

Chapitre 2.

Fonctions et équations logiques

Un circuit logique est un ensemble d'éléments physiques connectés qui réalise une fonction logique. Les notions de fonctions logiques et équations logiques seront développées dans ce chapitre.

2.1. Fonctions logiques**2.1.1. Domaine de définition**

B_2 étant l'ensemble des valeurs logiques ($B_2 = \{0,1\}$), B_2^n est l'ensemble formé de quantités $X = (x_1, x_2, \dots, x_n)$ ou les « x_i » sont des variables booléennes. B_2^n est un ensemble comprenant 2^n points (figure 2.1).

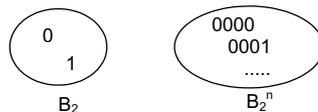


Figure 2.1. *Domaine de définition*

On appelle **combinaison** ou encore **point** de B_2^n une quantité $X = (x_1, x_2, \dots, x_n)$ ou « x_i » est une variable booléenne

On définit une distance sur B_2^n appelée **distance de hamming** : soit X_0 et X_1 deux points, la distance de X_0 à X_1 notée $D(X_0, X_1)$ est le nombre de composantes différentes dans X_0 et X_1 . C'est le nombre d'arêtes séparant 2 points de l'hypercube.

Exemple : $D(00101, 01001) = 2$.

D a bien les propriétés d'une distance, en particulier D vérifie bien la propriété d'inégalité triangulaire :

$$D(X_0, X_1) \leq D(X_0, X_2) + D(X_2, X_1)$$

Lorsque la distance entre deux combinaisons de B_2^n est égale à 1 les deux points sont dits adjacents. Toute combinaison de n variables est adjacente à n autres combinaisons obtenues en changeant l'une des composantes.

2.1.2. Fonctions logiques simples

Une **fonction logique simple** de n variables est une application qui à toute combinaison X appartenant à B_2^n , fait correspondre un élément y appartenant à B_2 . Elle sera notée $y = f(X)$.

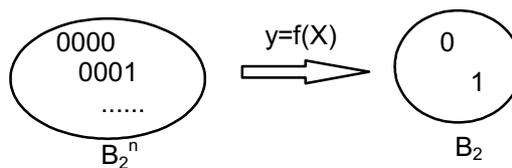


Figure 2.2. Fonctions logiques simples

Une fonction simple divise l'ensemble des combinaisons de B_2^n en deux parties : l'ensemble des points "vrais" qui sont les combinaisons X_i telles que $f(X_i)=1$ et l'ensemble des points "faux" qui sont les combinaisons X_j telles que $f(X_j)=0$.

Un point X tel que $f(X) = 1$ est dit **couvert** par la fonction. La **couverture** de la fonction est l'ensemble des points couverts.

Le nombre de fonctions logiques de n variables est 2^{2^n} puisqu'à chacune des 2^n combinaisons on peut faire correspondre l'une des deux valeurs 0,1. Ce nombre comprend toutes les fonctions de n variables au plus c'est à dire toutes les fonctions de 0,1,2,...,n variables.

La **fonction complémentaire** d'une fonction $f(X)$ notée $f'(X)$ est une fonction qui prend la valeur complémentaire de $f(X)$ pour toutes les combinaisons X_i .

Un ordre partiel est défini sur les fonctions. On dit que $f \leq g$ si et seulement si pour tout X tel que $f(X) = 1$ alors $g(X) = 1$, i.e. l'ensemble des points couverts par g contient l'ensemble des points couverts par f .

2.1.3. Fonctions logiques multiples

Une **fonction logique multiple** (ou simultanée) est une application qui à toute combinaison X_i de B_2^n fait correspondre un point Y_i appartenant à B_2^q . Nous emploierons alors la notation $Y=f(X)$.

Exemple1 : $n=3, q=2$

$$y_1 = a.b + c$$

$$y_2 = a.c$$

Exemple2 : Additionneur binaire $n=3, q=2$

$$S = a \oplus b \oplus r$$

$$R_{\text{sortante}} = r . (a \oplus b) + a.b$$

2.1.4. Fonctions logiques incomplètement spécifiées

Lors de la définition d'une fonction, la valeur de cette fonction peut être indifférente pour certaines combinaisons qui sont appelées combinaisons non-spécifiées. On conviendra de noter ϕ la valeur de la fonction en ces points.

Nous appellerons **variable ϕ -booléenne** une variable susceptible de prendre ses valeurs dans l'ensemble $B_2\phi=(0,1,\phi)$.

Une **fonction incomplète** est une application qui à toute combinaison X_i de B_2^n fait correspondre un élément de $B_2\phi^q$ ou $B_2\phi=(0,1,\phi)$.

On peut caractériser une fonction incomplète d'une façon qui en montre mieux sa nature. Pour cela définissons une relation d'ordre total sur $B_2\phi$ par $0 \leq \phi \leq 1$.

Cette relation d'ordre total permet de définir une relation d'ordre partiel sur l'ensemble des combinaisons en posant $X \leq Y$ si pour tout i on a $x_i \leq y_i$.

Exemple : $(0\phi 1\phi 01) \leq (0\phi 11\phi 1)$

Cette relation d'ordre partiel induit alors une relation d'ordre partiel sur l'ensemble des fonctions incomplètement spécifiées qui est : $f \leq g$ si pour tout X_i $f(X) \leq g(X)$.

Avec ces conventions on peut définir une fonction incomplètement spécifiée f à l'aide de deux fonctions complètes appelées « borne inférieure » (f_{inf}) et « borne supérieure » (f_{sup}).

La borne inférieure f_{inf} (supérieure) d'une fonction f est obtenue en remplaçant chacune des valeurs non spécifiées de f par la valeur 0 (1). On a donc la relation $f_{inf} \leq f \leq f_{sup}$.

2.2. Représentation des fonctions logiques

Il existe plusieurs manières de représenter les fonctions logiques. Nous verrons dans ce chapitre les représentations les plus usuelles à savoir :

- la représentation algébrique,
- les représentations tabulaires,
- les représentations implicites,
- les représentations graphiques.

Pour répondre au problème posé par l'accroissement de la taille des fonctions logiques à traiter (grand nombre de fonctions, de monômes et de variables), d'autres techniques de représentation des fonctions booléennes ont été développées (graphes de décision binaires et notation cube de position). Ces dernières, qui permettent une représentation efficace en machine, feront l'objet de chapitres spécifiques (cf. chapitre 3 et chapitre 4).

