

Chapitre II : Algèbre de Boole

II.1. Objectif de ce chapitre

A l'issu de ce chapitre, l'apprenant sera capable de :

- Comprendre les bases de l'algèbre de Boole, y compris les opérations logiques fondamentales (ET, OU, NON).
- Savoir représenter des fonctions logiques sous forme de tables de vérité.
- Être capable de simplifier des expressions booléennes en utilisant les lois de l'algèbre de Boole.
- Apprendre à exprimer des fonctions logiques sous forme canonique (somme de produits, produit de sommes).
- Utiliser les tables de Karnaugh pour simplifier des expressions logiques.
- Savoir concevoir des circuits logiques en utilisant des portes logiques de base.

II.2. Introduction

L'**algèbre de Boole**, ou calcul booléen, est la partie des mathématiques, de la logique et de l'électronique qui s'intéresse aux opérations et aux fonctions sur les variables logiques. Plus spécifiquement, l'algèbre booléenne permet d'utiliser des techniques algébriques pour traiter les expressions à deux valeurs du calcul des propositions. Elle fut initiée par le mathématicien britannique du milieu du XIX^e siècle George Boole [01].

II.3. Concepts de l'algèbre de Boole

a) Valeurs de vérité

On appelle valeurs de vérité l'ensemble B constitué de deux éléments VRAI et FAUX. Cet ensemble est noté aussi : $B = \{0,1\}$. Nous allons voir, dans les éléments suivants, quelques opérations sur l'ensemble B : complémentation (inversion), disjonction (union / ou / + / max), et conjonction (intersection / et / . / min), ...

b) Complémentation (inversion)

Soit a une variables logiques ($a \in B$). La complémentation (inversion) est une opérateur unaire (il ne porte que sur une seule variable d'entrée). l'inverse de la variable a , noté \bar{a} , est défini comme suit :

Tableau II.1 : Complémentation

a	\bar{a}
0	1
1	0

c) Disjonction (Union)

Soit a et b deux variables logiques ($a, b \in B$). La disjonction (l'union / + / max / OU / OR) est une opération binaire interne sur l'ensemble B , écrite comme suit : $a + b$, et elle est fautive uniquement si a est FAUX et b est FAUX.

La table de vérité de la disjonction est comme suit :

Tableau II.2 : La disjonction (union)

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

d) Conjonction (Intersection)

Soit a et b deux variables logiques ($a, b \in B$). La conjonction (l'intersection / . / min / ET / AND) est une opération binaire interne sur l'ensemble B , écrite comme suit : $a . b$, et elle est vraie uniquement si a est VRAI et b est VRAI.

La table de vérité de la conjonction est :

Tableau II.3 : La conjonction (Intersection)

a	b	$a . b$
0	0	0
0	1	0
1	0	0
1	1	1

e) Ou exclusif (XOR)

Soit a et b deux variables logiques ($a, b \in B$). Le OU exclusif, dit aussi XOR, noté \oplus , est une opération binaire interne sur l'ensemble B , écrite comme suit : $a \oplus b$, et elle est vraie uniquement si a est différents de b .

La table de vérité du XOR est :

Tableau II.4 : Ou exclusif (XOR)

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

f) Coïncidence (XNOR)

Soit a et b deux variables logiques ($a, b \in B$). L'opération coïncidence (XNOR) est l'inverse du Ou exclusif, notée \odot , est une opération binaire interne sur l'ensemble B , écrite comme suit : $a \odot b$, et elle est vrai uniquement si a est égale de b .

La table de vérité de la conjonction est :

Tableau II.5 : Coïncidence (XNOR)

a	b	$a \odot b$
0	0	1
0	1	0
1	0	0
1	1	1

g) Propriétés

Soient x, y et z deux variables logiques ($(x, y, z) \in B^3$).

