CHAPITRE 4: Simplification des fonctions

- Fonctions logiques et formes canoniques
- Simplification des fonctions
  - Simplification algébrique
  - Simplification graphique
    - *Karnaugh*
    - *McCluskey*

Fonctions logiques et formes canoniques

f fonction logique de n variables

– On appelle « ***minterme*** » de n variables, l'un des produits de ces variables ou de leurs complémentaires.

– On appelle « ***maxterme*** » de n variables, l'une des sommes de ces variables ou de leurs complémentaires.

exemple $n = 4$ variables $\{a, b, c, d\}$

$m = a \cdot b \cdot c \cdot d$ est un minterme

$m = \bar{a} \cdot \bar{b} \cdot c \cdot d$ est un autre minterme

$m = a \cdot \bar{b} \cdot c$ n'est pas un minterme

$M = a + b + c + d$ est un maxterme

$M = \bar{a} + \bar{b} + c + d$ est un autre maxterme

$M = a + \bar{b} + c$ n'est pas un maxterme

Indexation et nombre de mintermes et maxtermes

– Pour chaque « m--terme » on construit un code binaire en posant 1 si une variable est présente, 0 si son complémentaire est présent.

– On convertit ce code binaire en base décimal pour obtenir l'indice du m—terme

$$m_i = a \cdot \bar{b} \cdot c \cdot \bar{d} \quad \text{code binaire associé } (1010)_2 \text{ donc } i = 10$$

- Si deux m--termes sont différents, leurs indices sont différents.
- Le nombre de m--terme de *n* variables vaut 2^n

Propriétés des mintermes et maxtermes

Le complémentaire d'un (max)minterme est un (min)maxterme

$$\bar{m}_i = M_{2^n-1-i} \quad et \quad \bar{M}_j = m_{2^n-1-j}$$

Théorèmes (3 définitions équivalentes) :

1. $$\sum_{i=0}^{2^n-1} m_i = 1 \quad et \quad \prod_{j=0}^{2^n-1} M_j = 0$$

2. Le produit de deux mintermes différents vaut 0 et la somme de deux maxtermes différents vaut 1

3. Soit *f* une expression booléenne écrite sous la forme d'une somme de mintermes respectivement d'un produit de maxtermes), son complément *f* est la somme de tous les mintermes (respectivement le produit de tous les maxtermes) qui ne figurent pas dans *f*.

Formes canoniques (1)

Une fonction est sous **forme canonique** (ou **normale**) si chaque terme contient toutes les variables. L'écriture sous forme canonique est unique.

Exemples :

$$f(x,y,z) = \underbrace{x.y.z}_{\text{Minterme}} + \bar{x}.\bar{y}.z + \bar{x}.y.z$$

Minterme ou intersection de base

Première forme canonique ou forme normale disjonctive

$$f(x,y,z) = \underbrace{(x+\bar{y}+z)}_{\text{Maxterme}}.(\bar{x}+y+\bar{z})$$

Maxterme ou réunion de base

Deuxième forme canonique ou forme normale conjonctive

Formes canoniques (2)

Si la fonction n'est pas sous forme normale

i.e. une des variables (au moins) ne figure pas dans un des termes

$\longrightarrow$ La fonction est sous une forme simplifiée

$f(x,y,z) = xyz + \overline{x}y\overline{z} + xy\overline{z}$ Première forme canonique

$= xy(z+\overline{z}) + \overline{x}y\overline{z}$ Forme simplifiée

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

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

Première forme : obtention (1)

Premier théorème d’expansion de Shannon :

$$F(a,b,c.,..) = a.F(1,b,c,...) + \overline{a}.F(0,b,c...)$$

Première forme : obtention (2)

Premier théorème d'expansion de Shannon :

$$F(a,b,c.,..) = a.F(1,b,c,\ldots) + \overline{a}.F(0,b,c\ldots)$$

Si a = 1 : $F(1,b,c.,..) = 1.F(1,b,c,\ldots) + 0.F(0,b,c\ldots)$

Première forme : obtention (3)

