Spécialité NSI

LYCÉE de l'ÉLORN
le choix de votre réussite

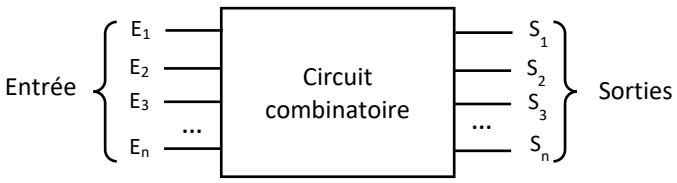
Logique Combinatoire
Algèbre de Boole

Introduction aux systèmes logiques

1.1 Système logique, définitions

Un système *logique* est un système dont les entrées et les sorties sont de type ***vrai*** ou ***faux***, **0** ou **1**, ou encore **Tout Ou Rien (TOR)** ⇨ Une lampe est allumée ou non, un interrupteur est appuyé ou non, ouvert ou fermé...

Un système est à **logique combinatoire** si un signal d'entrée logique (ou une combinaison de signaux d'entrée) conduit invariablement au même signal de sortie. **La même cause produit toujours le même effet**.



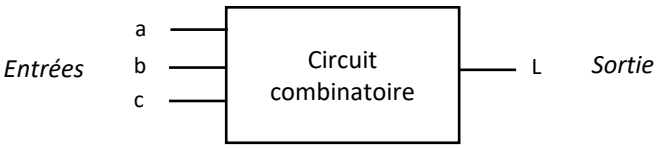
1.2 Les méthodes de représentation d'une fonction logique.

a) Par une phrase explicitant la fonction qu'elle réalise :

Exemple : La lampe « L » s'allume si le bouton « a » est actionné et qu'en même temps le bouton « b » n'est pas actionné, ou alors si le bouton « c » est actionné.

b) Par une table de vérité (TDV) :

Elle indique toutes les combinaisons possibles des états logiques des entrées ainsi que le résultat de la sortie.

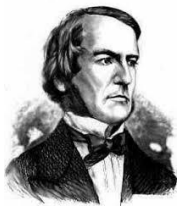


	entrées		*sortie*
a	**b**	**c**	**L**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Une fonction de sortie peut comporter n variables d'entrée. Avec ces n variables on dénombre 2^n combinaisons possibles.

c) Par une équation logique :

Les deux états possibles (0 ou 1) de la fonction logique sont toujours **le résultat d'opérations logiques**. Nous utiliserons les règles mathématiques régies par l'algèbre initié par le mathématicien britannique George Boole (1815-1864).



Les 4 opérations de base entre 1 ou 2 variables binaires a et b sont :

L'opération OUI	notée S = a	qui donne la valeur 1 à S, si et seulement si	a = 1
L'opération NON *appelée aussi complément, inversion ou négation logique*	notée S = ā		a = 0
L'opération OU *disjonction logique*	notée S = a + b		a = 1 OU b = 1
L'opération ET *conjonction logique*	notée S = a · b		a = 1 ET b = 1

Le symbole de l'opérateur ET peut être omis : L'écriture S=a·b est équivalente à *S=ab*

Principales règles de l'algèbre de Boole :

OU		ET		NON
$0+0=0$	$a+1=1$	$0\cdot 0=0$	$a\cdot 1=a$	$\overline{1}=0$
$0+1=1$	$a+0=a$	$0\cdot 1=0$	$a\cdot 0=0$	$\overline{0}=1$
$1+0=1$	$a+a=a$	$1\cdot 0=0$	$a\cdot a=a$	$\bar{\bar{a}}=a$
$1+1=1$	$a+\bar{a}=1$	$1\cdot 1=1$	$a\cdot \bar{a}=0$	
Commutativité		**Associativité**		**Distributivité**
$a\cdot b=b\cdot a$		$a\cdot(b\cdot c)=(a\cdot b)\cdot c$		$a\cdot(b+c)=a\cdot b+a\cdot c$
$a+b=b+a$		$a+(b+c)=(a+b)+c$		$a+(b\cdot c)=(a+b)\cdot(a+c)$

L'opérateur ET est prioritaire par rapport à l'opérateur OU.