2.2.1. Représentation algébrique

Une fonction peut être définie à partir d'une expression logique comme étant la valeur de cette expression logique. A toute expression logique correspond donc une fonction logique et une seule.

Exemple : $f(a,b,c) = a.b' + b.c$

Cette représentation d'une fonction est appelée représentation algébrique.

Réciproquement on peut démontrer qu'il est toujours possible d'associer une expression logique à une fonction.

Exemple : Soit f /

(0,0)	-> 0
(1,1)	-> 0
(0,1)	-> 1
(1,0)	-> 1

$$f(a,b) = ab' + a'b = a \oplus b$$

Les termes $a.b'$, $a'.b$, ... sont les monômes booléens de la fonction. De manière plus générale, on appelle **monôme booléen** une quantité $M = A.x_1^* .x_2^* \dots .x_p^*$

- ou :
- « A » est le coefficient du monôme (variable booléenne)
 - « x_i » sont des variables booléennes
 - « x_i^* » sont des littéraux (signifie que l'on ne précise pas si on a x_i ou x_i')
 - « $x_1^* .x_2^* \dots .x_p^*$ » est le corps du monôme
 - « p » est le degré du monôme

Si l'on raisonne dans un espace de dimension n , on peut avoir des monômes de dimension $1,2,\dots,n$. D'autre part, un monôme n 'a de signification que si son coefficient est égal à 1 ($A=1$).

On appellera *monôme canonique* ou *minterme* un monôme de dimension n ($p=n$).

La représentation algébrique d'une fonction incomplètement spécifiée requiert d'exprimer les bornes inférieures et supérieures de la fonction, c'est-à-dire d'exprimer la fonction d'une part pour $\phi = 0$ et d'autre part pour $\phi = 1$.

Exemple : Soit $f /$

(0,0) \rightarrow 0
 (1,1) \rightarrow ϕ
 (0,1) \rightarrow 1
 (1,0) \rightarrow 1

$f_{\text{inf}}(a,b) = ab' + a'b$
 $f_{\text{sup}}(a,b) = ab' + a'b + ab$

2.2.2. Représentations tabulaires

2.2.2.a. Table de vérité

La table de vérité est l'écriture directe sous forme tabulaire de la correspondance entre B_2^n et B_2^p ou $(B_2^*)^p$ définie par la fonction. La liste des points de B_2^n est en général donnée dans l'ordre naturel de numération en base 2 (figure 2.3).

	abc	f	f _{inf}	f _{sup}
X₀	000	ϕ	0	1
X₁	001	1	1	1
X₂	010	1	1	1
X₃	011	0	0	0
X₄	100	1	1	1
X₅	101	ϕ	0	1
X₆	110	1	1	1
X₇	111	1	1	1

Figure 2.3. Table de vérité d'une fonction incomplète $f(a,b,c)$.

2.2.2.b. Table de Karnaugh

Pour de nombreux problèmes de logique, en particulier pour la simplification des fonctions, on utilise la propriété d'adjacence.

Afin de mettre en évidence cette propriété, il est commode d'utiliser une table à deux entrées avec un codage non plus binaire naturel mais binaire réfléchi (code de Gray). Une telle table est appelée table de Karnaugh.

		bc			
		00	01	11	10
a	0	ϕ	1	0	1
	1	1	ϕ	1	1

Figure 2.4. Écriture de la fonction f précédente sous forme de table de Karnaugh.

Ainsi, les tables de Karnaugh sont telles que deux cases contiguës de la table correspondent à deux combinaisons adjacentes. Malheureusement la réciproque n'est pas vraie et il existe sur une telle table des cases

non contiguës dont les codes sont adjacents. Par exemple, sur la table présentée sur la figure 2.5, le codage des cases « x » est adjacent au codage de la case X.

	cd	00	01	11	10
ab	00	X	x		x
	01				
	11				
	10	x			

Figure 2.5. Adjacences sur un table de Karnaugh

Même si ces relations d'adjacence ne sont pas toujours directes, elles sont facilement identifiables. Ces adjacences pouvant conduire des regroupements de monômes, elles permettront de réduire le nombre de monômes caractérisant la fonction. Les tables présentées sur la figure 2.6 montrent la relation entre adjacence et monômes sur la table de Karnaugh.

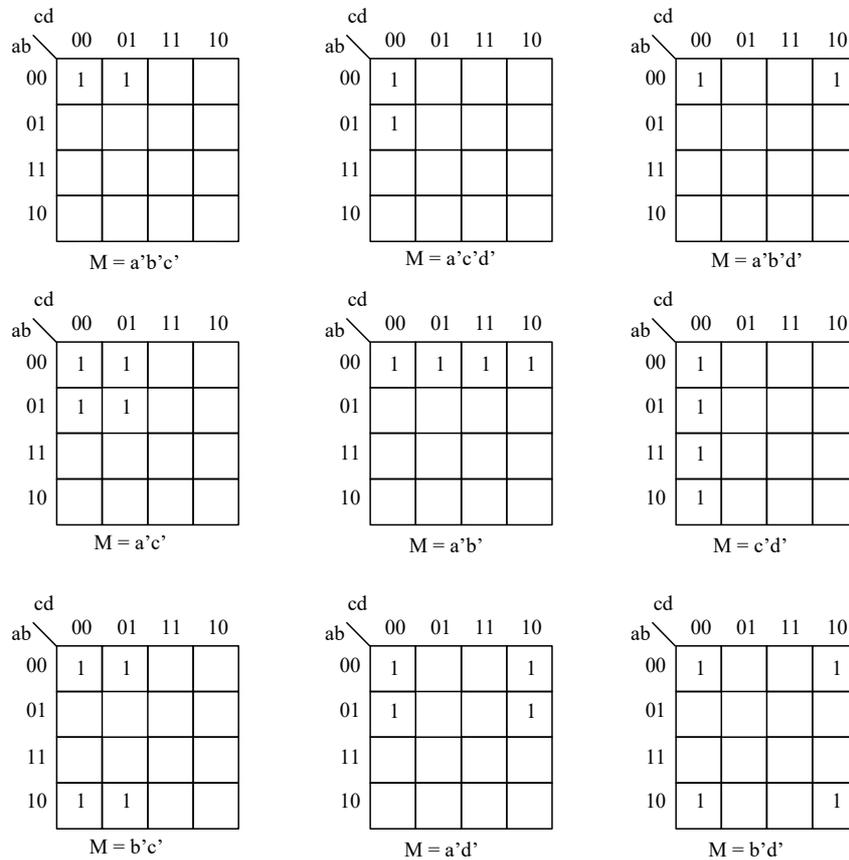


Figure 2.6. Relation entre monômes et adjacences

2.2.3. Représentations implicites

2.2.3.a. Image caractéristique

L'image caractéristique d'une fonction est constituée des 2^n valeurs de la fonction, données sous forme d'une ligne. Ces valeurs sont rangées dans l'ordre naturel de numération en base 2 croissant de la gauche vers la droite.