- Pour la disjonction nous avons :
 - $x + 0 = x$
 - $x + 1 = 1$
 - $x + \bar{x} = 1$
 - $x + x = x$
- Pour la conjonction nous avons :
 - $x \cdot 0 = 0$
 - $x \cdot 1 = x$
 - $x \cdot \bar{x} = 0$
 - $x \cdot x = x$
- Inversion
 - $\bar{\bar{x}} = x$

- La commutativité :

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

$$x \otimes y = y \otimes x$$

$$x \oplus y = y \oplus x$$
- L'associativité :

$$x + (y + z) = (x + y) + z$$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$
- La distributivité :

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$
- Théorème d'absorption :

$$x \cdot (x + y) = x$$

$$x + (x \cdot z) = x$$

Pour démontrer le théorème d'absorption, on réalise le table de vérité, comme suit :

Tableau II.6 : Table de vérité de : $x \cdot (x + y)$

x	y	$x + y$	$x \cdot (x + y)$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

Vous remarquez que les valeurs de $x \cdot (x + y)$ sont les même que x .

De la même façon, vous pouvez démontrer que $x + (x \cdot z) = x$?

- Théorème d'allongement :

$$x + (\bar{x} \cdot y) = x + y$$

$$x \cdot (\bar{x} + y) = x \cdot y$$

Utiliser la table de vérité pour vérifier les deux égalité du théorème d'allongement ?

- Théorème de Morgan :

$$\overline{x + y} = \bar{x} \cdot \bar{y}$$

$$\overline{\bar{x} \cdot \bar{y}} = x + y$$

L'opération $\overline{x + y}$ est appelée *NOR*, et l'opération $\overline{\bar{x} \cdot \bar{y}}$ est dite *NAND*.

II.4. Fonctions logiques simples et leur représentation (Portes logiques)

II.4.1. Définition

Une fonction logique (fonction binaire) est une fonction qui utilise des variables logiques (appartenant aux valeurs de vérité B) et prends comme résultat une valeur logique (VRAI ou

FAUX). Soit f une fonction logique de n variables binaires (logiques), on écrit :

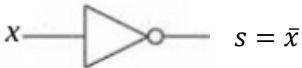


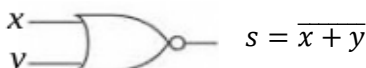

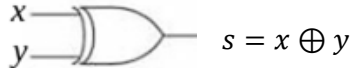
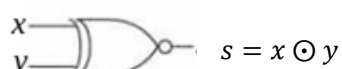
$$f: B^n \rightarrow B$$

Les opérateurs vu dans la section II.3. (*inversion, disjonction, conjonction, XOR, coïncidence*) sont considérés comme des fonctions logiques de base. Et chacun possède une représentation dans la logique combinatoire.

II.4.2. Portes logiques

Le tableau suivant illustre cinq portes logiques des opérations de l'algèbre de Boole, tel-que, x et y sont des variables booléenne, et s est la sortie de la fonction logique :

Tableau II.7 : Portes logiques

Opérations	Expression algébrique	Portes logiques
NOT (inversion)	$s = \bar{x}$	
OR (disjonction)	$s = x + y$	
AND (conjonction)	$s = x \cdot y$	
NOR	$s = \overline{x + y}$	
NAND	$s = \overline{x \cdot y}$	
XOR	$s = x \oplus y$	
XNOR	$s = x \odot y$	

Avec ces portes logiques de base, on peut construire des circuits combinatoires (comme le demi-additionneur, l'additionneur complet, demi-soustracteur, ...).

II.4.3. Expression des fonctions logiques

Une fonction logique peut être exprimée soit :

- **Par sa table de vérité** : qui permet de représenter l'état de sortie (résultat de la fonction) en fonction de toutes les combinaisons des variables d'entrée. Pour une fonction logique avec n variables logiques en entrée, nous aurons 2^n combinaisons possibles pour ces variables d'entrée.
- **Par son équation logique** : utiliser les opérations de l'algèbre de Boole pour exprimer la fonction logique pour les sorties (résultats) correspondant à la valeur 1 (VRAI). On appelle cette équation l'expression algébrique.

Prenons l'exemple suivant : soit $S(a,b,c)$ une fonction logique définie par la table de vérité suivante :

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

On prend les valeurs de sortie qui sont égales à 1.

$\bar{a} \cdot b \cdot \bar{c}$
 $\bar{a} \cdot b \cdot c$
 $a \cdot b \cdot \bar{c}$

Équation logique de S est :

$$S = \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot b \cdot c + a \cdot b \cdot \bar{c}$$

Vous remarquez que la fonction S est exprimée sous forme de la somme de produits, on appelle cela la première forme canonique de la fonction S .

