A Brief Introduction to Ternary Logic

Jorge Pedraza Arpasi *

7th November 2003

1 Introduction

Based on the work of Ivan Guzmán de Rojas [1] and some basic notions of binary logic and algebra of Boole, we will introduce and show some elementary properties of ternary logic. These properties can be resumed as follows: **a)** Ternary logic is a generalization of binary logic, **b)** it has not a structure to be a Boolean algebra, **c)** it is based on more than three basic operations, and **d)** its tautologies and contradictions are more complicated for find out. The generalization is in the sense that if one proposition p is true(false) under the rules of binary logic then it is true(false) under the ternary logic. The lack of Boolean structure, in ternary logic, is compensated by powerful tools for inferential analysis [1].

In this note, for the binary case, we will use the values $\{0, 1\}$ which means true=1 and false=0. Whereas that for ternary case we will use the values $\{0, 1, 2\}$ which means true=1, false=2, and "perhaps true perhaps false"=0. This notation is the most adequate for a later algebraic analysis. By x we will mean a simple statement or proposition whereas by p or f we will mean a composed proposition which depends on other simple propositions, hence we also refer they as functions.

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

2 Binary Logic(Classical Logic)

We can see the binary logic as a system $\mathcal{L}$ whose elements called **propositions** or **statements** are valued on the set $\{0, 1\}$ [2]. This set $\{0, 1\}$ we denote as being $\mathbb{Z}_2$. Thus, if x is a proposition, the value of x can be seen as a mapping $\nu : \mathcal{L} \to \{0, 1\}$ such that

$$\nu(x) = \begin{cases} 1; & \text{if} \quad x \quad \text{is true} \\ 0; & \text{if} \quad x \quad \text{is false} \end{cases}$$

It is standard, in almost all the logical literature, to ignore the mapping ν. Therefore, for the sack of practical purposes, it is assumed that $\nu(x) = x$ and from this, $x = 1$ means x is true and $x = 0$ means x is false. Over $\mathcal{L}$ are defined the following basic operations

- The negation $\neg$ (unary operation "not")
- The disjunction $\vee$ (binary operation "or")
- The conjunction $\wedge$ (binary operation "and")

The system $\mathcal{L}$ is closed under any of these three operations, in the sense that if $x, y \in \mathcal{L}$, then both $\neg x \in \mathcal{L}$, $x \vee y \in \mathcal{L}$, and $x \wedge y \in \mathcal{L}$. From these three operations we can derive 16 binary operations, among them the implication $Imp(x, y) = x \to y$ and the equivalence $Equiv(x, y) = x \leftrightarrow y$. The value of these basic operations and any other composed propositions depend on the value of each component x and y as can be seen in the **true table** for these operations shown in Table 1

Another way to describe propositions based on the above basic operations is by considering them as functions. In this way, the unary operator "negation" is a function $f : \mathbb{Z}_2 \to \mathbb{Z}_2$, and a binary operation such as the "disjunction" is a function $f : \mathbb{Z}_2^2 \to \mathbb{Z}_2$. In general, by combining the three basic operations we can define **binary logic functions** as mappings $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$.

When $n = 1$ we have one-variable functions $f(x)$, and there are $2^{2^1} = 4$ of these kind of functions which are: the Identity or Affirmation $id(x)$, the Negation $N(x)$, the Tautology

x	y	$Neg(x)$ $\neg x$	$Disj(x,y)$ $x \vee y$	$Conj(x,y)$ $x \wedge y$	$Imp(x,y)$ $x \to y$	$Equiv(x,y)$ $x \leftrightarrow y$
1	1	0	1	1	1	1
1	0		1	0	0	0
0	1	1	1	0	1	0
0	0		0	0	0	1

Table 1: Basic operations or functions in binary logic

$\tau(x)$ and the Contradiction $\gamma(x)$. All these four functions are also called **modal** functions of x and they are shown in the Table 2

x	$id(x)$	$N(x)$	$\tau(x)$	$\gamma(x)$
1	1	0	1	0
0	0	1	1	0

Table 2: All the one-variable binary functions

When $n = 2$ we have two-variable functions $f(x, y)$, and there are $2^{2^2} = 16$ of these kind of functions which are shown in the Table 3. Notice, in that Table, that $f_2(x, y)$ is the disjunction, $f_8(x, y)$ is the conjunction, $f_5(x, y)$ is the implication, f_7 is the equivalence, whereas f_1, f_{16} are the tautology and the contradiction functions, respectively.

x	y	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Table 3: All the two-variable binary functions

For three variables we will have $2^{2^3} = 256$ different functions $f(x, y, z)$. In general there exist 2^{2^n} different binary logical functions $f(x_1, x_2, \ldots, x_n)$ of n variables.