Ainsi, la première valeur correspond à celle de la première ligne de la table de vérité, la seconde à celle de la deuxième ligne etc...

Exemple : L'image caractéristique de la fonction f précédente est $I_c[f(a,b,c)] = \phi 1101\phi 11$

Cette représentation n'amène rien de plus que la table de vérité si ce n'est qu'elle permet de condenser son écriture en évitant d'avoir à expliciter chaque combinaison.

2.2.3.b. Image décimale

Une combinaison X_i peut être considérée comme une représentation en base 2 d'un nombre décimal N . Dans ce cas, il est nécessaire de d'établir une convention sur l'ordre des variables (poids fort à gauche : $f(a,b,c,d) \Rightarrow a$ de poids fort).

On peut alors définir la fonction en précisant l'équivalent décimal des combinaisons X_i telles que $f(X)=1$, l'ensemble des combinaisons X_i telles que $f(X)=0$ étant quant à lui précisé par défaut. Dans le cas des fonctions incomplètes il est par contre également nécessaire de préciser les combinaisons X_i telles que $f(X)=\phi$.

Exemple : L'image décimale de la fonction f précédente avec a poids fort est :

$$I_d[f(a,b,c)] = R_1(1,2,4,6,7) + R_\phi(0,6)$$

2.2.4. Représentations graphiques

2.2.4.a. Hypercube

Une combinaison de B_2^n peut être représentée par un point dans un espace de dimension n . Ces points constituent les sommets d'un hypercube de dimension n .

Une fonction simple peut être représentée graphiquement en marquant les sommets de cet hypercube correspondant aux points vrais de la fonction (figure 2.7).

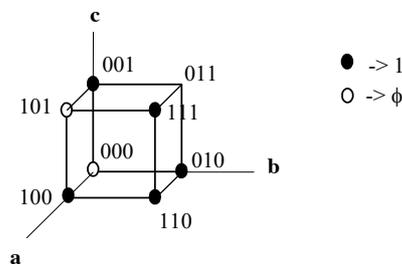


Figure 2.7. Représentation d'une fonction dans l'hypercube

Comme pour la table de Karnaugh, une telle représentation permet de mettre en évidence les adjacences entre les différents points de B_2^n . Elle pourra ainsi être utilisée pour regrouper des mintermes et ainsi simplifier l'expression des fonctions comme illustré sur la figure 2.8,

- un minterme est un sommet de l'hypercube
- un monôme est une "face" de l'hypercube.
- dans B_2^n , un monôme de degré p couvre 2^{n-p} points.

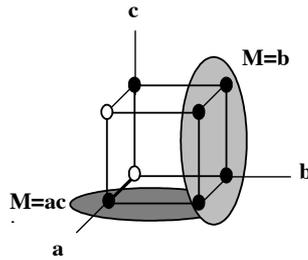


Figure 2.8. Visualisation des monômes sur l'hypercube.

Cette représentation met effectivement en évidence les distances entre les points de B_2^n mais, tout comme la table de Karnaugh, reste limitée à de petites fonctions. Elle n'est pas vraiment utilisable pour $n > 4$.

2.2.4.b. Treillis

Une combinaison de B_2^n peut être représentée par une notation indicielle indiquant la position des 1. Par exemple, en partant de la gauche, la combinaison 0101 peut être représentée par l'élément (2,4).

En utilisant cette notation, l'hypercube peut être transformé en un treillis dont la 1^{ère} colonne ne comprend aucun indice (point 0), la seconde en comprend 1 (point ne comportant qu'un seul 1), la 3^{ème} en comprend 2 (points comportant deux 1) et ainsi de suite. Cette représentation illustrée sur la figure 2.9 peut permettre de représenter des fonctions comportant plus de variables que la représentation hypercube tout en conservant la caractéristique de ce dernier à savoir faire apparaître les distances entre les points B_2^n (et notamment les points adjacents). Pour cela, il suffit de relier les points d'une colonne i aux points de la colonne suivante incluant les indices d'origine.

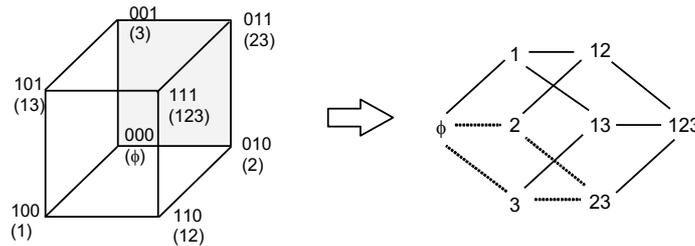


Figure 2.9. Représentation d'une fonction sous forme de treillis.

Dans ce cas, les mintermes sont les sommets du treillis et les monômes un sous-treillis composé de l'ensemble des liaisons reliant un point d'une colonne i à un point d'une colonne j . Le monôme correspondant à la face grisée de l'hypercube soit « a » se retrouve dans le sous-treillis [(ϕ), (2,3)]

A titre d'exemple sur un espace de dimension 4 (figure 2.10) le point (12) représente le minterme « abc'd' » et le sous-treillis allant du point (1) au point (1234) représente le monôme « a ». Le sous-treillis allant du point (1) au point (123) représente quant à lui le monôme « ad' ».