Théorème de De Morgan (1806-1871) :

En substance, il s'agit de convertir les fonctions en y changeant les opérateurs. On démontre que :

$\overline{a+b}=\overline{a}\cdot\overline{b}$ et $\overline{a\cdot b}=\overline{a}+\overline{b}$

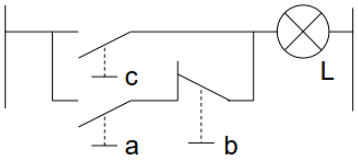
Par extension : $\overline{a+b+c}=\overline{a}\cdot\overline{b}\cdot\overline{c}$ $\overline{a+\overline{b}}=\overline{a}\cdot b$ $\overline{a\cdot b\cdot c}=\overline{a}+\overline{b}+\overline{c}$ $\overline{a\cdot\overline{b}}=\overline{a}+b$

Ex : Vérifier, en complétant ce tableau, la validité du théorème de De Morgan.

a	b	$a\cdot b$	$\overline{a\cdot b}$	$a+b$	$\overline{a+b}$	$\overline{a}$	$\overline{b}$	$\overline{a}+\overline{b}$	$\overline{a}\cdot\overline{b}$

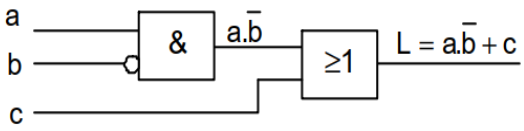
d) Par un schéma à contact :

Dans celui-ci, chaque contact concrétise, par ses deux positions, les deux états d'une variable d'entrée. La lampe symbolise la variable de sortie.



e) Par un logigramme :

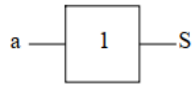
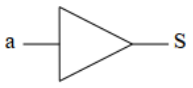
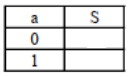
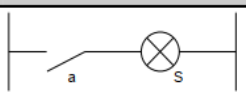
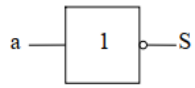
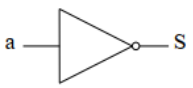
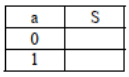
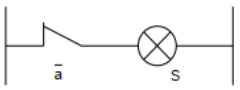
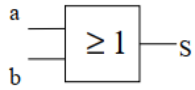
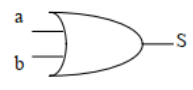
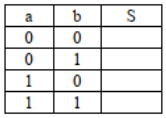
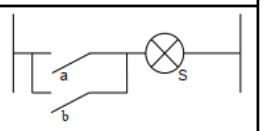
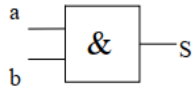
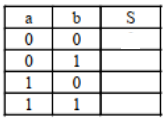
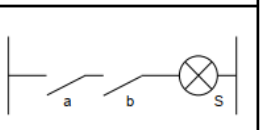
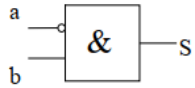
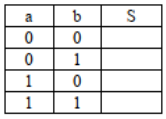
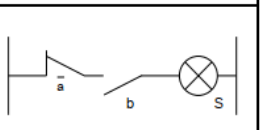
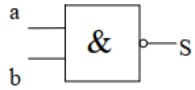
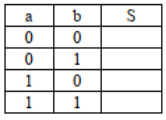
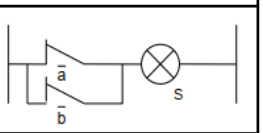
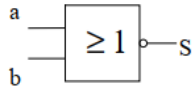
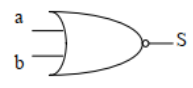
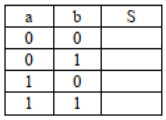
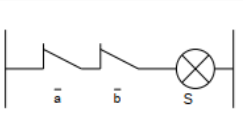
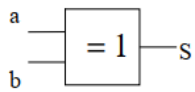
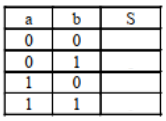
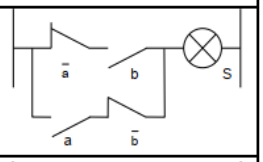
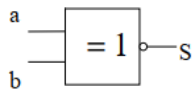
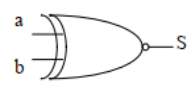
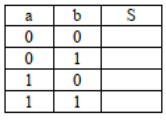
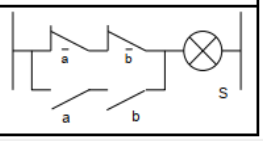
Il donne une représentation graphique du comportement du système en mettant en œuvre des symboles logiques.



