ALGÈBRE DE BOOLE ET FONCTIONS BOOLÉENNES

1 PROPRIÉTÉS

L'algèbre de Boole est définie sur l'ensemble E2 constitué des éléments {0,1}. Il existe une relation d'ordre 0 < 1, et trois opérations de base. La complémentation, définie en Table 1 est une application de E2 sur E2. Les opérations union (Table 2, gauche) appelée encore ou, max et qui est notée +, et intersection (Table 2, droite) appelée encore et, min, qui est notée . sont des applications de E2 X E2 -> E2

x	$\overline{x}$
0	1
1	0

Table 1 : complémentation

x	y	S
0	0	0
0	1	1
1	0	1
1	1	1

Table 2 : Union, +, ou, max

x	y	S
0	0	0
0	1	0
1	0	0
1	1	1

Intersection, ., et, min

Pour tout a, b, c $\in E_2$, les propriétés suivantes sont vérifiées :

1) 0 est l'élément minimum, 1 est l'élément maximum
   - a.1 = a car min (a,1) = a
   - a+0 = a car max (a,0) = a
   - a.0 = 0
   - a+1 = 1

2) complément :
   - $a.\overline{a} = 0$ car min (0,1) = 0
   - $a + \overline{a} = 1$ car max (0,1) = 1

3) Commutativité
   - a.b = b.a
   - a+b = b+a
   - car les fonctions min et max sont commutatives

4) Associativité
   - a.(b.c) = (a.b).c= a.b.c
   - a+(b+c) = (a+b)+c= a+b+c
   - car les fonctions min et max sont associatives

5) Distributivité
   - a.(b+c) = a.b+a.c
   - a+(b.c) = (a+b).(a+c)

6) THÉORÈME DE MORGAN

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

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

La Table 3 constitue une démonstration de ce théorème.

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

Table 3 : théorème de Morgan

1.1 OPÉRATEURS NAND ET NOR

Les opérateurs NAND et NOR ont la définition suivante.
NAND (a, b) = $\overline{a.b} = \overline{a} + \overline{b}$
NOR (a,b) = $\overline{a+b} = \overline{a}.\overline{b}$
Ces opérateurs sont fonctionnellement complets : avec un de ces opérateurs, on peut implanter les fonctions complément, min et max de l'algèbre de Boole.
La démonstration pour l'opérateur NAND est la suivante :

$$\overline{x} = \overline{x.1} = \overline{x.x}$$

$$x.y = \overline{1.\overline{x.y}}$$

$$x+y = \overline{\overline{1.x}.\overline{1.y}}$$

La Figure 1 donne la représentation symbolique des différents opérateurs, sous forme de portes logiques. L'inverseur *(NOT)* correspond à la fonction complémentation. Les autres portes ont le même nom que les fonctions logiques correspondantes.

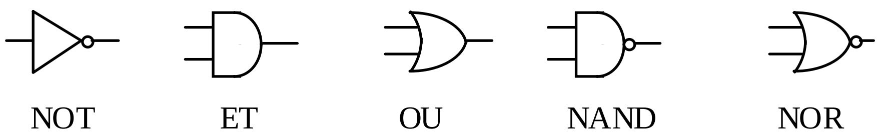

Figure 1 : Opérateurs logiques.

La Figure 2 donne les deux représentations graphiques du théorème de Morgan.