Il y a quatre formes canoniques, à savoir :

1^{ère} forme Canonique : Somme des Produits $\Sigma\Pi$

2^{ème} forme canonique : Produit des sommes $\Pi\Sigma$

3^{ème} forme canonique : Utiliser l'opération NAND uniquement

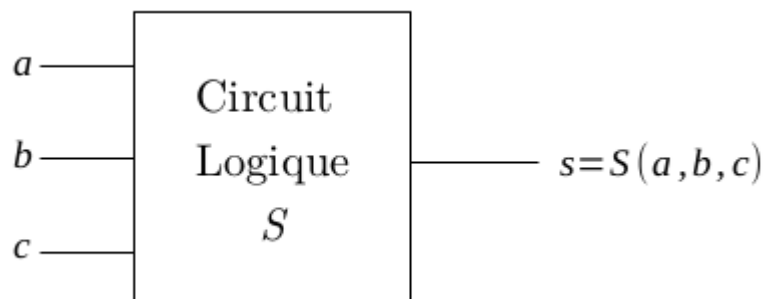
4^{ème} forme canonique : Utiliser l'opération NOR uniquement.

II.4.4. Circuits logiques : schématisation des fonctions logiques

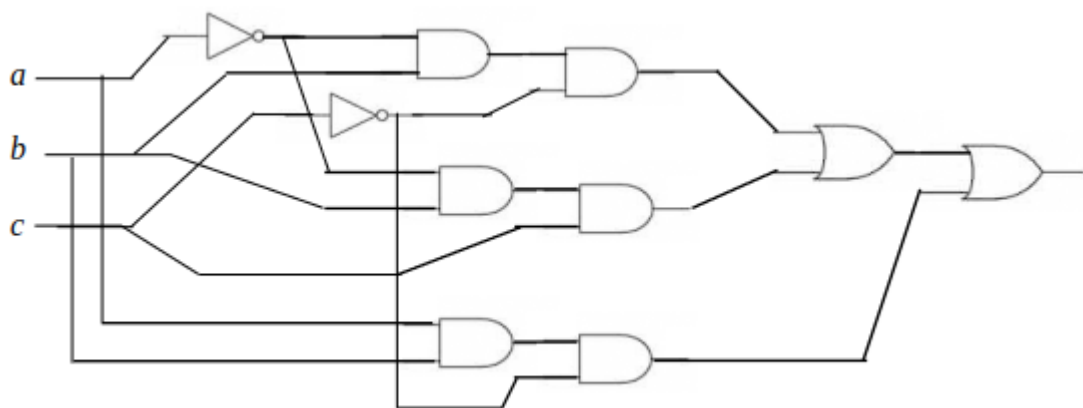
Les fonctions booléennes (logiques) peuvent être schématisées en utilisant les portes logiques vues à la section II.4.2. Prenons la fonction logique S précédente :

$$S(a, b, c) = \bar{a}.b.\bar{c} + \bar{a}.b.c + a.b.\bar{c}$$

Nous voulons réaliser un circuit logique qui permet de calculer la valeur $s = S(a, b, c)$, en fonction des valeurs des variables a , b et c , comme indiqué ci-dessous :



Le circuit logique suivant, schématise l'équation ci-dessus :



Vous remarquez que le schéma est compliqué, pour cela, plusieurs techniques sont utilisées pour rendre le schéma plus simple. Ceci sera réalisé en simplifiant les fonctions booléennes. (voir la section suivante).

II.4.5. Simplification des fonctions booléennes

Les fonctions booléennes (logiques) sont utilisées pour construire des circuits logiques, et dans l'objectif d'avoir des circuit logiques :

- à faible coût
- dans les bref délais
- avec moins d'erreurs
- avec peu de composants

L'expression algébrique des fonctions logiques doit être simplifiée au maximum. Il y a plusieurs méthodes et techniques pour simplifier les fonctions logiques : simplification algébrique, méthode de Karnaugh, ...

