Introduction :

En L'algèbre de Boole, ou calcul booléen, est la partie des mathématiques qui s'intéresse aux opérations et aux fonctions sur les variables logiques. Elle fut inventée par le mathématicien britannique George Boole. Aujourd'hui, l'algèbre de Boole trouve de nombreuses applications en informatique et dans la conception des circuits électroniques.

2.1 Conception de l'algèbre de Boole

2.1.1 Variable booléen

De manière générale, une variable peut être associée à un évènement. Si ce dernier est vrais, elle prend la valeur « 1 », si non (elle est faux) elle prend la valeur « 0 ».

2.1.2 La fonction booléenne

Est une fonction qui peut avoir une ou plusieurs variables booléennes et retourne l'une des deux valeurs « 0 » ou « 1 »

2.1.3 La table de vérité

La table de vérité d'une fonction logique ***F*** a « n » variables booléenne est un tableau de « m » colonnes et « k » lignes tel que :

$$\begin{cases} \mathbf{m=n+1} & \text{(chaque variable associé à une colonne, la dernière est réservée pour F)} \\ \mathbf{k=2^n} & \text{(chaque ligne représente une combinaison des « n » variable)} \end{cases}$$

Exemple : Soit F(x, y) une fonction logique à deux variable avec :

Deux variables : n= 2

$m = n+1 = 2+1 \longrightarrow 3$ colonnes

$k = 2^n = 2^2 \longrightarrow 4$ combinaisons

$$\longrightarrow \begin{cases} x = y = 0 \longrightarrow F=1 \\ x = 0,\ y = 1 \longrightarrow F=0 \\ x = 1,\ y = 0 \longrightarrow F=0 \\ x = 1,\ y = 1 \longrightarrow F=0 \end{cases}$$

La table de vérité de la fonction F est :

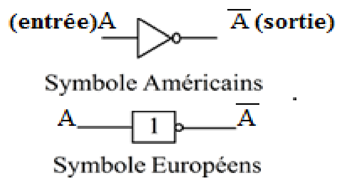
x	y	F
0	0	1
0	1	0
1	0	0
1	0	0

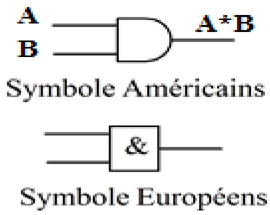
2.1.4 Fonctions logiques de base

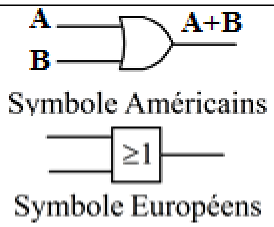
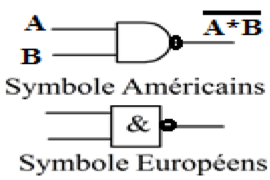
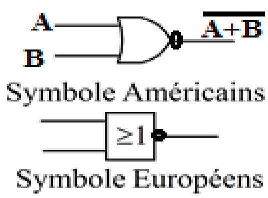
Le tableau suivant représente les différentes fonctions logiques de base, leurs symboles et leur table de vérité.

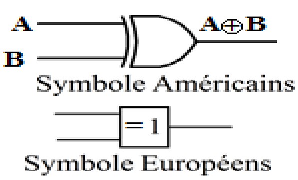
Soient A, B deux variables logique

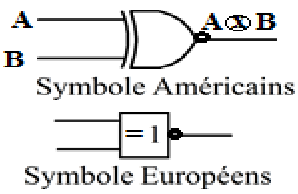
<table>
<tr><th>Nom de fonction</th><th>Définition</th><th>Symbole de la porte logique</th><th>Opération booléenne</th><th>Table de vérité</th></tr>
<tr><td>Complément (inverseur)

NON (NOT)</td><td>$\overline{A}$ est le complément de A On lit A barre</td><td>
</td><td>$\overline{A}$</td><td><table><tr><th>Entrée</th><th>Sortie</th></tr><tr><td>A</td><td>NOT A</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>Produit logique (intersection)

ET (AND)</td><td>La sortie (A*B) est vrai si toutes les entrées (A et B) sont à l'état vrai</td><td>
</td><td>A*B</td><td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>A AND B</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>Somme logique (réunion)

