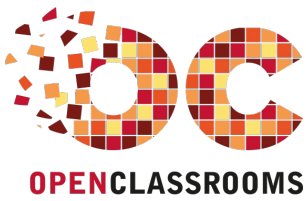
L'algèbre de Boole

Par alcinos



www.openclassrooms.com

Dernière mise à jour le 7/03/2012

Sommaire

L'algèbre de Boole

Par alcinos
Mise à jour : 07/03/2012
Difficulté : Facile Durée d'étude : 1 jour

Nous allons dans ce tuto étudier l'algèbre de boole. Derrière ce nom barbare se cache en fait quelque chose que tout programmeur, même débutant, utilise sans le savoir, dans les conditions.
Il est assez peu utile dans la plupart des cas , mais il peut avoir son utilité en algorithmique, pour la recherche de performances au niveau des conditions, en SQL, pour les requêtes, ou en électronique, pour limiter le nombre de câblage de portes logiques.
Sommaire du tutoriel :

Les variables booléennes

L'algèbre de Boole est une structure algébrique qui ne contient que deux éléments, que l'on appelle couramment variables booléennes. Ces variables ne peuvent avoir que deux états, 1 ou 0 (true ou false dans certains langages de programmation), et respectent quelques règles de calcul que nous détaillerons plus loin.

Mais quel est le rapport avec les conditions ?

J'y viens . En fait n'importe quel test exécuté dans une condition est une variable booléenne.
Prenons un exemple

Code : Autre

```
SI (A=B) ALORS ...
```

Le test (A=B) peut avoir deux valeurs : 1 si A est réellement égal à B et 0 sinon. On dit que c'est la variable booléenne associée au test A=B.

Les opérateurs logiques

L'algèbre de Boole utilise plusieurs opérateurs que l'on nomme opérateurs booléens, opérateurs logiques, ou encore fonctions logiques ou portes logiques (terme plus propre à l'électronique).
En voici les principaux.

L'opérateur OU

Il correspond à la **réunion** de deux conditions. Par exemple : je souhaite acheter une voiture bleue OU climatisée. Les voitures correspondant à mon choix seront donc soit bleue, soit climatisée, soit les deux à la fois.

On symbolise l'opérateur OU dans les conditions par un OR ou ||
En algèbre de Boole, il est symbolisée par un +

La matheux préfèrent quant à eux la symbolisation $\vee$

Exemple :

Code : C

```
if(a==1 OR a==2){
```

Équivaut à

Code : Autre

```
SI((a=1)+(a=2)) alors
```

Si on prend des variables booléennes, on obtient :
$a \,\mathrm{OU}\, b \Leftrightarrow a + b$

Table de vérité :

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

Dans ce tableau on voit le résultat du test $a + b$ en fonction des valeurs de a et de b

L'opérateur ET

Il correspond à l'**intersection** de deux conditions. Par exemple : je souhaite acheter une voiture bleue ET climatisée. correspondant à mon choix seront donc à la fois bleues et climatisées.

On symbolise l'opérateur ET dans les conditions par un AND ou &&
En algèbre de Boole, il est symbolisée par un ·
En maths on écrira plutôt $\wedge$

Exemple :

Code : C

```
if(a==1 AND a==2){
```

Équivaut à

Code : Autre

```
SI((a=1).(a=2)) alors
```

Si on prend des variables booléennes, on obtient :
$a \,\mathrm{ET}\, b \Leftrightarrow a \cdot b$

Table de vérité :

a	b	a.b
0	0	0

1	0	0
0	1	0
1	1	1

Dans ce tableau on voit le résultat du test $a \cdot b$ en fonction des valeurs de a et de b

L'opérateur NON

Il correspond au **complément à 1** d'une condition. Par exemple : je souhaite acheter une voiture NON polluante. Les voitures correspondant à mon choix seront donc écologiques.

L'opérateur NON est souvent représenté dans les conditions par un !
En algèbre de Boole, il est symbolisée par une barre au dessus de(s) variable(s)
En maths on utilise le symbole ¬
Exemple :

Code : C

```
if(a!=1){
```

Équivaut à

SI $\left(\overline{a = 1}\right)$ alors ...

Si on prend des variables booléennes, on obtient :
!a équivaut à $\overline{a}$

Table de vérité :

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

Dans ce tableau on voit le résultat du test $\overline{a}$ en fonction des valeurs de a

L'opérateur XOR (OU exclusif)

Il correspond à l'**intersection** de deux conditions, **privée de la réunion** de ces deux conditions. Par exemple : le menu propose du fromage OU EXCLUSIF un dessert : Je peux prendre soit l'un soit l'autre, mais pas les deux.