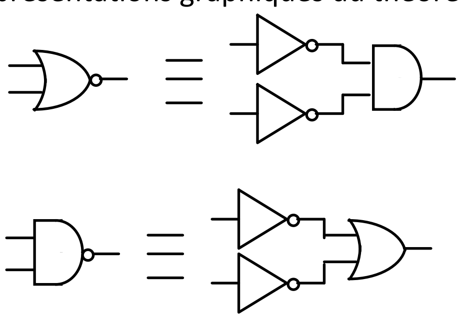

Figure 2 : Représentation graphique du théorème de Morgan

Les portes logiques que nous avons présentées travaillent sur les valeurs logiques 0 et 1. Elles supposent un fonctionnement instantané, c'est à dire un retard nul entre entrée et sorties. Ces portes sont implantées avec des circuits électriques, qui travaillent sur des variables continues. Il y a toujours un retard entre entrée et sortie. Il est important de souligner que toutes les propriétés de l'algèbre de Boole ne sont pas **toujours** vérifiées avec les circuits réels. Les deux propriétés $a.\overline{a}=0$ et $a+\overline{a}=1$ ne sont pas toujours vérifiées. La Figure 3 montre qu'à cause des temps de retard entre l'entrée et la sortie d'un inverseur, il y a deux périodes pendant lesquelles les deux relations ne sont pas vérifiées : c'est le cas lorsque E = $\overline{E}$. Cette situation correspond à ce que l'on appelle un **aléa**. Les signaux des circuits physiques ne sont donc valides que lorsque les lois de l'algèbre de Boole sont vérifiées, c'est à dire en dehors des aléas.

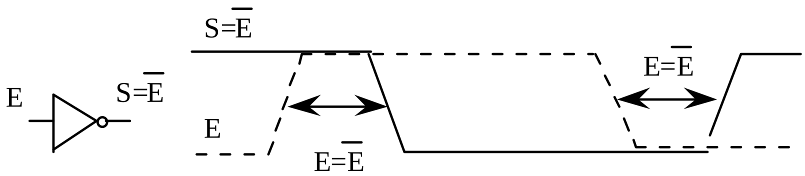

Figure 3 : Les aléas liés aux temps de retard dans un inverseur

1.2 FONCTIONS BOOLÉENNES

Dans le cas général, les fonctions booléennes sont une application de E_i x E_j x E_k ..x E_p -> E_2 où E_i= {0, 1, 2,...,i-1}. Les variables d'entrée ont un nombre fini de valeurs entières. La Table 4 donne la table de vérité d'une fonction booléenne pour laquelle la variable x est binaire et la variable y est ternaire (3 états possibles).

x	y	S
0	0	1
0	1	0
0	2	1
1	0	0
1	1	0
1	2	1

Table 4 : Exemple de fonction booléenne

Comme les fonctions utilisées pratiquement ont des variables d'entrée de même nature que les variables de sortie, on se restreint au cas particulier des fonctions booléennes applications de E_2 x E_2 x E_2...x E_2 -> E_2. La Table 5 donne l'exemple d'une telle fonction de deux variables x et y. Cette manière de représenter une fonction booléenne est appelée table de vérité. Les tables de vérité illustrent les deux problèmes rencontrés lors du traitement d'une fonction booléenne : il faut être capable de repérer une entrée de la table, et il faut être capable d'associer une valeur de la fonction à chaque entrée de la table.

	x	y	S
m_0	0	0	0
m_1	0	1	1
m_2	1	0	1
m_3	1	1	0

Table 5 : Exemple de fonction booléenne de deux variables.

Il existe différentes manières d'exprimer une fonction booléenne.

1.2.1 Forme disjonctive normale

A chaque entrée de la table, on associe une variable binaire m_i appelée terme produit *(minterm).* m_0 est associé à la ligne 0, m_1 est associé à la ligne 1, etc.

m_0 = 1 si x = 0 ET y = 0, soit $\overline{x} = 1$ ET $\overline{y} = 1$, soit $\overline{x}.\overline{y} = 1$ et $m_0 = \overline{x}.\overline{y}$

On repère de cette manière chaque ligne de la Table 6.

x	y	m_0	m_1	m_2	m_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Table 6 : Termes produit

Pour une table de vérité à deux entrées, les termes produit sont :

$$m_0 = \overline{x}.\overline{y}$$
$$m_1 = \overline{x}.y$$
$$m_2 = x.\overline{y}$$
$$m_3 = x.y$$

Un terme produit est donc constitué de l'intersection (et) de toutes les variables d'entrées, complémentées si leur valeur est 0, non complémentées si leur valeur est 1. Puis, à chaque terme produit m_i, on associe la valeur S_i de la fonction booléenne S (Table 7).

Ceci peut être réalisé sous la forme d'une union de produits, de la manière suivante :

$$S = m_0.S_0 + m_1.S_1 + m_2.S_2 + m_3.S_3.$$

Pour une configuration d'entrée, un seul terme m_i est égal à 1 et tous les autres sont à 0. On a donc automatiquement $S = m_i.S_i = S_i$ pour le terme produit m_i à 1.

