Algèbre de BOOLE

SOMMAIRE :

Outils informatiques :

.
.
.

Sites « Web » :

.
.
.

Bibliographie :

.

.

1. PRESENTATION

L'algèbre de BOOLE est la logique utilisée par les ordinateurs. En automatique, que l'on soit en « combinatoire » ou en séquentiel, on prend en compte, on traite, on donne des ordres sous forme binaire (0 ou 1).
Les variables qui permettent de traiter ces informations peuvent s'organiser sous forme de fonctions.

Fonctions de base : OUI, NON, ET, OU.
Fonctions spécialisées : NON-OU (NOR), NON-ET (NAND), OU « exclusif » (XOR)

2. PROPRIETES

Somme : fonction OU		Produit : fonction ET		Négation : fonction NON
$0+0=0$	$a+1=1$	$0.0=0$	$a.1=a$	$\overline{1}=1$
$0+1=1$	$a+0=a$	$0.1=0$	$a.0=0$	$\overline{0}=0$
$1+0=1$	$a+a=a$	$1.0=0$	$a.a=a$	$\overline{a}=a$
$1+1=1$	$a+\overline{a}=1$	$1.1=1$	$a.\overline{a}=0$	

Commutativité : $a.b = b.a$
$a+b = b+a$

Associativité : $a.(b.c) = (a.b).c$
$a+(b+c) = (a+b)+c$

Distributivité : $a.(b+c) = a.b + a.c$ $(a+b).(c+d) = a.c + a.d + b.c + b.d$
$a+(b.c) = (a+b) . (a+c)$

Idempotence : $a+a = a$ $\overline{a}+\overline{a}=\overline{a}$

Absorption : $a+a.b=a$

Identités remarquables :
$a+a.b=a$
$a.b+a.c=a.(b+c)$
$a+\overline{a}.b=a+b$
$a.\overline{b}+b.c=(a+b).(\overline{b}+c)=a.\overline{b}+b.c+a.c$

Théorèmes de DE MORGAN :

1er Théorème : Le complément d'une somme logique est équivalent au produit logique des termes de ce produit, eux-mêmes complémentés.

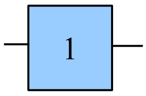
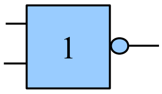
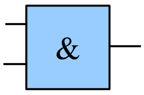
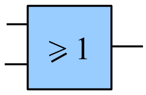
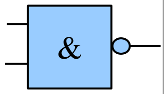
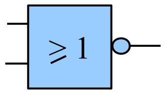
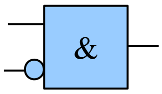
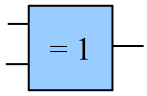
$$\overline{(a+b)}=\overline{a}.\overline{b}$$

2ème Théorème : Le complément d'un produit logique est équivalent à la somme logique des termes de cette produit, eux-mêmes complémentés.

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

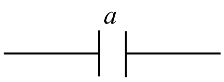
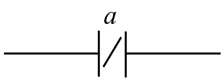
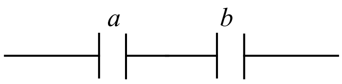
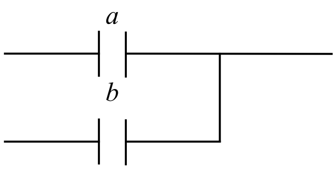
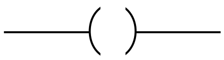
3. Représentations GRAPHIQUES

3.1. Logigrammes :

Symboles logiques	
Fonction OUI 	Fonction NON 
Fonction ET 	Fonction OU 
Fonction NON-ET (NAND) 	Fonction NON-OU (NOR) 
Fonction INHIBITION 	Fonction OU Exclusif : 

3.2. Schémas à contacts :

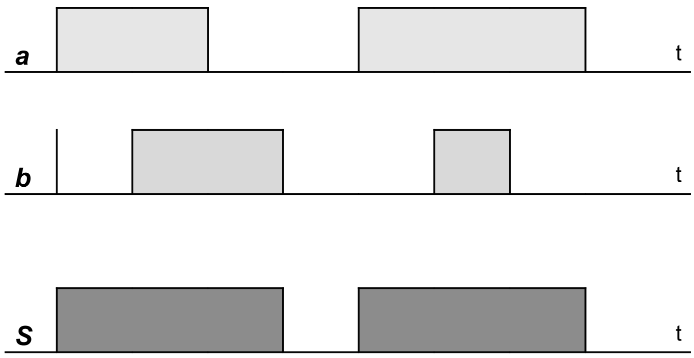
C'est la représentation la plus utilisée comme langage de programmationdans les automates programmables. L'analogie directe avec les schémas de circuits électriques permet une compréhension aisée.
Les contacts représentant les « entrées » sont toujours placés à gauche et les « sorties » à droite.