L'opérateur XOR est représenté dans les conditions par un XOR
En algèbre de Boole, il est symbolisée par un $\oplus$
Exemple :

$a \,\mathrm{XOR}\, b \Leftrightarrow a \oplus b$

Table de vérité :

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

Dans ce tableau on voit le résultat du test $a \oplus b$ en fonction des valeurs de a et de b

Note : $a \oplus b = \overline{a} \cdot b + a \cdot \overline{b}$

L'opérateur NXOR (NON OU exclusif)

Cet opérateur n'a pas de symbolisation propre, on le note simplement NXOR

Table de vérité :

a	b	a NXOR b
0	0	1
1	0	0
0	1	0
1	1	1

Cet opérateur est appelé opérateur **coincidence**. En effet le test "$a \,\mathrm{NXOR}\, b$" n'est vrai que si a et b sont dans le même état logique (0 ou 1)

Note : $a \,\mathrm{NXOR}\, b = a \cdot b + \overline{a} \cdot \overline{b}$

L'opérateur NAND (NON ET) ou connecteur de Sheffer

La fonction NAND est l'enchainement de la fonction ET et de la fonction NON. On la symbolise par ↑

Table de vérité :

a	b	a↑b
0	0	1
1	0	1
0	1	1
1	1	0

L'opérateur NAND est appelé opérateur universel : il est possible de recréer n'importe quelle fonction logique uniquement en utilisant la fonction NAND. Nous verrons plus loin dans ce tuto comment.

Note : $a \uparrow b = \overline{a \cdot b} = \overline{a} + \overline{b}$

L'opérateur NOR (NON OU)

La fonction NOR est l'enchainement de la fonction OU et de la fonction NON. On la symbolise par ↓

Table de vérité :

a	b	a↓b
0	0	1
1	0	0
0	1	0
1	1	0

L'opérateur NOR est également un opérateur universel.

Note : $a \downarrow b = \overline{a + b} = \overline{a} \cdot \overline{b}$

Régles de simplification

Entrons maintenant dans le vif du sujet en apprenant les règles de l'algèbre, qui vont nous permettre de simplifier nos conditions.

Règle 1 : la commutativité

Même si elle parait évidente, il me semble bon de la préciser :

$$a + b = b + a$$
$$a \cdot b = b \cdot a$$

Règle 2 : l'idempotence

Un nom barbare pour une propriété simple :

$$a + a = a$$
$$a \cdot a = a$$

Règle 3 : l'élément nul

Comme son nom l'indique, il prend le dessus dans l'égalité :

$$a + 1 = 1$$
$$a \cdot 0 = 0$$

Règle 4 : l'élément neutre

Il ne change rien à l'égalité :

$$a \cdot 1 = a$$
$$a + 0 = a$$

Règle 5 : l'associativité

Elle parait évidente aussi :

$$a + (b + c) = (a + b) + c$$
$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

Règle 6 : la complémentarité

$$\overline{\overline{a}} = a \quad (\text{Non Non} = \text{Oui})$$
$$a + \overline{a} = 1$$
$$a \cdot \overline{a} = 0$$

Règle 7 : la distributivité

Attention, les deux lois sont distributives l'une par rapport à l'autre, contrairement à l'algèbre classique

$$a \cdot (b + c) = a \cdot b + a \cdot c$$
$$a + (b \cdot c) = (a + b) \cdot (a + c)$$

Règle 8 : théorème de De Morgan

Règle très utile, mais attention aux erreurs de changement d'opérateur

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

Et les autres

Il existe d'autres règles qui découlent plus ou moins simplement des précédentes. Notamment :
$a + \overline{a}.b = a + b$

Démonstration :

$$\begin{aligned}
\overline{a + \overline{a} \cdot b} &= \overline{a} \cdot \overline{\overline{a} \cdot b} && \text{(Règle 8)} \\
&= \overline{a} \cdot (\overline{\overline{a}} + \overline{b}) && \text{(Règle 8)} \\
&= \overline{a} \cdot (a + \overline{b}) && \text{(Règle 6)} \\
&= \overline{a} \cdot a + \overline{a} \cdot \overline{b} && \text{(Règle 7)} \\
&= 0 + \overline{a} \cdot \overline{b} && \text{(Règle 6)} \\
&= \overline{a} \cdot \overline{b} && \text{(Règle 4)} \\
&= \overline{a + b} && \text{(Règle 4)}
\end{aligned}$$