Il utilisera de préférence les symboles logiques NF ISO 5784.

Ces symboles sont aussi appelés **opérateurs**, **cellules** ou **portes logiques**.

Tableau des opérateurs les plus usités :

Fonction	équation logique	symbole AFNOR	symbole US	table de vérité	schéma à contact
OUI	$S = a$				
NON	$S = \bar{a}$				
OU	$S = a + b$				
ET	$S = a.b$				
INHIBITION	$S = \bar{a}.b$				
NAND (NON ET)	$S = \overline{a.b} = \bar{a} + \bar{b}$				
NOR (NON OU)	$S = \overline{a + b} = \bar{a}.\bar{b}$				
OU EXCLUSIF	$S = a \oplus b$ $= \bar{a}.b + a.\bar{b}$				
ET INCLUSIF (IDENTITE)	$S = a \odot b$ $= \bar{a}.\bar{b} + a.b$				

Remarquez la pertinence du repère figurant dans les symboles AFNOR.

f) par une implémentation dans un langage de programmation :

En Python, une variable booléenne est soit ***True*** soit ***False*** et les opérateurs booléens sont les mots clés :

- *and* pour le ET
- *or* pour le OU
- *not* pour le NON

Exemples :

$s = a \cdot b$

```
a= True
b= False
if a and b:
    s = 1
else:
    s = 0
```

s = 0 ici

$s = \overline{a} \cdot b$

```
a= True
b= False
s = not(a) and b
```

s = False ici

$s = \overline{a + \overline{b}}$

```
def fonction(a,b):
    if not(a or not(b)) :
        return  1
    else:
        return  0

print(fonction(False, False))
print(fonction(False, True))
print(fonction(True, False))
print(fonction(True, True))
```

Renvoie : ……

2 Écriture d'une fonction logique

2.1 Système logique, forme canonique d'une fonction

Une fonction logique sera parfaitement déterminée par la liste ordonnée de ses variables développée dans une table de vérité.

a	b	*S*
0	0	0
0	1	1
1	0	1
1	1	0

Pour écrire l'équation de X en fonction des 2 variables a et b on peut dire que :
S=1 si *a=0* **et** b=1 **ou** si *a=1* **et** *b=0*

On répète ceci autant de fois que la fonction est égale à 1
Ce qui donne : $S = \overline{a} \cdot b + a \cdot \overline{b}$

Cette forme d'écriture est appelée **forme canonique**.

Par extension on peut aussi écrire que S=0 si a=0 **et** b=0 **ou** si a=1 **et** b=1 soit $\overline{S} = \overline{a} \cdot \overline{b} + a \cdot b$ ou encore $S = \overline{\overline{a} \cdot \overline{b} + a \cdot b}$

2.2 Ecriture d'une fonction logique à partir d'une table de vérité :

a	b	f_1
0	0	0
0	1	1
1	0	0
1	1	1

a	b	c	f_2
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	1

$f_1 =$

$f_2 =$

Remarquez-vous des simplifications évidentes ?

2.3 Remplissage d'une table de vérité à partir d'une fonction logique

$f_1 = \overline{a}.b$

$f_2 = \overline{a} \cdot \overline{b} + a \cdot b$

$f_3 = \overline{a} \cdot \overline{b} + a \cdot b \cdot c + \overline{a}$

a	b	f_1
0	0	
0	1	
1	0	
1	1	

a	b	f_2
0	0	
0	1	
1	0	
1	1	

a	b	c	f_3
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

2.4 Simplification algébrique d'une fonction logique

On réalise les simplifications en utilisant les propriétés de la page 2 mais il existe d'autres types de simplification.