Premier théorème d'expansion de Shannon :

$$F(a,b,c.,..) = a.F(1,b,c,\dots) + \bar{a}.F(0,b,c\dots)$$

Si a = 1 : $F(1,b,c.,..) = \boxed{}\,F(1,b,c,\dots) + \cancel{0.F(0,b,c\dots)}$

Si a = 0 : $F(0,b,c.,..) = \cancel{0.F(1,b,c,\dots)} + \boxed{}.F(0,b,c\dots)$

Première forme : obtention (4)

Premier théorème d'expansion de shannon :

$$F(a,b,c.,..) = a.F(1,b,c,...) + \overline{a}.F(0,b,c...)$$

Pour 2 variables :

$$F(a,b) = a.F(1,b) + \overline{a}.F(0,b)$$

$$F(a,b) = a.(b.F(1,1) + \overline{b}.F(1,0)) + \overline{a}.(b.F(0,1) + \overline{b}.F(0,0)$$

$$F(a,b) = a.b.F(1,1) + a.\overline{b}.F(1,0) + \overline{a}.b.F(0,1) + \overline{a}.\overline{b}.F(0,0)$$

Point particulier de la fonction F vaut 0 ou 1

Première forme : mise en oeuvre

$$F(a,b) = a.b.F(1,1) + a.\bar{b}.F(1,0) + \bar{a}.b.F(0,1) + \bar{a}.\bar{b}.F(0,0)$$

Pour chaque i,j le point de la fonction F(i,j) dépend du problème

a	b	F	
0	0	0	F(0,0)
0	1	1	F(0,1)
1	0	1	F(1,0)
1	1	0	F(1,1)

$$\longrightarrow \quad F(a,b) = \bar{a}.b + a.\bar{b}$$

La première forme canonique ne laisse apparaître que les termes qui valent 1

Il y a 2^N mintermes possibles. La somme des 2^N mintermes vaut 1. (fonction valant 1 partout)

Deuxième forme : obtention

Deuxième théorème d'expansion de Shannon :

$$F(a,b,c,\ldots) = (a + F(0,b,c,\ldots)).(\overline{a} + F(1,b,c,\ldots))$$

Si a=0 : $F(0,b,c,\ldots) = (0 + F(0,b,c\ldots)).(1 + F(1,b,c,\ldots))$

neutre +

absorbant +
neutre .

Pour deux variables :

$$F(a,b) = (a + b + F(0,0)).(\overline{a} + b + F(1,0)).$$

$$(a + \overline{b} + F(0,1)).(\overline{a} + \overline{b} + F(1,1))$$

Deuxième forme : mise en oeuvre

$$F(a,b) = (a + b + F(0,0)).(\bar{a} + b + F(1,0)).$$
$$(a + \bar{b} + F(0,1)).(\bar{a} + \bar{b} + F(1,1))$$

a	b	F	
0	0	0	F(0,0)
0	1	1	F(0,1)
1	0	1	F(1,0)
1	1	0	F(1,1)

$$F(a,b) = (a + b).(\bar{a} + \bar{b})$$

Que les termes valant 0

Il y a 2^N maxtermes possibles. La somme des 2^N maxtermes vaut 0. (fonction valant 0 partout)

Les formes numériques (illustration de l'indexation)

C'est un Raccourci d'écriture !

a	b	F	
0	0	0	F(0,0)
0	1	1	F(0,1)
1	0	1	F(1,0)
1	1	0	F(1,1)

1ère forme : F(a,b) = R(1,2)
2ème forme : F(a,b) = I(0,3)

R = Réunion I = Intersection

a = msb (Most Significant Bit)
b = lsb (least Significant Bit)
bit = BInary digIT

$$(abcd)_2 = a*2^3 + b*2^2 + c*2^1 + d*2^0$$

Passage d'une forme canonique à une autre

– On utilise

$$\overline{\overline{f}} = f$$

Et le théorème :

Soit *f* une expression booléenne écrite sous la forme d'une somme de mintermes respectivement d'un produit de maxtermes), son complément $\overline{f}$ est la somme de tous les mintermes (respectivement le produit de tous les maxtermes) qui ne figurent pas dans *f*.