Donc $\overline{a + \overline{a}.b} = \overline{a + b}$, soit $a + \overline{a}.b = a + b$

Autre règle :
$a + a.b = a$

Démonstration :

$$\begin{aligned}
a + a \cdot b &= a \cdot 1 + a \cdot b && \text{(Règle 4)} \\
&= a \cdot (1 + b) && \text{(Règle 7)} \\
&= a \cdot 1 && \text{(Règle 3)} \\
&= a && \text{(Règle 4)}
\end{aligned}$$

Voilà, vous connaissez maintenant les principales règles de l'algèbre de boole, vous en savez assez pour simplifier la grande majorité des équations. Pour les plus récalcitrantes, un autre outil existe, il s'agit du tableau de Karnaugh

Spécificités des portes composées

Dans cette partie nous allons vois les règles qui s'appliquent (ou pas) aux portes logiques composées, c'est à dire XOR, NXOR, NAND, et NOR.

L'universalité des portes NAND et NOR

La porte NAND

Comme je vous l'ai dit précédemment, il est possible de réaliser n'importe quelle fonction logique de base en utilisant uniquement des portes NAND. C'est notamment très utile en électronique, puisqu'on peut alors produire en masse (et donc moins coûteusement) uniquement ces portes la, pour câbler ensuite n'importe quelle équation logique.

La fonction NON : $a \uparrow 1 = \overline{a \cdot 1} = \overline{a}$

La fonction ET : $(a \uparrow b) \uparrow 1 = \overline{\overline{a \cdot b} \cdot 1} = \overline{\overline{a \cdot b}} = a.b$

La fonction OU : $(a \uparrow 1) \uparrow (b \uparrow 1) = \overline{\overline{a \cdot 1} \cdot \overline{b \cdot 1}} = \overline{\overline{a} \cdot \overline{b}} = a + b$

La porte NOR

La porte NOR possède les mêmes spécificités :

La fonction NON : $a \downarrow 0 = \overline{a+0} = \overline{a}$

La fonction ET : $(a \downarrow 0) \downarrow (b \downarrow 0) = \overline{\overline{a+0}+\overline{b+0}} = \overline{\overline{a}+\overline{b}} = a.b$

La fonction OU : $(a \downarrow b) \downarrow 0 = \overline{\overline{a+b}+0} = \overline{\overline{a+b}} = a+b$

Règles de calcul spécifiques

Pour effectuer des simplifications avec des portes composées, il existe deux solutions. La première consiste a transformer la fonction composée de façon à n'utiliser que les portes de base (NON, OU, ET). Il faut pour cela utiliser les équivalences que j'ai indiqué à la définition de chaque porte. Par exemple : $a \oplus b \Leftrightarrow a.\overline{b} + \overline{a}.b$ On peut alors utiliser les règles de simplifications listées plus haut.

Le deuxième choix qui s'offre à nous est d'utiliser les règles de calcul spécifiques aux portes composées (utile par exemple s'il faut garder un seul type de porte dans les équations)

Règles spécifiques à la porte XOR

Quelques règles qui sont en fait des propriétés aisément déductibles des règles précédentes

- $a \oplus 1 = \overline{a}$
- $a \oplus 0 = a$
- $a \oplus a = 0$
- $a \oplus \overline{a} = 1$

Règles spécifiques à la porte NAND

Voici les propriétés de la porte NAND

- $a \uparrow 1 = \overline{a}$
- $a \uparrow 0 = 1$
- $a \uparrow a = \overline{a}$
- $a \uparrow \overline{a} = 1$

Il n'y a PAS d'associativité avec les portes NAND : $a \uparrow (b \uparrow c) \neq (a \uparrow b) \uparrow c$

Règles spécifiques à la porte NOR

Voici les propriétés de la porte NOR

- $a \downarrow 1 = 0$
- $a \downarrow 0 = \overline{a}$
- $a \downarrow a = \overline{a}$
- $a \downarrow \overline{a} = 0$

Il n'y a PAS non plus d'associativité avec les portes NOR : $a \downarrow (b \downarrow c) \neq (a \downarrow b) \downarrow c$

Exercices

Bon voilà, ça fait beaucoup de connaissances à ingurgiter d'un coup . Il va falloir s'entrainer un peu pour vérifier que tout est bien acquis .

Exercice 1

Bon on va commencer doucement . Munissez vous d'un papier, d'un crayon et surtout de patience, car ce n'est pas forcement évident au début .

