Advances in Theoretical and Applied Mathematics.
ISSN 0973-4554 Volume 21, Number 1 (2026), pp. 1-10

http://www.ripublication.com/atam.htm

A Study of Logic Gates on Pre A*-Algebra

***1 Dr.Y.Praroopa , 2 Mr. P. Chenchu Babu**
1 Department of Mathematics, SDM Siddhartha Mahila Degree College, Vijayawada, Andhra Pradesh
2 Department of Mathematics,Andhra Loyola College, Vijayawada, AP
** Corresponding author Email id: praroopaphd@gmail.com,drpraroopay@siddharthamahila.ac.in*

Abstract

This paper analyze the concept of logic gates and its real life applications in various fields.This paper describes the concept of *logic gates on Pre A*-algebra***.** We analyze the concept of logic on Pre A*-algebra.This paper studies logic gates on Pre A*- algebra.

Keywords:
Logic,Boolean algebra, Pre A*-algebra,switching algebra, Logic Gates, OR GATE, AND GATE,,NOT GATE

AMS Subject Classification 2010 : 03-XX,03B70,03B80

I. INTRODUCTION:

Logic gates are the fundamental building blocks of digital systems.the name logic gate is derived from the ability of such a device to make decisions,in the sence that it produces one output level when some combinations of input levels are present,and a different output level when other combinations of input levels are present.

There are three basic logic gates that are OR Gate, AND gate ,NOT gate which are described below.The fact that computers areable to perform very complex logic operations,stems from the way these elementary gates are interconnected.The interconnection of gates to perform a variety of logical operations is called logic design.

Pre A*-algebra is regular extension of Boolean logic to 3 truth-values, where 0 stands for false,1 stands for true but the third truth-value stands for an undefined truth-value.By well known definitions of logic gates,AND gate,OR gate,NOT gate,NAND gate,NOR gates in the Boolean algebra, we define logic gates,AND gate,OR gate,NOT gate,NAND gate,NOR gates in Pre A*-algebra**.**We use the operations $\wedge,\vee$ for AND gate,OR gate respectively where the complementation is used by NOT gate.

Boolean algebras, essentially introduced by Boole in 1850's to codify the laws of thought, have been a popular topic of research since then. A major breakthrough was the duality of Boolean algebras and Boolean spaces as discovered by Stone in 1930's. Stone also proved that Boolean algebras and Boolean rings are essentially the same in the sense that one can convert via terms from one to the other. Since every Boolean algebra can be represented as a field of sets, the class of Boolean algebras is sometimes regarded as being rather uncomplicated. However, when one starts to look at basic questions concerning decidability, rigidity, direct products etc., they are associated with some of the most challenging results.

In a draft paper [5], *The Equational theory of Disjoint Alternatives*, around 1989, E.G.Manes introduced the concept of Ada **EQ**(Algebra of disjoint alternatives) $(A, \wedge, V, (-)^I, (-)_\pi, 0, 1, 2)$(Where $\wedge$,V are binary operations on A, $(-)^I$, $(-)_\pi$ are unary operations and 0,1,2 are distinguished elements on A)which is however differ from the definition of the Ada of his later paper[6] *Adas and the equational theory of if-then-else* in 1993. While the Ada of the earlier draft seems to be based on extending the If-Then-Else concept more on the basis of Boolean algebras and the later concept is based on C-algebras $(A, \wedge,V,(-)^{\sim})$)(where $\wedge$,V are binary operations onA,$(-)^{\sim}$ is a unary operation) introduced by Fernando Guzman and Craig C. Squir[2]. In 1994, P.Koteswara Rao[3] first introduced the concept of A*-algebra $(A, \wedge, V, *, (-)^{\sim}, (-)_\pi, 0, 1, 2)$)(where $\wedge$,V,* are binary operations on A, $(-)^{\sim}$,$(-)_\pi$ are unary operations and 0,1,2 are distinguished elements on A) not only studied the equivalence with Ada, C-algebra, Ada's connection with 3-Ring, Stone type representation but also introduced the concept of A*-clone, the If-Then-Else structure over A*-algebra and Ideal of A*-algebra. In 2000, J.Venkateswara Rao[8] introduced the concept Pre A*-algebra $(A, \vee, \wedge, (-)^{\sim})$(where $\wedge$,V are binary operations on A,$(-)^{\sim}$ is a unary operation on A)analogous to C-algebra as a reduct of A*- algebra, studied their subdirect representations, obtained the results that **2**= {0, 1} and **3**= {0, 1, 2} are the subdirectly irreducible Pre-A*-algebras and every Pre-A*-algebra can be imbedded in $\mathbf{3}^X$ **(** where $\mathbf{3}^X$ is the set of all mappings from a nonempty set X into **3**= {0, 1, 2}**).**