Exemples : 1ère forme vers 2ème forme

1er exemple : 2 variables, 2 mintermes et 2 maxtermes

$$\overline{\overline{f}} = \overline{\overline{\bar{a}\cdot b + a\cdot\bar{b}}} = \overline{\left(a+\bar{b}\right)\cdot\left(\overline{a+b}\right)} = \overline{\cancel{a}\bar{a} + ab + \bar{a}\bar{b} + \cancel{b}\bar{b}} = \overline{a\cdot b + \bar{a}\cdot\bar{b}} = (\bar{a}+\bar{b})\cdot(a+b)$$

2ème exemple : 3 variables, 3 mintermes et 5 maxtermes

$$f = \underbrace{(x+y+\bar{z})}_{6}\cdot\underbrace{(\bar{x}+y+z)}_{3}\cdot\underbrace{(\bar{x}+\bar{y}+z)}_{1}$$

2ème forme

Indices des maxtermes présents

$$f = \overline{\underbrace{(x+y+z)}_{7}\cdot\underbrace{(x+\bar{y}+z)}_{5}\cdot\underbrace{(x+\bar{y}+\bar{z})}_{4}\cdot\underbrace{(x+\bar{y}+z)}_{2}\cdot\underbrace{(x+y+z)}_{0}}$$

Indices des maxtermes manquants

$$= \bar{x}\cdot\bar{y}\cdot\bar{z} + \bar{x}\cdot y\cdot\bar{z} + \bar{x}\cdot y\cdot z + \bar{x}\cdot y\cdot\bar{z} + x\cdot y\cdot z$$

1ère forme

Formes canoniques : Choix

Première forme canonique = expression des 1 de la fonction

Deuxième forme canonique = expression des 0 de la fonction

Les deux formes canoniques sont équivalentes

On choisit celle qui donne le résultat le plus simple
peu de 0 => deuxième forme / peu de 1 => première forme

Fonction incomplètement définie

Si une combinaison d'entrée ne peut pas se présenter
ou si pour cette combinaison la valeur de la fonction n'est pas importante, on dit que la fonction n'est pas définie en ce point.

$$F(a,b,c) = \phi \quad \text{(ou x ou -)}$$

'x' ou '-' se lit « don't care »

Ce point peut être remplacé par 1 ou 0 en fonction des besoins de simplification.

Simplification des fonctions

Objectif : Fabriquer un système

et/ou
- à moindre coût
- rapide
- fiable
- peu consommateur

Méthodes : Algébriques
Graphiques
Programmables

Résultat : on cherche la forme minimale d'une fonction
nombre minimal de monômes/nombre minimal de lettre par monôme

Possibilité de plusieurs formes minimales : formes équivalentes

Simplification : avertissement

La forme mathématique la plus simple ne correspond
pas toujours à la réalisation la plus simple
et/ou la plus rapide.

La prise en compte de contraintes technologiques peut
imposer une complexification d'écriture de l'expression.

Simplification : définitions

Décomposition : Problème général de la réalisation d'une fonction logique à l'aide d'opérateurs

Transformation : Passage d'une forme à une autre forme équivalente

Simplification : cas particulier d'une transformation quand on passe d'une forme canonique à une forme minimale.

Simplification algébrique (1)

Applications des principes et propriétés de l'algèbre de Boole

Identités remarquables :

1	$a.b + \overline{a}.b = b$	$(a+b).(\overline{a}+b)=b$
2	$a + a.b = a$	$a.(a+b) = a$
3	$a + \overline{a}.b = a+b$	$a.(\overline{a}+b) = a.b$

Démonstrations : 1 et 2 trivial

3 : $a + \overline{a}.b = \underbrace{a.a + a.b}_{a} + \underbrace{a.\overline{a}}_{0} + \overline{a}.b = (a+\overline{a}).(a+b) = a+b$

Simplification algébrique (2)

Règles de simplification :

(Mintermes adjacents = 1 seule variable qui change)

1 : Deux mintermes adjacents $\longrightarrow$ Il reste l'intersection commune

1': Deux maxtermes adjacents $\longrightarrow$ Il reste la réunion commune

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

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

2 : On ajoute des termes neutres ou déjà existant (idempotence)

3 : théorème du consensus

4 : On simplifie la forme canonique ayant le moins de termes

Méthode algébrique toujours possible mais démarche intuitive qui dépend de l'habileté et de l'expérience.