Symboles de base	
	: entrée a
	: entrée $\overline{a}$
	: $a \,.\, b$; $(a \text{ et } b)$
	: $a + b$; $(a \text{ ou } b)$
	: sortie S

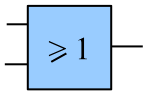
3.3. Chronogrammes :

Les chronogrammes sont une représentation graphique de l'évolution de l'état de variables d'entrées et de sorties, mais également les variables « internes » d'un système automatisé.

L'abscisse représente le temps et l'ordonnée l'état logique (0 ou 1) de la variable. Les graphes sont placés les uns au-dessus des autres avec une même échelle des temps.



$S = a + b$



4. Simplifications des expressions booléennes

A partir des tables de vérité, on obtient des relations ou équations logiques souvent complexes et difficilement traduisibles en schémas ou programmes. Il est donc nécessaire de simplifier ces équations logiques.

4.1. Méthode algébrique :

La simplification d'une équation logique se fait très souvent par « calcul » algébrique en cherchant à mettre en facteur les variables et en utilisant les propriétés des fonctions logiques vues au chapitre 2.

Exemple 1 :

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

$$S = a.b + c$$

Exemple 2 :

$$S = \bar{a}.b.\bar{c} + a.b.\bar{c} + b.\bar{c}.d$$

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

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

$$S = \bar{c}.(b.(1 + d))$$

$$S = \bar{c}.(b.1)$$

$$S = \bar{c}.b = b.\bar{c}$$

4.2. Méthode des tableaux de Karnaugh :

La table de vérité donne un aperçu rapide de l'état des variables de sorties en fonction de l'état des variables d'entrées. On tire de cette table de vérité une équation booléenne qu'il faut simplifier.

Les tableaux de Karnaugh sont utiles pour simplifier les équations logiques tirées de tables de vérité.

Principe : Chaque case du tableau de Karnaugh correspond à un état de la variable de sortie.

A	B	S
0	0	1
0	1	0
1	1	1
1	0	0

Simplification : On tire des cases l'équation de la sortie. On regroupe les cases adjacentes pour simplifier.

	A = 0	A = 1
B = 0	1	0
B = 1	0	1

Ici, $S = \bar{A}.\bar{B} + A.B$

On simplifie également les équations en regroupant les cases de 1 en carré ou en lignes de 4 cases. On peut regrouper en 8 cases. Les groupements de cases peuvent aussi être réalisés sur les cotés des tableaux qui sont assimilés à des cylindres : « boites à Camembert ».

<table>
<tr><th colspan="4">Exemples de simplification par tableaux de Karnaugh</th></tr>
<tr><td>Équation à simplifier</td><td>$S=\bar{a}.\bar{b}+a.\bar{b}$</td><td>$S=a.\bar{b}+a.b$</td><td>$S=\bar{a}.\bar{b}+a.\bar{b}+a.b$</td></tr>
<tr><td>Tableau de Karnaugh</td>
<td><table><tr><td></td><td>$a=0$</td><td>$a=1$</td></tr><tr><td>$b=0$</td><td>1</td><td>1</td></tr><tr><td>$b=1$</td><td>0</td><td>0</td></tr></table></td>
<td><table><tr><td></td><td>$a=0$</td><td>$a=1$</td></tr><tr><td>$b=0$</td><td>0</td><td>1</td></tr><tr><td>$b=1$</td><td>0</td><td>1</td></tr></table></td>
<td><table><tr><td></td><td>$a=0$</td><td>$a=1$</td></tr><tr><td>$b=0$</td><td>1</td><td>1</td></tr><tr><td>$b=1$</td><td>0</td><td>1</td></tr></table></td></tr>
<tr><td>Équation simplifiée</td><td>$S=\bar{b}$</td><td>$S=a$</td><td>$S=a+\bar{b}$</td></tr>
</table>

Remarque : Certaines combinaisons des entrées ne correspondent à aucun cas de fonctionnement du système. Dans la mesure où ces cas sont indifférents ou sans effet, ils peuvent être affectés de la valeur 0 ou 1 en fonction des besoins de la simplification.