OU (OR)</td><td>La sortie (A + B) est vrai si au minimum une des entrées (A ou B) est à l'état vrai.</td><td>
</td><td>A+B</td><td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>A OR B</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>NAND (NON- ET)</td><td>La sortie ($\overline{A*B}$) est à l'état faux si toutes les entrées (A et B) sont à l'état vrai.</td><td>
</td><td>$\overline{A*B}$</td><td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>A NAND B</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>NON-OU (NOR)</td><td>La sortie ($\overline{A+B}$) est vrai si toutes les entrées (A et B) sont à l'état faux.</td><td>
</td><td>$\overline{A+B}$</td><td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>A NOR B</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>OU Exclusif

(Exclusive OR)
XOR</td><td>La sortie ($A\oplus B$) n'est vaut que si les deux entrées (A et B) sont différentes</td><td>
</td><td>$A\oplus B$
=
$\overline{A}B+A\overline{B}$</td><td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>A XOR B</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>Coïncidence (identité) (Exclusive NOR)

NXOR</td><td>La sortie (A θ B) est vrai que dans les cas où les deux entrées (A et B) sont égales.</td><td>
</td><td>A θ B
=
$AB+\overline{A}\,\overline{B}$
=
$\overline{A\oplus B}$</td><td><table><tr><th colspan="2">Entrées</th><th>Sortie</th></tr><tr><td>A</td><td>B</td><td>A NXOR B</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
</table>

Tab.1 : Fonctions logiques de base

2.1.4 Propriétés des fonctions logiques de base

Soient A, B et C trois variables logiques.

2.1.4.1 Associativité

Comme avec les opérations habituelles, certaines parenthèses sont inutiles :

- (A+ B) + C= A+ (B + C) = A + B + C
- (A*B)*C = A*(B*C) = A*B*C

Démonstration via la table de vérité

A	B	C	A+B	B+A	(A+B) + C	A+B+C		A*B	B*A	(A*B)*C	A*(B*C)
0	0	0	0	0	0	0		0	0	0	0
0	0	1	0	0	1	1		0	0	0	0
0	1	0	1	1	1	1		0	0	0	0
0	1	1	1	1	1	1		0	0	0	0
1	0	0	1	1	1	1		0	0	0	0
1	0	1	1	1	1	1		0	0	0	0
1	1	0	1	1	1	1		1	1	0	0
1	1	1	1	1	1	1		1	1	1	1

2.1.4.2 Commutativité

L'ordre est sans importance :

A+ B = B + A

A*B = B*A

Démonstration via la table de vérité

A	B	A+B	B+A	A*B	B*A
0	0	0	0	0	0
0	1	1	1	0	0
1	0	1	1	0	0
1	1	1	1	1	1

2.1.4.3 Distributivité

Comme avec les opérations mathématiques habituelles, il est possible de distribuer :

A*(B + C) = A*B + A*C

Attention : comportement différent par rapport aux opérateurs + et * habituels :

A + (B*C) = (A+ B)*(A + C)

A	B	C	A*B	A*C	(A*B) + (A*C)	A*(B+C)		A+B	A+C	(A+B)(A+C)	A+ (B* C)
0	0	0	0	0	0	0		0	0	0	0
0	0	1	0	0	0	0		0	1	0	0
0	1	0	0	0	0	0		1	0	0	0
0	1	1	0	0	0	0		1	1	1	1
1	0	0	0	0	0	0		1	1	1	1
1	0	1	0	1	1	1		1	1	1	1
1	1	0	1	0	1	1		1	1	1	1
1	1	1	1	1	1	1		1	1	1	1

2.1.4.4 Élément neutre

$$A + 0 = A$$

$$A*1 = A$$

2.1.4.5 Élément nul (absorbant)

$$0*A = 0$$

$$1 + A = 1$$

2.1.4.6 Idempotence

$$A + A + A+ [...] + A = A$$

$$A*A*A*[...]*A = A$$

2.1.4.7 Complémentarité

$$A= \overline{\overline{A}}$$

$$A + \overline{A} = 1$$

$$A* \overline{A} = 0$$

2.1.4.8 Lois de De Morgan

1 Complément d'un produit $\overline{A*B}=\overline{A}+\overline{B}$

2 Complément d'une somme $\overline{A+ B}= \overline{A}*\overline{B}$

2.1.5 Quelques relations utiles

En application des propriétés qui ont été énoncées précédemment, on peut démontrer un certain nombre de relation très utiles tel :

$$X+X*Y= X$$

$$X+\overline{X}*Y = X+Y$$

$$X*(\overline{X}+Y) = X*Y$$

$$\left.\begin{array}{l} X*A + \overline{X}*B + A*B = X*A + \overline{X}*B \\ (X+A)*(\overline{X}+B)*(A+B) = (X+A)*(\overline{X}+B) \end{array}\right\}$$ ***Relations du consensus***