Karnaugh – simplification graphique

- La méthode de Karnaugh permet de visualiser une fonction et d'en tirer naturellement une écriture simplifiée.
- L'élément de base de cette méthode est la table de Karnaugh qui représente toutes les combinaisons d'états possibles pour un nombre de variables donné.
- La table de Karnaugh est un outil graphique qui permet de simplifier de manière méthodique des expressions booléennes. Elle offre une méthode de simplification pratique suivant une démarche systématique semblable à une recette de cuisine.
- La construction des tables de Karnaugh exploite le codage de l'information et la notion d'adjacence

Simplification graphique (1)

Principe : Mettre en évidence sur un graphique les mintermes (ou maxtermes) adjacents. Transformer les adjacences logiques en adjacences «géométriques».

Trois phases : transcrire la fonction dans un tableau codé
recherche des adjacents pour simplification
équations des groupements effectués

Description : Table de vérité vs Tableau de Karnaugh

Table de vérité	Tableau de Karnaugh
1 ligne	1 case
n variables	2^n cases

Simplification graphique (2)

Code Gray ou binaire réfléchi : code à symétries multiples

Construction

0
1

Simplification graphique (3)

Code Gray ou binaire réfléchi : code à symétries multiples

Construction

0
1

1
0

Symétrie

«Miroir»

Simplification graphique (4)

Code Gray ou binaire réfléchi : code à symétries multiples

Construction

0 0 0 1	On complète par des 0
1 1 1 0	On complète par des 1

Simplification graphique (5)

Code Gray ou binaire réfléchi : code à symétries multiples

Construction

0 0
0 1
1 1
1 0
---
1 0
1 1
0 1
0 0

Symétrie

Simplification graphique (6)

Code Gray ou binaire réfléchi : code à symétries multiples

Construction

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0

On complète par des 0

On complète par des 1

Simplification graphique (7)

Code Gray ou binaire réfléchi : code à symétries multiples

Propriétés :

distance unité

0 0 0
0 0 1
0 1 1
0 1 0
$\boxed{1\ 1\ 0}$
1 1 1
1 0 1
1 0 0

Cyclique

Monôme de n symboles = n adjacents $\longrightarrow$ multisymétrie du code

Tableau de Karnaugh : codage lignes et colonnes par code Gray

Simplification graphique (8)

Exemple 1: Depuis une table de vérité

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

a \ bc	00	01	11	10
0		1		
1				

Simplification graphique (9)

Exemple 1 Depuis une table de vérité

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

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

Simplification graphique (10)

Exemple 1 : Depuis une table de vérité

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

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

3 lettres = 3 adjacents

!Attention! Haut et Bas / Gauche et Droite liés (tore de Karnaugh)

Simplification graphique (11)

Exemple 2 : Par une première forme canonique (Par les 1)

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

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

Simplification graphique (12)

Exemple 2 : Par une deuxième forme canonique (Par les 0)

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

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

Simplification graphique (13)

Règles de simplification

1 : Les groupements comportent une puissance de deux cases,
2 : Les 2^k cases forment un rectangle,
3 : Un groupement de 2^k cases correspond à une simplification de k variables et s'écrit avec (n-k) lettres,
4 : Il faut utiliser au moins une fois chaque 1, le résultat est donné par la réunion logique de chaque groupement,
5 : Expression minimale si :

- les groupements les plus grands possibles
- utiliser les 1 un minimum de fois

6 : Codage d'un groupe par les 1 :

- n'apparaît que les variables fixes dans le groupement
- forme simple si la variable vaut 1/ complémentée sinon

Simplification graphique (14)

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

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

Choix d'un 1 et recherche
des adjacents contenant un 1

Simplification graphique (15)

Il faut essayer de maximiser les groupements

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

Recherche d’un ensemble de deux cases adjacent contenant des 1

Echec

Simplification graphique (16)

Autre groupement

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

On choisit un des 1 restant et recherche des 1 adjacents

Simplification graphique (17)

Maximisation groupement

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

Choix d'un 1 adjacent et recherche
d'un groupement adjacent
OK !