0n peut remarquer que les valeurs 0 de la fonction (S_i=0) ne contribuent pas à l'expression de S (car m_i. 0 = 0, et 0 est absorbé dans l'union logique). On remarque d'autre part que lorsque S_i=1, on a $m_i.S_i = m_i$. On peut en déduire la règle pratique suivante, qui donne la **forme disjonctive normale** d'une fonction booléenne : **la forme disjonctive normale d'une fonction booléenne est obtenue par union logique des termes produits pour lesquels la fonction a pour valeur 1**.

	x	y	S
m_0	0	0	S_0
m_1	0	1	S_1
m_2	1	0	S_2
m_3	1	1	S_3

Table 7 : Termes produit et sorties

x	y	m_0	m_1	m_2	m_3	m_1+m_2	S
0	0	1	0	0	0	0	0
0	1	0	1	0	0	1	1
1	0	0	0	1	0	1	1
1	1	0	0	0	1	0	0

Table 8 : Exemple de fonction

S = 1 si $m_1 = 1$ ou si $m_2 = 1$, soit $m_1 + m_2 = 1$ **==>** $S = m_1 + m_2$

$$S = \bar{x}.y + x.\bar{y}$$

On peut utiliser cette propriété de la forme disjonctive normale pour remplacer la table de vérité par une forme plus condensée de représentation. Une fonction peut être représentée sous la forme

$f=\sum m$ (liste des termes produit pour lesquels la fonction est égale à 1).

Par exemple, la fonction de la Table 5 s'écrira $f=\sum m(1,2)$.
Soit l'exemple de la Table 8, qui utilise la fonction de la Table 5:
La fonction particulière que nous avons prise comme exemple s'appelle **OU exclusif** et se note ⊕. Son schéma logique est donné en Figure 4.

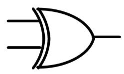

Figure 4 : Porte logique Ou exclusif

L'implémentation de la fonction Ou exclusif sous forme de Ou de Et qui résulte de la forme disjonctive normale est présentée en Figure 5.

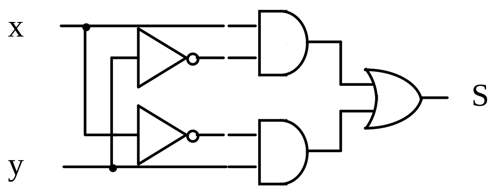

Figure 5 : Ou exclusif résultant de la forme disjonctive

1.2.2 Forme NAND de NAND.

La forme disjonctive normale peut se transformer en forme NAND de NAND, par application du théorème de Morgan. La Figure 6 l'illustre graphiquement. La forme disjonctive normale se transforme automatiquement[1] en forme NAND de NAND en remplaçant les portes Et par des portes NAND et les portes Ou par des portes NAND.

1.2.3 Forme conjonctive normale

Il existe une autre forme de représentation : la forme conjonctive normale. On définit des termes somme *(maxtermes)*, dont on fait l'intersection. La Table 9 présente les termes somme pour une fonction à deux entrées.

[1] Attention : Lorsqu'une variable d'entrée entre directement sur la porte Ou, on doit considérer qu'elle traverse une porte Et à une entrée, qui se transforme en un inverseur (porte Nand à une entrée).

x	y	M_0	M_1	M_2	M_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Table 9 : Termes somme

