Algèbre de Boole

Eric Cariou

Université de Pau et des Pays de l'Adour
UFR Sciences Pau - Département Informatique

Eric.Cariou@univ-pau.fr

Algèbre de Boole

- Système algébrique constitué de l'ensemble { 0, 1 }
  - Variable booléenne : prend une valeur 0 (faux) ou 1 (vrai)
- Origine
  - Mathématicien anglais Georges Boole, 1815 – 1864
- Trois opérateurs de base
  - NON / NOT, noté $\overline{a}$
    - Inverse/complémente la valeur de la variable a
    - Vrai devient faux et faux devient vrai
  - ET / AND, noté $a \, . \, b$ ou ab
    - Retourne 1 si a et b sont à 1, sinon retourne 0
    - Les deux variables doivent être vraies ensemble, sinon le ET est faux
  - OU / OR, noté $a + b$
    - Retourne 1 si a ou b est à 1, sinon retourne 0
    - Au moins une des deux variables est vraie, sinon le OU est faux

Propriétés de base

- Involution : $\overline{\overline{a}} = a$
- Idempotence : $a + a = a \qquad a \,.\, a = a$
- Complémentarité : $a \,.\, \overline{a} = 0 \qquad a + \overline{a} = 1$
- Éléments neutres : $a = a \,.\, 1 = 1 \,.\, a = a$
  $a + 0 = 0 + a = a$
- Absorbants : $a + 1 = 1 \qquad a.0 = 0$

Propriétés de base

- Associativité : $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
  $(a + b) + c = a + (b + c)$
- Distributivité : $a \cdot (b + c) = a \cdot b + a \cdot c$
  $a + (b \cdot c) = (a + b) \cdot (a + c)$
- Règles de De Morgan : $\overline{a + b} = \overline{a} \cdot \overline{b}$
  $\overline{a \cdot b} = \overline{a} + \overline{b}$
- Optimisations : $a + \overline{a}\, b = a + b$
  $a + b\, c = (a + b)(a + c)$

Fonction logique

- Fonction logique
  - Prend en entrée une ou plusieurs variables booléennes
  - Retourne une valeur booléenne fonction des variables d'entrée
- Définition d'une fonction logique : deux méthodes
  - Par une expression logique
    - Combinaison des variables de la fonction via les opérateurs de base de l’algèbre de Boole
    - Exemple : fonction `f` de trois variables `a`, `b` et `c`
      $$f(a,b,c)=ab+\overline{b}c+a\overline{c}$$
    - Peut avoir plusieurs expressions logiques pour une même fonction
  - Par sa table de vérité
    - Table unique qui définit la valeur de la fonction pour chaque combinaison de valeurs possibles en entrée

Tables de vérité

- Table de vérité pour une fonction à *p* variables
  - Pour chacune des combinaisons différentes de *p* valeurs, on précise la valeur retournée par la fonction
- Table de vérité des opérateurs de base

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

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

a	b	a.b
0	0	0
0	1	0
1	0	0
1	1	1

Fonction logique

- Équivalence/passage entre une expression logique et la table de vérité de la fonction
  - On peut toujours déterminer l'une à partir de l'autre
- Deux fonctions logiques sont identiques si
  - On peut montrer via les propriétés de l'algèbre de Boole que leurs expressions logiques sont identiques
  - Leurs tables de vérité sont identiques
- Note
  - Quand on parle de fonction logique, on parle souvent par abus de langage de la forme correspondant à une expression logique

Formes canoniques d'une fonction

- Pour une fonction logique à x variables
  - Un minterme : groupe des x variables (pouvant être complémentées) liées par des ET
  - Un maxterme : groupe des x variables (pouvant être complémentées) liées par des OU
- Forme canonique d'une fonction logique
  - Première forme : union (OU) de mintermes
  - Seconde forme : intersection (ET) de maxtermes
- Il n'y a qu'une seule expression d'une forme canonique de chaque type pour une fonction donnée