Example 1 *Functions or propositions of one variable*

Let x be the statement *"Peter is tall"*. Then, we can construct all the four one-variable functions;

- $id(x) =$*Peter is tall,*
- $N(x) = \neg x =$ *Peter is not tall,*
- $\tau(x) = x \vee \neg x =$ *Peter is tall or he is not tall,*
- $\gamma(x) = x \wedge \neg x =$ *Peter is tall and he is not tall*

Example 2 *Some functions or propositions of two variables*

Let x, y be the statements *"Peter is tall"* and *"Peter is thin"*, respectively. Then we can construct the following functions, among all the 16 possible ones (Table 3;

- $f_2(x, y) = x \vee y =$ *Peter is tall or thin* (disjunction)
- $f_8(x, y) = x \wedge y =$ *Peter is tall and thin* (conjunction)
- $f_5(x, y) = x \rightarrow y =$ *If Peter is tall then he is thin* (implication)
- $f_7(x, y) = x \leftrightarrow y =$ *Peter is tall if only if he is thin* (equivalence)
- $f_1(x, y) = \tau(x, y) = (x \wedge y) \vee (\neg x \vee \neg y) =$ *Peter is tall and thin, or he is not tall or he is not thin* (tautology)
- $f_{16}(x, y) = \gamma(x, y) = (x \vee y) \wedge (\neg x \wedge \neg y) =$ *Peter is tall or thin, and he is not tall and he is not thin* (contradiction)

1	$x \vee y = y \vee x$	5	$x \wedge y = y \wedge x$
2	$(x \vee y) \vee z = x \vee (y \vee z)$	6	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$
3	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee r)$	7	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
4	$x \vee \gamma = x$	8	$x \wedge \tau = x$
9	$x \vee \neg x = \tau$	10	$x \wedge \neg x = \gamma$

Table 4: Boolean properties of binary logic, where τ=tautology and γ=contradiction

3 Algebra of Boole

Definition 1 *Let S be a finite set. Then the* **power set** *of S, denoted by $\mathcal{P}(S)$, is the set of all subsets of S.*

If the set S has $|S|$ elements then this power set $\mathcal{P}(S)$ has $2^{|S|}$ elements (subsets)

Example 3 *Let S be the set by $S = \{a, b, c\}$.*

We have that S has three elements, then the power set $\mathcal{P}(S)$ has $2^3 = 8$ elements(subsets). In effect, $\mathcal{P}(S) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$, where $\emptyset$ is the empty set and $\{a, b, c\}$ is the full set S itself.

Definition 2 *An Algebra of Boole is a non empty set $\mathcal{B}$ with two binary operations, the sum $(+)$, and the product $(\cdot)$, and one unary operation, the complement $(')$; satisfying the following conditions:*

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

disjunction	$\vee$	$\mapsto$	$+$	Boolean sum
conjunction	$\wedge$	$\mapsto$	$\cdot$	Boolean product
negation	$\neg$	$\mapsto$	$'$	Boolean complement
tautology	τ	$\mapsto$	1	product identity
contradiction	γ	$\mapsto$	0	sum identity

Table 5: Operations and identities of binary logic and Boolean algebra

6. *The product is associative, that is,* $(x.y).z = x.(y.z)$*, for all* $x, y, z \in \mathcal{B}$.

7. *The product is distributive with respect the sum, that is,* $x.(y + z) = (x.y) + (x.z)$*, for all* $x, y, z \in \mathcal{B}$.

8. *There exists an neutral element for the product,* $1 \in \mathcal{B}$*, such that* $x.1 = x$ *for all* $x \in \mathcal{B}$.

9. $x + x' = 1$ *for all* $x \in \mathcal{B}$

10. $x.x' = 0$ *for all* $x \in \mathcal{B}$

3.1 Examples

Example 4 *Consider a class* $\mathcal{L}$ *of binary logical propositions, that is, with values on* $\mathbb{Z}_2 = \{0, 1\}$*, with the operations negation* $(\neg)$*, disjunction* $(\vee)$*, and conjunction* $(\wedge)$*, then* $\mathcal{L}$ *with these operations is an infinite boolean algebra*

By making the correspondences of the Table 5 we can verify that the properties of $\mathcal{L}$ shown in the Table 4 satisfy the definition of Boolean algebra.

Example 5 *Consider a finite set* S*, then the power set* $\mathcal{P}(S)$*; with the set operations: union of sets* $\cup$*, the intersection of sets* $\cap$*, and the complement of a set* $'$*; is a finite boolean algebra.*

In effect, any subsets A, B and C of S hold the properties of the Table 6 and therefore $\mathcal{P}(S)$ is a finite Boolean Algebra.