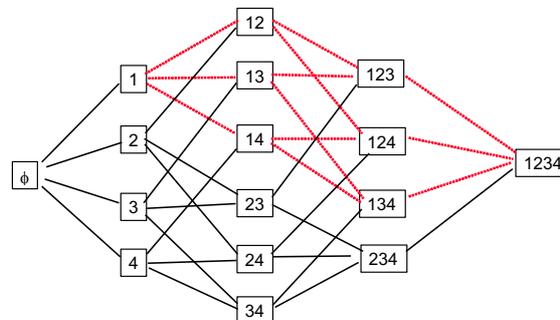


Figure 2.10. Représentation d'une fonction de dimension 4

2.2.4.c. Logigramme

C'est à la fois une représentation graphique et symbolique. Elle s'effectue à l'aide des représentations symboliques des opérateurs binaires reliées entre elles. Chaque opérateur est caractérisé par un ensemble d'entrées, une sortie et une règle opératoire précisée soit par un signe placé à l'intérieur de l'opérateur soit par sa forme même. La règle opératoire permet de calculer l'expression logique associée à la sortie d'un opérateur dès que l'on connaît les expressions logiques attachées à toutes ses entrées.

Une sortie d'opérateur est reliée à des entrées d'autres opérateurs. Les sorties libres sont appelées « sorties primaires », les entrées libres « entrées primaires ». A chaque entrée libre est associée une variable logique.

Un logigramme permet ainsi de représenter une fonction logique. Un exemple de logigramme est donné sur la figure 2.11.

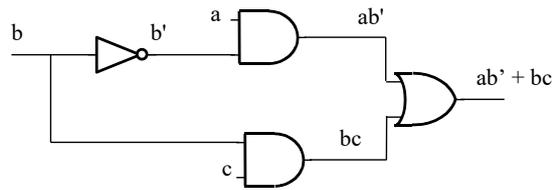


Figure 2.11. Logigramme associé à la fonction $f(a,b,c) = a.b' + b.c$

2.3. Fonctions remarquables

2.3.1. Fonctions linéaires

Définition : On dira qu'une fonction est *linéaire* si elle peut s'exprimer uniquement avec des opérateurs OU exclusifs, c'est-à-dire, si son expression dans le champ de Galois (\oplus , ET) s'écrit sans l'opérateur ET. Une fonction linéaire peut donc se mettre sous la forme d'une somme modulo 2 de variables. Par exemple, $a \oplus b \oplus c$ et $1 \oplus a \oplus b \oplus c$ sont des fonctions linéaires.

Exemple : Soit $f(a,b,c) = a'.b'.c' + a.b.c' + a.b'.c + a'.b.c$

$$\begin{aligned}
 &= c'.(a'.b' + a.b) + c.(a.b' + a'.b) \\
 &= c'.(a \oplus b)' + c.(a \oplus b) \\
 &= (c \oplus (a \oplus b))' \\
 &= 1 \oplus c \oplus a \oplus b
 \end{aligned}$$

$\Rightarrow f(a,b,c)$ est une fonction linéaire

2.3.2. Fonctions paires et impaires

Soit une fonction $f(X)$. Nous noterons $f_d(X)$ la fonction duale de $f(X)$ et $f(X')$ la fonction obtenue à partir d'une expression quelconque de $f(X)$ en remplaçant chacune des variables par son complément.

2.3.2.a. Fonction paire

Une fonction est *paire* si $f(X') = f(X)$

Exemple : Soit $f(a,b,c) = a.b + b'.c + a'.c'$

$$\begin{aligned}
 f(a',b',c') &= a'.b' + b.c' + a.c \\
 &= a'.b'.c + a'.b'.c' + a.b.c' + a'.b.c' + a.b.c + a.b'.c \\
 &= a.b.(c+c') + b'.c.(a+a') + a'.c'.(b.b') \\
 &= a.b + b'.c + a'.c' \\
 &= f(a,b,c)
 \end{aligned}$$

$\Rightarrow f(a,b,c)$ est une fonction paire

Remarque : Une fonction paire comporte nécessairement un nombre pair de sommets à 1 (et donc aussi un nombre pair de sommets à 0). Si une fonction est paire, sa fonction complémentaire est également paire. L'ensemble des fonctions paires est fermé pour la somme et le produit (la somme de deux fonctions paires est paire, le produit de deux fonctions paires est pair).

2.3.2.b. Fonction impaire

Une fonction est **impaire** si $f(X)=f'(X')$ ou $f'(X)=f(X')$ ou $f(X)=f_d(X)$, c'est-à-dire si elle est égale à sa duale (fonction autoduale).

Exemple : Soit $f(a,b,c) = a.b + b.c + a.c$

$$\begin{aligned} f_d(a,b,c) &= (a+b)(b+c).(a+c) \\ &= (a.b + a.c + b + b.c).(a+c) \\ &= (a.c + b).(a+c) \\ &= (a.c + a.c + a.b + b.c) \\ &= (a.c + a.b + b.c) \\ &= f(a,b,c) \end{aligned}$$

$\Rightarrow f(a,b,c)$ est une fonction impaire

Remarque : Une fonction impaire de n variable a nécessairement 2^{n-1} sommets à 0 et 2^{n-1} sommets à 1. (Si un sommet X est à 0, le sommet X' est nécessairement à 1 et inversement).

2.3.3. Fonctions symétriques

Une fonction est **symétrique** si elle est telle que toute permutation de variables laisse la fonction inchangée.

On démontre qu'une condition nécessaire et suffisante pour qu'une fonction soit symétrique est qu'elle prenne la valeur 1 si et seulement si k variables ont la valeur 1 ou k prend ses valeurs dans l'ensemble (k_1, \dots, k_p) avec $0 \leq k_i \leq n$.

Exemple : la fonction de trois variables définie par l'expression $f(a,b,c) = a.b.c + a'.b'.c + a.b'.c' + a'.b.c'$ est symétrique.

Elle prend la valeur 1 lorsque et seulement lorsque une ou trois variables prennent la valeur 1.

On dit qu'une fonction est **symétrique par rapport à un ensemble de littéraux** (variables ou variables complémentées) si elle est telle que toute permutation des littéraux de l'ensemble laisse la fonction inchangée.

Exemple : soit $f(a,b,c) = a'.b'.c' + a.b'.c + a'.b.c$

La fonction $f(a,b,c)$ n'est pas symétrique par rapport aux variables a,b,c mais l'est par rapport aux littéraux a,b,c' .

Les variables par rapport auxquelles la fonction est symétrique sont appelées « variables de symétrie ».

L'ensemble des fonctions symétriques par rapport aux mêmes littéraux est fermé pour la somme et le produit (la somme de deux fonctions symétriques est symétrique, le produit de deux fonctions symétriques est symétrique).