Exemples de formes canoniques

- Fonction à 3 variables *a*, *b* et *c*, exemples :
  - Mintermes : $a\,b\,c\,,\;a\,\overline{b}\,c\,,\;a\,\overline{b}\,\overline{c}\,,\;\overline{a}\,b\,\overline{c}\,,\;...$
  - Maxtermes :
    $a+b+c\,,\;a+\overline{b}+c\,,\;a+\overline{b}+\overline{c}\,,\;\overline{a}+b+\overline{c}\,,\;...$
  - Première forme canonique :
    $f(a,b,c)=a\,b\,c+a\,\overline{b}\,c+a\,\overline{b}\,\overline{c}+\overline{a}\,b\,\overline{c}$
  - Seconde forme canonique :
    $g(a,b,c)=(a+b+c).(a+\overline{b}+c).(a+\overline{b}+\overline{c}).(\overline{a}+b+\overline{c})$

Passage aux formes canoniques

- Partir de la fonction et la transformer pour faire apparaître des mintermes/maxtermes complets
- Pour la transformation
  - On s'appuie sur les propriétés de l'algèbre de Boole, et notamment l'invariant :
    - $x + \overline{x} = 1$
    - Il permettra de rajouter des variables manquantes dans des termes

Exemple de passage à la première forme canonique

- Soit $f(a,b,c)=ab+\bar{b}c+a\bar{c}$
- Premier minterme ab
  - Il manque la variable c
  - Transforme ab en $ab.(c+\bar{c})$ car $c+\bar{c}=1$
- Même chose pour les 2 autres mintermes
- D'où :

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

Exemple de passage à la seconde forme canonique

- Soit $f(a,b,c) = ab + \overline{b}c + a\overline{c}$
- On passe par l'involution $\overline{\overline{x}} = x$
- Après développement :
  $\overline{f(a,b,c)} = \overline{a}b + \overline{a}bc + \overline{a}\,\overline{c} + \overline{a}\,\overline{b}\,\overline{c}$
- Reste à transformer les mintermes à 2 variables : $\overline{a}b + \overline{a}\,\overline{c} = \overline{a}b(c + \overline{c}) + \overline{a}\,\overline{c}(b + \overline{b})$
- Au final $\overline{f(a,b,c)} = \overline{a}bc + \overline{a}\,\overline{b}\,\overline{c} + \overline{a}b\overline{c}$
- Et donc $f(a,b,c) = (a + \overline{b} + \overline{c})(a + b + c)(a + \overline{b} + c)$

Passage de la fonction logique à la table de vérité

- Pour chaque combinaison de valeurs possibles pour les variables, on détermine la valeur booléenne de *f(X)* (X = ensemble des variables)
- Exemple : $f(a,b,c) = ab + \bar{b}c + a\bar{c}$

a	b	c	$\bar{b}$	$\bar{c}$	ab	$\bar{b}$c	a$\bar{c}$	f(a,b,c)
0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	1	0	1
0	1	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	1	0	1
1	1	0	0	1	1	0	1	1
1	1	1	0	0	1	0	0	1

Passage de la table de vérité à la fonction logique

- A partir de la table de vérité : fonction sous première forme canonique
  - Pour chaque valeur de *f(X)* égale à 1
    - On définit un minterme de toutes les variables tel que
    - Si une variable $X_i = 1$ on note X_i, sinon on note $\overline{X}_i$
  - La première forme canonique de *f(X)* est le OU de ces mintermes

Passage de la table de vérité à la fonction logique

- A partir de la table de vérité : fonction sous seconde forme canonique
  - Pour chaque valeur de *f(X)* égale à 0
    - On définit un minterme de toutes les variables tel que
    - Si une variable $X_i = 1$ on note X_i, sinon on note $\overline{X_i}$
  - Le OU de ces mintermes = $\overline{f(X_i)}$
  - Après calcul de $\overline{\overline{f(X_i)}}$, on obtient la seconde forme canonique

Exemple de calcul de la fonction logique sous première forme

- A partir de la table de vérité de l'exemple précédent
  - f(a,b,c) = 1 quand :
    - a = 0, b = 0 et c = 1 d'où le minterme $\overline{a}\,\overline{b}\,c$
    - a = 1, b = 0 et c = 0 d'où le minterme $a\,\overline{b}\,\overline{c}$
    - a = 1, b = 0 et c = 1 d'où le minterme $a\,\overline{b}\,c$
    - a = 1, b = 1 et c = 0 d'où le minterme $a\,b\,\overline{c}$
    - a = 1, b = 1 et c = 1 d'où le minterme $a\,b\,c$
  - On fait le OU de ces mintermes
    - $f(a,b,c) = a\,b\,c + a\,b\,\overline{c} + a\,\overline{b}\,c + \overline{a}\,\overline{b}\,c + a\,\overline{b}\,\overline{c}$

