement si p alors q notée p $\leftrightarrow$ q est une proposition.

p $\leftrightarrow$ q est définie par la table de vérité suivante :

p	q	$p \leftrightarrow q$
F	F	
F	V	
V	F	
V	V	

Priorité des opérateurs logiques:

D'une manière générale, il y a une règle de **priorité** entre les opérateurs logiques quand on n'utilise pas les parenthèses. Cette règle de priorité est la suivante :

1. Négation (non)
2. Conjonction (et)
3. Disjonction (ou)
4. Biconditionnelle (si et seulement si)

Les parenthèses ont bien sûr une priorité supérieure.

Tautologies :

Une tautologie est une proposition toujours vraie. Une tautologie sera notée par « ***t*** ».

Ex: $\mathbf{p \vee (\neg p \vee q) \leftrightarrow \ldots \leftrightarrow t}$

Autres exemples de tautologie :

1. $\neg (p \wedge q) \leftrightarrow (\neg p) \vee (\neg q)$
2. $p \vee \neg p$
3. $(p \vee q) \leftrightarrow (q \vee p)$

Vérifions la première tautologie :

p	q	$p \wedge q$	$\neg(p \wedge q)$	$\neg p$	$\neg q$	$\neg (p \wedge q) \leftrightarrow (\neg p) \vee (\neg q)$
F	F					
F	V					
V	F					
V	V					

Contradictions (antilogies)

Une contradiction est une forme propositionnelle qui est toujours fausse. Une contradiction sera notée par « **c** ».

Ex: $\mathbf{p \wedge (\neg p \wedge q) \leftrightarrow \ldots \leftrightarrow c}$

Autres exemples de contradiction :

- $p \wedge \neg p$
- $(p \vee q) \wedge (\neg p) \wedge (\neg q)$

Vérifions le deuxième énoncé:

p	q	$p \vee q$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$(p \vee q) \wedge (\neg p) \wedge (\neg q)$
F	F					
F	V					
V	F					
V	V					

Une forme propositionnelle qui n’est ni une tautologie ni une contradiction est une

contingence.

Équivalence logique:

On dit qu'une proposition P est équivalente à une proposition Q si la proposition $P \leftrightarrow Q$ est une **tautologie**. On notera l'équivalence par $P \Leftrightarrow Q$ ou $P \equiv Q$.

Ex: $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$

Autres exemples :

1. $(p \wedge q) \vee (\neg p \wedge q) \Leftrightarrow q$
2. $p \vee q \Leftrightarrow q \vee p$
3. $p \vee (\neg p \vee q) \Leftrightarrow t$
4. $p \wedge (\neg p \wedge q) \Leftrightarrow c$

Vérifions le premier énoncé:

p	q	$p \wedge q$	$\neg p$	$(\neg p \wedge q)$	$P := (p \wedge q) \vee (\neg p \wedge q)$	$P \leftrightarrow q$
F	F					
F	V					
V	F					
V	V					

Exercices :

1. $p \vee q \Leftrightarrow q \vee p$
2. $p \vee (\neg p \vee q) \Leftrightarrow t$
3. $p \wedge (\neg p \wedge q) \Leftrightarrow c$

p	*q*					
F	*F*					
F	*V*					
V	*F*					
V	*V*					

Propriétés :

La commutativité :

$p \vee q \Leftrightarrow q \vee p$

$p \wedge q \Leftrightarrow q \wedge p$

L'associativité :

$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$

$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$

La distributivité :

$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$

$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

L'élément neutre :

$p \vee c \Leftrightarrow p$

$p \wedge t \Leftrightarrow p$

Complémentarité :

$p \vee \neg p \Leftrightarrow t$

$p \wedge \neg p \Leftrightarrow c$

L'idempotence :

$p \vee p \Leftrightarrow p$

$p \wedge p \Leftrightarrow p$

Involution :

$\neg \neg p \Leftrightarrow p$

L'élément absorbant :

$p \vee t \Leftrightarrow t$

$p \wedge c \Leftrightarrow c$

L'absorption :

$p \wedge (p \vee q) \Leftrightarrow p$

$p \vee (p \wedge q) \Leftrightarrow p$

RÈGLES DE DeMORGAN:

$\neg (p \vee q) \Leftrightarrow (\neg p) \wedge (\neg q)$

$\neg (p \wedge q) \Leftrightarrow (\neg p) \vee (\neg q)$

Règle de DeMorgan

Il existe des équivalences utiles concernant la distribution d'une négation devant une parenthèse avec des opérateurs logiques. Elles peuvent aider à simplifier ou développer les expressions.

$\neg(p \vee q) \equiv \neg p \wedge \neg q$ $\qquad$ $\neg(p \wedge q) \equiv \neg p \vee \neg q$

- Exemples : simplifier les expressions suivantes

a) $p \wedge \neg\big(q \vee (\neg p)\big)$

b) $(\neg A \wedge \neg(B \vee A))$

Exercices :

1- Simplifier la forme propositionnelle

$$(\neg A \wedge \neg (B \vee A))$$

2- Donner la négation de l'expression $p \wedge (r \vee \neg q)$ et les formes équivalentes.

Négation des symboles d'équation et d'inéquation

Voici les négations de propositions construites à l'aide des symboles mathématiques $=, \neq, <, >, \leq, \geq$;

- $\neg(A = B) \equiv (A \neq B)$
- $\neg(A \neq B) \equiv (A = B)$
- $\neg(A < B) \equiv (A \geq B)$
- $\neg(A > B) \equiv (A \leq B)$
- $\neg(A \leq B) \equiv (A > B)$
- $\neg(A \geq B) \equiv (A < B)$

Exemple : quelle est la valeur de vérité de cette expression? $(7 < 6) \vee \neg\left(\frac{1}{0} = 0\right)$

Exemple de l'utilisation des connecteurs logiques en programmation:

1. LIRE a,b,c
2. d:=b*b-4*a*c
3. **Si d<0**, écrire « pas de racine réelles »
4. **Sinon si d=0 alors** x:=-b/(2*a)
5. Écrire x
6. **Sinon** x1:=(-b+(d^(1/2)))/(2*a)
7. x2:= (-b-(d^(1/2)))/(2*a)
8. Écrire x1, x2

```
int x;
cout << "Entrez un entier positif";
cin >> x;
if ((x % 4 == 0) && (x % 3 == 0))
{
    cout << "La fusée vole" <<endl;
}
else
{
    cout << "La fusée explose" <<endl ";
}
```

Que verra-t-on s'afficher si :

a) x = 12
b) x = 15
c) x = 37

```
int y;
cout << "Entrez un entier positif";
cin >> y;
if ((y % 2 == 0) xor (y >= 65))
{
    cout << " Dessiner un triangle" <<endl;
}
else
{
    cout << " Dessiner un carré" <<endl ";
}
```

Que verra-t-on s'afficher si :

a) y = 20
b) y = 77
c) y = 200

```
int z;
cout << "Entrez un entier positif";
cin >> z;
if ((z % 2 == 0) | | (z >= 65))
{
    cout << " Dessiner un triangle" <<endl;
}
else
{
    cout << " Dessiner un carré" <<endl ";
}
```

Que verra-t-on s'afficher si :

a) z = 20
b) z = 77
c) z = 200

Devoir rencontre 10

- Faire les exercices du fichier (sur Omnivox):

  Exercices logique propositionnelle

- Écouter, si nécessaire, les capsules vidéo dans la section ALGÈBRE DE BOOLE (sur prodafor.com):
  - **Boole01** **jusqu'à** **Boole10**