2.4. Formes canoniques des fonctions logiques

2.4.1. Théorème d'expansion de Shannon

Le théorème d'expansion de Shannon est un théorème extrêmement important. C'est en effet sur ce théorème que s'appuient la plupart des techniques de manipulation et d'optimisation des « grosses » fonctions logiques. Il permet, en décomposant une fonction par rapport à ses variables, de traiter cette fonction de manière récursive.

Théorème : Soit $f(x_1, x_2, \dots, x_i, \dots, x_n)$ une fonction logique de n variables

$$(1) \quad f(x_1, x_2, \dots, x_i, \dots, x_n) = x_i' \cdot f(x_1, x_2, \dots, 0, \dots, x_n) + x_i \cdot f(x_1, x_2, \dots, 1, \dots, x_n)$$

$$(2) \quad f(x_1, x_2, \dots, x_i, \dots, x_n) = [f(x_1, x_2, \dots, 0, \dots, x_n) + x_i] \cdot [f(x_1, x_2, \dots, 1, \dots, x_n) + x_i']$$

Preuve : Vérifions l'égalité en posant $x_i = 0$ puis $x_i = 1$

Soit $x_i = 0$

$$(1) \Rightarrow x_i' \cdot f(x_1, x_2, \dots, 0, \dots, x_n) + x_i \cdot f(x_1, x_2, \dots, 1, \dots, x_n) = f(x_1, x_2, \dots, 0, \dots, x_n)$$

$$(2) \Rightarrow [f(x_1, x_2, \dots, 0, \dots, x_n) + x_i] \cdot [f(x_1, x_2, \dots, 1, \dots, x_n) + x_i'] = f(x_1, x_2, \dots, 0, \dots, x_n)$$

Soit $x_i = 1$

$$(1) \Rightarrow x_i' \cdot f(x_1, x_2, \dots, 0, \dots, x_n) + x_i \cdot f(x_1, x_2, \dots, 1, \dots, x_n) = f(x_1, x_2, \dots, 1, \dots, x_n)$$

$$(2) \Rightarrow [f(x_1, x_2, \dots, 0, \dots, x_n) + x_i] \cdot [f(x_1, x_2, \dots, 1, \dots, x_n) + x_i'] = f(x_1, x_2, \dots, 1, \dots, x_n)$$

$f(x_1, x_2, \dots, 0, \dots, x_n)$ également noté $f_{x_i'}$ est appelé **cofacteur de f par rapport à x_i'** . Ce terme est obtenu en remplaçant dans f chaque occurrence de x_i par 0.

$f(x_1, x_2, \dots, 1, \dots, x_n)$ également noté f_{x_i} est appelé **cofacteur de f par rapport à x_i** . Ce terme est obtenu en remplaçant dans f chaque occurrence de x_i par 1.

En notant $f_{x_i'}$ et f_{x_i} les cofacteurs de f par rapport à 0 et 1 les relations (1) et (2) s'écrivent :

$$(1) \quad f = x_i' \cdot f_{x_i'} + x_i \cdot f_{x_i}$$

$$(2) \quad f = [f_{x_i'} + x_i] \cdot [f_{x_i} + x_i']$$

Exemple d'utilisation du théorème de Shannon :

$$f(a, b, c) = a \cdot b + a' \cdot b \cdot c + b' \cdot c$$

$$f_a = b + b' \cdot c$$

$$f_a' = b \cdot c + b' \cdot c$$

$$(1) \Rightarrow f(a, b, c) = a \cdot (b + b' \cdot c) + a' \cdot (b \cdot c + b' \cdot c)$$

$$(2) \Rightarrow f(a, b, c) = [b \cdot c + b' \cdot c + a] \cdot [b + b' \cdot c + a']$$

Corollaire : Si une fonction est monoforme par rapport à une variable x alors le théorème de Shannon peut s'exprimer de la manière suivante :

$$\text{Monoforme / } x \Rightarrow f = x \cdot f_x + f_{x'}$$

$$\Rightarrow f = (x + f_{x'}) \cdot f_x$$

$$\text{Monoforme / } x' \Rightarrow f = f_x + x' \cdot f_{x'}$$

$$\Rightarrow f = f_{x'} \cdot (x' + f_x)$$

Corollaire : Soit f et g deux fonctions logiques de variables x_i ($i=1, \dots, n$) et Θ une fonction arbitraire.

$$f \Theta g = x \cdot (f_x \Theta g_x) + x' \cdot (f_{x'} \Theta g_{x'})$$

Corollaire : Soit f une fonction logique de variables x_i ($i=1, \dots, n$)

$$f = x' \cdot f_{x'} \oplus x \cdot f_x$$

Corollaire : Soit f une fonction logique de variables x_i ($i=1, \dots, n$)

$$f = 1 \Leftrightarrow f_x = 1 \text{ et } f_{x'} = 1$$

2.4.2. Formes de Lagrange

Soit une fonction logique de 2 variables $f(x, y)$. Appliquons le théorème de Shannon par rapport à x puis par rapport à y .

$$(1) \Rightarrow f(x, y) = x' \cdot f(0, y) + x \cdot f(1, y)$$

$$= x' \cdot [y' \cdot f(0, 0) + y \cdot f(0, 1)] + x \cdot [y' \cdot f(1, 0) + y \cdot f(1, 1)]$$

$$= x' \cdot y' \cdot f(0, 0) + x' \cdot y \cdot f(0, 1) + x \cdot y' \cdot f(1, 0) + x \cdot y \cdot f(1, 1)$$

$$\begin{aligned}
 (2) \Rightarrow f(x,y) &= [x' + f(1,y)].[x + f(0,y)] \\
 &= \{x' + [y' + f(1,1)].[y + f(1,0)].\} \cdot \{x + [y' + f(0,1)].[y + f(0,0)]\} \\
 &= [x' + y' + f(1,1)].[x' + y + f(1,0)].[x + y' + f(0,1)].[x + y + f(0,0)]
 \end{aligned}$$

En itérant le processus précédant sur une fonction de n variables les relations précédentes se généralisent de la manière suivante :

$$(1) f(x_1, x_2, \dots, x_n) = [f(0,0,\dots,0)x_1' \cdot x_2' \dots x_n'] + [f(1,0,\dots,0)x_1 \cdot x_2' \dots x_n'] + \dots + [f(1,1,\dots,1)x_1 \cdot x_2 \dots x_n]$$

$$(2) f(x_1, x_2, \dots, x_n) = [f(0,0,\dots,0) + x_1 + x_2 + \dots + x_n] \cdot [f(1,0,\dots,0) + x_1' + x_2' + \dots + x_n'] \dots [f(1,1,\dots,1) + x_1' + x_2' + \dots + x_n']$$

Ces deux expressions d'une fonction sont uniques. Elles sont connues sous les noms de **première et deuxième forme canonique** ou **première et deuxième forme de Lagrange**.