Exemple de calcul de la fonction logique sous seconde forme

- A partir de la table de vérité de l'exemple précédent
  - f(a,b,c) = 0 quand :
    - a = 0, b = 0 et c = 0 d'où le minterme $\bar{a}\,\bar{b}\,\bar{c}$
    - a = 0, b = 1 et c = 0 d'où le minterme $\bar{a}\,b\,\bar{c}$
    - a = 0, b = 1 et c = 1 d'où le minterme $\bar{a}\,b\,c$
  - On fait le OU de ces mintermes
    - $\overline{f(a,b,c)} = \bar{a}\,\bar{b}\,\bar{c} + \bar{a}\,b\,\bar{c} + \bar{a}\,b\,c$
  - Au final :

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

Minimisation des fonctions logiques

- Les formes canoniques d'une fonction logique sont une définition correcte de la fonction, mais elles peuvent être simplifiées
  - Pour écrire la même fonction avec le moins de termes et le moins d'occurences des variables
  - Pour réaliser la fonction avec moins d'éléments électroniques (portes logiques)
- Deux méthodes pour simplifier l'écriture d'une fonction logique
  - Utiliser les propriétés de l'algèbre de Boole
  - Utiliser la méthode des tableaux de Karnaugh

Simplification via algèbre de Boole

- A partir des propriétés de l'algèbre de Boole, transformer la fonction pour la simplifier
- Principes généraux
  - Simplifier la fonction initiale à l'aide des propriétés de l'algèbre de Boole
    - Appliquer la propriété d'involution ($x = \overline{\overline{x}}$) à la fonction simplifiée est parfois intéressant, mais calculs longs...
  - Essayer de déduire d'autres simplifications après chaque simplification
  - Il est compliqué d'être sur dans certains cas qu'on a bien obtenu une forme la plus simplifiée possible

Exemple de simplification via algèbre de Boole

- Soit $f(a,b,c) = abc + ab\overline{c} + a\overline{b}c + \overline{a}\,\overline{b}c + a\overline{b}\,\overline{c}$
- En factorisant, on obtient :

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

$$= a + \overline{a}\,\overline{b}c$$

$$= a + \overline{b}c \qquad \text{(car } x+\overline{x}y=x+y\text{)}$$

  - On ne peut pas simplifier plus

Exemple de simplification via algèbre de Boole

- Autre exemple : $f(a,b,c)=\overline{(a+b)c}+\overline{b}c$
- On distribue et calcule le non :

  $f(a,b,c)=\overline{a}\,\overline{b}+\overline{c}+\overline{b}c$
- En utilisant l'involution :

  $\overline{f(a,b,c)}=bc$
- D'où : $f(a,b,c)=\overline{b}+\overline{c}$
  - On aurait pu aussi simplifier en remarquant que $\overline{c}+\overline{b}c=\overline{c}+\overline{b}$ (car $x+\overline{x}y=x+y$ et donc $\overline{x}+x\overline{y}=\overline{x}+\overline{y}$)

Simplification par la méthode des tableaux de Karnaugh

- Principes généraux
  - Représentation sous une forme particulière de la table de vérité d'une fonction logique
  - Détermination de blocs rectangulaires de taille 2^n (avec n=1, 2, 4, ...) de bits adjacents à 1
  - On déduit de ces blocs la fonction simplifiée associée à la table de vérité

Simplification par la méthode des tableaux de Karnaugh

- On représente un tableau à 2 dimensions
- Chaque dimension concerne une ou 2 variables
- Le passage d'une colonne à une colonne adjacente ou d'une ligne à une ligne adjacente modifie la valeur d'une seule variable
- Le tableau se referme sur lui-même : la colonne la plus à gauche est voisine de la colonne la plus à droite, idem pour les lignes du haut et du bas
  - Pour les 2 colonnes (2 lignes) extrêmes, là aussi, une seule variable doit changer de valeur entre ces 2 colonnes (lignes)
- Une case du tableau contient une valeur booléenne, déterminée à partir de la table de vérité et des valeurs des variables