Boolean algebra depends on two-element logic. C-algebra, Ada, A*-algebra and our Pre A*-algebra are regular extensions of Boolean logic to 3 truth-values, where the third truth-value stands for an undefined truth-value. The Pre A*-algebra structure is denoted by $(A, \vee, \wedge, (-)^{\sim})$where A is non-empty set,$\wedge,\vee$ are binary operations and $^{\sim}$ is a unary operation.

II Pre A* - Algebra:

2.1 Definition: An algebra $(A, \vee, \wedge, (-)^{\sim})$ satisfying:

(a) $(x^{\sim})^{\sim} = x, \forall x \in A$

(b) $x \wedge x = x, \forall x \in A$

(c) $x \wedge y = y \wedge x, \forall x,y \in A$

(d) $(x \wedge y)^{\sim} = x^{\sim} \vee y^{\sim}, \forall x,y \in A$

(e) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$, $\forall$ x,y,z $\in$ A
(f) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$, $\forall$ x,y,z $\in$ A
(g) $x \wedge y = x \wedge (x^{\sim} \vee y)$, $\forall$ x,y $\in$ A
is called a Pre A* - algebra.

2.2.1 Examples : **3** = {0,1,2} with $\vee$, $\wedge$, $(-)^{\sim}$ defined below is a Pre A* - algebra

$\wedge$	0	1	2
0	0	0	2
1	0	1	2
2	2	2	2

$\vee$	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

Pre

x	$x^{\sim}$
0	1
1	0

1	0	1
$\vee$	0	1
0	0	1
1	1	1

x	$x^{\sim}$
0	1
1	0

Actually (B, $\vee$, $\wedge$, $(-)^{\sim}$) is a Boolean algebra. So every Boolean algebra is a Pre A* - algebra.

2.2.3 Note :- The elements 0,1,2 in examples satisfy the following laws.
(a) $2^{\sim} = 2$; (b) $1 \wedge x = x$, $\forall$ x $\in$ **3** ('1' the identity for $\wedge$)
(c) $1^{\sim} = 0$; (d) $2 \wedge x = 2$, $\forall$ x $\in$ **3**
(e) $0 \vee x = x$, $\forall$ x $\in$ **3** ('0' is the identity for $\vee$)

2.3 Logic on Pre A* – algebra:

We will use digital logic in_ Pre A* – algebra for the elements 0,1,2. in_ Pre A* – algebra we can use logic signals 0,1,2.Suppose 0 for False,1 for True,2 for some specific signal.

For Pre A* – algebra to implement a system,we will choose one of the physical manifestations to represent each value.The inputs and outputs of a digital system represent real quantities,these are 0,1,2 each for different signals.Inputs and outputs of the logic gates can occur in two levels termed as HIGH,LOW or TRUE,FALSEor ON,OFF or simply 1,0 where as for Pre A*-algebra,we can use 3 levels in which 1 for HIGH or TRUEor ON,2 for LOW or FALSE or OFF and 2 for any other indication.

We adopt the connection that the lines entering the gate symbol from the left are input lines and the single line on the right is the output line.