Simplifier : $a \cdot (a + b)$

Solution (cherchez d'abord !)

Secret (cliquez pour afficher)

$$\begin{aligned} a \cdot (a + b) &= a \cdot a + a \cdot b && \text{(Règle 7)} \\ &= a + a \cdot b && \text{(Règle 2)} \\ &= a && \text{(Voir la démonstration de la règle complémentaire)} \end{aligned}$$

Facile non ?
Bon, donc on peut compliquer un peu

Exercice 2

Légèrement plus difficile :

Simplifier : $a + \overline{a}.b + \overline{a}.\overline{b}$

Solution :

Secret (cliquez pour afficher)

Pour cette équation (comme souvent), il y a plusieurs façon de procéder. Je vais en donner 3 cette fois ci mais à l'avenir je n'en donnerai qu'une : pas d'inquiétude donc si ce n'est pas la votre, c'est le résultat qui compte

Solution 1 :

$$\begin{aligned} a + \overline{a} \cdot b + \overline{a} \cdot \overline{b} &= a + \overline{a} \cdot (b + \overline{b}) && \text{(Règle 7)} \\ &= a + \overline{a} \cdot 1 && \text{(Règle 6)} \\ &= a + \overline{a} && \text{(Règle 4)} \\ &= 1 && \text{(Règle 6)} \end{aligned}$$

Solution 2 :

$$\begin{aligned} a + \overline{a} \cdot b + \overline{a} \cdot \overline{b} &= a + b + \overline{a} \cdot \overline{b} && \text{(Règle Complémentaire)} \\ &= a + b + \overline{b} && \text{(Règle Complémentaire)} \\ &= a + 1 && \text{(Règle 6)} \\ &= 1 && \text{(Règle 3)} \end{aligned}$$

Solution 3 :

$$\begin{aligned} a+\overline{a}\cdot b+\overline{a}\cdot\overline{b} &= a+b+\overline{a}\cdot\overline{b} && \text{(Règle Complémentaire)}\\ &= a+b+\overline{a+b} && \text{(Règle 8)}\\ &= (a+b)+\overline{(a+b)} &&\\ &= 1 && \text{(Règle 6)} \end{aligned}$$

Si vous avez des difficulté à comprendre ces exercices, un petit retour à la partie précédente s'impose

Remarque : Dans l'exemple précédent, on remarque que quelles que soient les états logiques de a et b, l'équation proposée vaut toujours 1. On parle alors de **tautologie**

Exercice 3

Plus difficile :

Simplifier : $a+b\cdot\overline{c}+\overline{a}\cdot\overline{(b\cdot\overline{c})}\cdot(a\cdot d+b)$, puis transformer l'expression obtenue de manière à n'utiliser que l'opérateur NAND

Secret (cliquez pour afficher)

$$\begin{aligned} a+b\cdot\overline{c}+\overline{a}\cdot\overline{(b\cdot\overline{c})}\cdot(a\cdot d+b) &= a+b\cdot\overline{c}+\overline{a}\cdot(\overline{b}+c)\cdot(a\cdot d+b) && \text{(Règle 8)}\\ &= a+b\cdot\overline{c}+(\overline{a}\cdot\overline{b}+\overline{a}\cdot c)\cdot(a\cdot d+b) && \text{(Règle 7)}\\ &= a+b\cdot\overline{c}+\overline{a}\cdot\overline{b}\cdot a\cdot d+\overline{a}\cdot\overline{b}\cdot b+\overline{a}\cdot c\cdot a\cdot d+\overline{a}\cdot c\cdot b) && \text{(Règle 7)}\\ &= a+b\cdot\overline{c}+0\cdot\overline{b}\cdot d+\overline{a}\cdot 0+0\cdot c\cdot d+\overline{a}\cdot c\cdot b) && \text{(Règle 6)}\\ &= a+b\cdot\overline{c}+0+0+0+\overline{a}\cdot c\cdot b && \text{(Règle 3)}\\ &= a+b\cdot\overline{c}+\overline{a}\cdot(c\cdot b) && \text{(Règle 4)}\\ &= b\cdot\overline{c}+a+\overline{a}\cdot(c\cdot b) && \text{(Règle 1)}\\ &= b\cdot\overline{c}+a+(c\cdot b) && \text{(Règle Complémentaire)}\\ &= b\cdot(\overline{c}+c)+a && \text{(Règle 7)}\\ &= b\cdot(1)+a && \text{(Règle 6)}\\ &= b+a && \text{(Règle 3)}\\ &= (a\uparrow 1)\uparrow(b\uparrow 1) && \text{(Voir porte NAND)} \end{aligned}$$

Ne désespérez pas si vous n'y arrivez pas du premier coup, car l'algèbre de Boole, malgré ses ressemblances, est assez différent de l'algèbre classique.

