Logique booléenne

I. Définitions

I.1. Variable binaire

On appelle **variable binaire** (ou logique), une variable prenant ses valeurs dans l'ensemble {0, 1}.

Exemple : état d'un interrupteur, d'un bouton poussoir, la présence d'une tension,...

Soit ***a*** la variable associée à l'état d'un bouton poussoir, alors $a = 0$ (faux ou bas) signifie qu'il n'est pas actionné, $a = 1$ (vrai ou haut) signifie qu'il est actionné.

I.2. Equation logique

On appelle **équation logique** une combinaison de plusieurs variables logiques donnant l'état d'une variable dite de sortie associée. Cette combinaison est réalisée à l'aide d'opérations logiques :

Soit x_i ($i \in [1, n]$) les variables d'entrée. L'équation $A = f(x_i)$ définit l'état de la variable de sortie A.

I.3. Table de vérité

La **table de vérité** représente l'état de la variable de sortie pour chacune des combinaisons des n variables d'entrée (2^n lignes).

II. Les opérations logiques élémentaires

II.1. Opérateur OUI

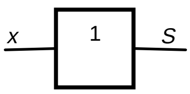
L'opération (ou opérateur) OUI est dite **unaire** (ne s'applique qu'à une seule opérande). Elle affecte à la variable de sortie l'état logique de la variable d'entrée.

Equation : x est la l'entrée, S la sortie : $S = x$.

x	S
0	0
1	1

Table de vérité

Symbole (norme IEC[1])



Symbole : norme IEEE[2]

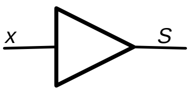
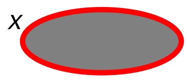


Diagramme de Venn
(représentation ensembliste)



Remarque : les anglo-américains notent H (*High*) le niveau haut et L (*Low*) le niveau bas.

II.2. Opérateur NON

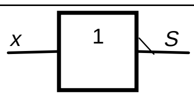
L'opération (ou opérateur) NON est la fonction unaire qui affecte à la variable de sortie l'état complémentaire de la variable d'entrée.

Equation : x est la l'entrée, S la sortie, $S = \overline{x}$ (prononcer « x barre »).

x	S
0	1
1	0

Table de vérité

Symbole (norme IEC)



Symbole (norme IEEE)

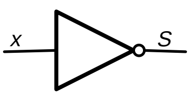
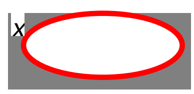


Diagramme de Venn



[1] IEC, *International Electrotechnical Commission* (CEI en français).

[2] IEEE, *Institute of Electrical and Electronics Engineers*.

II.3. Opérateur ET

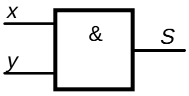
L'opération ET est le **produit logique**. Le signe est celui de la multiplication (un point), mais on lit « et ». C'est un opérateur **binaire** qui affecte à la variable de sortie l'état 1 si et seulement si les variables d'entrée sont à 1 simultanément.

Equation : x et y les entrées, S la sortie, $S = x.y = xy$.

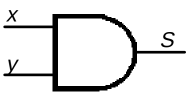
On note aussi l'opération ET par un V retourné :

$x.y = x \wedge y$ (penser à l'intersection d'ensembles).

Symbole (norme IEC)



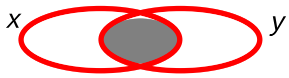
Symbole (norme IEEE)



x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Table de vérité

Diagramme de Venn



II.4. Opérateur OU

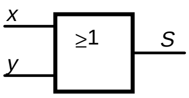
L'opération OU est la **somme logique**. Le signe est celui de l'addition (+), mais on lit « ou ». C'est un opérateur binaire qui affecte à la variable de sortie l'état 1 si et seulement si une variable d'entrée est à 1. Cette définition induit directement le symbole ≥1.

Equation : x et y les entrées, S la sortie, $S = x.y = xy$.

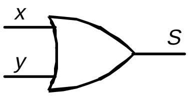
On note aussi l'opération OU par un V :

$x.y = x \vee y$ (penser à l'union d'ensembles).