Pour notre cas, on va se baser sur la méthode de Karnaugh.

a) Table de Karnaugh

Dans la table de vérité, d'une fonction logique S , chaque ligne représente une possibilité de combinaison des variables d'entrée. Si la fonction S possède n variables en entrée, donc, nous aurons 2^n lignes dans la table de vérité.

Dans l'exemple précédent $S(a, b, c) = \bar{a}.b.\bar{c} + \bar{a}.b.c + a.b.\bar{c}$ nous avons eu la table de vérité suivante :

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$2^n = 2^3 = 8$
lignes

Les tables de vérité ne sont pas adéquates pour les simplifications, pour cela, les tables de Karnaugh sont destinées à faire apparaître visuellement les simplifications possibles d'une fonction logique.

Dans la table de Karnaugh, pour une fonction logique S à n variables booléennes en entrée, on n'affecte pas les possibilités des valeurs des entrées aux lignes, on les affectent aux cellules (cases) du tableau : chaque cellule représente le résultat correspondant à une possibilité de combinaison de valeur de n variables d'entrée. Donc, le nombre de cellule qui représente les résultats de la fonction S est 2^n . Pour réaliser ça, on affecte la moitié des variables aux colonnes

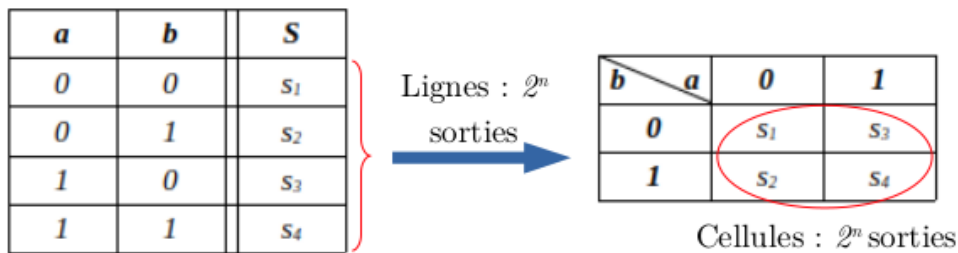
($n/2$ ou $n/2+1$ si n est impair) et l'autre moitié ($n/2$) aux lignes. Nous aurons :

- Le nombre de colonnes : Si n est pair : $2^{(n/2)}$, sinon : $2^{(n/2+1)}$
- Le nombre de lignes : $2^{(n/2)}$.

Pour illustrer ça, nous donnons les exemples suivants.

Exemples :

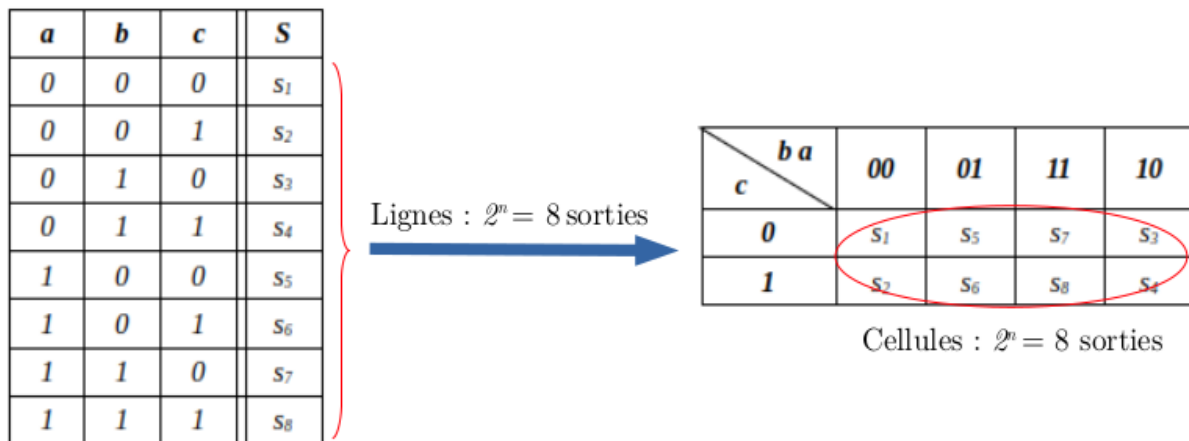
- 1) Soit $S(a,b)$ une fonction logique avec deux variables d'entrée ($n=2$), et s_1, s_2, s_3 et s_4 les $2^n=4$ résultats possibles de la fonctions S , sa table de vérité et sa table de Karnaugh seront comme suit :



