Laboratorio de Altos Estudios en Ciencias Informáticas
Programa de Entrenamiento Académico (PEA) en Ciencias Informáticas

Cátedra de Introducción a la Lógica y Métodos Científicos
Prof. Eugenia Bahit

Tema V
Lógica Simbólica y Tablas de Verdad

Índice de contenidos

Lógica Simbólica y Tablas de Verdad....1
- Lógica simbólica....2
- Variables sentenciales....2
- Conjunciones....2
- Tablas de verdad. 3
- Negación y agrupación....4
- Disyunción....4
  - Disyunción inclusiva....4
  - Disyunción exclusiva....5
- Condicionales....6
- Bibliografía....7

Lógica simbólica.

A medida que se avanza en el estudio de la lógica comienza a hacerse absolutamente indispensable la aplicación de un lenguaje formal y de métodos legibles, sintéticos e inequívocos para la verificación de la validez de los argumentos.

La **lógica simbólica** (o lógica matemática) es la que incorpora el uso de signos para generar un lenguaje universal, denominado **lenguaje formal**, que evita las confusiones y "malos entendidos" que pueden producirse en el lenguaje natural y cotidiano.

En el estudio que haremos de la lógica, los signos se encuentran ya definidos. El estudio particular de éstos corresponde a la ***semiótica*** y es menester de todo científico que pretenda proponer sus propias teorías, profundizar en el estudio de los procesos semióticos. Por el momento, solo nos limitaremos al uso de los símbolos.

Variables sentenciales.

Para la sustitución de enunciados utilizaremos las letras medias del alfabeto, *"**p**"*, *"**q**"*, *"**r**"*, *"**s**"*, …. De esta forma, el enunciado hipotético:

> Si el sábado llueve entonces el concierto se suspende.

Se simbolizaría como:

> Si ***p*** entonces ***q***.

Conjunciones.

Para unir de forma conjuntiva utilizaremos el punto "**.**".

De esta forma, el enunciado:

> El sábado llueve y se suspende el concierto.

se escribiría como:

$$p \cdot q$$

Tablas de verdad.

A fin de simplificar la forma de evaluar el valor de verdad de un enunciado compuesto, vamos a incorporar las *tablas de verdad*. Las **tablas de verdad** se confeccionan a base de un encabezado que contiene enunciados y a continuación, una fila por cada combinación de "verdaderos y falsos" que pueda haber. Incorporamos las letras "**V**" y "**F**" para simbolizar *verdadero*[1] y *falso* respectivamente.

Así el enunciado $p \cdot q$ expuesto en la conjunción anterior, quedaría representado por la siguiente tabla de verdad:

p	q	$p \cdot q$
V	**V**	**V**
V	**F**	**F**
F	**V**	**F**
F	**F**	**F**

Para armar una tabla de verdad, se descompone un enunciado "desde adentro hacia afuera", o menos metafóricamente, desde sus partes más pequeñas hasta las más grandes, yendo paso a paso, es decir, respetando las conexiones lógicas entre los componentes.

Por eso en la tabla anterior vemos las tres columnas p , q y $p \cdot q$, porque las partes más pequeñas de $p \cdot q$ son p , q .

Luego, en cada fila, se van poniendo los resultados de la evaluación por separado. Primero, el de los componentes más pequeños y luego, el de los más grandes.

1 En muchas traducciones latinoamericanas al español de libros de autores como, entre otros, Irving M. Copi, se suele utilizar la letra "T" del inglés "true" que significa "verdadero" en idioma español.

Negación y agrupación.

Para negar un enunciado utilizaremos la tilde “**~**”.

Suponiendo los mismos enunciados componentes *p* y *q* del ejemplo anterior, si quisiéramos negar uno de ellos (*q*, por ejemplo), diríamos:

$p \cdot \sim q$ (esperando *p* verdadera y *q* falsa)

Para negar *p*:

$\sim p \cdot q$ (esperando *p* falsa y *q* verdadera)

Para negar el enunciado completo, incorporamos los paréntesis “**( )**” para agrupar:

$\sim (p \cdot q)$ (esperando que sea falso que *q* y *p* sean ambas verdaderas)

Para negar *p* y *q*:

$(\sim p) \cdot (\sim q)$ (esperando *p* falsa y *q* falsa)

Disyunción.

Para escribir enunciados disyuntivos utilizaremos letra “**v**” ya que la palabra *“vel”* en latín, representa el modo inclusivo de la conjunción disyuntiva “o” (y *“aut”*, el de la exclusiva).