Symbole (norme IEC)



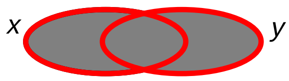
Symbole (norme IEEE)



x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Table de vérité

Diagramme de Venn



II.5. Remarques et compléments

Il est possible d'étendre la notion d'opération logique en utilisant des concepts plus « algébriques » :

- pour le NON logique : $\overline{x} = 1 - x$ avec $x \in \{0, 1\}$,
- pour le ET logique : $x.y = \mathrm{Min}(x,y)$ avec $(x,y) \in \{0, 1\}\times\{0, 1\}$,
- pour le OU logique : $x + y = \mathrm{Max}(x,y)$ avec $(x,y) \in \{0, 1\}\times\{0, 1\}$,

Ces notations sont aisément vérifiables à l'aide de tables de vérité.

III. Les opérations logiques induites

III.1. L'opération NON ET ou NAND

Cette fonction logique est le résultat de l'association d'un NON et d'un ET. C'est un opérateur binaire qui affecte à la variable de sortie l'état 0 si et seulement si les variables d'entrée sont à 1 simultanément

Equation : x et y les entrées, S la sortie, $S = \overline{x.y}$.

On note aussi l'opération NAND par une flèche montante :

$S = \overline{x.y} = x \uparrow y$ (penser $\wedge$).

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité

Symbole (norme IEC)

Symbole (norme IEEE)

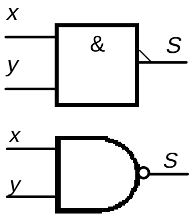
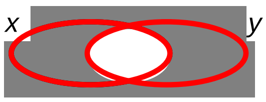


Diagramme de Venn



III.2. L'opération NON OU ou NOR

Cette fonction logique est le résultat de l'association d'un NON et d'un OU. C'est un opérateur binaire qui affecte à la variable de sortie l'état 1 si et seulement si les variables d'entrée sont à 0 simultanément.

x	y	$x \downarrow y$
0	0	1
0	1	0
1	0	0
1	1	0

Table de vérité

Equation : x et y les entrées, S la sortie, $S = \overline{x+y}$.

On note aussi l'opération NOR par une flèche descendante : $S = \overline{x+y} = x \downarrow y$ (penser $\vee$).

Symbole (norme IEC)

Symbole (norme IEEE)

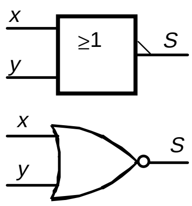
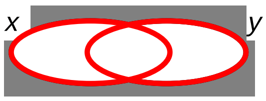


Diagramme de Venn



III.3. L'opération OU EXCLUSIF ou XOR

Cet opérateur logique binaire ne prend la valeur 1 que si une seule des entrées est à 1.

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Table de vérité

Equation : x et y les entrées, S la sortie, $S = x \oplus y$.

Symbole (norme IEC)

Symbole (norme IEEE)

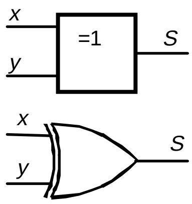
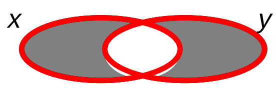


Diagramme de Venn



Généralisation

L'opérateur XOR se généralise à un ensemble de n variables d'entrée par la définition suivante :

La sortie vaut 1 si et seulement si le nombre d'entrées à 1 est impair.

Cet opérateur peut donc aisément faire fonction de contrôleur de parité (ou d'imparité).

IV. Les expressions logiques et leur simplification

Tous les opérateurs précédents permettent de combiner des variables pour en construire de nouvelles.

Exemple :

$$\left.\begin{array}{r} c = a + b.d \\ d = e + f \end{array}\right\} \Rightarrow c = a + b.(e + f)$$

IV.1. Propriétés

- Commutativité

  $a + b = b + a$ (commutativité de l'opération OU)

  $a.b = b.a$ (commutativité de l'opération ET)

- Associativité

  $a + (b + c) = (a + b) + c = a + b + c$ (associativité de l'opération OU)

  $(ab)c = a(bc) = abc$ (associativité de l'opération ET)

- Distributivité

  $(a + b).c = ac + bc$ (distributivité du produit logique sur la somme logique)

  $ab + c = (a + c).(b + c)$ (distributivité de la somme logique sur le produit logique)

Les parenthèses imposent une priorité supérieure.

