

Logique multivaluée  
Systèmes logiques et physique : calculs combinatoire (QLC)

Noémie PERROT  
Mathieu HA-SUM

16 juin 2016

# Table des matières

<b>1</b>	<b>Algèbre de Boole et logique multivaluée</b>	<b>2</b>
1.1	Définition usuelle (binaire) . . . . .	2
1.1.1	Historique . . . . .	2
1.1.2	Axiomes et postulats . . . . .	2
1.2	Extension à la logique multivaluée . . . . .	3
1.3	Opérateurs multivalués . . . . .	3
1.4	Représentations et décompositions . . . . .	4
1.4.1	Les tables de vérité . . . . .	4
1.4.2	Décomposition de Shanon . . . . .	4
1.4.3	Somme de produits (SOP) . . . . .	4
1.4.4	Produit de sommes (POS) . . . . .	4
1.4.5	Reed-Muller-Fourier Transform . . . . .	4
1.4.6	Forme Canonique . . . . .	4
1.4.7	Forme Polynomiale . . . . .	4
<b>2</b>	<b>Eigenlogic : logique multivaluée</b>	<b>5</b>
2.1	Logique trivaluée . . . . .	5
2.1.1	Logique trivaluée $\{0, 1, 2\}$ . . . . .	5
2.1.2	Logique trivaluée $\{-1, 0, 1\}$ . . . . .	8
2.1.3	Lien avec la logique binaire . . . . .	8
2.1.4	Somme de polynômes interpolateurs de Lagrange . . . . .	9
2.2	Logique multivaluée . . . . .	10
2.2.1	Fonctions <i>Min</i> et <i>Max</i> multivaluées : généralisation . . . . .	10
2.2.2	Un résultat intéressant . . . . .	13
2.2.3	Peut-on se ramener d'une dimension $n$ à une dimension $m < n$ ? . . . . .	14
<b>3</b>	<b>Applications</b>	<b>16</b>
3.1	Circuits électroniques . . . . .	16
3.1.1	VHDL . . . . .	16
3.1.2	Additionneur sans retenue . . . . .	16
3.1.3	Mémoires . . . . .	17
3.2	Ordinateurs quantiques . . . . .	17
3.2.1	Notations et postulats . . . . .	18
3.2.2	Bits quantiques . . . . .	18
3.2.3	Les opérations en quantique . . . . .	18
3.2.4	Exemple du SWAP . . . . .	19

# Chapitre 1

## Algèbre de Boole et logique multivaluée

### 1.1 Définition usuelle (binaire)

#### 1.1.1 Historique

George Boole (1815-1864), mathématicien autodidacte anglais, a développé une algèbre permettant de manipuler les propositions logiques au moyen d'équations mathématiques où les énoncés *VRAI* et *FAUX* sont représentés par les valeurs **1** et **0**, tandis que les opérateurs *ET* et *OU* deviennent des opérateurs algébriques de multiplication et d'addition.

L'algèbre de Boole est ainsi usuellement définie comme un système logique algébrique fondé à partir de l'ensemble  $\{0, 1\}$ . En particulier, ces deux valeurs suffisent à représenter les états des circuits électriques à 2 états (courant passant et non passant).

#### 1.1.2 Axiomes et postulats

Une algèbre de Boole est constituée :

- d'un ensemble  $E$ ,
- de deux éléments particuliers de  $E$  : **0** et **1** (correspondant respectivement à *FAUX* et *VRAI*),
- de deux opérations binaires sur  $E$  :  $+$  et  $\cdot$  (correspondant respectivement au *OU* et *ET* logiques),
- d'une opération unaire sur  $E$  :  $\neg$  (correspondant à la négation logique).

On acceptera les postulats suivant :

- |                                |                        |
|--------------------------------|------------------------|
| 1. $0 \cdot 0 = 0$             | 5. $1 + 1 = 1$         |
| 2. $0 \cdot 1 = 1 \cdot 0 = 0$ | 6. $1 + 0 = 0 + 1 = 1$ |
| 3. $1 \cdot 1 = 1$             | 7. $0 + 0 = 0$         |
| 4. $\bar{0} = 1$               | 8. $\bar{1} = 0$       |

Ces postulats correspondent, à toute fin utile, aux définitions des opérations logiques sur les éléments  $\mathbf{0}$  et  $\mathbf{1}$  de  $E$ . On pourra noter que dans le cas des opérations binaires  $\cdot$  et  $+$ , ces postulats correspondent aux résultats de l'arithmétique usuelle, sauf pour  $1 + 1 = 1$ .

De ces postulats découlent les axiomes suivants. Soient  $a, b, c$  des éléments de  $E$  :

<i>Commutativité</i>	$a + b = b + a$	$a \cdot b = b \cdot a$
<i>Associativité</i>	$(a + b) + c = a + (b + c)$	$(a \cdot b) \cdot c = a \cdot (b \cdot c)$
<i>Distributivité</i>	$a \cdot (b + c) = a \cdot b + a \cdot c$	$a + (b \cdot c) = (a + b) \cdot (a + c)$
<i>Élément neutre</i>	$a + 0 = a$	$a \cdot 1 = a$
<i>Complémentation</i>	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$

## 1.2 Extension à la logique multivaluée

Les concepts de la logique peuvent néanmoins s'étendre à des systèmes composés de plus de deux valeurs. Ainsi, dans toute notre étude, nous utilisons une définition plus large de l'algèbre de Boole.