2.4 Logic Gates on Pre A* – algebra:

2.4.1 (a) OR Gate :-

OR Gate may havetwo or more inputs but only one output.The output assumes the logic state 1,even if one of its input is logic 1 state.Its output assumes the logic state0 state,only when each one of its input is logic 0 state.OR gate may takes output 1,even if one of its input is 1.

In Pre A*-algebra,OR gate is used for v operation.OR gate may takes output 2,even if one of its input is 2.

Fig (i) shows an OR Gate with inputs x and y and output A = x v y where V is defined by the truth table in Fig. (ii).

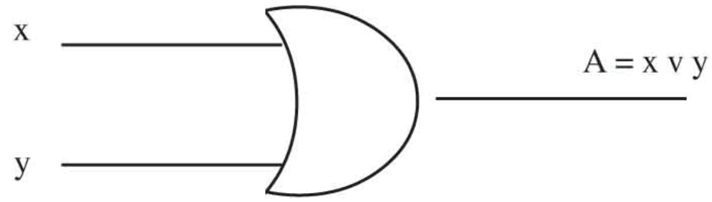

Fig (i)

x	y	x v y
1	1	1
1	0	1
1	2	2
0	1	1
0	0	0
0	2	2
2	1	2
2	0	2
2	2	2

Fig (ii)

Logic symbol and truth table for OR gate in Pre A*-algebra

Thus the output A = 0 only, when input x = 0 and y = 0, otherwise A = 1 or 2 gate may have mere than two inputs Fig (iii) shows an OR gate with four inputs p, q, r, s and output A = pvqvrvs

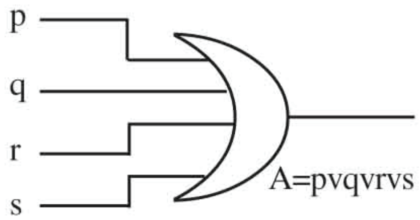

Fig (iii)

The output A = 0 if and only if all the inputs are 0.
Suppose for instance, the input data for the OR gate in fig (iii) are the following sequences:

P =	1 0 2 0 1 2
q =	0 0 1 0 0 1
r =	2 1 0 0 0 2
s =	0 1 2 0 1 0
pvqvrvs=	2 1 2 0 1 2

The OR gate only yields 0 when all input bits are 0. This occurs only in the 4^{th} position (reading from left to right).
Thus the output is the sequence A = 212012

2.4.2 AND Gate :

AND Gate may have two or more inputs but only one output.The output assumes the logic state 1,even if all of its input is logic 1 state.Its output assumes the logic state 0 state,only when one of its input is logic 0 state.AND gate may takes output 1,even if each one of its input is 1.
In Pre A*-algebra, AND gate is used for Λ operation, AND gate may takes output 2,even if one of its input is 2.
Fig (iv) shows an AND gate with inputs p and q and output A = p Λ q where multiplication is defined by pq, the truth table is shown in Fig (v).

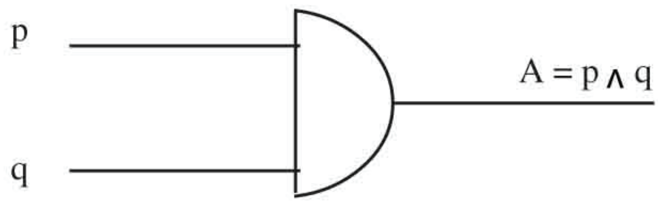

Fig (iv)

p	q	$p \wedge q$
1	1	1
1	0	0
1	2	2
0	1	0
0	0	0
0	2	2
2	1	2
2	0	2
2	2	2

Fig (v)

Logic symbol and truth table for AND gate on Pre A*-algebra

Thus the output A = 1 when input p = 1 and q = 1 otherwise A = 0 or 2
Such as AND gate may have more than two inputs.
Fig (vi) shows an AND gate with four inputs p, q, r, s and output $A = p\wedge q\wedge r\wedge s$.