1	$A \cup B = B \cup A$	5	$A \cap B = B \cap A$
2	$(A \cup B) \cup C = A \cup (B \cup C)$	6	$(A \cap B) \cap C = A \cap (B \cap C)$
3	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	7	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
4	$A \cup \emptyset = A$	8	$A \cap \mathcal{P}(S) = A$
9	$A \cup A' = \mathcal{P}(S)$	10	$A \cap A' = \emptyset$

Table 6: The Boolean structure for sets with the operations $'$, $\cup$, and $\cap$

4 Ternary logic

As ternary logic we will mean a system $\mathcal{L}$ whose elements called **propositions** or **statements** are valued in the set $\{0, 1, 2\}$. This set we denote by $\mathbb{Z}_3$. If x is a proposition, the value of x can be seen as a mapping $\nu : \mathcal{L} \to \{0, 1, 2\}$ such that;

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

From this, we have that if $\nu(x) = 1$ (true) under the rules of binary logic then also $\nu(x) = 1$(true) under the ternary logic laws. Analogously for the false value. On the other hand, for the same considerations made for binary logic case, we can avoid ν by making $\nu(x) = x$. Then over $\mathcal{L}$ are defined the following basic operations, [1];

- The negation $\neg$ (unary operation "not")
- The disjunction $\vee$ (binary operation "or")
- The conjunction $\wedge$ (binary operation "and")
- The implication $\to$ (binary operation "if...then")

The system $\mathcal{L}$ is closed under any of these four operations, in the sense that if $x, y \in \mathcal{L}$ then $\neg x \in \mathcal{L}$, $x \vee y \in \mathcal{L}$, $x \wedge y \in \mathcal{L}$, and $x \to y \in \mathcal{L}$. Notice that the implication, in this case, is not derived from de other three basic operations as it happens in the binary logic.

		$Neg(x)$	$Conj(x,y)$	$Disj(x,y)$	$Imp(x,y)$	$Equiv(x,y)$
x	y	$\neg x$	$x \wedge y$	$x \vee y$	$x \to y$	$x \leftrightarrow y$
1	1	2	1	1	1	1
1	0		0	1	0	0
1	2		2	1	2	2
0	1	0	0	1	1	0
0	0		0	0	1	1
0	2		2	0	0	0
2	1	1	2	1	1	2
2	0		2	0	1	0
2	2		2	2	1	1

Table 7: Basic operations in the ternary logic

The value of $\neg x$, $x \vee y$, $x \wedge y$, and $x \to y$ and other composed operations depend on the value of each component x and y. These values can be obtained by using the **true table** as the Table 7 shows. In such Table 7 also it is shown the equivalence operation which can be derived from the conjunction and implication.

Another way to describe the above basic operations is by considering them as functions. The unary operator negation is a function $f : \mathbb{Z}_3 \to \mathbb{Z}_3$, and a binary operator such as the disjunction is a function $f : \mathbb{Z}_3^2 \to \mathbb{Z}_3$. In general, we can define **ternary logical functions** as mappings $f : \mathbb{Z}_3^n \to \mathbb{Z}_3$.

When $n = 1$ we have one-variable functions $f(x)$, and there are $3^{3^1} = 27$ of these functions, among them are, the Identity or Affirmation $id(x)$, the negation $N(x)$, the Tautology $\tau(x)$ and the contradiction $\gamma(x)$. All these 27 functions are also called **modal** functions of x and they are shown in the Table 8

When $n = 2$ we have two-variable functions $f(x, y)$, and there are $3^{3^2} = 19683$ of them. It is impossible, in a single page, to show the true table for each one.

x	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	2	2	2	1	1	1	0	0	0	2	2	2	1	1	1
2	2	1	0	2	1	0	2	1	0	2	1	0	2	1	0	2	1	0

x	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
1	2	2	2	2	2	2	2	2	2
0	0	0	0	2	2	2	1	1	1
2	2	1	0	2	1	0	2	1	0

Table 8: All the one-variable ternary functions

In the same way we can compute that there are $3^{3^3} = 7625597484987$ three-variable different functions. In general there exist 3^{3^n} different ternary logical functions $f(x_1, x_2, \ldots, x_n)$ of n variables.