**Définition.** Une algèbre de Boole est un sextuple  $\langle B, +, \cdot, \neg, 0, 1 \rangle$  où  $B$  est un ensemble d'éléments contenant 0 et 1,  $+$  et  $\cdot$  étant des opérateurs binaires et  $\neg$  un opérateur unaire. L'opérateur  $\cdot$  est prioritaire devant  $+$  et  $\neg$  prioritaire devant les deux autres. Le  $\cdot$  est souvent sous-entendu. Une algèbre de Boole suit les axiomes suivants :

- Si  $a \in B$ , alors  $\bar{a} \in B$
- Si  $a \in B$  et  $b \in B$ , alors  $a + b \in B$  et  $ab \in B$
- Si  $a \in B$  et  $b \in B$ , alors  $a + b = b + a$  et  $ab = ba$
- Si  $a \in B$ ,  $b \in B$ , et  $c \in B$ , alors  $a(b+c) = ab+ac$  et  $a+bc = (a+b)(a+c)$
- Si  $a \in B$ , alors  $a1 = a$  et  $a + 0 = a$
- Si  $a \in B$ , alors  $a\bar{a} = 0$  et  $a + \bar{a} = 1$
- $0 \neq 1$

## 1.3 Opérateurs multivalués

Les opérateurs qui nous intéressent le plus, a priori, sont ceux qui permettent de représenter de manière efficace le comportement de certains circuits électriques.

Dans le cas de deux variables, il y a  $p^{p^2}$  fonction binaires  $p$ -valuées et  $p^p$  fonctions unaires  $p$ -valuées : il y a donc un nombre de possibilités considérable pour le choix des opérateurs algébriques.

Par exemple, on pourrait prendre, en prolongement de la logique binaire usuelle, les opérateurs suivants (figure 1.1). Les tables de vérité sont un moyen de représenter ces opérateurs.

	$P = 2$	$P = 3$	$P = 4$
Min	$\begin{array}{c cc} \cdot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$	$\begin{array}{c ccc} \cdot & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 2 \end{array}$	$\begin{array}{c cccc} \cdot & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 2 & 2 \\ 3 & 0 & 1 & 2 & 3 \end{array}$
Max	$\begin{array}{c cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array}$	$\begin{array}{c ccc} + & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 \end{array}$	$\begin{array}{c cccc} + & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 2 & 3 \\ 2 & 2 & 2 & 2 & 3 \\ 3 & 3 & 3 & 3 & 3 \end{array}$
Mod-sum	$\begin{array}{c cc} \oplus & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array}$	$\begin{array}{c ccc} \oplus & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 2 & 0 \\ 2 & 2 & 0 & 1 \end{array}$	$\begin{array}{c cccc} \oplus & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 1 & 1 & 2 & 3 & 0 \\ 2 & 2 & 3 & 0 & 1 \\ 3 & 3 & 0 & 1 & 2 \end{array}$
Mod-difference	$\begin{array}{c cc} \ominus & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array}$	$\begin{array}{c ccc} \ominus & 0 & 1 & 2 \\ \hline 0 & 0 & 2 & 1 \\ 1 & 1 & 0 & 2 \\ 2 & 2 & 1 & 0 \end{array}$	$\begin{array}{c cccc} \ominus & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 3 & 2 & 1 \\ 1 & 1 & 0 & 3 & 2 \\ 2 & 2 & 1 & 0 & 3 \\ 3 & 3 & 2 & 1 & 0 \end{array}$
Truncated sum	$\begin{array}{c cc} +_r & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array}$	$\begin{array}{c ccc} +_r & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{array}$	$\begin{array}{c cccc} +_r & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 1 & 1 & 2 & 3 & 3 \\ 2 & 2 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 \end{array}$

FIGURE 1.1 – Opérateurs *usuels* pour  $p=2,3,4$  représentés à l'aide de tables de vérité

## 1.4 Représentations et décompositions

### 1.4.1 Les tables de vérité

### 1.4.2 Décomposition de Shanon

### 1.4.3 Somme de produits (SOP)

### 1.4.4 Produit de sommes (POS)

### 1.4.5 Reed-Muller-Fourier Transform

### 1.4.6 Forme Canonique

### 1.4.7 Forme Polynomiale

## Chapitre 2

# Eigenlogic : logique multivaluée

Nous allons nous intéresser à un système logique particulier, qui constituera notre cadre d'étude pour la suite : la **Eigenlogic**.

### 2.1 Logique trivaluée

#### 2.1.1 Logique trivaluée $\{0, 1, 2\}$

On se place dans le cadre de la physique quantique. On suppose que les états accessibles puissent prendre trois valeurs entières, mettons  $\{0, 1, 2\}$ . On écrit l'observable du moment cinétique comme :

$$L = \hbar\Gamma = \hbar \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

On va décomposer tous les états sur la base des vecteurs d'espace, calquée sur les polynômes interpolateurs de Lagrange (qui sont toujours une base dans l'espace considéré).

Les vecteurs d'espace seront :

$$\Pi_2 = \frac{1}{2}\Gamma(\Gamma - 1) \quad \Pi_1 = \Gamma(2 - \Gamma) \quad \Pi_0 = \frac{1}{2}(\Gamma - 2)(\Gamma - 1)$$

Et la décomposition de toute fonction  $F$  dans cette base s'écrira :

$$F(U, V) = \sum_{i,j=0}^2 f_{ij}(U, V) \Pi_i(U) \otimes \Pi_j(V)$$

Il existe  $n^{n^2}$  fonctions  $f_{ij}(U, V)$ . On va s'intéresser aux fonctions *Max* et *Min*, dont on dresse les tables de vérité :

<i>Max</i>	2	1	0
2	2	2	2
1	2	1	1
0	2	1	0

<i>Min</i>	2	1	0
2	2	1	0
1	1	1	0
0	0	0	0

On a les fonctions d'interpolations suivantes :

$$f_0(x) = \frac{1}{2}(x-1)(x-2)$$

$$f_1(x) = \frac{1}{2}x(2-x)$$

$$f_2(x) = \frac{1}{2}x(x-1)$$

On utilise dans les calculs  $f_0 + f_1 + f_2 = 1$  ainsi que  $2f_2(x) = x - f_1(x)$ .

