Ternary Logic: Some basic identities

Jorge Pedraza Arpasi [*]

9th October 2002

Contents

1 Introduction

Based on the work of Ivan Guzmán de Rojas [1] and some elementary notions of Boolean algebra we will expose and show some basic ternary logic identities. Our purpose for that is

---

[*] The author is maintainer of the website Aymara Uta Internet: www.aymara.org.

to give some parallel Boolean structure for this logic with three values. A Boolean algebra structure $\mathcal{A}$ is, basically, a non empty set with two binary operations a third unary operation satisfying a set of rules. From these conditions follows that the cardinality, the number of elements of $\mathcal{A}$, nust be a power of two, simbolically that means $|\mathcal{A}| = 2^k$ for some natural number $k = 1, 2, \ldots,$. On the other hand any logic structure based on a three valued logic must have a cardinality of power of three, that is $|\mathcal{T}| = 3^k$, where T is a set obeying a set of three valued logic. Therefore a logic sistem based on a three valued logic would not be a Boolean algebra.

In accordance with the **excluded middle law**, fundamental for the binary logic, each proposition only can have one of the two values: or **true (1)** or **false (0)**, never either. The ternary logic do not take into account this excluded middle law, we can say that it is included. Then the ternary logic have three values **true (1)**, **false (2)** and **perhaps true, perhaps false (0)**.

The notation $\mathbb{Z}_2 = \{0, 1\}$ for the binary case, and $\mathbb{Z}_3 = \{0, 1, 2\}$ for the ternary case, is the most adequate for a later algebraic analysis. As we will see in the true tables of the conjunction, disjunction, implication and the equivalence the ternary logic is a generalization of the binary one, in the sense that if one proposition x is true (1) under the binary logic then it is true (1) under the ternary logic, analogously for the false case, x false (0) under the binary analysis implies it is false under the ternary case (2). The advantage of the the ternary logic is that there are many statements for which is not possible to decide if they are true or false, then they can be analyzed by the rules of the ternary logic.

2 Boolean Algebra

Definition 1 *An Algebra of Boole is a non empty set $\mathcal{A}$ with two binary operations $+$, "the sum", and $\cdot$, "the product" and one unary operation $'$, "the complement", satisfying the following conditions;*

1. *The sum is commutative, that is, $x + y = y + x$, for all $x, y \in \mathcal{A}$.*

2. *The sum is associative, that is,* $(x + y) + z = x + (y + z)$*, for all* $x, y, z \in \mathcal{A}$*.*
3. *The sum is distributive with respect the product, that is,* $x + (y.z) = (x + y).(x + z)$*, for all* $x, y, z \in \mathcal{A}$*.*
4. *There exists an an neutral element for the sum* $0 \in \mathcal{A}$ *such that* $x + 0 = x$ *for all* $x \in \mathcal{A}$*.*
5. *The product is commutative, that is,* $x.y = y.x$*, for all* $x, y \in \mathcal{A}$*.*
6. *The product is associative, that is,* $(x.y).z = x.(y.z)$*, for all* $x, y, z \in \mathcal{A}$*.*
7. *The product is distributive with respect the sum, that is,* $x.(y + z) = (x.y) + (x.z)$*, for all* $x, y, z \in \mathcal{A}$*.*
8. *There exists an an neutral element fro the product* $1 \in \mathcal{A}$ *such that* $x.1 = x$ *for all* $x \in \mathcal{A}$*.*
9. $x + x' = 1$ *for all* $x \in \mathcal{A}$
10. $x.x' = 0$ *for all* $x \in \mathcal{A}$

2.1 Examples

Example 1 *Consider the class of the logical propositions* $\mathcal{P}$*, which satisfies the* ***excluded middle law****, and the logical functors the disjunction* $\vee$ *(or), the conjunction* $\wedge$ *(and) and the negation* $\sim$ *(not). Then* $\mathcal{P}$ *is an Algebra of Boole*

Example 2 *Consider the class of the sets* $\mathcal{C}$ *and the set operators: union of sets* $\cup$*, intersection of sets* $\cap$ *and the set complement* $'$*. Then* $\mathcal{C}$ *is an Algebra of Boole*

3 The ternary logic: elementary definitions

3.1 Unary operator: Negation

Let x be a logic proposition then the negation x' of this proposition is given by

x	x'
1	2
0	0
2	1

3.2 Binary operators

Let x, y be ternary logic propositions, that is, $x = \begin{cases} 1; & \text{if} \quad x \quad \text{is true} \\ 0; & \text{if} \quad x \quad \text{is perhaps true, perhaps false} \\ 2; & \text{if} \quad x \quad \text{is false} \end{cases}$

and

$$y = \begin{cases} 1; & \text{if} \quad x \quad \text{is true} \\ 0; & \text{if} \quad x \quad \text{is perhaps true, perhaps false} \\ 2; & \text{if} \quad x \quad \text{is false} \end{cases}$$

Then the binary functor associated with the ordered pair (x, y) is the application $f : \{1, 0, 2\} \times \{1, 0, 2\} \to \{1, 0, 2\}$ which maps in according to the basic rules given for the

- Conjunction $Conj(x, y)$
- Disjunction $Disj(x, y)$
- Implication $Imp(x, y)$
- Equivalence $Equiv(x, y)$

which are shown in the following table.