Nous avons $n = 2$ (n est pair) :

- On affecte $n/2 = 1$ variable aux colonnes : la variable $a \rightarrow 2^1$ colonnes
- On affecte $n/2 = 1$ variable aux lignes : la variable $b \rightarrow 2^1$ lignes

- 2) Soit $S(a,b,c)$ une fonction logique avec trois variables d'entrée ($n=3$). Nous aurons 2^3 sorties possibles : s_1, s_2, \dots, s_8 . La table de vérité et celle de Karnaugh de la fonction S seront comme suit :

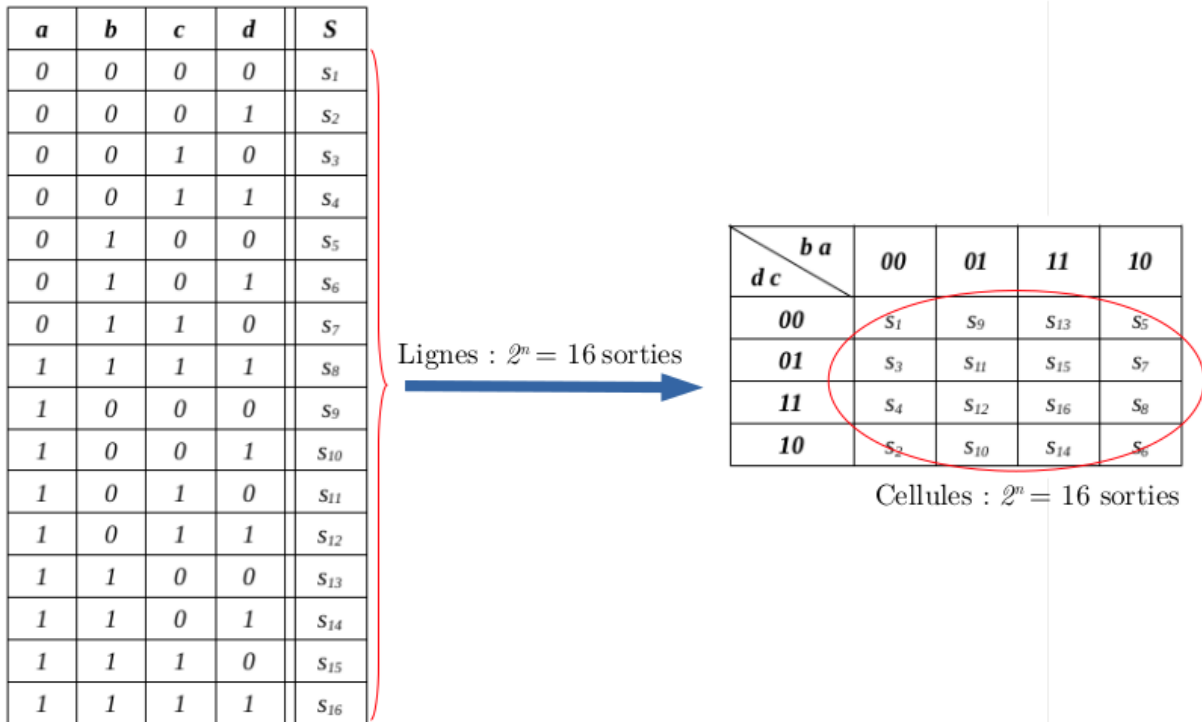


Nous avons $n = 3$ (n est impair):

- On affecte $n/2 + 1 = 2$ variables aux colonnes : la variable a et $b \rightarrow 2^2$ colonnes
- On affecte $n/2 = 1$ variables aux lignes : la variable $c \rightarrow 2^1$ lignes

Vous remarquez que les valeurs des colonnes : 00 01 11 00 (*pourquoi ?*)

3) Soit $S(a,b,c,d)$ une fonction logique avec quatre variables d'entrée ($n=4$). Nous aurons 2^4 sorties possibles : s_1, s_2, \dots, s_{16} . La table de vérité et celle de Karnaugh de la fonction S seront comme suit :



Nous avons $n = 4$ (n est pair):

- On affecte $n/2 = 2$ variables aux colonnes : la variable a et $b \rightarrow 2^2$ colonnes
- On affecte $n/2 = 2$ variables aux lignes : la variable c et $d \rightarrow 2^2$ lignes

Vous remarquez, aussi, que les valeurs des colonnes et des lignes : 00 01 11 10
(pourquoi ?)