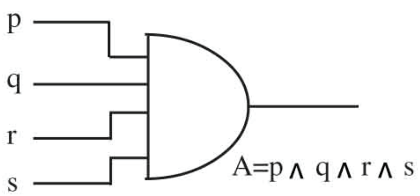

Fig (vi)

The output A = 1 if and only if all the inputs are 1
Suppose, for instance, the input data for the AND gate in Fig (vi) have the following sequences.

P =	1 1 1 0 2 1
q =	0 2 1 1 0 1
r =	2 0 1 2 2 1
s =	1 0 1 2 0 1
$p\wedge q\wedge r\wedge s$ =	2 2 1 2 2 1

The AND gate only yields 1 when all input bits are 1. This occurs in 3rd and 6th positions. Thus the output in the sequence 2 2 1 2 2 1

2.4.3 NOT gate :

NOT gate has only one input and only one output.It is a device whose output is always the complement of its input. That is In Pre A*-algebra NOT gate is used for ~ operation. In Pre A*-algebra the output of the NOT gate assumes the logic 1 state,when its input is in logic 0 state and assumes the logic 0 state,when its input is in logic 1 state and assumes the logic 2 state,when its input is in logic 2 state.

Fig (vii) shows a NOT gate, also called a complement, with input x and output $x^{\sim}$ where complement denoted by the ~, in defined by the truth table in Fig (viii).

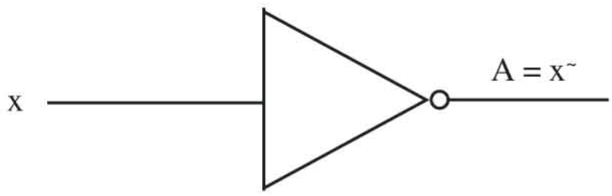

Fig (vii)

x	$x^{\sim}$
0	1
1	0
2	2

Fig (viii)

Logic symbol and truth table for NOT gate on Pre A*-algebra

The value of the output $A = x^{\sim}$ is the complement of the input x.

i.e., $x^{\sim} = 1$ when $x = 0$

$x^{\sim} = 0$ when $x = 1$

$x^{\sim} = 2$ when $x = 2$

We emphasize that a NOT gate can have only one input, whereas the OR ad AND gates may have two or more inputs.

Suppose, for instance a NOT gate is asked to process the following three sequences.

A_1 = 0 2 0 0 0 1

A_2 = 1 0 2 0 0 1

A_3 = 2 0 1 0 0 1

The NOT gate changes 0 to 1, 1 to 0

Note that 2 does not changes. Thus

$A_1^{\sim}$ = 1 2 1 1 1 0

$A_2^{\sim}$ = 0 1 2 1 1 0

$A_3^{\sim}$ = 2 1 0 1 1 0

are the three corresponding outputs.

2.4.4 NAND and NOR gates on Pre A* – algebra:

The negation of AND gate is called as the NAND gate in Pre A* – algebra and the negation of OR gate is called as the NOR gate in Pre A* – algebra.

There are two additional gates which are equivalent to combinations of the above basic gates.

a. NAND gate, pictured in Fig (XI) is equivalent to an AND gate followed by a NOT gate.
b. NOR gate, pictured in Fig (XII) is equivalent to an OR gate followed by a NOT gate.

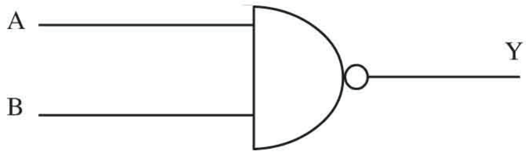

Fig (XI)

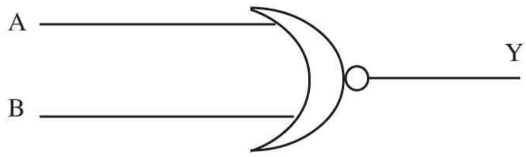

Fig (XII)

The truth tables for these gates (using two inputs A ad B) appear in Fig (c). The NAND and NOR gates can actually have two or more inputs just like the corresponding AND ad OR gates. Furthermore, the output of a NAND gate is 0 if and only if all the inputs are1, and the output of a NOR gate is 1 if ad only if all inputs are 0. also the output of a NAND and NOR gate is 2 if one of the input is 2.