Remarque : ces relations permettent, avec un peu de pratique, de simplifier l'écriture des fonctions logiques (méthodes algébrique).

2.1.6 Formes canoniques des expressions logiques

Une expression logique F peut s'écrire sous un grand nombre de formes différentes. Deux d'entre elles, dites formes canoniques, sont particulièrement utiles.

- ***La première Formes canoniques :***

  Cette forme est basée sur les mintermes des fonctions logiques. L'expression correspondante à cette forme canonique d'une fonction logique est obtenue par la ***somme*** des différents mintermes, correspondant à chaque 1 de la variable de sortie de la fonction.

  Dite aussi : forme disjonctive, ou somme des produits.

Un mintermes : Noté $\mathbf{m_i}$ est le produit logique des variables de la même ligne de la table de vérité**.**

Exemple : les mintermes d'une fonction à trois variables F (A, B, C)

A	B	C		F		Mintermes
0	0	0		**0**		$\mathbf{m_0 = \bar{A}\bar{B}\bar{C}}$
0	0	1		0		$\mathbf{m_1 = \bar{A}\bar{B}C}$
0	1	0		0		$\mathbf{m_2 = \bar{A}B\bar{C}}$
0	1	1		1		$\mathbf{m_3 = \bar{A}BC}$
1	0	0		0		$\mathbf{m_4 = A\bar{B}\bar{C}}$
1	0	1		1		$\mathbf{m_5 = A\bar{B}C}$
1	1	0		1		$\mathbf{m_6 = AB\bar{C}}$
1	1	1		1		$\mathbf{m_7 = ABC}$

L'expression de la fonction F (A, B, C) = $\sum$ ($\mathbf{m_3, m_5, m_6, m_7}$)

. $= \mathbf{\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC}$

- **La deuxième Forme canoniques**

Cette forme est basée sur les maxtermes des fonctions logiques. L'expression correspondante à cette forme canonique d'une fonction logique est obtenue par le ***produit*** des différents maxtermes, correspondant à chaque 0 de la variable de sortie de la fonction.

Dite aussi : conjonctive, ou produit des sommes

Un maxtermes : noté $\mathbf{M_i}$, s'obtient en faisant la somme logique des variables ***sous forme inversée*** de la même ligne de la table de vérité.

Exemple : les maxtermes d'une fonction à trois variables F (A, B, C).

A	B	C		F		Maxtermes
0	0	0		0		$\mathbf{M_0 = A + B + C}$
0	0	1		0		$\mathbf{M_1 = A + B + \bar{C}}$
0	1	0		0		$\mathbf{M_2 = A + \bar{B} + C}$
0	1	1		1		$\mathbf{M_3 = A + \bar{B} + \bar{C}}$
1	0	0		0		$\mathbf{M_4 = \bar{A} + B + C}$
1	0	1		1		$\mathbf{M_5 = \bar{A} + B + \bar{C}}$
1	1	0		1		$\mathbf{M_6 = \bar{A} + \bar{B} + C}$
1	1	1		1		$\mathbf{M_7 = \bar{A} + \bar{B} + \bar{C}}$

L'expression de la fonction F (A, B, C) = $\prod$ ($\mathbf{M_0, M_1, M_2, M_4}$)

F (A, B, C) $= (A + B + C) * (A + B + \overline{C}) * (A + \overline{B} + C) * (\overline{A} + B + C)$