Le développement donne pour  $Max$  :

$$\begin{aligned}
Max(U, V) &= 2 (f_2(U) \otimes f_2(V) + f_2(U) \otimes f_1(V) \\
&\quad + f_2(U) \otimes f_0(V) + f_1(U) \otimes f_2(V) + f_0(U) \otimes f_2(V)) \\
&\quad + f_1(U) \otimes f_1(V) + f_1(U) \otimes f_0(V) + f_0(U) \otimes f_1(V) \\
&= 2 (f_2(U) \otimes \mathbb{1}(V) + \mathbb{1}(U) \otimes f_2(V) - f_2(U) \otimes f_2(V)) \\
&\quad + \mathbb{1}(U) \otimes f_1(V) - f_2(U) \otimes f_1(V) + f_1(U) \otimes f_0(V) \\
&= 2 \left( \frac{1}{2}U(U-1) + \frac{1}{2}V(V-1) - \frac{1}{4}UV(U-1)(V-1) \right) \\
&\quad + V(2-V) - \frac{1}{2}U(U-1)V(2-V) + \frac{1}{2}U(2-U)(V-1)(V-2) \\
&= U^2 - U + V^2 - V - \frac{1}{2}UV(UV - U - V + 1) + 2V - V^2 \\
&\quad - \frac{1}{2}UV(2U - 2V + V) + \frac{1}{2}(2U - U^2)(V^2 - 3V + 2) \\
&= U^2 - U + V - \frac{1}{2}UV(U-1) + \frac{1}{2}(2-U)(V^2 - 3V + 2) \\
&= U^2 - U + V + V + \frac{1}{2}U(-5V + 2UV + 2V^2 - UV^2 - 2U + 4) \\
&= -U + V - \frac{5}{2}UV + U^2V + V^2U - \frac{1}{2}U^2V^2 + 2U \\
&= U + V + U^2V + V^2U - \frac{1}{2}U^2V^2 - \frac{5}{2}UV
\end{aligned}$$

De même :

$$\begin{aligned}
Min(U, V) &= 2 f_2(U)f_2(V) + f_2(U)f_1(V) + f_1(U)f_2(V) + f_1(U)f_1(V) \\
&= Uf_2(V) - f_1(U)f_2(V) + f_1(U)f_2(V) + f_1(U)f_1(V) + f_2(U)f_1(V) \\
&= \frac{1}{2}(UV - f_1(U)f_1(V)) + f_1(U)f_1(V) \\
&= \frac{1}{2}(UV + UV(U-2)(V-2)) \\
&= \frac{1}{2}UV(1 - 2U - 2V + UV + 4) \\
&= \frac{5}{2}UV + \frac{1}{2}U^2V^2 - U^2V - V^2U
\end{aligned}$$

### 2.1.2 Logique trivaluée $\{-1, 0, 1\}$

On prend comme second exemple une logique trivaluée prenant les valeurs  $\{-1, 0, 1\}$ .

De la même manière, on obtient :

$$Max(U, V) = \frac{1}{2}(U^2 + V^2 + U + V - U^2V^2 - UV)$$

$$Min(U, V) = \frac{1}{2}(-U^2 - V^2 + U + V + U^2V^2 + UV)$$

Comme nous le verrons plus loin, cette logique présente plusieurs avantages : elle s'approche des deux logiques les plus communément utilisées en binaire  $\{-1, 1\}$  et  $\{0, 1\}$ , qui ne sont en fait que des cas particuliers de cette logique ; de plus, ses valeurs sont centrées en zéro, assurant ainsi a priori une simplification des résultats et des propriétés intéressantes de symétrie.

### 2.1.3 Lien avec la logique binaire

L'étude de systèmes trivalués permet dans les deux cas précédents de retrouver certains résultats de logique binaire ; de manière plus générale, l'établissement des fonctions sous forme de polynômes peut être grandement simplifié dans cette approche par quelques propriétés vérifiées par les fonctions utilisées.

Retrouvons l'expression du  $OU$  et du  $ET$  de la logique binaire grâce à nos fonctions.

**Dans la logique  $\{0, 1\}$**

Pour cette logique, on a la propriété particulière suivante  $U^2 = U$ . D'où, en supprimant la valeur  $\{-1\}$  dans l'exemple 2 :

$$\begin{aligned} Min(U, V) &= \frac{1}{2}(-U^2 - V^2 + U + V + U^2V^2 + UV) \\ &= \frac{1}{2}(U + V - U - V + UV + UV) \\ &= UV \end{aligned}$$

$$\begin{aligned} Max(U, V) &= \frac{1}{2}(U^2 + V^2 + U + V - U^2V^2 - UV) \\ &= \frac{1}{2}(U + V + U + V - UV - UV) \\ &= U + V - UV \end{aligned}$$

Il est aisé de montrer qu'en supprimant la valeur 2 dans l'exemple 1, on obtient le même résultat. On retrouve bien les résultats classiques.

### Dans la logique $\{-1, 1\}$

Pour cette logique, on a  $U^2 = 1$ . Dans l'exemple 2, on enlève la valeur  $\{0\}$ , et on obtient :

$$\begin{aligned} \text{Min}(U, V) &= \frac{1}{2}(-U^2 - V^2 + U + V + U^2V^2 + UV) \\ &= \frac{1}{2}(-1 - 1 + U + V + 1 + UV) \\ &= \frac{1}{2}(-1 + U + V + UV) \end{aligned}$$

$$\begin{aligned} \text{Max}(U, V) &= \frac{1}{2}(U^2 + V^2 + U + V - U^2V^2 - UV) \\ &= \frac{1}{2}(U + V + 1 + 1 - UV - 1) \\ &= \frac{1}{2}(U + V + 1 - UV) \end{aligned}$$

On vérifie la cohérence du résultat avec les tableaux de vérité de  $ET$  et  $OU$  :

U	V	ET	OU
-1	-1	-1	-1
-1	1	-1	1
1	-1	-1	1
1	1	1	1

On assimilera donc dans la suite la fonction  $Max$  à l'opération  $OU$  et la fonction  $Min$  à l'opération  $ET$ .

### 2.1.4 Somme de polynômes interpolateurs de Lagrange

On peut remarquer que dans les deux exemples précédents :  $f_{-1} + f_0 + f_1 = 1$  et  $f_0 + f_1 + f_2 = 1$ . Cette propriété est-elle généralisable ?