Simplification graphique (18)

Tous les 1 sont groupés !

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

Equation :

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

Simplification graphique (19)

Par les 0

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

Equation :

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

Limites de la méthode

Difficile avec plus de 6 variables

- Intérêt pédagogique
- Les problèmes sont toujours découpables en pb plus petits

Pas programmable (autres méthodes : McCluskey, Sheinman, Tison)

Temps pas pris en compte

Difficile minimiser plusieurs fonctions conjointement

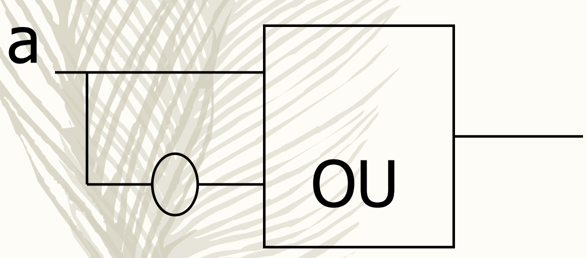
Problème de l’aléa de propagation : si deux groupes sont adjacents

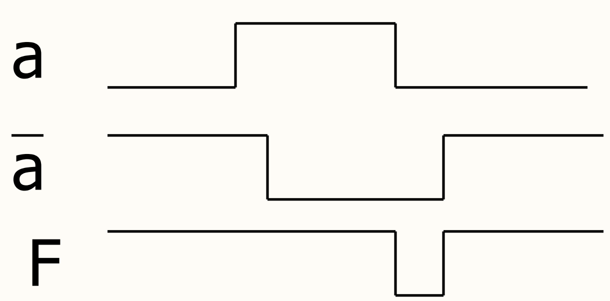
L’aléa de propagation : problème

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

$$F = a.\bar{b} + \bar{a}.c$$

Si b=0, c=1 $F = a + \bar{a}$ = 1 mathématiquement, mais …





Retard techno

L’aléa de propagation : problème

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

$$F = a.\bar{b} + \bar{a}.c + \underline{\bar{b}.c}$$

Si b=0, c=1 $F = a + \bar{a} + 1$ = 1 mathématiquement et technologiquement

C’est le théorème du consensus

Exercice 1

abc \ def	000	001	011	010	110	111	101	100
000	-	0	0	1	-	0	0	1
001	0	0	-	0	0	0	0	0
011	0	0	-	0	0	0	0	0
010	-	0	0	1	1	-	0	-
110	-	0	0	1	-	0	0	-
111	0	0	-	0	0	-	0	0
101	0	-	0	0	0	0	0	-
100	-	0	0	1	1	0	0	1

Exercice 1 : Démarche (1)

abc \ def	000	001	011	010	110	111	101	100
000	-	0	0	1	-	0	0	1
001	0	0	-	0	0	0	0	0
011	0	0	-	0	0	0	0	0
010	-	0	0	1	1	-	0	-
110	-	0	0	1	-	0	0	-
111	0	0	-	0	0	-	0	0
101	0	-	0	0	0	0	0	-
100	-	0	0	1	1	0	0	1

- ✓ Plus de 1 ou de 0 ?
- ✓ Choix d'un 1
- ✓ Recherche des adjacents

Adjacents éligibles ?

Si oui groupement

Groupement «augmentable» ?

Exercice 1 : Démarche (2)

abc \ def	000	001	011	010	110	111	101	100
000	-	0	0	1	-	0	0	1
001	0	0	-	0	0	0	0	0
011	0	0	-	0	0	0	0	0
010	-	0	0	1	1	-	0	-
110	-	0	0	1	-	0	0	-
111	0	0	-	0	0	-	0	0
101	0	-	0	0	0	0	0	-
100	-	0	0	1	1	0	0	1

- ✓ Plus de 1 ou de 0 ?
- ✓ Choix d'un 1
- ✓ Recherche des adjacents

Adjacents éligibles ?

