

COMUNICACION BREVE CONSTRUCCIÓN DE UNA LÓGICA TRINARIA CON LA HOJA DE CÁLCULO DE EXCEL

William Jiménez

Magíster en Docencia de las matemáticas.

williamaj@hotmail.com

Sergio Pachón, Paola Martínez, Paula Moreno, Daniel Duque

Estudiantes Educación Básica Secundaria.

huesosoriginal@hotmail.com, pao_03401@hotmail.com, paumm11@hotmail.com, danielfd_94@hotmail.com

Instituto Pedagógico Nacional

Resumen

Nuestro trabajo está basado en el proyecto realizado por los estudiantes de la promoción 2010 del Instituto Pedagógico Nacional del énfasis de matemáticas, que consistió en analizar los 16 conectores lógicos generados a partir de la disyunción (V) y la negación (–). Este trabajo dio las herramientas para dar inicio a nuestro proyecto, que radica en generar un sistema “trinario”, investigando el comportamiento y propiedades de cada uno de los conectores lógicos resultantes de tres valores de verdad y sus negaciones existentes. Buscamos cuál de las posibles negaciones mantiene las mismas propiedades en el sistema trinario con respecto al binario.

Palabras clave: Sistema trinario, lógica.

Abstract

Our work is based on the project by students from the Class of 2010 National Pedagogical Institute of the emphasis on mathematics, which was to analyze the logical connectors 16 generated from the disjunction (V) and negation (–). This work gave the tools to begin our project, which lies in generating a "trinary" investigating the behavior and properties of each of the logical connectors resulting from three values of truth and his denials exist. We are looking for which of the possible negatives maintains the same properties in the trinary system with respect to binary.

Key words: Trinary system, logic.

INTRODUCCIÓN

Para comenzar a generar el sistema trinario, definimos la operación \vee , con tres valores de verdad, tomando 0 como verdadero, 1 como neutro y 2 como falso, tratando de mantener las propiedades que se dan con sólo dos valores de verdad. Teniendo esta tabla, se buscó representar esta operación en Excel por medio de la función RESIDUO (multiplica los números, luego lo divide entre tres y finalmente da el residuo en módulo 3) con la intención de programar las otras tablas, con la cual logramos determinar la primera tabla de disyunción (\vee):

Se puede observar que la tabla 1 cumple las características de la disyunción (\vee) como: conmutativa, modulativa, asociativa.

Tabla 1

	V		
v	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

DESARROLLO

Después de observar estas propiedades, definimos una negación inicial para programar, de modo que si esta tabla de negación es afectada cambiando sus términos entonces todas las otras tablas definidas por ésta y por la tabla de disyunción, también se verán afectadas. Utilizando estas dos tablas comenzamos a definir nuestras 15 tablas de verdad faltantes. Primero, comenzamos con las que necesitaban de pocas transformaciones como por ejemplo las operaciones número: 1-2-4-7 las cuales seguimos trabajando y con ellas generamos las otras 15 proposiciones:

	v				Negación		
v	0	1	2		0	1	2
0	0	0	0		2	1	0
1	0	1	2		2	1	0
2	0	2	1		2	1	0

(pVq) V ~p				pVq				pV~q				~((~pVq))V~p			
P0Q	0	1	2	P1Q	0	1	2	P2Q	0	1	2	P3Q	0	1	2
0	0	0	0	0	0	0	0	0	0	2	1	0	1	0	2
1	0	1	0	1	0	1	2	1	0	1	2	1	1	1	2
2	0	2	0	2	0	2	1	2	0	0	0	2	1	2	2

~((~pVq))V~q				~((~pVq))V(~pV~q))				~(~pV~q)				~pV~q			
P5Q	0	1	2	P6Q	0	1	2	P7Q	0	1	2	P8Q	0	1	2
0	1	1	1	0	1	2	0	0	1	0	2	0	1	2	0
1	0	1	2	1	2	1	2	1	0	1	2	1	2	1	0
2	2	2	2	2	0	2	1	2	2	2	2	2	0	0	0

~(pVq))V~q				~(pVq)				~(pVq))V~p				~pVq			
P10Q	0	1	2	P11Q	0	1	2	P12Q	0	1	2	P4Q	0	1	2
0	1	1	1	0	2	2	2	0	1	2	0	0	0	0	0
1	2	1	0	1	0	1	2	1	1	1	0	1	2	1	0
2	0	0	0	2	1	0	2	2	1	0	0	2	1	2	0

~(pV~q)				~(pVq)				~((pVq) V ~p)				~((~pVq))V(~pV~q))			
P13Q	0	1	2	P14Q	0	1	2	P15Q	0	1	2	P9Q	0	1	2
0	2	0	1	0	2	2	2	0	2	2	2	0	1	0	2
1	2	1	0	1	2	1	0	1	2	1	2	1	0	1	0
2	2	2	2	2	2	0	1	2	2	0	2	2	2	0	1