Le calcul nous montre que oui :

Soit  $x_1, x_2, x_3$  trois réels distincts.

$$\begin{aligned} (f_{x_1} + f_{x_2} + f_{x_3})(x) &= \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} + \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} \\ &= \frac{P(x)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} \end{aligned}$$

$$P(X) = (X - x_2)(X - x_3)(x_2 - x_3) + (X - x_1)(X - x_3)(x_3 - x_1) + (X - x_2)(X - x_1)(x_1 - x_2)$$

Il est facile de voir que :

$$\begin{aligned} P'(X) &= 2X(x_2 - x_3) - (x_2 + x_3)(x_2 - x_3) + 2X(x_3 - x_1) - (x_1 + x_3)(x_3 - x_1) \\ &\quad + 2X(x_1 - x_2) - (x_2 + x_1)(x_1 - x_2) \\ &= 2\alpha X - \beta \end{aligned}$$

avec :  $\alpha = x_2 - x_3 + x_3 - x_1 + x_1 - x_2 = 0$  et  $\beta = x_2^2 - x_3^2 + x_3^2 - x_1^2 + x_1^2 - x_2^2 = 0$   
 Donc  $P'(X) = 0$  et  $P(X)$  constant. On l'évalue en  $X = x_1$  :

$$P(x_1) = (x_1 - x_2)(x_1 - x_3)(x_2 - x_3)$$

D'où finalement :

$$(f_{x_1} + f_{x_2} + f_{x_3})(x) = 1$$

Cette expression permet la simplification des résultats. On pourrait également trouver des propriétés certes un peu moins évidentes mais toujours utiles avec les polynômes d'interpolation de Lagrange d'ordre supérieur pour des calculs à la main. Par exemple en 4 valeurs :  $(f_{-1} + f_0 + f_1 + f_2)(x) = 1 - x^2$ .

Fort heureusement, des logiciels de calcul formel, tel Mathematica, existent et vont nous permettre d'obtenir des résultats sans calculs fastidieux...

## 2.2 Logique multivaluée

### 2.2.1 Fonctions *Min* et *Max* multivaluées : généralisation

On décide d'écrire un script Mathematica qui permet d'obtenir les fonctions  $Min(U, V)$  et  $Max(U, V)$  en ne donnant en entrée que les valeurs propres voulues, quels que soient leur nombre et leur valeur.

Le code est donné ci-dessous (pour  $\{0, 1, 2\}$ ) :

```
liste = {0, 1, 2};

(*On va créer :
- un tableau de dimension n*2
  'fonceval' où on met les fonctions d'interpolations évaluées en u et en v
- une liste de dimension n
  'fonctions' où on met juste les fonctions d'interpolations*)

fonctions = {};
fonceval = {};

Do[
  AppendTo[fonceval, {0, 0}],
  {k, Length[liste]}]
Do [
  f[x_] := 1;
  Do[
    If[i == j, Continue[]];
    f[x_] = f[x] * ((liste[[j]] - x) / (liste[[j]] - liste[[i]])),
    {j, Length[liste]};
  fonceval[[i]][[1]] = Expand[f[u]];
  fonceval[[i]][[2]] = Expand[f[v]];
  AppendTo[fonctions, Expand[f[x_]]], {i, 1, Length[liste]}]
```

Si on veut les afficher, les fonctions d'interpolations sont accessibles dans "*fonctions*".

```

|(*On va créer :
- un tableau de dimension n2
  'tableaumax' qui donne le max de deux valeurs
- un tableau de dimension n2
  'tableaumin' qui donne le min de deux valeurs
- une liste de dimension 2
  'tableAeval' qui stocke les tableaux à évaluer (tableaumin, tableaumax...)
*)

tableAeval = {};
tableaumax = {};
tableaumin = {};

Do[
  lignemax = {};
  Do[
    AppendTo[lignemax, Max[liste[[i]], liste[[j]]]],
    {j, Length[liste]};
  AppendTo[tableaumax, lignemax],
  {i, Length[liste]}]
AppendTo[tableAeval, tableaumax];

Do[
  lignemin = {};
  Do[
    AppendTo[lignemin, Min[liste[[i]], liste[[j]]]],
    {j, Length[liste]};
  AppendTo[tableaumin, lignemin],
  {i, Length[liste]}]
AppendTo[tableAeval, tableaumin];

```

A noter qu'on pourrait ici insérer n'importe quel tableau, et donc n'importe laquelle des  $n^{n^2}$  fonctions de la logique considérée. On a choisi de s'intéresser uniquement à *Max* et *Min* mais le programme permet d'étudier d'autres options... et donc éventuellement des logiques différentes de l'Eigenlogic !

```

(*On va créer:
- une liste de dimension 2
  'sortie' qui sort les fonctions max et min*)

sortie = {};
Do[
  eval = 0;
  Do[
    Do[
      eval = eval + tableAeval[[k]][[i]][[j]] * fonceval[[i]][[1]] * fonceval[[j]][[2]],
      {j, Length[tableaumax]],
      {i, Length[tableaumax]};
    Expand[eval];
    AppendTo[sortie, Expand[eval]],
    {k, Length[tableAeval]}]
  sortie

```

Les résultats donnés par ce programme pour quelques logiques multivaluées sont les suivants :

(Debug) Out[225]=

{0, 1, 2}

(Debug) Out[239]=

$$\left\{ u + v - \frac{5uv}{2} + u^2v + uv^2 - \frac{u^2v^2}{2}, \frac{5uv}{2} - u^2v - uv^2 + \frac{u^2v^2}{2} \right\}$$

(Debug) Out[242]=

u + v

(Debug) Out[243]=

{-1, 0, 1}

(Debug) Out[257]=

$$\left\{ \frac{u}{2} + \frac{u^2}{2} + \frac{v}{2} - \frac{uv}{2} + \frac{v^2}{2} - \frac{u^2v^2}{2}, \frac{u}{2} - \frac{u^2}{2} + \frac{v}{2} + \frac{uv}{2} - \frac{v^2}{2} + \frac{u^2v^2}{2} \right\}$$

(Debug) Out[260]=

u + v

Mais très vite, les fonctions deviennent peu aisément exploitables :

