

# TD - NSI

## Variables booléennes



### Objectif

Dresser la table d'une expression booléenne.

## 1 Variables booléennes

### Définition 1

Un **booléen** est une variable informatique qui ne peut prendre que deux valeurs : **True** ou **False**.  
Un booléen est représenté en machine par un bit, qui vaut :

- 1 pour la valeur **True**;
- 0 pour la valeur **False**.

### Définition 2

Il existe trois opérations élémentaires sur les booléens : **disjonction**, **conjonction** et **négation**.

- **OU** : La **disjonction de A et B** est notée :  $A \text{ or } B$ ,  $A + B$ ,  $A \vee B$ .  
 $A \text{ or } B$  vaut **True** si et seulement si A vaut **True** ou B vaut **True**.

A	B	$A \text{ or } B = A + B$
0	0	
0	1	
1	0	
1	1	

- **ET** : La **conjonction de A et B** est notée :  $A \text{ and } B$ ,  $A \cdot B$ ,  $A \wedge B$ .  
 $A \text{ and } B$  vaut **True** si et seulement si A vaut **True** et B vaut **True**.

A	B	$A \text{ and } B = A \cdot B$
0	0	
0	1	
1	0	
1	1	

- **La négation de A** est notée  $\text{not } A$  ou encore  $\overline{A}$ .  
 $\overline{A}$  vaut **True** si et seulement si A vaut **False**.

A	$\text{not } A = \overline{A}$
0	
1	

**Définition 3** (Expression booléenne)

Une **expression booléenne** est une combinaison d'opérations élémentaires (**or**, **and**, **not**) portant sur une ou plusieurs variables booléennes.

**Exercice 1.1**

Compléter les égalités suivantes.

$$A \text{ or } A = \dots \quad A \text{ and } A = \dots \quad \text{not}(\text{not } A) = \dots \quad A \text{ or } (\text{not } A) = \dots \quad A \text{ and } (\text{not } A) = \dots$$

**Exercice 1.2****Formules de Morgan**

1. Compléter les tables de vérité ci-dessous.

$A$	$B$	$A \text{ or } B$	$\overline{A \text{ or } B}$	$A$	$B$	$\overline{A}$	$\overline{B}$	$\overline{A \text{ and } B}$
0	0			0	0			
0	1			0	1			
1	0			1	0			
1	1			1	1			

2. En déduire que  $\overline{A \text{ or } B} = \overline{A} \text{ and } \overline{B}$ .  
 3. Montrer de même que  $\overline{A \text{ and } B} = \overline{A} \text{ or } \overline{B}$ .

**Exercice 1.3**

1. Dresser la table de vérité de l'expression  $S = (A \text{ or } B) \text{ and } (\overline{A} \text{ or } \overline{B})$ .

$A$	$B$	$A \text{ or } B$	$\overline{A}$	$(\overline{A} \text{ or } \overline{B})$	$S$
0	0				
0	1				
1	0				
1	1				

2. Quelle égalité booléenne peut en déduire?

**Exercice 1.4**

Dresser la table de vérité de l'expression  $S = (A \text{ and } B) \text{ or } (A \text{ and not } C) \text{ or } (\text{not } B \text{ and } C)$ .

A	B	C	A and B	A and not C	not B and C	S
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

**Exercice 1.5**

On donne ci-dessous les tables de vérité de différentes expressions booléennes  $U$ ,  $V$  et  $W$ .

Retrouver les expressions de  $U$ ,  $V$  et  $W$  en fonction de  $A$  et  $B$ .