x	y	$Conj(x,y)$ $x \wedge y$	$Disj(x,y)$ $x \vee y$	$Imp(x,y)$ $x \Rightarrow y$	$Equiv(x,y)$ $x \Leftrightarrow y$
1	1	1	1	1	1
1	0	0	1	0	0
1	2	2	1	2	2
0	1	0	1	1	0
0	0	0	0	1	1
0	2	2	0	0	0
2	1	2	1	1	2
2	0	2	0	1	0
2	2	2	2	1	1

3.3 Examples

Example 3 *Let* $f(x,y) = (x \Rightarrow y) \wedge (y \Rightarrow x)$ *be a functor. Then* $f(x,y)$ *is equivalent to* $Equiv(x,y)$.

x	y	$x \Rightarrow y$	$y \Rightarrow x$	$f(x,y)$	$Equiv(x,y)$
1	1	1	1	1	1
1	0	0	1	0	0
1	2	2	1	2	2
0	1	1	0	0	0
0	0	1	1	1	1
0	2	0	1	0	0
2	1	1	2	2	2
2	0	1	0	0	0
2	2	1	1	1	1

Example 4 *Let* $f(x,y) = y' \Rightarrow x'$ *be a functor, then* $f(x,y)$ *is equivalent to* $Imp(x,y)$.

x	y	y'	x'	$f(x,y)$	$Imp(x,y)$
1	1	2	2	1	1
1	0	0	2	0	0
1	2	1	2	2	2
0	1	2	0	1	1
0	0	0	0	1	1
0	2	1	0	0	0
2	1	2	1	1	1
2	0	0	1	1	1
2	2	1	1	1	1

Example 5 *Let $f(x,y) = (x \vee y)'$ be a functor, then $f(x,y)$ is equivalent to $g(x,y) = x' \wedge y'$. (The first Morgan's law)*

x	y	$x \vee y$	$f(x,y)$	y'	x'	$g(x,y)$
1	1	1	2	2	2	2
1	0	1	2	2	0	2
1	2	1	2	2	1	2
0	1	1	2	0	2	2
0	0	0	0	0	0	0
0	2	0	0	0	1	0
2	1	1	2	1	2	2
2	0	0	0	1	0	0
2	2	2	1	1	1	1

Example 6 *Let $f(x,y) = (x \wedge y)'$ be a functor, then $f(x,y)$ is equivalent to $g(x,y) = x' \vee y'$. (The second Morgan's law)*

x	y	$x \vee y$	$f(x,y)$	y'	x'	$g(x,y)$
1	1	1	2	2	2	2
1	0	0	0	2	0	0
1	2	2	1	2	1	1
0	1	0	0	0	2	0
0	0	0	0	0	0	0
0	2	2	1	0	1	1
2	1	2	1	1	2	1
2	0	2	1	1	0	1
2	2	2	1	1	1	1

Example 7 *Let* $f(x,y) = x' \vee y$ *be a functor, then* $f(x,y)$ *is not equivalent to* $Imp(x,y) = x \Rightarrow y$ *as it happens in the binary logic.*

x	y	x'	$f(x,y)$	$Imp(x,y)$
1	1	2	1	1
1	0	2	0	0
1	2	2	2	2
0	1	0	1	1
0	0	0	0	1
0	2	0	0	0
2	1	1	1	1
2	0	1	1	1
2	2	1	1	1

3.4 Why a ternay logic can not be a Boolean Algebra

We can show that a ternary logic holds all the firts eight properties, of a Boolean Algebra *commutative, associative, distributive* and *identity* properties, for sum and product, but fails in the last two properties, which work with the unary complement operator, as we show now;

x	x'	$x+x'$
1	2	1
0	0	0
2	1	1

therefore $x+x' \neq 1$, and also

x	x'	$x.x'$
1	2	2
0	0	0
2	1	2

therefore $x.x' \neq 2$

4 Modal Analysis

Amodal	x	1	0	2
Possibility		1	1	2
Certitude		1	2	2

Negation	$N(x)$	2	0	1
Neg(Certitude)=Doubt		2	1	1
Neg(Possibility)		2	2	1

		1	0	1
Tautology	$T(x)$	1	1	1
Plausibility(+)		1	2	1

		2	0	2
Neg(Plausibility(+))		2	1	2
Contradiction	$C(x)$	2	2	2

Likelihood		1	0	0
Feasibility		1	1	0
Evident		1	2	0

Neg(Likelihood)		2	0	0
Neg(Evident)		2	1	0
Neg(Feasibility)=difficulty		2	2	0

		0	0	1
Plausibility(-)		0	1	1
		0	2	1

		0	0	2
		0	1	2
Neg(Plausibility(-))		0	2	2

Abduction		0	0	0
Contingency		0	1	0
Neg(contingency)		0	2	0

References

[1] Guzmán de Rojas, Ivan; *Logical and Linguistic Problems of Social Communication with Aymara People*; International Development Research Centre (IDRC), Ottawa, Canada, 1984.

[2] Gersting, Judith L. *Mathematical Structures for Computer Science*, Fourth Ed., W. H. Freeman, New York, 1998.