(Debug) Out[261]=

{-2, -1, 0, 1, 2}

(Debug) Out[275]=

$$\left\{ \frac{u}{2}, \frac{7u^2}{12}, \frac{u^4}{12}, \frac{v}{2}, \frac{25uv}{36}, \frac{u^2v}{9}, \frac{7v^2}{12}, \frac{113u^2v^2}{144}, \frac{23u^4v^2}{144}, \frac{uv^4}{9}, \frac{u^2v^4}{36}, \frac{v^4}{12}, \frac{23u^2v^4}{144}, \frac{5u^4v^4}{144}, \frac{u}{2}, \frac{7u^2}{12}, \frac{u^4}{12}, \frac{v}{2}, \frac{25uv}{36}, \frac{u^2v}{9}, \frac{7v^2}{12}, \frac{113u^2v^2}{144}, \frac{23u^4v^2}{144}, \frac{uv^4}{9}, \frac{u^2v^4}{36}, \frac{v^4}{12}, \frac{23u^2v^4}{144}, \frac{5u^4v^4}{144} \right\}$$

(Debug) Out[278]=

u - v

Voire complètement tordues :

```
(Debug) Out[276]:
{-7, -4, 0, 1, 2, 3, 8, 12}

(Debug) Out[281]:
8297 u 81283 u2 2217 u3 218359 u4 1979 u5 4183 u6 229 u7 8297 v 3291089 u v 14496682583 u2 v 49740353 u3 v 10396973101 u4 v 498591341 u5 v
8550 1436400 2462400 34473600 11491200 34473600 34473600 8550 877800 6636168000 541728000 53089344000 53089344000
10565011 u2 v 418069 u3 v 81283 u4 14496682583 u5 v 132898976549 u6 v 12775123663 u7 v 284572072331 u8 v 208884587 u9 v 393122473 u10 v 63424507 u11 v
3539289600 1966272000 1436400 6636168000 79434016000 955608192000 1911216384000 25482851200 173746944000 382243276800
82217 u2 49740353 u3 12775123663 u4 503160179 u5 7934825309 u6 161171669 u7 40796359 u8 59111 u9 218359 u10 10396973101 u11
2462400 541728000 955608192000 604735336000 7644865536000 1528973107200 254828512000 138997555200 34473600 53089344000
284572072331 u2 v 7934825309 u3 v 204585043301 u4 v 11337116599 u5 v 444735163 u6 v 228678977 u7 v 1979 v 498591341 u8 v 208884587 u9 v 161171669 u10 v
1911216384000 7644865536000 15289731072000 15289731072000 2184247296000 15289731072000 11491200 53089344000 2548285120 1528973107200
11337116599 u2 v 8375417 u3 v 172527431 u4 v 167603 u5 v 4183 v 10565011 u6 v 393122473 u7 v 40796359 u8 v 444735163 u9 v 172527431 u10 v
15289731072000 185762112000 15289731072000 188567936000 34473600 3539289600 173746944000 254828512000 2184247296000 15289731072000
3159131 u2 v 3481033 u3 v 229 v 418069 u4 v 63424507 u5 v 59111 u6 v 228678977 u7 v 167603 u8 v 3481033 u9 v 256793 u10 v
1019315404800 15289731072000 34473600 1966272000 382243276800 138997555200 15289731072000 198567936000 1528973107200 15289731072000
253 u 81283 u2 82217 u3 218359 u4 1979 u5 4183 u6 229 u7 253 v 14496682583 u8 v 49740353 u9 v 10396973101 u10 v 498591341 u11 v
8550 1436400 2462400 34473600 11491200 34473600 34473600 8550 877800 6636168000 541728000 53089344000 53089344000
10565011 u2 v 418069 u3 v 81283 u4 14496682583 u5 v 132898976549 u6 v 12775123663 u7 v 284572072331 u8 v 208884587 u9 v 393122473 u10 v 63424507 u11 v
3539289600 1966272000 1436400 6636168000 79434016000 955608192000 1911216384000 25482851200 173746944000 382243276800
82217 u2 49740353 u3 12775123663 u4 503160179 u5 7934825309 u6 161171669 u7 40796359 u8 59111 u9 218359 u10 10396973101 u11
2462400 541728000 955608192000 604735336000 7644865536000 1528973107200 254828512000 138997555200 34473600 53089344000
284572072331 u2 v 7934825309 u3 v 204585043301 u4 v 11337116599 u5 v 444735163 u6 v 228678977 u7 v 1979 v 498591341 u8 v 208884587 u9 v 161171669 u10 v
1911216384000 7644865536000 15289731072000 15289731072000 2184247296000 15289731072000 11491200 53089344000 2548285120 1528973107200
11337116599 u2 v 8375417 u3 v 172527431 u4 v 167603 u5 v 4183 v 10565011 u6 v 393122473 u7 v 40796359 u8 v 444735163 u9 v 172527431 u10 v
15289731072000 185762112000 15289731072000 188567936000 34473600 3539289600 173746944000 254828512000 2184247296000 15289731072000
3159131 u2 v 3481033 u3 v 229 v 418069 u4 v 63424507 u5 v 59111 u6 v 228678977 u7 v 167603 u8 v 3481033 u9 v 256793 u10 v
1019315404800 15289731072000 34473600 1966272000 382243276800 138997555200 15289731072000 198567936000 1528973107200 15289731072000
(Debug) Out[289]:
u + v
```

On a donc accès aux fonctions souhaitées. Les formes développées offrent l'avantage d'explicitier les symétries et de pouvoir vérifier simplement la compatibilité du résultat avec les modèles simplifiés connus (vérifier qu'on peut retrouver les résultats de logique binaire par exemple).

### 2.2.2 Un résultat intéressant

On s'intéresse ici à la somme des fonctions *Min* et *Max*.

Le tableau correspondant à cette fonction est symétrique. On note que ce n'est pas une table de vérité car les valeurs prises par la fonction ne sont pas nécessairement contenues dans l'ensemble des valeurs propres.