La première forme canonique est une forme $\sum\Pi$. Chaque intersection contient les n variables. Ces intersections sont appelées « mintermes ».

La deuxième forme canonique est une forme $\Pi\Sigma$. Chaque réunion contient les n variables. Ces réunions sont appelées « maxtermes ».

Lorsqu'une fonction est donnée par l'une des représentations tabulaires ou numériques ses formes canoniques s'obtiennent directement par application du théorème d'expansion de Shannon.

- 1^{ère} forme canonique : Somme des mintermes vrais de la fonction
- 2^{ème} forme canonique : Produit des maxtermes complémentaires des points faux de la fonction.

Exemple : Les 2 formes canoniques de la fonction représentées sur table de vérité de la figure 2.12 sont les suivantes :

a b c	f
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	1

$$\begin{aligned}
 f(a,b,c) &= a' \cdot b' \cdot c + a' \cdot b \cdot c' + a \cdot b' \cdot c' + a \cdot b \cdot c' + a \cdot b \cdot c \\
 f(a,b,c) &= (a+b+c) \cdot (a+b'+c') \cdot (a'+b+c')
 \end{aligned}$$

Figure 2.12. 1^{ère} et 2^{ème} formes canoniques

La deuxième forme canonique d'une fonction f peut ainsi être obtenue en appliquant directement la relation $f(X) = D[f'(X')]$. Elle peut également être obtenue en complétant la 1^{ère} forme canonique de la fonction f' et en appliquant le théorème de De Morgan.

Exemple : Reprenons l'exemple précédent

$$\begin{aligned}
 f'(a,b,c) &= a' \cdot b' \cdot c' + a' \cdot b \cdot c + a \cdot b' \cdot c \\
 f(a,b,c) &= (a' \cdot b' \cdot c' + a' \cdot b \cdot c + a \cdot b' \cdot c)' \\
 &= (a' \cdot b' \cdot c')' \cdot (a' \cdot b \cdot c)' \cdot (a \cdot b' \cdot c)' \\
 &= (a+b+c) \cdot (a+b'+c') \cdot (a'+b+c')
 \end{aligned}$$

Les formes canoniques d'une fonction étant uniques, pour démontrer que deux fonctions sont identiques, il suffit de déterminer une de leurs formes canoniques et d'en montrer l'identité.

Exemple : soit $f_1(a,b,c) = a.b + a'.b.c$ et $f_2(a,b,c) = b.c + a.b.c'$

1^{ère} forme canonique de $f_1(a,b,c)$: $a.b.c + a.b.c' + a'.b.c$

1^{ère} forme canonique de $f_2(a,b,c)$: $a.b.c + a'.b.c + a.b.c'$

$\Rightarrow f_1(a,b,c) = f_2(a,b,c)$

2.4.3. Formes de Reed-Muller

La **notation galoisienne d'une fonction** est une écriture de la fonction qui utilise uniquement les opérateurs OU exclusif et ET. Par exemple $xy \oplus xz$ et $1 \oplus xy \oplus xz$ sont des formes galoisiennes.

Rappel : pour écrire une fonction logique sous la forme galoisienne on peut utiliser les égalités suivantes :

$$x' = 1 \oplus x \qquad x + y = x \oplus y \oplus xy$$

La **forme de Reed-Muller** est la forme canonique de la fonction dans l'algèbre de Galois. Cette forme de Reed-Muller peut être obtenue à partir de la 1^{ère} forme canonique de la fonction transformée dans le champ de Galois.

Pour une fonction de deux variables, le processus est le suivant :

$$\begin{aligned} f(x,y) &= x.f(1,y) + x'.f(0,y) \\ &= x.f(1,y) \oplus x'.f(0,y) \\ &= x.[y.f(1,1) \oplus y'.f(1,0)] \oplus x'.[y.f(0,1) \oplus y'.f(0,0)] \\ &= x.y.f(1,1) \oplus x.y'.f(1,0) \oplus x'.y.f(0,1) \oplus x'.y'.f(0,0) \\ &= \dots\dots \text{en remplaçant } x' \text{ par } 1 \oplus x \text{ et } y' \text{ par } 1 \oplus y \\ &= f(0,0) \oplus x.[f(0,0) \oplus f(1,0)] \oplus y.[f(0,0) \oplus f(0,1)] \oplus x.y[f(0,0) \oplus f(0,1) \oplus f(1,0) \oplus f(1,1)] \end{aligned}$$

En généralisant ce développement à une fonction de n variables, la forme de Reed-Muller peut être obtenue à partir de la relation suivante :

$$f(x_1, x_2, \dots, x_n) = g_0 \oplus x_1 g_1 \oplus x_2 g_2 \oplus \dots \oplus x_1.x_2 \dots g_{n+1} \oplus \dots \oplus x_1.x_2 \dots x_n g_{2^n-1}$$

avec g_i = somme exclusive des valeurs de f correspondant à toutes les lignes de la table de vérité pour lesquelles toutes les variables non impliquées ont la valeur 0

Exemple : La figure 2.13 présente le processus d'obtention de la forme de Reed Muller d'une fonction f de 3 variables a,b,c.

a b c	f	
0 0 0	0	terme constant : $g_0 = f_0 = 0$ (a et b et c sont nuls)
0 0 1	1	en a : $g_1 =$ somme des f_i pour lesquelles b et c sont nuls = $0 \oplus 1 = 1$
0 1 0	1	en b : $g_2 =$ somme des f_i pour lesquelles a et c sont nuls = $0 \oplus 1 = 1$
0 1 1	1	en c : $g_3 =$ somme des f_i pour lesquelles a et b sont nuls = $0 \oplus 1 = 1$
1 0 0	1	en a.b : $g_4 =$ somme des f_i pour lesquelles c est nul = $0 \oplus 1 \oplus 1 \oplus 0 = 0$
1 0 1	0	en a.c : $g_5 =$ somme des f_i pour lesquelles b est nul = $0 \oplus 1 \oplus 1 \oplus 0 = 0$
1 1 0	0	en b.c : $g_6 =$ somme des f_i pour lesquelles a est nul = $0 \oplus 1 \oplus 1 \oplus 1 = 1$
1 1 1	0	en a.b.c : $g_7 =$ somme des $f_i = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$
		Donc $f(a,b,c) = a \oplus b \oplus c \oplus b.c$: forme canonique de Reed-Muller