Exercice 4

Exercice sur le théorème de De Morgan :

Simplifier : $\overline{\overline{a}\cdot\overline{\overline{b}\cdot\overline{c}}\cdot\overline{b}\cdot\overline{a}}$, puis transformer l'expression obtenue de manière à n'utiliser que l'opérateur NAND

Indice :

Secret (cliquez pour afficher)

Simplifiez les barres de haut en bas

Solution

Secret (cliquez pour afficher)

$$\begin{aligned}
\overline{\overline{a}\cdot\overline{\overline{b}\cdot\overline{c}\cdot\overline{b}}\cdot\overline{a}} &= a+\overline{\overline{\overline{b}\cdot\overline{c}\cdot\overline{b}}\cdot\overline{a}} && \text{(Règle 8)}\\
&= a+\left(\overline{b}\cdot\overline{c}+b\right)\cdot\overline{a} && \text{(Règle 8)}\\
&= a+\left(\overline{c}+b\right)\cdot\overline{a} && \text{(Règle Complémentaire)}\\
&= a+\left(\overline{c}+b\right) && \text{(Règle Complémentaire)}\\
&= a+\overline{c}+b\\
&= (a\uparrow 1)\uparrow(b\uparrow 1)\uparrow c && \text{(Voir porte NAND)}
\end{aligned}$$

La table de vérité

Nous avons déjà abordé les tables de vérité dans la 2ème partie, sans trop s'y attarder, chose que nous allons faire à présent.
Une table de vérité permet de connaitre l'état logique S de l'équation, en fonction des états des variables.
La table de vérité la plus simple est la suivante :

a	S
0	0
1	1

Ici on voit que quand la variable a vaut 1 S vaut 1 et quand a vaut 0, S vaut 0.

Pour remplir, une table de vérité, plusieurs étapes sont à respecter :

1. Placer en haut de chaque colonne le nom de toutes les variables logiques et dans la dernière le nom de la variable résultat (généralement S)
2. Placer dans les colonnes sous les noms de variables TOUTES les combinaisons des variables dont on connait le résultat logique (il ne doit pas y avoir 2 fois la même combinaison)

Exemple :

On veut faire la table de vérité d'une équation S inconnue a 2 variables logique, que l'on nomme ici a et b.

On commence à remplir le tableau

a	b	S

Admettons maintenant qu'on connaisse S quel que soit l'état de a et de b. On rempli donc le tableau avec toutes les combinaisons de a et de b possibles :

a	b	S
0	0	?
0	1	?
1	0	?
1	1	?

Maintenant on remplit avec les valeurs de S que l'on connait (je vous les donne) :

a	b	S
0	0	1
0	1	0

1	0	0
1	1	1

Euh, c'est bien beau tout ça mais à quoi ça sert ??

Minute papillon, j'y viens

Grâce à cette table de vérité, on va pouvoir retrouver l'équation logique reliant a et b !

1ère méthode : la forme normale disjonctive (DNF)

Il va s'agir d'écrire l'équation sous la forme d'une somme de produits. Pour cela, il faut repérer dans la table de vérité toutes les combinaisons pour lesquelles S vaut 1.
On les surligne :

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

On a 2 lignes surlignées : on aura donc une somme de 2 produits.
On écrit ces produits en fonction de la combinaison des variables : si une variable vaut 0 alors on écrit : $\overline{nom\ \ de\ \ la\ \ variable}$, sinon on écrit simplement le nom de la variable.

Dans notre cas, on obtient donc :

- 1ère ligne
  - a vaut 0 donc on écrit $\bar{a}$
  - b vaut 0 donc on écrit $\bar{b}$
- 4ème ligne
  - a vaut 1 donc on écrit a
  - b vaut 1 donc on écrit b

En français, pour que S soit égal à 1, il faut que a=0 ET b=0, OU que a=1 ET b=1
On traduit donc par l'égalité suivante :

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

On a bien une somme de 2 produits.

On appelle parfois les produits issus d'une seule case de la table de vérité des **mintermes**. Il est possible de démontrer que la DNF (non simplifiée) est unique à l'ordre des termes près.