U/V	0	1	2
0	0	1	2
1	1	2	3
2	2	3	4

La symétrie de ce tableau entraîne un résultat bien particulier. On écrit un script pour en faire la somme :

(\* On étudie une propriété des fonctions max et min : leur somme\*)

```
sortie2 = 0;
Do[
Do[
Do[
sortie2 += tableAeval[[k]][[i]][[j]] * fonceval[[i]][[1]] * fonceval[[j]][[2]],
{j, Length[tableaumax}},
{i, Length[tableaumax}},
{k, Length[tableAeval]]}
Expand[sortie2]
```

En effectuant la somme de  $Min$  et  $Max$  grâce à ce programme, on obtient :

```
(Debug) Out[225]=
      {0, 1, 2}
```

```
(Debug) Out[239]=
      {u + v - \frac{5 u v}{2} + u^2 v + u v^2 - \frac{u^2 v^2}{2}, \frac{5 u v}{2} - u^2 v - u v^2 + \frac{u^2 v^2}{2}}
```

```
(Debug) Out[242]=
      u + v
```

Ce résultat est général (même si nous l'avons seulement testé empiriquement). Il est d'ailleurs très intuitif : la somme du minimum et du maximum de deux nombres est la somme de ces deux nombres. On peut ainsi se simplifier la tâche dans le calcul de  $Min$  et  $Max$  en observant qu'obtenir une des deux fonctions revient à obtenir l'autre par décalage de  $U + V$ .

### 2.2.3 Peut-on se ramener d'une dimension $n$ à une dimension $m < n$ ?

**Le cas  $\{-1, 0, 1\}$**

On essaye de se ramener aux expressions de  $Min$  et  $Max$  dans la logique  $\{-1, 0, 1\}$  à partir de celles déterminées dans la logique  $\{-1, 0, 1, 2\}$  :

$Min$  :

$$\left| \frac{2u}{3} + \frac{u^2}{2} - \frac{u^3}{6} + \frac{2v}{3} - \frac{5uv}{6} - \frac{u^2v}{4} + \frac{u^3v}{4} + \frac{v^2}{2} - \frac{uv^2}{4} - \frac{u^2v^2}{2} + \frac{u^3v^2}{4} - \frac{v^3}{6} + \frac{uv^3}{4} + \frac{u^2v^3}{4} - \frac{u^3v^3}{6} \right|$$

$Max$  :

$$\left| \frac{u}{3} - \frac{u^2}{2} + \frac{u^3}{6} + \frac{v}{3} + \frac{5uv}{6} + \frac{u^2v}{4} - \frac{u^3v}{4} - \frac{v^2}{2} + \frac{uv^2}{4} + \frac{u^2v^2}{2} - \frac{u^3v^2}{4} + \frac{v^3}{6} - \frac{uv^3}{4} - \frac{u^2v^3}{4} + \frac{u^3v^3}{6} \right|$$

La relation vérifiée dans la logique  $\{-1, 0, 1\}$  est :

$$x(x-1)(x+1) = 0 \quad \text{soit} \quad x^3 = x$$

On remplace dans les expressions de *Min* et *Max* déterminées par Mathematica :

```
(Debug) In[24]:=
fMAX[u3_, v3_] :=  $\frac{2u}{3} + \frac{u^2}{2} - \frac{u^3}{6} + \frac{2v}{3} - \frac{5uv}{6} - \frac{u^2v}{4} + \frac{u^3v}{4} + \frac{v^2}{2} - \frac{uv^2}{4} - \frac{u^2v^2}{2} + \frac{u^3v^2}{4} - \frac{v^3}{6} + \frac{uv^3}{4} + \frac{u^2v^3}{4} - \frac{u^3v^3}{6}$ 
fMIN[u3_, v3_] :=  $\frac{u}{3} - \frac{u^2}{2} + \frac{u^3}{6} + \frac{v}{3} + \frac{5uv}{6} + \frac{u^2v}{4} - \frac{u^3v}{4} - \frac{v^2}{2} + \frac{uv^2}{4} + \frac{u^2v^2}{2} - \frac{u^3v^2}{4} + \frac{v^3}{6} - \frac{uv^3}{4} - \frac{u^2v^3}{4} + \frac{u^3v^3}{6}$ 
relation = Expand[u*(u-1)*(u+1)]
resMAX = fMAX[Expand[u], Expand[v]]
resMIN = fMIN[Expand[u], Expand[v]]

(Debug) Out[26]=
-u + u^3

(Debug) Out[27]=
 $\frac{u}{2} + \frac{u^2}{2} + \frac{v}{2} - \frac{uv}{2} + \frac{v^2}{2} - \frac{u^2v^2}{2}$ 

(Debug) Out[28]=
 $\frac{u}{2} - \frac{u^2}{2} + \frac{v}{2} + \frac{uv}{2} - \frac{v^2}{2} + \frac{u^2v^2}{2}$ 
```

On obtient bien les expressions attendues.

### Généralisation

On peut en fait toujours se ramener d'une dimension à n'importe quelle autre, avec une méthodologie très simple :

- Soit  $\{x_1, x_2, \dots, x_n\}$  les n valeurs propres de la logique considérée. On a les fonctions *Min* et *Max* pour cette logique. On souhaite obtenir ces fonctions pour une logique ramenée à m de ces n valeurs :  $\{y_1, y_2, \dots, y_m\}$
- On écrit le polynôme minimal annulateur de ces valeurs :

$$P_{\{y_1, y_2, \dots, y_m\}}(X) = \prod_{k=1}^m (X - y_k)$$

- $P_{y_1, y_2, \dots, y_m}$  est unitaire de degré m.
- $P_{y_1, y_2, \dots, y_m}(y_m) = 0$  (annulateur pour toutes les valeurs considérées). Ce polynôme nous donne donc une expression de  $X^m$  en fonction des valeurs des  $X^k, k < m$ .
- On remplace dans  $Min_{logiqueN}(U, V)$  et  $Max_{logiqueN}(U, V)$  les  $U^k$  et  $V^k$  de degré trop élevé jusqu'à obtenir une fonction dont les arguments ne sont pas élevés à une puissance supérieure à m.
- On a alors les fonctions souhaitées, dans la nouvelle logique.

L'efficacité de cette méthodologie peut être montrée aisément par récurrence sur le nombre de valeurs propres.