Figure 2.13. Obtention de la forme de Reed Muller

Notons que la forme de Reed-Muller peut également être obtenue simplement en exprimant la fonction dans le champ de Galois sous forme $\sum_{\oplus} \prod$ puis en faisant toutes les simplifications de type $x \oplus x = 0$

Exemple : Soit $f(a,b,c) = a' b' c + a' b c' + a' b c + a b' c'$

$$\begin{aligned}
 &= a' b' c \oplus a' b c' \oplus a' b c \oplus a b' c' \\
 &= a' (b' c \oplus b c) \oplus a b' c' \\
 &= (1 \oplus a) (b' c \oplus b c) \oplus a (1 \oplus b) (1 \oplus c) \\
 &= b' c \oplus b c \oplus a b' c' \oplus a b c \\
 &= a \oplus b \oplus c \oplus b' c
 \end{aligned}$$

2.4.4. Formes de Davio

Les formes de Davio (forme positive et forme négative) sont des écritures de la fonction dans le champ de Galois. Ces formes sont particulièrement intéressantes pour certaines applications. Elles sont obtenues à partir des théorèmes suivants :

Théorème : Soit $f(x_1, x_2, \dots, x_i, \dots, x_n)$ une fonction logique de n variables

- (1) $f = f_{x'} \oplus x.(f_x \oplus f_{x'})$ (forme de Davio positive)
- (2) $f = f_x \oplus x'.(f_x \oplus f_{x'})$ (forme de Davio négative)

Preuve : Soit $f(x_1, x_2, \dots, x_i, \dots, x_n)$ une fonction logique de n variables. F peut s'écrire sous la forme suivante :

$$f = x'.f_{x'} \oplus x.f_x$$

(1) En appliquant le théorème de l'algèbre de Boole suivant : $a \oplus ab = a'$

$$f = (f_{x'} \oplus x.f_{x'}) \oplus x.f_x$$

$$f = f_{x'} \oplus x.(f_x \oplus f_{x'})$$

(2) En appliquant le même théorème de l'algèbre de Boole sur le deuxième terme

$$f = x'.f_{x'} \oplus (f_x \oplus x'.f_x)$$

$$f = f_x \oplus x'.(f_x \oplus f_{x'})$$

2.5. Théorème d'inclusion

Savoir si un monôme est inclus dans l'expression de la fonction, c'est à dire couvert par une fonction, peut être un problème complexe. Ce problème peut en fait être réduit à un problème de vérification de tautologie en utilisant le théorème d'inclusion.

Théorème : Une expression F contient un monôme m si et seulement si le cofacteur de F par rapport à m (F_m) est une tautologie.

$$m \subset F \Leftrightarrow F_m = 1$$

Preuve : Remarquons que $m \subseteq F \Leftrightarrow F \cdot m = m$

Supposons que $m \subseteq F$

$$\Rightarrow F \cdot m = m$$

Considérons les cofacteurs par rapport à m des 2 termes. Les termes étant égaux, les cofacteurs sont égaux.

$$\Rightarrow (F \cdot m)_m = m_m$$

$$m_m = 1 \Rightarrow (F \cdot m)_m = 1$$

$$\Rightarrow F_m \cdot m_m = 1$$

$$\Rightarrow F_m = 1$$

$$\Rightarrow F_m \text{ est une tautologie}$$

Réciproquement, supposons que F_m soit une tautologie

$$\Rightarrow F_m = 1$$

$$\begin{aligned} \Rightarrow F_m \cdot m &= m \\ \Rightarrow F \cdot m &= [(m \cdot F_m) + (m' \cdot F_m)] \cdot m = F_m \cdot m = m \\ \Rightarrow m &\subseteq F \end{aligned}$$

Exemple : Soit la fonction $f(a,b,c) = a.b + a.c + a'$

Cherchons à savoir si le monôme "bc" est couvert par cette expression.

Le cofacteur de f par rapport à bc est : $f_{bc} = a + a' = 1$

Le monôme bc est donc inclus dans f comme le montre la figure 2.14

		bc			
		00	01	11	10
a	0	1	1	1	1
	1	0	1	1	1

Figure 2.14. Inclusion d'un monôme dans une fonction

2.6. Equations logiques

Une équation logique est une égalité de la forme $E1(X)=E2(X)$, ou E1 et E2 sont des expressions logiques de n variables $X=(x_1, \dots, x_n)$ et pour lesquelles on se propose de trouver les combinaisons de X qui satisfont cette égalité.

L'équation $E1=E2$ est équivalente à :

- (1) $E1 \oplus E2 = 0$ c'est à dire $E1.E2' + E1'.E2 = 0$
- (2) $(E1 \oplus E2)' = 1$ c'est à dire $E1.E2 + E1'.E2' = 1$

On peut donc toujours mettre une équation logique sous des formes $E(X)=0$ ou $E(X)=1$.

2.6.1. Résolution explicite

Une première méthode de résolution consiste à écrire l'expression $E(X)$ sous la forme d'une somme de produit.

$$E(X) = \sum m_i$$

On obtient alors toutes les solutions de l'équation sous la forme de familles de points définies par m_i .

Si $E(X)=(E1 \oplus E2)'$ les solutions sont directement données par $m_i=1$

Exemple :

$$\begin{aligned} E1(a,b,c,d) &= a.b + d & E2(a,b,c,d) &= a'.b'.c \\ E1 = E2 &\Rightarrow (E1 \oplus E2)' = 1 \\ &\Rightarrow E1.E2 + E1'.E2' = 1 \\ &\Rightarrow (a.b + d)(a'.b'.c) + (a' + b').d'.(a + b + c') = 1 \\ &\Rightarrow a'.b'.c.d + (a'.d' + b'.d')(a + b + c') = 1 \\ &\Rightarrow a'.b'.c.d + a'.b.d' + a'.c'.d' + a.b'.d' + b'.c'.d' = 1 \end{aligned}$$

Les solutions sont donc les combinaisons qui vérifient l'une quelconque des égalités suivantes :

$$\begin{aligned} a'.b'.c.d &= 1 \\ a'.b.d' &= 1 \\ a'.c'.d' &= 1 \end{aligned}$$

$$a.b'.d' = 1$$

$$b'.c'.d' = 1$$

La liste des solutions de l'équation est donc :

$$abcd = (0011), (01\phi 0), (0\phi 00), (10\phi 0), (\phi 000)$$

Pour montrer que deux expressions logiques sont identiques il vaut mieux utiliser la forme $E(X)=E1 \oplus E2$. Si $E(X)=0$ quelles que soient les valeurs de X, les expressions E1 et E2 sont identiques.