2ème méthode : la forme normale conjonctive

On veut cette fois obtenir un produit de sommes. Il faut pour cela repérer toutes les combinaisons pour lesquelles S vaut 0 :
On les surligne dans le tableau :

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

On a 2 lignes surlignées : on aura donc un produit de 2 sommes.
On écrit ces somme en fonction de la combinaison des variables : si une variable vaut 0 alors on écrit : $\overline{nom\ \ de\ \ la\ \ variable}$, sinon on écrit simplement le nom de la variable.

Dans notre cas, on obtient donc :

- 2ème ligne
  - a vaut 0 donc on écrit $\bar{a}$
  - b vaut 1 donc on écrit b
- 3ème ligne
  - a vaut 1 donc on écrit a
  - b vaut 1 donc on écrit $\bar{b}$

En français, pour que S soit égal à 1, il ne faut PAS que a=0 ET b=1, OU que a=1 ET b=0
On traduit donc par l'égalité suivante :
$\overline{\bar{a}.b} + \overline{a.\bar{b}}$
La grande barre symbolise le "PAS". en effet, on ne veut que les cas ou S vaut 1
On simplifie ensuite en utilisant le théorème de De Morgan :
$\overline{\bar{a}.b} + \overline{a.\bar{b}} = \overline{(\bar{a}.b)}.\overline{(a.\bar{b})}$
$= (a + \bar{b}).(\bar{a} + b)$
On a bien un produit de 2 sommes.
On peut vérifier que cette égalité est équivalente à la précédente :
On distribue :
$= a.\bar{a} + a.b + \bar{b}.\bar{a} + \bar{b}.b$
On simplifie ce qui doit l'être :
$= a.\bar{b} + b.\bar{a}$
Et on retrouve bien l'égalité du 1)

Les formes normales, du fait de leur unicité, permettent de vérifier très rapidement l'égalité deux deux équations booléennes. En revanche, elles sont en générales très lourdes à manier puisque faisant intervenir tous les termes : il convient donc de les simplifier pour pouvoir les exploiter.
On se rend en revanche assez vite compte qu'une simplification à l'aide des règles usuelles est très fastidieuse, et source de nombreuses erreurs. C'est pourquoi nous allons maintenant étudier un outil puissant pour simplifier les tables de vérités complexes : le tableau de Karnaugh.

Le tableau de Karnaugh

Le tableau de Karnaugh est un outil qui permet de simplifier des équations booléennes, avec moins de chances de se tromper qu'en utilisant les règles de l'algèbre de Boole classique. On peut l'utiliser avec au maximum 5-6 variables, au-delà il devient rapidement inefficace.

Créer et remplir un tableau de Karnaugh

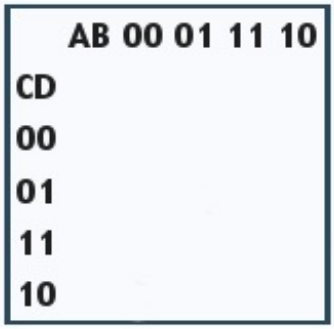
A partir d'une table de vérité

Prenons la table de vérité suivante :

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

Avec 4 variables, on a 2^4=16 combinaisons. On va donc créer un tableau de 4 colonnes et 4 lignes (4*4=16), pour ainsi avoir toutes les combinaisons :



C'est un peu comme une bataille navale, sauf que chaque case a 4 indices : l'état des 4 variables. Les cases 'AB' et 'CD' indiquent l'ordre des variables sur la ligne ou colonne associée : par exemple, sur la ligne 1, on voit la case 'AB', ce qui implique que '00' signifie A=0 et B=0 ou encore $\overline{a}$ et $\overline{b}$

ATTENTION : c'est très important que d'une case consécutive à l'autre, une seule variable change d'état ! Sans cela, tous ce que vous pourrez entreprendre par la suite sera tout bonnement **faux**. Ce tableau donnera par exemple des résultats faux :

En effet, pour passer de la case C0 à la case C1, les A et B changent toutes deux d'état : A passe de 0 à 1, et B de 1 à 0.