Si oui groupement

Groupement «augmentable» ?

Exercice 1 : Démarche (3)

abc \ def	000	001	011	010	110	111	101	100
000	-	0	0	1	-	0	0	1
001	0	0	-	0	0	0	0	0
011	0	0	-	0	0	0	0	0
010	-	0	0	1	1	-	0	-
110	-	0	0	1	-	0	0	-
111	0	0	-	0	0	-	0	0
101	0	-	0	0	0	0	0	-
100	-	0	0	1	1	0	0	1

Exercice 1 : Démarche (4)

abc \ def	000	001	011	010	110	111	101	100
000	-	0	0	1	-	0	0	1
001	0	0	-	0	0	0	0	0
011	0	0	-	0	0	0	0	0
010	-	0	0	1	1	-	0	-
110	-	0	0	1	-	0	0	-
111	0	0	-	0	0	-	0	0
101	0	-	0	0	0	0	0	-
100	-	0	0	1	1	0	0	1

Exercice 1 : Démarche (5)

abc \ def	000	001	011	010	110	111	101	100
000	-	0	0	1	-	0	0	1
001	0	0	-	0	0	0	0	0
011	0	0	-	0	0	0	0	0
010	-	0	0	1	1	-	0	-
110	-	0	0	1	-	0	0	-
111	0	0	-	0	0	-	0	0
101	0	-	0	0	0	0	0	-
100	-	0	0	1	1	0	0	1

Exercice 1 : Solution

abc \ def	000	001	011	010	110	111	101	100
000	-	0	0	1	-	0	0	1
001	0	0	-	0	0	0	0	0
011	0	0	-	0	0	0	0	0
010	-	0	0	1	1	-	0	-
110	-	0	0	1	-	0	0	-
111	0	0	-	0	0	-	0	0
101	0	-	0	0	0	0	0	-
100	-	0	0	1	1	0	0	1

1 seul groupement

$$H = \overline{c}.\overline{f}$$

Exercice 2

abc \ def	000	001	011	010	110	111	101	100
000	1	1	0	0	1	1	1	-
001	1	1	1	1	-	1	1	1
011	1	1	1	1	1	1	-	1
010	1	1	0	0	-	1	1	1
110	1	1	0	0	1	1	-	1
111	1	1	1	1	-	1	1	1
101	1	1	1	1	1	1	1	1
100	1	1	0	0	1	1	1	1

Exercice 2 : Solution

abc \ def	000	001	011	010	110	111	101	100
000	1	1	0	0	1	1	1	-
001	1	1	1	1	-	1	1	1
011	1	1	1	1	1	1	-	1
010	1	1	0	0	-	1	1	1
110	1	1	0	0	1	1	-	1
111	1	1	1	1	-	1	1	1
101	1	1	1	1	1	1	1	1
100	1	1	0	0	1	1	1	1

$$H = c + d + \overline{e}$$

Tableau de Karnaugh à variable inscrite (KVI)

- La méthode de KVI tente de pallier à la difficulté liée aux tableaux de Karnaugh au delà de 4 variables.
- Elle rajoute un niveau d'abstraction à la table de vérité de la fonction à réduire en y inscrivant une ou plusieurs variables

a	b	c	f	
0	0	0	0	0
0	0	1	0	
0	1	0	0	c
0	1	1	1	
1	0	0	1	$\overline{c}$
1	0	1	0	
1	1	0	1	1
1	1	1	1	

a	b	f
0	0	0
0	1	c
1	0	$\overline{c}$
1	1	1

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

$$1 = c + \overline{c}$$

$$f = a\overline{c} + bc$$

KVI exemple (2)