Simplification par mise en facteur commun :

$$f = \overline{a} \cdot b + \overline{a} \cdot \overline{b}$$

Exemple: $f = \overline{a} \cdot (b + \overline{b})$ on met en facteur.

$$f = \overline{a}$$

On peut faire cette simplification si :	- On a une variable dans un terme et son inverse dans l'autre. - Et si le reste des variables est identique.

Simplification par absorption.

Exemple : $g = a + \overline{a} \cdot b$ On distribue le a :

$$g = (a + \overline{a}) \cdot (a + b)$$

$$g = 1 \cdot (a + b)$$

$$g = a + b$$

Directement :

$$g = a + \overline{a} \cdot b$$

$$g = a + b$$

a	b	$\overline{a}$	$\overline{a}b$	$a + \overline{a}b$	$a + b$
0	0				
0	1				
1	0				
1	1				

Nous avons une simplification en distribuant un terme, on appele cette simplification une simplification par absorption.

On peut faire cette simplification si :	- Les 2 termes n'ont pas le même nombre de variables. - Et s'il y a une variable dans un terme et son inverse dans l'autre.

Application :

Simplifier les équations suivantes en utilisant les propriétés de l'algèbre de Boole.

$$f_1 = \overline{a} + a \cdot b$$

$$f_2 = b + a \cdot \overline{b} \cdot c$$

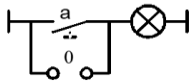
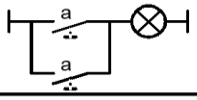
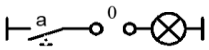
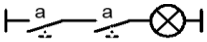
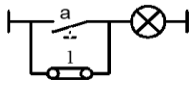
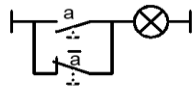
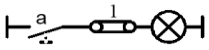
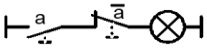
$$f_3 = \overline{a} \cdot \overline{b} \cdot \overline{c} + \overline{a} \cdot \overline{b} \cdot c + \overline{a} \cdot b \cdot c$$

$$f_4 = \overline{a} \cdot \overline{b} \cdot \overline{c} \cdot \overline{d} + \overline{a} \cdot \overline{b} \cdot c \cdot \overline{d} + a \cdot \overline{b} \cdot c \cdot \overline{d} + a \cdot b \cdot c \cdot \overline{d} + a \cdot \overline{b} \cdot \overline{c} \cdot \overline{d} + \overline{a} \cdot b \cdot c \cdot \overline{d}$$

3 Exercices d'applications :

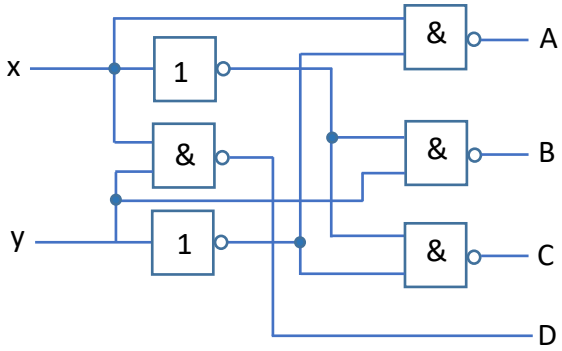
Exercice 1

Complétez le tableau ci-dessous afin de retrouver les principales propriétés de l'algèbre de Boole.