Ces mises en gardes terminées, on peut maintenant remplir le tableau en fonction de notre table de vérité : on commence par la première ligne de la, qui est, je vous le rappelle :

a	b	c	d	S
0	0	0	0	0

On va dans le tableau, et on cherche la case d'indice a=0, b=0,c=0 et d=0, c'est à dire la première case, et on y inscrit la valeur de S donnée par la table de vérité (ici 0) :

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

On continue avec la 2ème ligne de la table :

a	b	c	d	S
0	0	0	1	1

Ce qui donne :

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

Et ainsi de suite pour toutes les autres cases. On obtient donc :

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

A partir d'une équation booléenne

On peut également remplir un tableau directement à l'aide d'une équation booléenne. Cela permet par exemple de la simplifier plus qu'avec les règles de l'algèbre.
Pour cela il faut avoir une équation sous la forme d'une somme de produit.
Exemple : $\overline{a}.b.\overline{c}.\overline{d} + a.\overline{b}.c + a.c + b.d + a.b$
On isole le premier produit et on identifie les variables :
$\overline{a}.b.\overline{c}.\overline{d}$
On va donc chercher la case du tableau qui vérifie ces conditions : $a = 0$ $b = 1$ $c = 0$ $d = 0$, c'est à dire un cas où l'équation est vraie, et on ajoute un 1 dans la case

AB / CD	00	01	11	10
00		1		
01				
11				
10				

On passe au deuxième produit :

$a.\bar{b}.c$

On voit que dans ce produit la variable d n'apparait pas. Cela signifie que l'équation est vraie si :

$a = 1 \quad b = 0 \quad c = 1 \quad et \quad d = 0 \quad OU \quad d = 1$

AB / CD	00	01	11	10
00				
01				
11				1
10				1

On procède de même pour les trois autres produits, où il y a à chaque fois deux variables indéterminées, ce qui signifie qu'on rempli 4 cases pour chaque produit. On obtient finalement :

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

Remarque : il est fort probable que plusieurs produits conduisent à remplir la même case, c'est tout à fait normal

Retrouver l'équation booléenne simplifiée

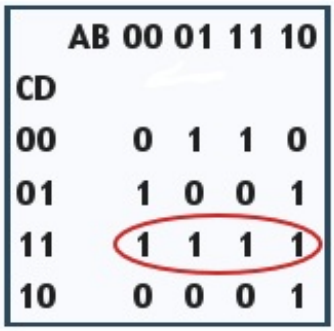
Réaliser les groupements

Il s'agit maintenant de réaliser des "groupements" de 1 : on va entourer des groupes de 1 dans le tableau en respectant quelques règles :

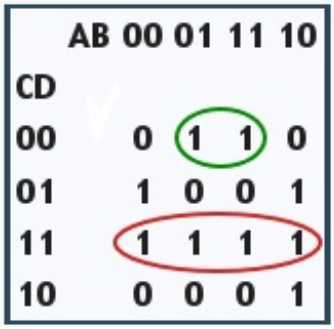
- Les groupements peuvent utiliser plusieurs fois le même 1
- Ils ne doivent pas contenir de 0
- Tous les 1 doivent être contenus dans un groupement
- Ils doivent être les plus grands possible
- Ils doivent être rectangulaire (ou carré)
- Leur dimensions doivent être des puissances entières de 2 (2^0=1, 2^1=2, 2^2=4, etc) Exemples : un carré de 2 par 2 ou un rectangle de 1 par 4
- On imagine que les lignes et colonnes sont bouclées : on peut utiliser par exemple un 1 de la première colonne avec un de la dernière colonne

Reprenons le tableau rempli précédemment avec la table de vérité. On commence par le groupement le plus évident, le rectangle

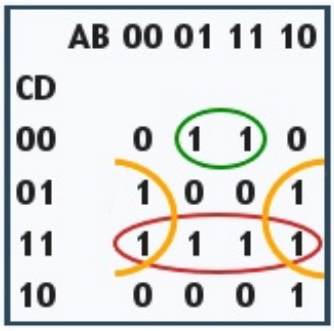
de 4 par 1 :



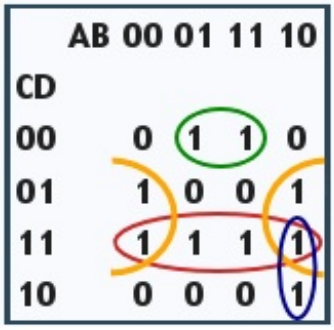
Ensuite vient le groupement de 2 par 1 du haut :



On peut se servir du fait que les lignes sont bouclées et qu'on peut réutiliser plusieurs fois le même 1 pour faire un groupement de 2 par 2 :



Il ne reste plus qu'un seul 1 que l'on place dans un groupement le plus grand possible, 2 par 1 (le groupement de 3 par 1 ne serait pas possible car 3 n'est pas une puissance entière de 2) :