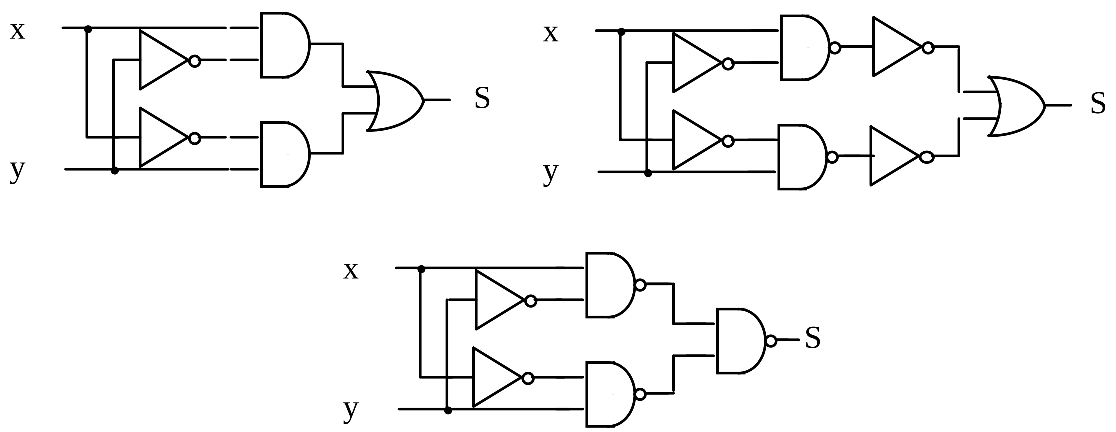

Figure 6 : Exemple de transformation de forme Ou de Et en forme NAND de NAND.

Pour une table de vérité à deux entrées, les termes somme sont :

$M_0 = x + y$

$M_1 = x + \overline{y}$

$M_2 = \overline{x} + y$

$M_3 = \overline{x} + \overline{y}$

Un terme somme est donc constitué de l'union (ou) de toutes les variables d'entrée, non complémentées si leur valeur est 0, complémentées si leur valeur est 1.
A chaque terme somme M_i, on associe la valeur S_i de la fonction (Table 10).

	x	y	S
M_0	0	0	S_0
M_1	0	1	S_1
M_2	1	0	S_2
M_3	1	1	S_3

Table 10

Ceci peut être réalisé sous la forme d'une intersection de sommes, de la manière suivante :

$S = (M_0+S_0) . (M_1+S_1) . (M_2+S_2) . (M_3+S_3)$

Pour une configuration d'entrée, un seul terme M_i est égal à 0 et tous les autres sont à 1. On a donc automatiquement S = M_i+S_i = S_i pour le terme produit M_i à 0. En effet, pour j≠i, on a M_j = 1 et donc M_j+S_j = 1, qui sont des termes neutres pour l'intersection.
On peut remarquer que les valeurs 1 de la fonction (S_i=1) ne contribuent pas à l'expression de S (car M_i + 1 = 1, et 1 est absorbé dans le produit logique). On remarque d'autre part que lorsque S_i=0, on a M_i+S_i = M_i. On peut en déduire la règle pratique suivante, qui donne la **forme conjonctive normale** d'une fonction booléenne : **la forme conjonctive normale d'une fonction booléenne est obtenue par produit logique des termes somme pour lesquels la fonction a pour valeur 0.**
Soit l'exemple de la Table 11, qui utilise la même fonction que la Table 5 :

x	y	M_0	M_1	M_2	M_3	$M_0.M_3$	S
0	0	0	1	1	1	0	0
0	1	1	0	1	1	1	1
1	0	1	1	0	1	1	1
1	1	1	1	1	0	0	0

Table 11

S = 0 si M_0 = 0 et si M_3 = 0, soit $M_0.M_3$ = 0 **==>** S = $M_0.M_3$
soit $S = (x + y).(\bar{x} + \bar{y})$
On peut montrer que cette forme est équivalente à celle qui résulte de la forme disjonctive normale.

1.2.4 Forme NOR de NOR

On montre de la même manière que toute fonction booléenne peut s'exprimer uniquement sous forme NOR de NOR. La forme NOR de NOR s'obtient application du théorème de Morgan sur la forme conjonctive normale : on remplace les portes Ou et les portes Et par des portes NOR[2].

1.3 Simplification des expressions booléennes.

Elles découlent de l'application des propriétés de l'algèbre de Boole définie en début de chapitre. Soit l'exemple de la fonction de 2 variables (Table 12)

	x	y	S
m_0	0	0	0
m_1	0	1	1
m_2	1	0	1
m_3	1	1	1

Table 12

[2] Attention : Lorsqu'une variable d'entrée entre directement sur la porte Et, on doit considérer qu'elle traverse une porte Ou à une entrée, qui se transforme en un inverseur (porte Nor à une entrée).