IV.2. Autres propriétés

a est une variable logique

$a+\bar{a} = 1$	$a + 0 = a$	$a + 1 = 1$	$a + a = a$
$a.\bar{a} = 0$	$a.0 = 0$	$a.1 = a$	$a.a = a$

IV.3. Exemples

$$f = a + \bar{a}.b = a + b$$

$$abc + \bar{a}bc + a\bar{b}c + ab\bar{c} = bc(a+\bar{a}) + a\bar{b}c + ab\bar{c} = c(b + a\bar{b}) + ab\bar{c} = ac + bc + ab\bar{c} = ab + bc + ac$$

$$(a+b).c + (a+c)ab + b\bar{c} + a = ac + bc + ab + abc + b\bar{c} + a = a(c + b + bc + 1) + bc + b\bar{c} = a + b$$

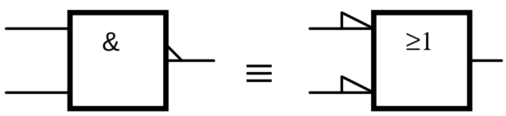
IV.4. Théorèmes de De Morgan[3]

<u>But</u> : exprimer les opérateurs ET, OU et NON exclusivement à l'aide d'opérateurs NOR seuls ou NAND seuls. On dit que les opérateurs NOR et NAND sont **universels** ou **complets**.

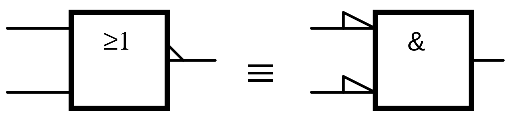
Premier théorème : $\overline{a+b} = \bar{a}\bar{b} \rightarrow$ généralisation $\boxed{\overline{\sum_{i=1}^{n} a_i} = \prod_{i=1}^{n} \bar{a}_i}$ où les les a_i sont les variables, $i \in [1,n]$.

Second théorème : $\overline{ab} = \bar{a} + \bar{b} \rightarrow$ généralisation $\boxed{\overline{\prod_{i=1}^{n} a_i} = \sum_{i=1}^{n} \bar{a}_i}$ où les les a_i sont les variables, $i \in [1,n]$.

Signification pratique



Un NAND est un OU à entrées complémentées



Un NOR est un ET à entrées complémentées

Expression des opérateurs de base à l'aide des seuls opérateurs universels.

- $\bar{a} = \overline{a.a} = a \uparrow a = \overline{a+a} = a \downarrow a$ Opérateur NON réalisé avec un NAND et avec un NOR
- $a.b = \overline{\overline{a.b}} = \overline{\bar{a}+\bar{b}} = \bar{a} \downarrow \bar{b}$ Trois opérateurs NOR (dont deux en NON)

  $a.b = \overline{\overline{a.b}} = \overline{a \uparrow b}$ Deux opérateurs NAND (dont un en NON) *Cas trivial*
- $a+b = \overline{\overline{a+b}} = \overline{\bar{a}.\bar{b}} = \bar{a} \uparrow \bar{b}$ Trois opérateurs NAND (dont deux en NON)

  $a+b = \overline{\overline{a+b}} = \overline{a \downarrow b}$ Deux opérateurs NOR (dont un en NON) *Cas trivial*

[3] De Morgan (Augustus), mathématicien et logicien britannique (1806-1871).

V. Ecriture des fonctions booléennes

V.1. Définitions

On appelle **minterme** de n variables, un produit logique de ces dernières (complémentées ou non). Avec n variables, on construit 2^n mintermes, c'est-à-dire autant que de combinaisons possibles de n éléments prenant deux états.

Exemple : pour 2 variables a et b, voici les 4 mintermes : $ab, \overline{a}b, a\overline{b}$ *et* $\overline{a}\overline{b}$.

On appelle **maxterme** de n variables, une somme logique de ces dernières (complémentées ou non). De la même manière que pour les mintermes, on construit 2^n maxtermes avec n variables.

Exemple : pour 2 variables a et b, voici les 4 maxtermes : $a+b, \overline{a}+b, a+\overline{b}$ *et* $\overline{a}+\overline{b}$.

V.2. Première forme canonique

La **première forme canonique** d'une expression booléenne est composée d'une somme de mintermes exclusivement. Pour une expression donnée cette forme est unique.

Exemple :

Remarque : la somme de tous les mintermes de n variables vaut toujours 1 puisqu'il existe toujours un minterme de n variables valant 1.