2.2 Simplification des functions logiques

2.2.1 Méthode algébrique

En application des propriétés qui ont été énoncées précédemment, on peut démontrer, par exemple :

$$A * B + A * \overline{B} = A \qquad \ldots(1)$$

$$A + A * B = A \qquad \ldots(2)$$

Démonstration de la relation (1) :

$$(1) \Rightarrow A * B + A * \overline{B} = A * (B + \overline{B})$$

$$= A * 1 \qquad \text{car} \qquad B + B = 1$$

$$= A \qquad \text{puisque} \qquad A \cdot 1 = A$$

$$(2) \Rightarrow A + A * B = A * (1 + B)$$

$$= A * 1 \qquad \text{car} \qquad 1 + B = 1$$

$$= A \qquad \text{puisque} \qquad A \cdot 1 = A$$

2.2.3 Méthode graphique (méthode de Karnaugh)

2.2.3.1 Définition de la table de Karnaugh

Il s'agit d'un tableau à double entrée dans lequel chaque combinaison des variables d'entrée est associée à une case qui contient la valeur de la fonction. Ce sont donc des tables de vérité !, cependant, la disposition des cases est telle que deux *cases voisines*, correspondent à des combinaisons adjacentes des variables d'entrée, c'est-à-dire des combinaisons ne différant que par la complémentation d'une seule variable.

Le nombre de case pour n variable est 2^n

Exemples :* Soit la fonction F (A, B) définie par la table de vérité (T.V) suivante :

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Correspondant aux tables de Karnaugh (TK) →

A\B	0	1
0	0	1
1	1	0

↔

A\B	0	1
0	$\overline{A}\overline{B}$	$\overline{A}B$
1	$A\overline{B}$	AB

* Soit la fonction F (A, B, C) définie par la table de vérité (TV) suivante :

TV

	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Correspondant des codes en décimal

Table Karnaugh

A\BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Comment remplir la table TK

A\BC	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	0	1	3	2
A	4	5	7	6

- Soit la fonction F (A, B, C, D) définie par la table de vérité (TV) suivante :
  Table de vérité TV F (A, B, C, D) Table de Karnaugh de F (A, B, C, D)

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

AB\CD	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	0	0	1	1
10	0	1	1	0

Comment remplir la table TK

AB\CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

L'utilité des tables de Karnaugh : ces tables permet d'obtenir les fonctions logiques sous leur première forme canonique la plus simple possible.

2.2.3.2 Simplification des fonctions logiques

Pour simplifier une fonction logique par une table de Karnaugh, on suit les étapes suivantes :

- Il faut regrouper les valeurs de la fonction égale à 1 dans des rectangles ayant comme nombre de case une puissance de 2 (16, 8, 4, 2, ou 1 case).
- Les groupes formés doivent être les moins nombreux possibles, mais ils doivent englober tous les 1. On peut faire des chevauchements.
- On a intérêt à dessiner des rectangles les plus grands possibles.

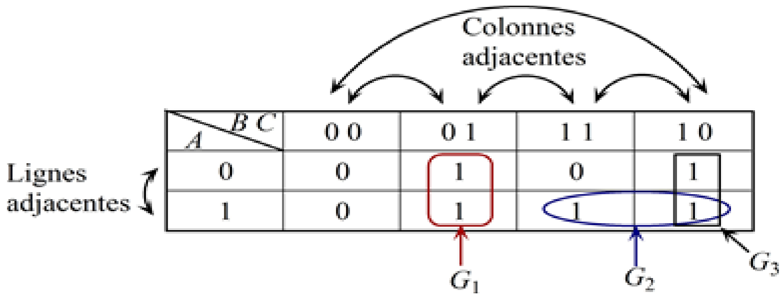
- On peut considérer la table de Karnaugh comme un tore : la dernière ligne est adjacente à la première, et la première colonne est adjacente à la dernière, ce que permet le regroupement des 1 se trouvant à ces emplacements.