# Chapitre 3

## Applications

Cela fait près de 50 ans que la logique multivaluée est un domaine de recherche qui intéresse les mathématiciens et les ingénieurs en circuits électroniques et informatiques. Plus récemment, la logique p-valuée s'est révélée cruciale dans l'étude des "ordinateurs quantiques". Afin de conforter l'intérêt des travaux formels sur la logique, nous avons reconstruit plusieurs applications qui illustrent l'apport du multivaleur aux connaissances actuelles.

### 3.1 Circuits électroniques

Dans le cadre des circuits électroniques, la logique multivaluée ne sert pas seulement à représenter des systèmes possédant plus de deux états logiques : en réalité, les méthodes de conception basées sur la logique multivaluée permettent également de réaliser un circuit final à l'aide de composants binaires CMOS (*complementary metal oxide semiconductor*). L'intérêt réside dans le fait que ces méthodes permettent d'obtenir de meilleures caractéristiques.

#### 3.1.1 VHDL

La langage VHDL (*VHSIC Hardware Description Language*) est un des plus utilisés pour la description des circuits électroniques (caractéristiques, simulation, synthèse, validation...). Il a été établi par l'IEEE (*Institute of electrical and Electronics Engineers*) en 1987. Le langage possédait alors un type de données pour la représentation à l'aide de 0, 1. Une des difficultés majeure que posait ce langage pour la simulation des circuits était l'existence d'états supplémentaires tels que "non-initialisé" ou bien "haute-impédance" qui n'étaient pas facilement représentables par l'outil. En 1993, une nouvelle norme adoptait 9 valeurs pour la logique multivaluée (voir figure 4.1).

#### 3.1.2 Additionneur sans retenue

L'additionneur est un des éléments fondamentaux en électronique. On peut, avec la logique multivaluée, essayer d'améliorer les performances de celui-ci en supprimant les délais parfois *aléatoires* causés par la propagation de la retenue, et ainsi se limiter à des coûts de calculs *fixes et définis*.

SYMBOL	NAME	MEANING
U	Uninitialized	Initial default value for literal
X	Forcing Unknown	Forced unresolved logic value
0	Forcing 0	Forced logic value 0
1	Forcing 1	Forced logic value 1
Z	High Impedance	Logic value of open circuit
W	Weak Unknown	Weak unresolved logic value
L	Weak 0	Weak logic value 0
H	Weak 1	Weak logic value 1
-	Don't Care	Used for synthesis

FIGURE 3.1 – Les différents états de la norme VHDL IEEE 1164

Par exemple, en utilisant le système  $\bar{1}, 0, 1$ , et une notation de longueur 2, on obtient un système dit *redondant*, où la valeur +1 peut être représentée de deux façons différentes. En effet :

$$+1 = (0) \times 2^1 + (1) \times 2^0 = (1) \times 2^1 + (-1) \times 2^0$$

Cette redondance peut être utilisée afin de se passer de la retenue. En effet, il suffit de remarquer qu'une retenue est non nulle lorsque 1 apparaît à la même position dans les deux membres de l'addition, ou bien si le  $\bar{1}$  répond à cette condition. Ainsi, de notre point de vue, en considérant qu'il s'agit d'une représentation de plusieurs valeurs logiques, cet exemple illustre l'utilisation de la logique multivaluée.

### 3.1.3 Mémoires

Une autre application réelle de la logique multivaluée est la mémoire flash (non-volatile) : Intel réalise un modèle de mémoire où chaque cellule mémoire est construite sur un transistor à grille flottante dont celle-ci peut être chargée sur plusieurs niveaux. On obtient la possibilité de stocker deux bits par cellule. Du point de vue de la logique, c'est une logique à quatre états : "11" correspond à l'état neutre, tandis que "01", "10", et "00" correspondent à différentes tensions.

## 3.2 Ordinateurs quantiques

On ne peut pas simplement dire qu'un ordinateur quantique est un ordinateur qui repose sur les lois de la mécanique quantique, puisque tout phénomène physique est gouverné par les lois de la physique quantique. Là où l'ordinateur quantique se distingue, c'est dans ses rapports à l'information logique, stockée sous forme de *bits* dans un ordinateur classique.

### 3.2.1 Notations et postulats

Nous utiliserons dans la suite, sans les rappeler, les notations usuelles de la physique quantique et de l'algèbre linéaire (notations "brackets", dirac...).

### 3.2.2 Bits quantiques

Dans l'informatique "classique", l'information est stockée sous formes de bits à deux états. On représente ces deux états par  $|0\rangle$  et  $|1\rangle$ . Une paire de bits peut ainsi prendre les valeurs suivantes

$$|01\rangle, |10\rangle, |00\rangle, |11\rangle$$

qui n'est d'autre qu'une notation équivalente à

$$|0\rangle|1\rangle, |1\rangle|0\rangle, |0\rangle|0\rangle, |1\rangle|1\rangle$$

On peut également les décomposer dans l'espace vectoriel à deux dimensions :

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Un bit quantique, en revanche, n'est pas limité à deux états. Un état  $|\psi\rangle$  associé peut s'assimiler à n'importe quel vecteur bidimensionnel de l'espace engendré par  $|0\rangle$  et  $|1\rangle$ . On note :

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

où  $\alpha_0$  et  $\alpha_1$  sont deux nombres complexes tels que  $\|\alpha_0\|^2 + \|\alpha_1\|^2 = 1$ .

Cet état est une superposition des états  $|0\rangle$  et  $|1\rangle$  d'amplitudes  $\alpha_0$  et  $\alpha_1$ . Si l'on a deux bits quantiques, l'un dans l'état  $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$  et l'autre  $|\phi\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$  alors l'état  $\Psi$  de la paire est donné par

$$\Psi = |\psi\rangle \otimes |\phi\rangle = \begin{pmatrix} \alpha_0 \beta_0 \\ \alpha_0 \beta_1 \\ \alpha_1 \beta_0 \\ \alpha_1 \beta_1 \end{pmatrix}$$

### 3.2.3 Les opérations en quantique

Nous allons voir comment effectuer des opérations sur les bits. En particulier, les ordinateurs quantiques effectuent des opérations *réversibles*, en transformant un bit quantique d'un état vers un autre à travers des opérations qui peuvent être inversées. La mesure de la valeur d'un bit quantique est une action **irréversible** : cette difficulté d'accès à l'information stockée dans le bit n'est pas présente en informatique classique mais joue un rôle fondamental dans l'informatique quantique.