$$f = R(2,4,7,12,10,15) + \phi(6,9,11,14)$$

a	b	c	d	*f*	*f*
0	0	0	0	0	0
0	0	0	1	0	
0	0	1	0	1	$\bar{d}$
0	0	1	1	0	
0	1	0	0	1	$\bar{d}$
0	1	0	1	0	
0	1	1	0	x	$d + \bar{d}\,x$
0	1	1	1	1	
1	0	0	0	0	$d\,x$
1	0	0	1	x	
1	0	1	0	1	$\bar{d} + d\,x$
1	0	1	1	x	
1	1	0	0	1	$\bar{d}$
1	1	0	1	0	
1	1	1	0	x	$d + \bar{d}\,x$
1	1	1	1	1	

a \ bc	00	01	11	10
0	0	$\bar{d}$	$d + \bar{d}\,x$	$\bar{d}$
1	$d\,x$	$\bar{d} + d\,x$	$d + \bar{d}\,x$	$\bar{d}$

$$f = c\bar{d} + bc + b\bar{d} = (b + c)\bar{d} + bc$$

Technique de Quine-McCluskey

La méthode de Quine part de la décomposition canonique disjonctive d'une expression Booléenne et utilise systématiquement la formule de simplification

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

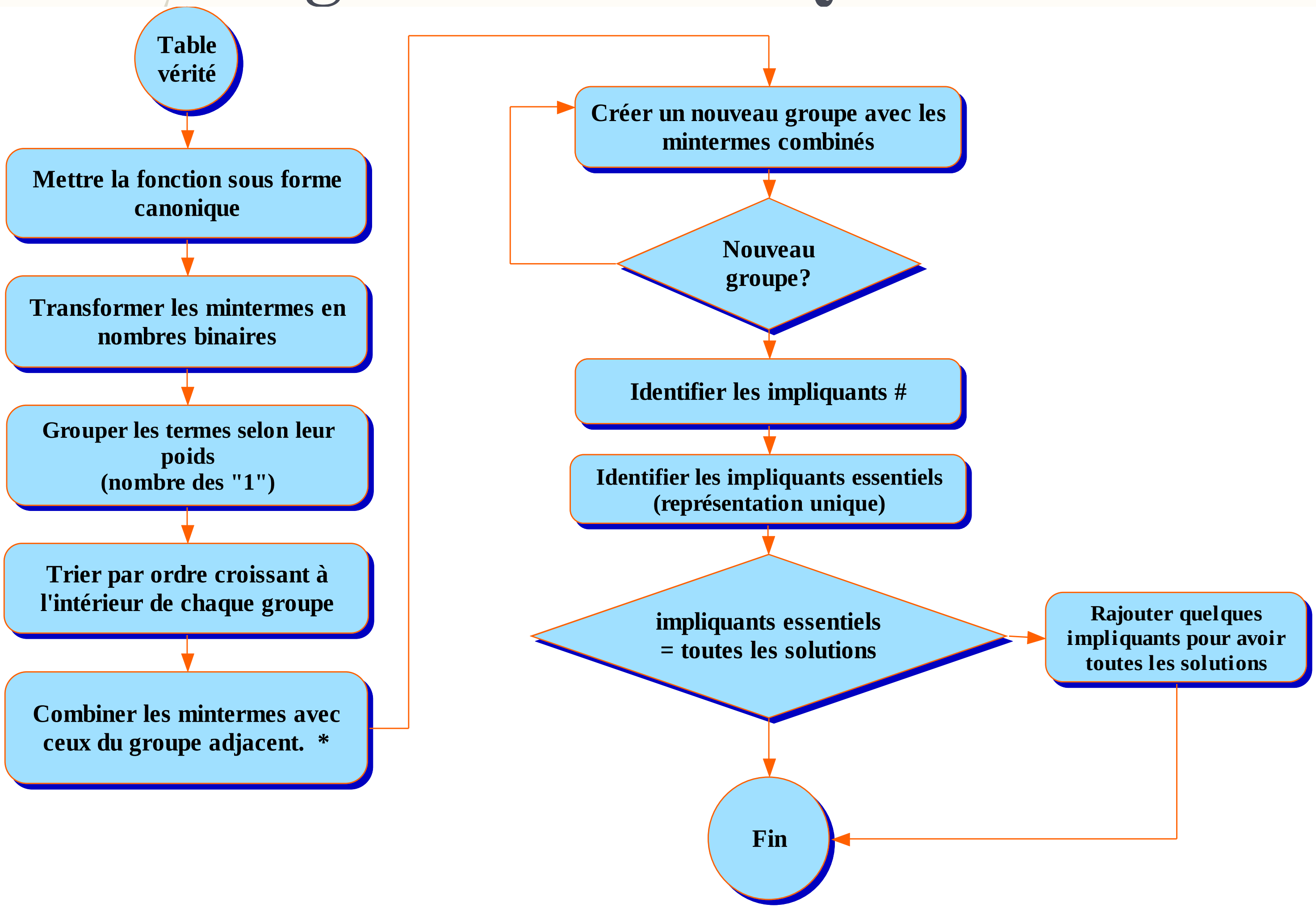
où x est un littéral et y un monôme.