Con base a las tablas anteriores logramos programar en Excel algunas tautologías básicas que se cumplen en el sistema binario, con la intención de analizar y comprobar si estas proposiciones siguen siendo tautologías dentro del sistema trinario, para lo cual hay que probar con cada una de las negaciones biyectivas y, así determinar en qué casos se cumplen más tautologías o si estas no se mantienen en ningún momento. Como tautologías básicas tomamos:

1. $\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$
2. $\neg(\neg p) \Leftrightarrow p$
3. $(p \rightarrow q) \Leftrightarrow (\neg p \vee q)$
4. $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$

El siguiente paso después de tener estas tautologías fue escribir los conectores en términos de la disyunción, por ejemplo $p \rightarrow q$ es equivalente a $\neg p \vee q$, con el objetivo de hacer la comprobación programando la hoja de Excel como lo veníamos haciendo. Las tautologías quedaron así:

1. $\neg(\neg(\neg(\neg(\neg p \vee \neg q)) \vee (\neg p \vee \neg q))) \vee (\neg(\neg(\neg p \vee \neg q) \vee (\neg(\neg p \vee \neg q))))$
2. $\neg(\neg(\neg p \vee p)) \vee (\neg p \vee \neg p)$
3. $\neg(\neg(\neg(\neg(\neg p \vee q)) \vee (\neg p \vee q))) \vee (\neg(\neg(\neg p \vee q) \vee (\neg(\neg p \vee q))))$
4. $\neg(\neg(\neg((\neg p \vee q) \vee (\neg p \vee q))) \vee (\neg(\neg((\neg p \vee q) \vee (\neg p \vee q))))$

Después de tener estas tautologías programadas en la hoja, seguimos a probar cada una de las 6 negaciones existentes, para observar cuál de ellas cumplía. (tabla 2)

Tabla 2

	0	1	2
$\neg p$	2	0	1
$\neg q$	2	0	1
\neg	2	0	1

La programación planteada por los estudiantes de undecimo que iniciaron este estudio, facilitó conjeturar que sólo la siguiente negación cumple con tres de las tautologías, lo que nos hace tomar a esta negación como esencial para nuestro sistema; por supuesto que al cambiar la negación los anteriores 15 conectores también variarían obteniendo como resultado final:

V				0				1				2			
V	0	1	2	$\neg p$	2	0	1	$\neg q$	2	0	1	\neg	2	0	1
0	0	0	0												
1	0	1	2												
2	0	2	1												

$(pVq) \vee \neg p$				pVq				$pV\neg q$				$\neg(\neg(pVq)) \vee \neg p$				$\neg pVq$			
P0Q	0	1	2	P1Q	0	1	2	P2Q	0	1	2	P3Q	0	1	2	P4Q	0	1	2
0	0	0	0	0	0	0	0	0	0	2	1	0	0	2	1	0	0	0	0
1	0	0	2	1	0	1	2	1	0	0	0	1	0	2	0	1	2	0	1
2	0	0	1	2	0	2	1	2	0	1	2	2	0	2	2	2	1	0	2

$\neg(\neg(pVq)) \vee \neg q$				$\neg(\neg(\neg pVq)) \vee (\neg(pVq))$				$\neg(\neg pV\neg q)$				$\neg pV\neg q$				$(\neg(\neg pVq)) \vee (\neg(pV\neg q))$			
P5Q	0	1	2	P6Q	0	1	2	P7Q	0	1	2	P8Q	0	1	2	P9Q	0	1	2
0	0	0	0	0	0	1	2	0	0	2	1	0	1	0	2	0	1	2	0
1	2	2	2	1	1	0	2	1	2	2	2	1	0	0	0	1	2	1	0
2	1	0	2	2	2	2	0	2	1	2	0	2	2	0	1	2	0	0	1

Podríamos después entrar a analizar en qué momento se pierde la tautología número 2, la cual en ninguna de las negaciones puede ser una tautología para el sistema trinario.

CONCLUSIONES

La programación en la hoja de cálculo de Excel permitió concluir que no existe una negación biyectiva en el sistema trinario que permita hacer una copia fiel de la lógica binaria, es decir, que al comparar las propiedades de las tablas obtenidas en el sistema binario con las construidas en el trinario, se llega a la conjetura que estas no se mantienen en los dos sistemas, lo que se debe quizás a la elección inicial del conector "V" y su tabla asociada, la cual se determinó como la función residuo.

BIBLIOGRAFÍA

Muñoz, Q. (2006). Introducción a la teoría de conjuntos. Ed. Universidad Nacional de Colombia.

Ostra A. (2001) Simetría y Lógico La notación de Peirce para los 16 conectivos binarios. En memorias XII Encuentro de Geómetra y sus Aplicaciones. Universidad Pedagógica Nacional.