La forme non simplifiée s'écrit $S = \bar{x}.y + x.\bar{y} + x.y$
L'application successive des règles conduit aux transformations suivantes :

$S = \bar{x}.y + x.\bar{y} + x.y + x.y$ car x.y = x.y + x.y puisque x.y + x.y = x.y (absorption)
$S = \bar{x}.y + x.y + x.\bar{y} + x.y$ par commutativité.
$S = (\bar{x} + x).y + x.(\bar{y} + y)$ par distributivité
S= 1.y + x.1 par absorption
S = y + x = x + y

Ces simplifications peuvent être réalisées graphiquement à l'aide de la méthode du **diagramme de Karnaugh.** Cette méthode se fonde sur une manière de représenter la table de vérité qui fait apparaître les symétries sur les variables. La Figure 7 : Diagramme de Karnaugh pour fonction à 2 entrées. présente l'exemple du diagramme de Karnaugh pour la fonction à 2 entrées de la table. Les quatre cases correspondent aux quatre termes produit m_0 à m_3. Les symétries selon x et y sont mises en évidence. Un regroupement de 2 éléments symétriques se traduit par la suppression d'une variable dans un terme. Un regroupement de 4 éléments pour lesquels existent 2 symétries se traduit par la suppression de 2 variables dans un terme, etc. Nous présentons le diagramme de Karnaugh (Figure 8) dans le cas d'une fonction de 4 variables, avec les numéros de case correspondant aux numéros de mintermes dans l'hypothèse d'une numération binaire pour les bits $e_3e_2e_1e_0$ où e_0 est le bit de poids faible.

Les règles pour la simplification des fonctions booléennes avec le diagramme de Karnaugh sont les suivantes :

- tous les termes produit pour lesquels la fonction est à 1 devront être pris au moins une fois dans un regroupement, ou seuls si aucun regroupement n'est possible.
- faire les regroupements de taille maximale, de manière à éliminer le plus grand nombre possible de variables dans les termes de l'expression.
- ne prendre que les regroupements ou termes produit nécessaires pour avoir au moins une fois chaque 1, sans redondance.

La méthode du diagramme de Karnaugh est efficace pour les expressions booléennes ayant au plus 4 entrées. Au delà, la représentation graphique devient complexe, il est difficile de mettre en évidence les symétries, et la méthode devient inutilisable. Dans ce cas, il fait utiliser des méthodes plus élaborées, comme celle de Quine - Mc Cluskey, qui est la base des heuristiques utilisées dans un certain nombre de logiciels spécialisés (Espresso, Mc Boole). D'autres logiciels utilisent des méthodes de réécriture d'expressions.

Il faut souligner que le problème de simplification d'expressions booléennes se pose, soit pour des expressions très simples à très peu de variables pour lesquelles le diagramme de Karnaugh est amplement suffisant, soit pour des expressions complexes à grand nombre de variables pour lesquelles les logiciels spécialisés sont inévitables.

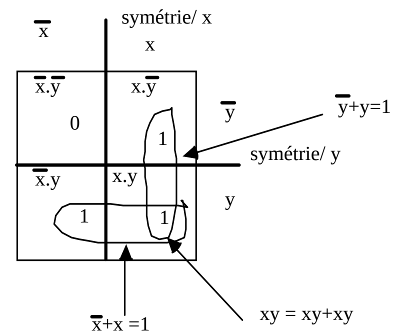

Figure 7 : Diagramme de Karnaugh pour fonction à 2 entrées.