Intérêts :

- Implémentation machine possible
- Nombre quelconque de variables
- Efficace, le résultat de la simplification est minimal

Peut nécessiter un temps exponentiel pour certains circuits !

Algorithme de Quine-McCluskey



* Deux mintermes se combinent s'ils diffèrent par un seul bit. Le minterme combiné contient un "x" à la place du bit différent

éléments non utilisés pour générer un élément du nouveau groupe

Procédure de simplification
Algorithme de Quine-McCluskey

Exemple – Quine (1)

Soit à simplifier

$$f(a,b,c,d) = ab + bc + a\bar{c} + \bar{a}cd + \bar{a}\bar{b}\bar{d} + \bar{a}\bar{b}c$$

$$f(a,b,c,d) = R(0,2,3,6,7,8,9,12,13,14,15)$$

$$f = abcd + abc\bar{d} + ab\bar{c}d + ab\bar{c}\bar{d} + a\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d} + \bar{a}bcd + \bar{a}bc\bar{d} + \bar{a}\bar{b}cd + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}\bar{c}\bar{d}$$

$$= 1111 + 1110 + 1101 + 1100 + 1001 + 1000 + 0111 + 0110 + 0011 + 0010 + 0000$$

Exemple – Quine (2)

classes	Etape 0	flag
0	0000	1
1	0010 1000	
2	0011 0110 1001 1100	
3	0111 1101 1110	
4	1111	

classes	Etape 0	flag	Etape1	flag
0	0000	1	00x0 x000	
1	0010 1000	1 1	001x 0x10 100x 1x00	
2	0011 0110 1001 1100	1 1 1 1	0x11 011x x110 1x01 110x 11x0	
3	0111 1101 1110	1 1 1	x111 11x1 111x	
4	1111	1		

Exemple – Quine (3)

Impliquants premiers

classes	Etape 0	flag	Etape 1	flag	Etape 2	flag
0	0000	1	00x0 x000	0 0		
1	0010 1000	1 1	001x 0x10 100x 1x00	1 1 1 1	0x1x 1x0x	0 0
2	0011 0110 1001 1100	1 1 1 1	0x11 011x x110 1x01 110x 11x0	1 1 1 1 1 1	x11x 11xx	0 0
3	0111 1101 1110	1 1 1	x111 11x1 111x	1 1 1		
4	1111	1				

Exemple – Quine (4)

* indique que l’impliquant couvre le terme
[*] l’impliquant est essentiel à ce terme
(*) terme couvert par un impliquant essentiel

	0000	0010	1000	0011	0110	1001	1100	0111	1101	1110	1111
00x0	*	*									
x000	*		*								
0x1x		(*)		[*]	(*)			(*)			
1x0x			(*)			[*]	*		(*)		
x11x					*			*		*	*
11xx							*		*	*	*

$$\boxed{\bar{a}c + a\bar{c}} \quad + \quad \begin{cases} ab + \bar{a}\bar{b}\bar{d} \\ ab + \bar{b}\bar{c}\bar{d} \\ bc + \bar{a}\bar{b}\bar{d} \\ bc + \bar{b}\bar{c}\bar{d} \end{cases}$$

F I N

Chapitre 4:

Simplification des fonctions