Disyunción inclusiva.

En el caso de un enunciado disyuntivo inclusivo, tal como:

Es un error de programación o los datos están corruptos.

quedaría simbolizado como:

$p \vee q$

y demostrado por la siguiente tabla de verdad:

p	q	$p \vee q$
V	**V**	**V**
V	**F**	**V**
F	**V**	**V**
F	**F**	**F**

Como bien puede observarse, $p \vee q$ solo será falsa cuando ambas, p y q individualmente sean falsas.

Disyunción exclusiva.

A diferencia del anterior, para indicar que no ambos componentes pueden ser verdaderos a la vez, como en el caso de un enunciado disyuntivo exclusivo tal como:

O es un error de programación o los datos están corruptos pero no ambos.

Será necesario expresar dicha exclusión de forma literal, es decir, diciendo que *"p o q pueden ser verdaderas pero que p y q (al mismo tiempo) deben ser falsas"*:

$$(p \vee q) \cdot \sim(p \cdot q)$$

Lo anterior, queda comprobado por la siguiente tabla de verdad:

p	q	$p \vee q$	$p \cdot q$	$\sim(p \cdot q)$	$(p \vee q) \cdot \sim(p \cdot q)$
V	**V**	**V**	**V**	**F**	**F**
V	**F**	**V**	**F**	**V**	**V**
F	**V**	**V**	**F**	**V**	**V**
F	**F**	**F**	**F**	**V**	**F**

Para comprender mejor la tabla, debe explicarse que:

- las columnas 1 y 2 son los enunciados componentes de los enunciados de las columnas 3 y 5;
- las columnas 3 y 5 son los enunciados componentes de la última columna;
- la evaluación de la columna 3 se hace sobre las columnas de sus enunciados componentes (columnas 1 y 2);
- la columna 4 solo es necesaria para la evaluación de la columna 5;

- las última columna es la comprobación del enunciado completo (evaluado a base de sus enunciados componentes de las columnas 3 y 5);
- las filas 2 y 3 demuestran los 2 únicos casos en los que $(p \vee q) \cdot \sim(p \cdot q)$ es verdadero.

Observación para programadores informáticos:Es de hacer notar que mientras que $p \vee q$ *es representado en la mayoría de los lenguajes de programación algebraicos por la instrucción* `(p or q)`*, muchos de ellos resuelve* $(p \vee q) \cdot \sim(p \cdot q)$ *mediante un token específico como* `(p xor q)` *por ejemplo, evitando así la instrucción completa* `(p or q) and not (p and q)`*. Sin embargo, a pesar de la "reducción", internamente el ordenador SIEMPRE ejecutará la versión formalmente lógica, es decir,* `(p or q) and not (p and q)`*.*

Condicionales.

Comentamos en el tema anterior que en un condicional no puede pasar que el consecuente sea falso y el antecedente verdadero porque si el consecuente es falso, necesariamente el antecedente también lo es (*modus tollens*). Por lo tanto, el condicional *"si p entonces q"* se representaría de la siguiente forma:

$$\sim(p \cdot \sim q)$$

Sin embargo, a fin de simplificar aún más dicha implicación, incorporamos el símbolo *herradura* $\supset$ obteniendo finalmente:

$$p \supset q$$

y dejando validado el condicional mediante la siguiente tabla de verdad:

p	q	$\sim q$	$p \cdot \sim q$	$\sim(p \cdot \sim q)$	$p \supset q$
V	**V**	**F**	**F**	**V**	**V**
V	**F**	**V**	**V**	**F**	**F**
F	**V**	**F**	**F**	**V**	**V**
F	**F**	**V**	**F**	**V**	**V**

Para entender mejor la tabla, debe observarse que:

- las últimas dos columnas son en realidad "el mismo enunciado" solo que escrito de diferentes formas;
- la quinta columna es necesaria porque la evaluación debe hacer en base a la negación de la cuarta.

Bibliografía.

Copi, Irving M., *Lógica Simbólica*, Ed. CECSA, México, 1992, cap. 2 y 3.
Copi, Irving M., *Introducción a la Lógica*, Ed. EUDEBA, Buenos Aires, 2014, cap. VII
Gianella, Alicia E., *La lógica simbólica y Elementos de Metodología Científica*, Ed. EUDEBA, Buenos Aires, 2002, cap. I y III