Exemple : $E1(X) = a.b + b'.c + a'.c'$

$$E2(X) = a'.b' + b.c' + a.c$$

$$E1 \oplus E2 = E1.E2' + E1'.E2$$

$$= (a.b + b'.c + a'.c').(a + b)(b' + c)(a' + c') + (a' + b')(b + c').(a + c).(a'.b' + b.c' + a.c)$$

$$= (a.b + b'.c + a'.c').(a.b' + a.c + b.c).(a' + c') + (a'.b + a'.c' + b'.c')(a + c).(a'.b' + b.c' + a.c)$$

$$= (a.b + b'.c + a'.c').(a.b'.c' + a'.b.c) + (a'.b.c + a.b'.c').(a'.b' + b.c' + a.c)$$

$$= 0$$

$$\Rightarrow E1=E2$$

On peut également montrer que deux fonctions sont identiques en établissant la forme canonique de chacune d'elles (la forme canonique est unique).

2.6.2. Résolution paramétrique

Une autre méthode de résolution consiste à éliminer successivement chacune des variables. Elle présente l'avantage d'aboutir à une écriture paramétrique de l'ensemble des solutions.

En effet, en vertu du théorème de Shannon, l'expression $E(X)=1$ peut se mettre sous la forme :

$$E_1 = E(x_1, x_2, \dots, x_{n-1}, 1)x_n + E(x_1, x_2, \dots, x_{n-1}, 0)x_n' = 1$$

La condition nécessaire pour que cette équation ait une solution en x_n est que :

$$E(x_1, x_2, \dots, x_{n-1}, 1) + E(x_1, x_2, \dots, x_{n-1}, 0) = 1$$

On a donc une équation de $n-1$ variables. Si l'on poursuit la procédure, on aboutit à une équation d'une seule variable :

$$E_n = a.x_1 + b.x_1' = 1$$

Si $a+b=1$ (condition nécessaire pour que cette équation ait une solution en x_1), on montre que la solution générale de cette équation est :

$$x_1 = b' + a.u$$

où u est une quantité logique pouvant prendre les valeurs 0 et 1. En reportant dans l'équation $E_{n-1}=1$, on peut alors résoudre en x_2 etc, jusqu'à obtenir la solution, si elle existe, de l'équation $E=1$.

Exemple : Soit à résoudre l'équation logique suivante par rapport aux variables x et y :

$$a.x'.y' + x.y = b$$

Mettons cette équation sous la forme $E(X)=1$.

$$a.b.x'.y' + b.x.y + b'(a' + x + y)(x' + y') = 1$$

$$a.b.x'.y' + b.x.y + b'(a'.x' + a'.y' + x.y' + x'.y) = 1$$

$$a.b.x'.y' + b.x.y + a'.b'.x' + a'.b'.y' + x.b'.y' + x'.b'.y = 1$$

Mettons cette équation sous le forme $x.E + x'.E = 1$.

$$x.(b.y + b'.y') + a'.b'.y' + x'.(a.b.y' + a'.b' + b'.y) = 1$$

$$x.(b.y + b'.y' + a'.b'.y') + x'.(a.b.y' + a'.b' + b'.y + a'.b'.y') = 1$$

$$x.(b.y + b'.y') + x'.(a.b.y' + a'.b' + b'.y) = 1 \quad (1)$$

La condition d'existence d'une solution en x s'écrit donc :

$$(b.y + b'.y') + (a.b.y' + a'.b' + b'.y) = 1$$

Mettons cette équation sous le forme $y.E + y'.E = 1$.

$$y.(b + b' + a'.b') + y'.(a.b + b' + a'.b') = 1$$

$$y + y'.(a.b + b') = 1$$

$$y + y'.(a + b') = 1$$

La condition d'existence d'une solution en x s'écrit donc :

$$1 + (a + b') = 1$$

Cette condition est toujours remplie. La solution en y s'écrit :

$$y = a'.b + u$$

En reportant dans (1) on obtient :

$$x.[b.(a'b+u) + b'.(a'b+u)] + x'.[a.b.(a'b+u) + a'.b' + b'.(a'b+u)] = 1$$

$$x.[b.(a'b+u) + b'.(a+b')u'] + x'.[a.b.(a+b')u' + a'.b' + b'.(a'b+u)] = 1$$

$$x.[a'.b + u.b + u'.b'] + x'.[a.b.u' + a'.b' + u.b'] = 1$$

D'où la solution :

$$x = [a.b.u' + a'.b' + u.b'] + [a'.b + u.b + u'.b'].v = 1$$

$$x = [(a'+b'+u)(a+b)(u'+b)] + [a'.b + u.b + u'.b'].v = 1$$

$$x = [(a'b+ab'+au+bu)(u'+b)] + [a'.b + u.b + u'.b'].v = 1$$

$$x = (a'.b + a.b'.u' + b.u) + (a'.b + u.b + u'.b').v = 1$$

On aboutit donc à quatre expressions de x et y en fonction de a et b en remplaçant les paramètres u et v par les quatre combinaisons possibles.

$$uv=00 \Rightarrow x = a'.b + a.b' \quad y = a'.b$$

$$uv=01 \Rightarrow x = a'.b + b' \quad y = a'.b$$

$$uv=10 \Rightarrow x = b \quad y = 1$$

$$uv=11 \Rightarrow x = b \quad y = 1$$

2.6.3. Résolution de systèmes d'équation

Un système d'équations logiques se présente sous la forme de p équations de n variables X. Chacune d'elles est équivalente à une relation de la forme $E_i(X)=1$ pour $i=1,2,\dots,p$. On ramène ce système à une seule équation logique qui est :

$$\sum_{i=1}^{i=p} E_i(X) = 1$$

Les procédures de résolution présentées précédemment sont alors applicables à cette équation.