V.3. Seconde forme canonique

La **seconde forme canonique** d'une expression booléenne est composée d'un produit de maxtermes exclusivement. Pour une expression donnée cette forme est unique.

Exemple :

Remarque : Le produit de tous les maxtermes de n variables vaut toujours 0 puisqu'il existe toujours un maxterme de n variables valant 0.

Pour changer de forme canonique on effectue d'une double complémentation (involution) de l'expression suivie de l'application de l'un des théorèmes de De Morgan.

V.4. Forme canonique décimale

L'écriture des expressions logique a cet inconvénient d'être assez longue. Chaque minterme parmi les 2^n de n variables correspond à un nombre représentant son ordre, c'est pourquoi on préfère parfois utiliser une écriture indiquant la liste classée des numéros des mintermes de la première forme canonique.

Exemple : $F = abcd + \overline{a}bc\overline{d} + a\overline{b}c\overline{d} + \overline{a}\overline{b}\overline{c}\overline{d}$ peut aussi s'écrire $F(a, b, c, d) = \Sigma$ 0, 6, 10, 15.

VI. Extraction d'une équation logique à partir d'une table de vérité

Une table de vérité recense l'ensemble des états d'une sortie pour **<u>toutes</u>** les combinaisons possibles des variables d'entrée.

Pour trouver une expression sous la première forme canonique, on applique la méthode suivante :

- on définit les mintermes de n variables qui sont les expressions logiques bâties sur la combinaison de ces n variables ;
- chaque minterme est associé à l'une des combinaisons de la table de vérité (en conservant la correspondance 1 pour la variable et 0 pour la variable complémentée),
- tous les mintermes valant 1 sont sommés logiquement pour obtenir l'expression de la sortie.
- Les simplifications sont effectuées par les procédés de calcul algébrique.

VII. Applications — Exercices (sans corrigé)

VII.1. Expressions logiques

$$F_1 = ab + \overline{c} + c(\overline{a} + \overline{b})$$

$$F_2 = (a+b+c)(\overline{a}+b+c)+ab+bc$$

$$F_3 = (x\overline{y}+z)(x+\overline{y})z$$

$$F_4 = (\overline{a}b + a\overline{b})(ab+\overline{a}\overline{b})$$

$$F_5 = \overline{a}\,\overline{b}c\overline{d} + \overline{a}b\overline{c}d + a\overline{b}c\overline{d} + \overline{a}\,\overline{b}\,\overline{c}\,\overline{d} + ab\overline{c}d + \overline{a}bcd + a\overline{b}\,\overline{c}\,\overline{d}$$

VII.2. Logigrammes

Tracer les logigrammes des expressions logiques suivantes :

$S_1 = b\overline{c}\overline{d} + ab\overline{d} + \overline{a}bc\overline{d}$ $\qquad$ $S_2 = x + \overline{y}z + xy\overline{t} \oplus \overline{z}yt$

VII.3. Chronogrammes

Dessiner la forme d'onde en sortie S du système logique de la ***Figure 1***.

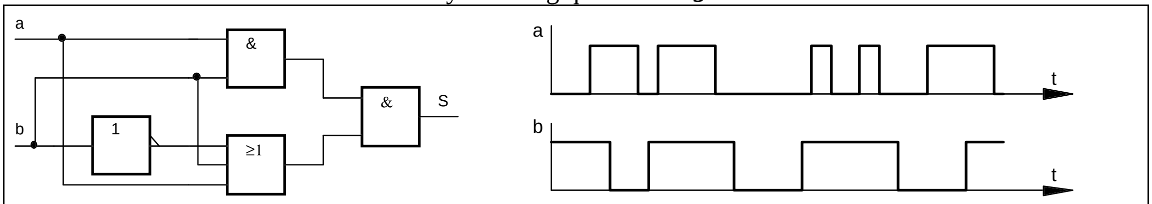


Figure 1

VII.4. Problème

Trois interrupteurs a, b et c commandent l'allumage de deux lampes R et S suivant les conditions :

- dès qu'un ou plusieurs interrupteurs sont activés la lampe R s'allume,
- la lampe S ne doit s'allumer que si au moins deux interrupteurs sont activés.

Trouver les expressions de R et S et dessiner les logigrammes.