A	B	NAND	NOR
1	1	0	0
1	0	1	0
1	2	2	2
0	1	1	0
0	0	1	1
0	2	2	2
2	1	2	2
2	0	2	2
2	2	2	2

Fig (XIII)

Observe that the only difference between the AND and NAND gates and between the OR and NOR gates is that the NAND ad NOR gates are each followed by a circle. We can also use such a small circle to indicate a complement before a gate. For example, the Boolean expressions corresponding to the two logic circuits in following Fig (XIV) are as follows.

a). $Y = (p^{\sim} \wedge q)^{\sim}$

b). Y = $(p^{\sim} \vee q^{\sim} \vee r)^{\sim}$

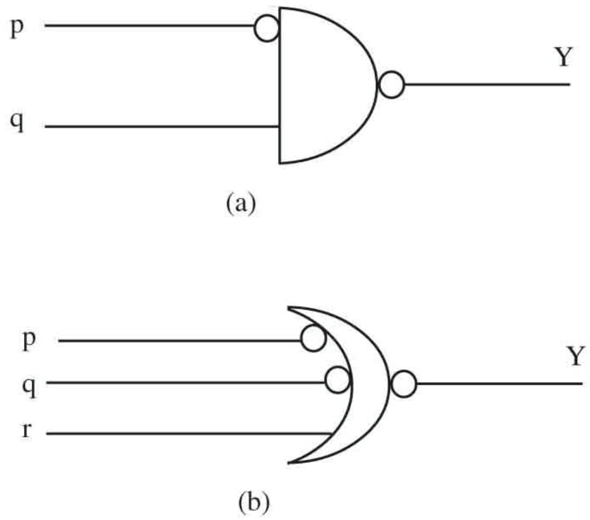

Fig (XIV)

Conclusion

This paper describes the concept of *logic gates on Pre A*-algebra***.** We analyze the concept of logic on Pre A*-algebra.This paper studies logic gates on Pre A*- algebra.

REFERENCES:

[1] Birkoff G: Lattice theory, American Mathematical Society, Colloquium Publications,Vol.25, New York, 1948.

[2] Fernando Guzman and Craig C.Squir: The Algebra of Conditional logic, Algebra Universalis 27(1990), 88-110.

[3] Koteswara Rao.P, A*-Algebra, an If-Then-Else structures(thesis) 1994, Nagarjuna University, A.P., India.

[4] Manes E.G: The Equational Theory of Disjoint Alternatives, Personal Communication to Prof.N.V.Subrahmanyam (1989).

[5] Manes E.G: Ada and the Equational Theory of If-Then-Else, Algebra Universalis 30(1993), 373-394.

[6] Venkateswara Rao.J., On A*-Algebras(Thesis) 2000, Nagarjuna University, A.P., India

[7] Venkateswara Rao.J,Praroopa.Y "Boolean algebras and Pre A*-Algebras", Acta Ciencia Indica (Mathematics), (ISSN: 0970-0455), (2006) 32: pp 71-76.)

[8] Venkateswara Rao.J and Praroopa.Y, " Pre A*-Algebra as a semilattice", Asian Journal of Algebra, (2011)Volume 4, Number 1, 12-22

[9] Venkateswara Rao.J and Praroopa Y., "Lattice in Pre A*-Algebra", Asian Journal of Algebra, ISSN 1994-540X Volume 4, Number 1, 1-11, 2010.

[10] Praroopa.Y, On Pre A*-Algebras(Thesis) 2012, Nagarjuna University, A.P., India
[11] Praroopa.Y,Concepts of Pre A* -Algebra in International Journal of Creative Research Thoughts in UGC approved journal 49018 with ISSN No.2320-2882 ,Volume 13, Issue 4, April 2025
http://www.ijcrt.org/papers/IJCRT2504018.pdf