Simplification par la méthode des tableaux de Karnaugh

- Regroupement en blocs rectangulaires des bits à 1 adjacents
  - Tous les bits à 1 du tableau doivent être englobés dans au moins un bloc (un bloc à une taille de 1, 2, 4, 8 ... bits)
  - On doit créer les blocs les plus gros possibles
  - Un bit à 1 peut appartenir à plusieurs blocs si cela permet de créer des blocs plus gros
- A chaque bloc correspond un terme formé comme suit
  - Si une une variable dans le bloc change de valeur (valeurs 0 et 1 pour des cases différentes), on ne la prend pas en compte
  - On ne conserve que les variables qui ne varient pas. Si une variable a reste à 1 : on note $\mathtt{a}$, si reste à 0 : on note $\overline{\mathtt{a}}$
  - Le terme logique du bloc correspond au ET de ces variables qui ne changent pas
- La fonction logique simplifiée est le OU de tous les termes des blocs trouvés

Exemple de tableau de Karnaugh

- Table pour 2 variables

```
                                  \ a
 a  b | f(a,b)              b \  0 | 1 |
------+--------                  +----+----+
 0  0 |   0                  0 | 0 | 1 |
 0  1 |   1               ---+----+----+
 1  0 |   1                  1 | 1 | 1 |
 1  1 |   1               ---+----+----+
```

- 2 groupes de 2 bits adjacents :
  - Pour le vertical : on a toujours `a = 1` donc cela donne le terme a
  - Pour l'horizontal : idem mais avec `b`
- `f(a,b) = a + b`

Exemple de tableau de Karnaugh

- Table pour 3 variables

```
 a b c | g
-------+---
 0 0 0 | 0           \ ab
 0 0 1 | 1          c \  00 | 01 | 11 | 10 |
 0 1 0 | 0              +----+----+----+----+
 0 1 1 | 0           0 | 0  | 0  | 1  | 1  |
 1 0 0 | 1          ---+----+----+----+----+
 1 0 1 | 1           1 | 1  | 0  | 1  | 1  |
 1 1 0 | 1          ---+----+----+----+----+
 1 1 1 | 1
```

- Bloc le plus petit
  - `a` passe de 0 à 1, on ne la prendra pas en compte
  - `b` reste à 0 et `c` reste à 1
  - Donne le terme $\bar{b}\,c$

Exemple de tableau de Karnaugh

- Bloc le plus gros : a reste à 1, b passe de 0 à 1 et c passe de 0 à 1
- On ne conserve que les variables qui ne changent pas, on a donc le terme a
- Au final : $g(a, b, c) = a + \bar{b}c$
- Pourquoi pour le bloc de 4 on obtient juste a ?
  - Si on fait le OU de tous les mintermes pour lequel la valeur est 1, cela donne pour ce bloc de 4 :

$$\begin{aligned} bloc &= abc + ab\bar{c} + a\bar{b}c + a\bar{b}\bar{c} \\ &= a(b(c+\bar{c}) + \bar{b}(c+\bar{c})) = a \end{aligned}$$

  - Les variables d'un bloc prenant les valeurs de 0 et 1 sont donc systématiquement non significatives car la simplification par factorisation les fait disparaître

Exemple de tableau de Karnaugh

- Tableau pour 4 variables

```
   \ ab
cd \  00 | 01 | 11 | 10 |
      +----+----+----+----+
 00 | 1  | 0  | 0  | 1  |
 ---+----+----+----+----+
 01 | 1  | 1  | 1  | 1  |
 ---+----+----+----+----+
 11 | 1  | 1  | 1  | 1  |
 ---+----+----+----+----+
 10 | 0  | 1  | 0  | 0  |
 ---+----+----+----+----+
```

3 blocs :

- 8 cases : d
- 4 cases : $\bar{b}\,\bar{c}$
- 2 cases : $\bar{a}bc$

Au final :

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

- On doit là aussi regrouper en les plus gros blocs possibles même si on recoupe d'autres blocs