Example 6 *Some functions or propositions of one variable*

Let x be the simple statement *"it is raining"*, then we show 6 of the 27 one-variable ternary functions $f(x)$

- $f_1(x) = id(x) =$ *it is raining* (affirmation)
- $f_{20}(x) = N(x) = \neg x =$ *it is not raining* (negation)
- $f_8(x) = \tau(x) = x \to x =$ *if it is raining then it is raining* (tautology)
- $f_{22}(x) = \gamma(x) = \neg(x \to x) =$ *it is not true that if it is raining then it is raining* (contradiction)
- $f_2(x) = x \vee \neg x =$ *it is raining, or it is not raining.*

  This function is not a tautology as it happens in the binary logic case. Also this example shows that condition 9 of the definition of Algebra of Boole can not be hold by the ternary logic system. Therefore, the ternary logic system can not have a Boolean Algebra structure.

- $f_{19}(x) = x \wedge \neg x =$ *it is raining, and it is not raining.*
  This function is not a contradiction as it happens in the binary logic case. Also this example shows that condition 10 of the definition of Algebra of Boole can not be hold by the ternary logic system. Therefore, the ternary logic system can not have a Boolean Algebra structure.

Example 7 *Some functions or propositions of two variables*

Let x, y be the propositions *"it is raining"* and *"the sun is shining"*, respectively. Then we write some of the 19683 two-variable functions

- $\tau(x, y) = \neg(x \vee y) \leftrightarrow (\neg x \wedge \neg y) =$*It is not true that it is raining or the sun is shining, if only if it is not raining and the sun is not shining.*
  By making $m_{11}(x, y) = \neg(x \vee y)$ and $m_{12}(x, y) = \neg x \wedge \neg y$ we can verify in the Table 9 that this is a tautology called the first law of the De Morgan. Analogously for $m_{21}(x, y) = \neg(x \wedge y)$, and $m_{22}(x, y) = \neg x \vee \neg y$ we can see that the second law of the De Morgan holds in ternary logic.

- $\alpha(x, y) = \neg x \vee y =$ *it is not raining or the sun is shining.*
  The true table of this function is shown in the Table 10 together the true table of the implication and we can verify over there, that $\alpha(x, y)$ and $imp(x, y)$ are not equivalent as it happens in binary logic

- $\beta(x, y) = \neg y \rightarrow \neg x =$ *If the sun is not shining then it not raining.*
  We can verify in the Table 10 that it is equivalent with the implication as the binary case.

x	y	$\neg x$	$\neg y$	$x \vee y$	$x \wedge y$	m_{11}	m_{12}	m_{21}	m_{22}
1	1	2	2	1	1	2	2	2	2
1	0	2	0	1	0	2	2	0	0
1	2	2	1	1	2	2	2	1	1
0	1	0	2	1	0	2	2	0	0
0	0	0	0	0	0	0	0	0	0
0	2	0	1	0	2	0	0	1	1
2	1	1	2	1	2	2	2	1	1
2	0	1	0	0	2	0	0	1	1
2	2	1	1	2	2	1	1	1	1

Table 9: The De Morgan laws in ternary logic

- $\delta(x, y) = (x \to y) \wedge (y \to x) =$ *If it not raining then the sun is shining, and if the sun is shining then it is raining.* We can see in the Table 11 that it is equivalent to the "equivalent" function.

5 Conclusions

We have shown that, at least, at this elementary level that there exist four main properties of ternary logic;

1. The ternary logic is one generalization of the binary(classic) case. Such generalization is in the sense of each proposition that is true under the rules of the binary logic will be true under the rules of the trivalent case. Analogously for the false propositions.

2. In ternary logic the construction of tautologies is more difficult than in binary logic. Worse for the construction of contradictions.

3. For the ternary logic, the implication $(x \to y)$ operation of two propositions x, y can

x	y	$\neg y$	$\neg x$	β	α	$imp(x,y)$
1	1	2	2	1	1	1
1	0	0	2	0	0	0
1	2	1	2	2	2	2
0	1	2	0	1	1	1
0	0	0	0	1	0	1
0	2	1	0	0	0	0
2	1	2	1	1	1	1
2	0	0	1	1	1	1
2	2	1	1	1	1	1

Table 10: $(x \to y) \leftrightarrow (\neg x \vee y)$ fails and $(\neg y \to \neg x) \leftrightarrow (x \to)$ holds in ternary logic

x	y	$x \to y$	$y \to x$	$\delta(x,y)$	$equiv(x,y)$
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

Table 11: The derivation of the equivalence law in ternary logic

not be derived from the basic operations ($\neg$), ($\vee$), and ($\wedge$) as it happens in the binary case.

4. The ternary logic can not have a Boolean algebra structure whereas the binary logic can have. The proof of this conclusion is given by the functions f_2 and f_{19} of the Table 8.

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.