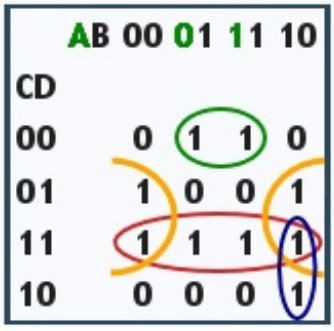


Déduire l'équation simplifiée

Maintenant on va déduire chaque groupement réalisé une partie de l'équation finale.
Pour cela, on va repérer pour chaque groupement les variables qui ne varient pas, et ce quelle que soit la case du groupement.
Cela peut paraître compliqué dit comme ça, mais vous verrez qu'avec un ou deux exemples, ça sera très facile

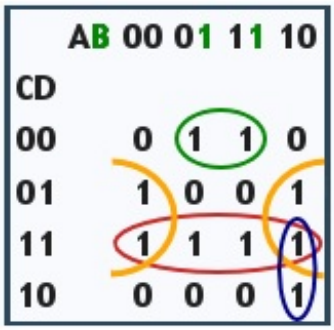
On va s'occuper en premier de notre groupement vert. On étudie chaque variable l'une après l'autre :

Commençons par la variable A :



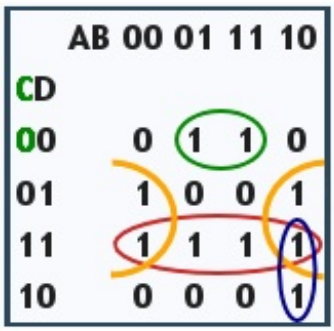
On s'aperçoit que pour les cases du groupement vert, la variable A n'est pas constante : elle prend comme valeur 0 et 1. On ne tiendra donc pas compte de cette variable.

Voyons maintenant la variable B :



Pour les deux cases du groupement, la variable B est constante : B=1. Cela signifie que pour que ce groupement soit vrai, il faut que B le soit aussi. On peut donc écrire provisoirement que : $groupementVert = B$

Pour la variable C :



C'est la même chose que pour la variable B, sauf que C est tout le temps égale à 0. Cela signifie que pour que ce groupement soit vrai, il faut en plus que C soit faux ($\overline{C}$) . On rajoute cette information à notre équation temporaire :
$groupementVert = B.\overline{C}$

La variable D se comporte exactement comme la variable C. On obtient donc $groupementVert = B.\overline{C}.\overline{D}$
On procède exactement pour le groupent bleu : $groupementBleu = A.\overline{B}.C$
Pour les groupements jaune et rouge, c'est encore similaire, à la nuance que les groupements sont plus grands, donc il y a une variable de moins qui ne varie pas (d'où l'intérêt de faire les groupements les plus grands possibles). Essayer de les retrouver par vous même pour vous entrainer.

Secret (cliquez pour afficher)

$groupementRouge = \overline{C}.D$
$groupementJaune = \overline{B}.D$

Bon et que fait-on avec tout ça ? Et bien on retrouve l'équation pardi !
On a trouvé les équations des groupements, c'est à dire les conditions au niveau des variables pour appartenir à un groupement. Comme les groupements regroupent toutes les combinaisons qui donnent un résultat vrai, il suffit qu'une combinaison vérifie

l'équation d'un des groupement pour qu'on soit sûrs que la sortie soit vraie. A l'inverse, on peut être sûrs que la sortie sera fausse. D'où l'équation finale :
$S = groupementJaune + groupementRouge + groupementBleu + groupementVert$
ce qui signifie que $S = B.\overline{C}.\overline{D} + A.\overline{B}.C + C.D + \overline{B}.D$

Et voilà ! Vous avez trouvé l'équation la plus simple possible de la table de vérité du début !
Essayer de simplifier avec les règles de l'algèbre de Boole, vous parviendrez tout au plus à factoriser mais en aucun cas à simplifier.

Exercice

Pour vous entrainer, vous pouvez essayer de retrouver l'équation du tableau de Karnaugh qu'on a rempli avec une équation. Le voici pour mémoire :

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

Vous devez obtenir :

Secret (cliquez pour afficher)

$S = A.C + B.\overline{C} + B.D$

Voilà, vous en savez assez pour vous débrouiller comme des grands avec l'algèbre de Boole
Je tiens à remercier particulièrement Talus pour son soutien et son travail de correction.

Pour ceux qui veulent s'entrainer ou qui sont intéressés, un solveur de l'algèbre de Boole est disponible ici. Attention, ce logiciel est encore en phase de test, il se peut donc qu'il reste quelques erreurs dans les équations retournées.

Bon courage

Partager