Pour coder les lignes et les colonnes dans le tableau de Karnaugh, nous utilisons le code GRAY (dit aussi, le code réfléchi), dans lequel, un seul bit qui sera basculé (changer d'état) de 0 à 1 ou de 1 à 0, comme suit :

Valeur	Code Gray	Valeur	Code Gray
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

b) Règles de simplification

Pour simplifier une fonction logique en utilisant le tableau de Karnaugh (diagramme de Karnaugh), nous suivons les étapes suivantes :

1. Ecrire la fonction logique sous la première forme canonique : $\sum \Pi$;
2. Remplir le tableau en mettant 1 dans les cases (cellules) correspondant aux termes présents dans la fonction logique, et 0 dans les autres cases;
3. Réunir en plus grands boucles (groupements) possibles les cases adjacentes contenant la valeur 1. Ces regroupements (boucles) sont réalisés en associant en puissance de 2 les cases adjacentes : en commençant par les plus grands boucles possibles (groupes de 8 cases, puis de 4 cases, de 2 cases sinon une seule cases). Ce regroupement est réalisé en minimisant le nombre de boucles et en construisant des boucles plus grandes ;
4. Ecrire l'expression booléenne de la fonction logique :
 - Eliminer la variable (les variables) qui apparaît avec son complément (inverse) dans une boucle
 - Faire l'addition logique des résultats obtenus pour les boucles.

En appliquant les quatre étapes ci-dessus, nous obtiendrons l'expression simplifiée de la fonction logique sous la forme d'une somme de produit (première forme canonique).

Remarques

- Dans le tableau de Karnaugh, les cases supérieures sont considérées adjacentes aux cases inférieures (sur la même colonne).
- De la même façon, les cases situées au bord gauche sont considérées adjacentes à celles situées à droite (sur la même ligne).
- Une case contenant 1 peut être utilisées dans plusieurs groupements (boucles).
- Toutes les cases contenant 1 doivent être entourées : soit isolée ou avec d'autres cases adjacentes contenant aussi 1.
- Dans le cas d'une fonction avec 4 variables, nous aurons un tableau de Karnaugh de $2^4 = 16$ cases :
 - Une case isolée contenant 1 correspond à un terme de 4 variables.
 - Une boucle de **deux** cases correspond à un terme de 3 variables.
 - Une boucle de **quatre** cases correspond à un terme de 2 variables.
 - Une boucle de **huit** cases correspond à un terme d'une seule variable.

c) Exemples pratiques

Soit la fonction logique S , de l'exemple précédent, définie comme suit : $S(a, b, c) = \bar{a}.b.\bar{c} + \bar{a}.b.c + a.b.\bar{c}$.

On peut écrire S sous forme de somme min-terme : $S(a, b, c) = \sum 2,3,6$

Sa table de vérité :

Tableau II.8 : Table de vérité de : $x.(x + y)$

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Le tableau de Karnaugh correspondant :

$c \backslash ba$	00	01	11	10
0	0	0	1	1
1	0	0	0	1

Les groupements (boucles) sont :

$c \backslash ba$	00	01	11	10
0	0	0	1	1
1	0	0	0	1

Donc, on peut écrire la fonction S comme suit :

$$S(a, b, c) = \bar{a}.b + b\bar{c}$$

Références

[01] <https://www.techno-science.net/glossaire-definition/Algebre-de-Boole-logique.html>

[02] Daniel Etiemble, 'ALGÈBRE DE BOOLE ET FONCTIONS BOOLÉENNES'. Notes de cours. (<https://www.lri.fr/>)

[03] Farid TAFININE, 'Electronique Numérique', Cours de l'université A.MIRA de Bejaia, département de Génie électrique. Année universitaire : 2016-2017.