Pour comprendre les opérations, on regarde souvent l'action sur les bits classiques. Les actions réversibles sur  $n$  bits classiques sont les permutations des  $2^n$  états. On peut étendre des opérations sur les bits quantiques, ce qui correspond



FIGURE 3.2 – Diagramme d'un circuit représentant l'action sur un bit quantique. L'état initial est  $|\psi\rangle$ , la ligne représente l'histoire de l'état, qui passe à  $U|\psi\rangle$

à permuter les amplitudes. Une grande partie des opérateurs sur les bits quantiques peut être définie de cette façon. En particulier, les transformations *NOT*, *SWAP*, qu'on définit sur les bits classiques, peuvent s'appliquer aux superpositions d'états  $\sum \alpha_x |x\rangle_n$ .

### 3.2.4 Exemple du SWAP

Un opérateur non trivial réversible unitaire est le NOT, représenté par le symbole  $X$  :

$$X : |x\rangle \rightarrow |\tilde{x}\rangle$$

Sa représentation matricielle est :

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

On introduit le projecteur sur l'état  $|1\rangle$ , noté  $n$  :

$$n|x\rangle = x|x\rangle, x = 0 \text{ ou } 1$$

On définit aussi l'opérateur complémentaire :

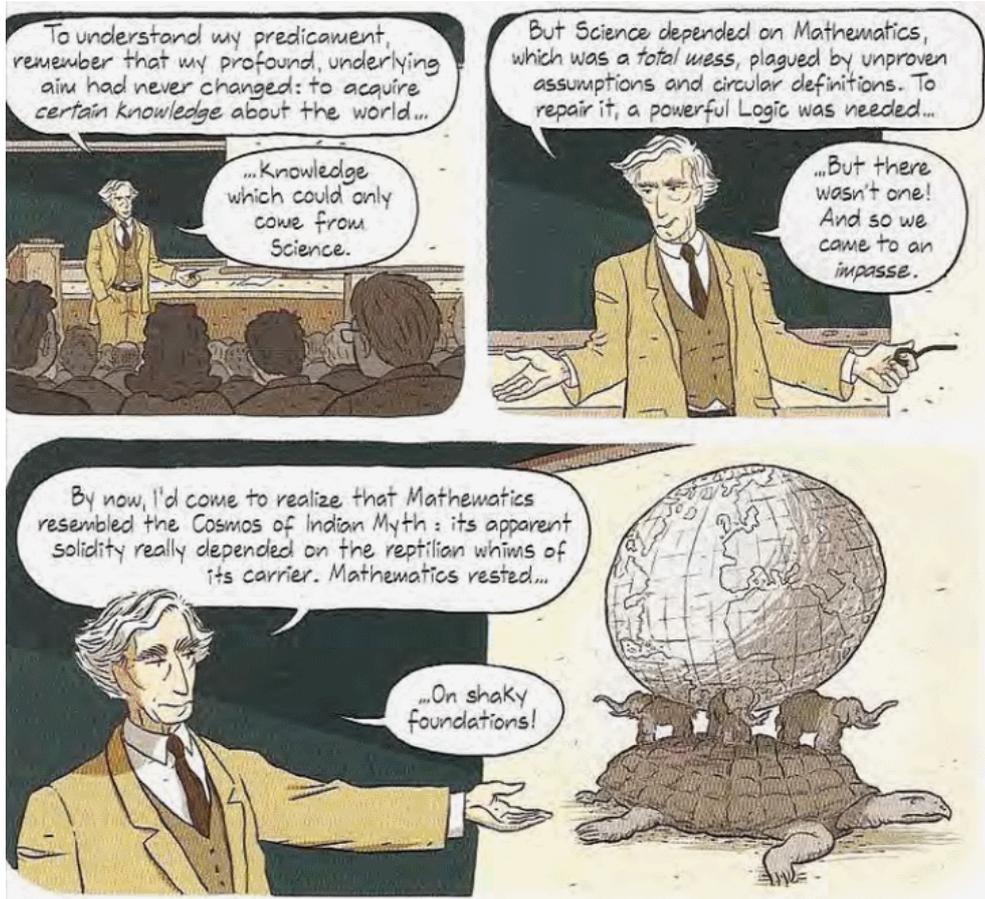
$$\tilde{n} = 1 - |n\rangle$$

Leurs représentations matricielles sont :

$$n = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad \tilde{n} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Maintenant, on peut par exemple définir un opérateur intéressant, et présent en quantique, l'opérateur *SWAP*,  $S_{ij}$ , qui "classiquement" regarde si les états de  $i$  et  $j$  sont différents et les inverse le cas échéant.

$$S_{ij} = n_i n_j + \tilde{n}_i \tilde{n}_j + (X_i X_j)(n_i \tilde{n}_j + \tilde{n}_i n_j)$$



# Bibliographie

- [1] François Dubois and Zeno Toffano. *Eigenlogic : a Quantum View for Multiple-Valued and Fuzzy Systems*. 2016.
- [2] Donald E. Knuth. *The Art of computer programming*. 2005.
- [3] N. David Mermin. *Quantum Computer Science*. 2007.
- [4] D. Michael Miller and Mitchell A. Thornton. *Multiple Valued Logic : Concepts and Representations*. 2008.
- [5] Emil S. Post. *Introduction to a General Theory of Elementary Propositions*. 1921.
- [6] David C. Rine. *Computer Science and Multiple-valued logic theory and Applications*. 1977.
- [7] Vlad P. Shmerko, Svetlana N. Yanushkevich, and Sergey Edward Lyshevski. *Computer Arithmetics for Nanoelectronics*. 2009.
- [8] Radomir S. Stankovic, Jaakko T. Astola, and Claudio Moraga. *Representation of Multiple-Valued Logic Functions*. 2012.
- [9] Zeno Toffano. *Eigenlogic in the spirit of George Boole*. 2015.