Exemple A :

Soit à simplifier par la table de Karnaugh la fonction F (A, B, C) définie par la table de vérité suivante :

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Correspondant au tableau de Karnaugh suivant :



De la table de Karnaugh on a : $G_1 = \overline{B}$, $G_2 = AB$, $G_3 = B\overline{C}$

F (A, B, C) = $G_1 + G_2 + G_3$

F (A, B, C) = $\overline{B} + AB + B\overline{C}$

Exemple B : Soit à simplifier la fonction F (A, B, C, D), donner sous sa première forme canonique suivante :

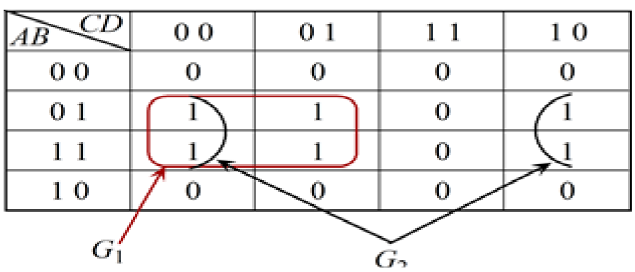
F (A, B, C, D) = ∑ (4, 5, 6, 12, 13, 14)

C.-à-d. : la fonction F prend la valeur 1 dans les case correspondante en décimal aux chiffres 4, 5, 6, 12, 13, et 14.

Remarque : pour les « 0 » (la deuxième forme canonique), la fonction F (A, B, C, D) s'écrit comme suit :

F (A, B, C, D) = ∏ (0, 1, 2, 3, 7, 8, 9, 10, 11, 15)

La table de Karnaugh :



$$G_1 = B\overline{C} \text{ et } G_2 = B\overline{D}$$

$$F\ (A, B, C, D) = B\overline{C} + B\overline{D}$$

Annexe : Quelques exemples sur la simplification des fonctions logiques

A-1 cas de fonctions a 2 variables :

a\b	0	1
0	0	1
1	0	1

F= b

a\b	0	1
0	1	1
1	0	0

F= $\bar{a}$

a\b	0	1
0	0	1
1	1	1

F = a + b

a\b	0	1
0	0	0
1	0	0

F= 0

a\b	0	1
0	1	1
1	1	1

F = 1

a\b	0	1
0		
1		

Cas interdit

Nb : Il y a d'autres exemples qu'on peut réaliser, bien sûr, on respectant la condition d'adjacence des cases

A-2 Cas de fonctions a 3 variables :

a\bc	00	01	11	10
0	0	0	1	1
1	0	0	1	1

F = b

a\bc	00	01	11	10
0	0	1	1	0
1	0	1	1	0

F = c

a\bc	00	01	11	10
0	1	0	0	1
1	1	0	0	1

F = $\bar{c}$

a\bc	00	01	11	10
0	0	0	0	0
1	1	1	1	1

F= a

a\bc	00	01	11	10
0	0	1	0	0
1	0	1	0	0

F = $\bar{b}$c

a\bc	00	01	11	10
0	1	0	0	1
1	0	0	0	0

F= $\bar{a}$ $\bar{c}$

a\bc	00	01	11	10
0	0	0	0	0
1	0	0	0	1

F = ab$\bar{c}$

Remarque-1 : On constate que, pour une fonction de 3 variables

- 4 cases regroupées : donne un produit d'une variable
- 2 cases regroupées : donne un produit de deux variables
- 1 case seule : donne un produit de 3 variables

- ***Comme aussi, il faut noter qu'on peut utiliser une case plusieurs fois, (x + …+ x = x)***

Par exemple :

a\bc	00	01	11	10
0	0	0	1	1
1	0	1	1	1

F = a c + b

A-2 Cas de fonctions a 4 variables :

ab\cd	00	01	11	10
00	**1**	**1**	**1**	**1**
01	**1**	**1**	**1**	**1**
11	**1**	**1**	**1**	**1**
10	**1**	**1**	**1**	**1**

F= 1

ab\cd	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

F = 0

ab\cd	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

F = c

ab\cd	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0

F = $\bar{a}$

ab\cd	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

F = $\bar{b}$

ab\cd	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

F = $\bar{b}\ \bar{d}$

ab\cd	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	0	0
10	0	0	0	0

F = $\bar{a}bd$

ab\cd	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	1	1	1
10	0	1	1	1

F = a d + c

Nb : Il y a d'autres exemples qu'on peut réaliser, bien sûr, on respectant la condition d'adjacence des cases

Remarque-2 : On constate que, pour une fonction de 4 variables

- 8 cases regroupées : donne un produit d'une variable
- 4 cases regroupées : donne un produit 2 variables
- 2 cases regroupées : donne un produit de 3 variables
- 1 case seule : donne un produit de 4 variables