Représentation électrique	Fonction	Représentation électrique	Fonction
			
			
			
			

Exercice 2

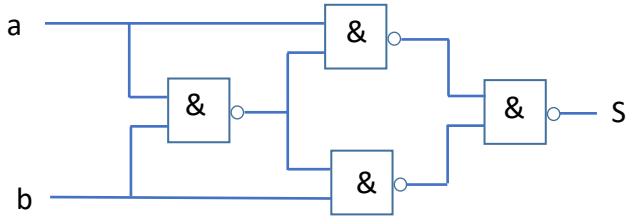
Donnez les équations de sorties **A**, **B**, **C** et **D** en fonction de **x** et **y** et **compléter** la table de vérité.
Pour pouvez procéder comme vous le souhaitez.



x	y	*A*	*B*	*C*	*D*
0	0				
0	1				
1	0				
1	1				

Exercice 3

Soit le logigramme suivant :



a	b	*S*
0	0	
0	1	
1	0	
1	1	

a) Compléter la table de vérité ci-dessus à partir du logigramme puis donner un nom à la fonction réalisée.
b) Extraire de ce logigramme l'équation logique de la sortie **S** en fonction de **a** et **b**. **Simplifier** la fonction à l'aide du théorème de De Morgan.

Exercice 4

Représentez le logigramme correspondant à la fonction ci-dessous avant puis après l'avoir développée et simplifiée.

$$L = (a + \bar{b})(\bar{a} + b) + a\bar{b}c$$

Exercice 5

Dans une banque, l'accès à la salle des coffres est réservé à 3 responsables possédant respectivement chacun une clé différente : clé **A**, clé **B**, clé **C**. Les 3 responsables, à l'exception de celui qui possède la clé **A** ne peuvent pénétrer seul, par contre l'accès à 2 ou 3 est autorisé.

a) Donner la table de vérité modélisant le système.

b) Trouver la fonction simplifiée de la variable **S** déclenchant l'ouverture du coffre.

c) Donner un logigramme puis un schéma à contact correspondant au fonctionnement du dispositif.

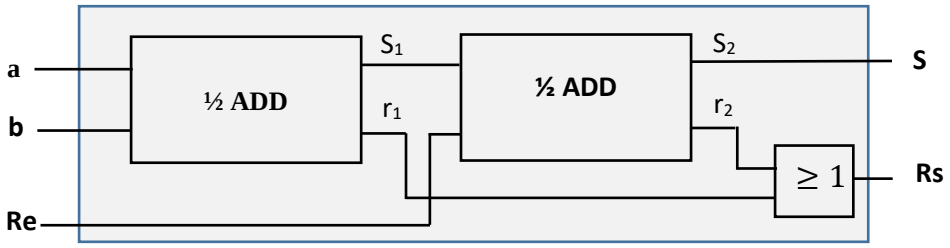
Exercice 6

Pour rappel, l'addition de deux nombres A et B peut s'écrire de façon générale :

$$\begin{array}{l} R_n\ R_{n-1} \quad R_i \qquad\qquad R_1\ R_0 \quad \} \textit{ Retenues} \\ a_n\ \ a_{n-1}.....a_i.....a_2\ a_1\ a_0 \\ \underline{b_n\ \ b_{n-1}.....b_i.....b_2\ b_1 b_0} \\ R_n\ S_n\ S_{n-1}\ .....S_i.....S_2 S_1 S_0 \end{array}$$

A- Câblage d'un additionneur complet

On veut réaliser un circuit électronique capable d'exécuter l'addition en binaire de 3 mots de 1 bit à partir du circuit ci-dessous. Dans cette structure nous retrouvons 2 additionneurs de 2 mots de 1 bit que nous appellerons ½ additionneur.



1. Donner une table de vérité du premier ½ additionneur.
2. En déduire les équations de S_1 et r_1 en fonction de a et b.
3. Donner le logigramme complet puis vérifier le fonctionnement.
4. Montrer que ce logigramme peut être réaliser à partir de 9 opérateurs NON-ET (NAND)

B- Câblage d'un additionneur de 2 mots de 8 bits

Donner le schéma de câblage d'un additionneur de 2 octets A et B à partir de schémas blocs d'additionneurs complets (on ne demande pas le logigramme complet).

$$\begin{array}{lll} \text{Mot A} \longrightarrow & & a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0 \\ \text{Mot B} \longrightarrow & + & b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0 \\ \text{Résultat A+B} \longrightarrow & & R\ S_7\ S_6\ S_5\ S_4\ S_3\ S_2\ S_1\ S_0 \end{array}$$