A	B	U
0	0	0
0	1	0
1	0	1
1	1	0

A	B	V
0	0	1
0	1	0
1	0	0
1	1	0

A	B	W
0	0	1
0	1	0
1	0	1
1	1	0

## 2 Opérations bit à bit

**Bit à bit**

On peut généraliser les opérations logiques **and** et **or** à des chaînes de bits ce que python peut faire facilement. Sous python :

- le **and** se note &
- le **or** se note |

— **Bit à bit : and se note &**

	1	0	1	1	0	1	1
&	1	0	1	0	1	0	1
	1	0	1	0	0	0	1

```
# Dans la console PYTHON
>>> bin(0b1011011&0b1010101)
'0b1010001'
```

— **Bit à bit : or se note |**

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

```
# Dans la console PYTHON
>>> bin(0b1011011|0b1010101)
'0b1011111'
```

**Exercice 2.6**

Calculer à la main puis avec python :

- 1111001 & 1110100
- 1111001 | 1110100

### 3 Obtention des formes normales

#### Propriété 1 (Non exigible)

1. **Forme normale conjonctive (FNC)** : Une fonction booléenne peut toujours s'exprimer sous forme d'une conjonction (**and**) de formes disjonctives (**or**) . Par exemple :

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

2. **Forme normale disjonctive (FND)** : Une fonction booléenne peut toujours s'exprimer sous forme d'une disjonction (**or**) de formes conjonctives (**and**) . Par exemple :

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



#### Méthode

Il existe cependant un moyen d'obtenir les FND ou FNC facilement en travaillant sur la table de vérité. On s'intéresse aux différents cas où la fonction retourne 0 ou 1.

$a$	$b$	$c$	$f(a, b, c)$
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

#### — Obtenir la FND

Pour la FND, on s'intéresse aux cas pour lesquels la fonction vaut 1 et on écrit la conjonction des littéraux (les *midterms*) correspondante :

1.  $a = b = 1$  et  $c = 0$ .

Ce premier cas donne la conjonction  $a.b.\bar{c}$ , et la table de vérité nous dit que celle-ci donne 1 lorsque  $a = 1$ ,  $b = 1$  et  $c = 0$ , or  $1.1.\bar{0} = 1$ .

2. On a aussi  $a = 1$  et  $b = 0$ , soit  $c = 1$  et on a la conjonction  $a.\bar{b}.c$ ;

3. On a aussi  $a = 1$  et  $b = 0$ , soit  $c = 0$  et on a la conjonction  $a.\bar{b}.\bar{c}$

Pour avoir la (FND) on prend la disjonction des conjonctions :

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

#### — Obtenir la FNC

On s'intéresse aux cas où la fonction vaut 0 (qu'on appelle des *maxterms*) , et de la même manière, on note les différentes conjonctions correspondantes.

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

Si on en prend la conjonction, la propriété assure que  $\overline{\overline{f(a, b, c)}} = f(a, b, c)$  :

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

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



### Exercice 3.7

L'opération « ou exclusif », noté **xor**, est défini par  $A \text{ xor } B = (A \text{ or } B) \text{ and } \overline{A \text{ and } B}$ .

1. Dresser la table de vérité de l'expression  $A \text{ xor } B$ .
2. En déduire la forme normale disjonctive de  $A \text{ xor } B$ .

$A$	$B$	$A \text{ or } B$	$A \text{ and } B$	$\overline{A \text{ and } B}$	$A \text{ xor } B = (A \text{ or } B) \text{ and } \overline{A \text{ and } B}$
0	0	1			
0	1	0			
1	0	1			
1	1	0			



### Exercice 3.8

On donne ci-dessous les tables de vérité de différentes expressions booléennes  $U$ ,  $V$  et  $W$ . Retrouver les expressions de  $U$ ,  $V$  et  $W$  en fonction de  $A$  et  $B$  et  $C$ .

$A$	$B$	$C$	$U$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$A$	$B$	$C$	$V$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$A$	$B$	$C$	$W$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



### Exercice 3.9

#### Question 1

Si  $A$  et  $B$  sont des variables booléennes, quelle est l'expression booléenne équivalente à  $(\text{not } A) \text{ or } B$ ?

- $(A \text{ and } B) \text{ or } (\text{not } A \text{ and } B) \text{ or } (\text{not } A \text{ and not } B)$
- $(A \text{ and } B) \text{ or } (\text{not } A \text{ and } B)$
- $(\text{not } A \text{ and } B) \text{ or } (\text{not } A \text{ and not } B)$
- $(A \text{ and } B) \text{ or } (\text{not } A \text{ and not } B)$