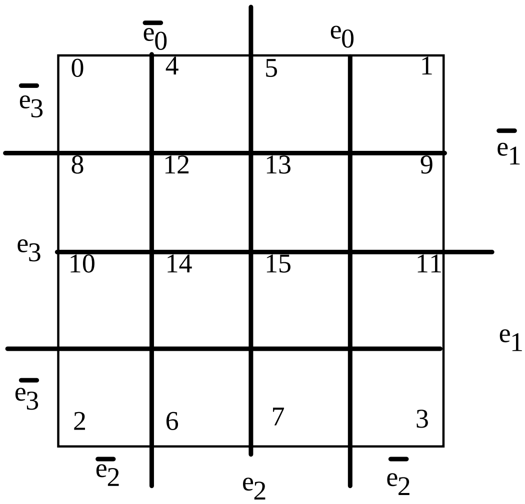

Figure 8 : Diagramme de Karnaugh pour une fonction à quatre entrées

1.3.1 Cas des fonctions booléennes incomplètement spécifiées.

Il existe des fonctions booléennes pour lesquelles il n'y a pas de valeurs associées à certains termes produit. Ceux-ci ne sont jamais "sélectionnés", et la valeur qui leur est associée peut être indifféremment 0 ou 1. On note d *(don't care)* ou Ø ce cas indifférent. L'afficheur 7 segments (Figure 9) est un exemple particulier de fonction booléenne incomplètement spécifiée. On veut afficher les 10 chiffres décimaux à l'aide de 7 segments, notés de a à g, qui peuvent être à 0 (éteint) ou 1 (allumé). Le codage des 10 chiffres décimaux nécessite 4 bits, que l'on peut noter e_3 à e_0. La Table 13donne les 7 fonctions booléennes traduisant l'allumage des 7 segments a à g en fonction des bits e_3 à e_0.

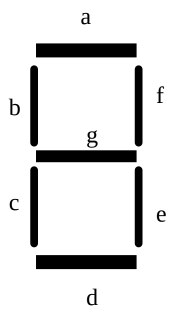

Figure 9 : Afficheur 7 segments

Pour simplifier de telles fonctions, on peut indifféremment associer à Ø la valeur 0 ou 1, pour simplifier au maximum. La Figure 10 montre le diagramme de Karnaugh associé à la fonction a de la

La valeur simplifiée de la fonction est $a = e_1 + e_3 + e_0.e_2 + \overline{e_0}.\overline{e_2}$

e_3	e_2	e_1	e_0		a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	0	0	0	1	1	0
0	0	1	0	2	1	0	1	1	0	1	1
0	0	1	1	3	1	0	1	1	0	1	1
0	1	0	0	4	0	1	0	0	1	1	1
0	1	0	1	5	1	1	0	1	1	0	1
0	1	1	0	6	1	1	1	1	1	0	1
0	1	1	1	7	1	0	0	0	1	1	0
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	1	0	1	1
1	0	1	0	10	Ø	Ø	Ø	Ø	Ø	Ø	Ø
1	0	1	1	11	Ø	Ø	Ø	Ø	Ø	Ø	Ø
1	1	0	0	12	Ø	Ø	Ø	Ø	Ø	Ø	Ø
1	1	0	1	13	Ø	Ø	Ø	Ø	Ø	Ø	Ø
1	1	1	0	14	Ø	Ø	Ø	Ø	Ø	Ø	Ø
1	1	1	1	15	Ø	Ø	Ø	Ø	Ø	Ø	Ø

Table 13 : afficheur 7 segments

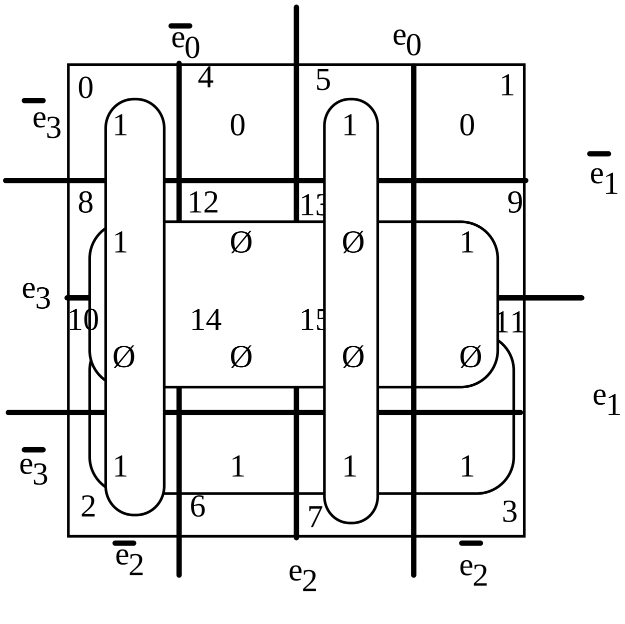

Figure 10 : Diagramme de Karnaugh avec cas indifférents