iscarena-cba.infd.edu.ar

Instituto Superior Dr. Carlos María Carena [Unidades]

~1 minute

Unidad N°2

Buenas tardes, comparto la presentación de la clase de hoy!! Iniciamos con LOGICA!! Les pido que realicen la lectura del texto y luego miren los 2 videos. Anoten dudas que puedan surgir y lo hablamos en la proxima clase presencial!

CLASE N°6 LOGICA Introducción 14/5/25 VIRTUAL

Conceptos básicos de Lógica

Proposiciones lógicas

Una **proposición** es básicamente una declaración que puede ser verdadera o falsa (pero no ambas cosas al mismo tiempo). La veracidad o falsedad de una proposición es lo que se suele llamar su **valor de verdad**.

Ejemplos y contraejemplos:

Ya que pueden ser verdaderas o falsas, las siguientes son proposiciones:

- $10 + 10 = 30$
- Hoy es sábado.
- Está lloviendo.
- El led indicador de señal está apagado.
- Hay conexiones disponibles.
- No hay alimentación en los puertos 1, 2 y 3.

Las siguientes, **no** son proposiciones:

- $10 + 10$
- Andá a buscarlo.
- Quizás hoy llueva.
- Altas llantas.
- Ojalá que llueva.
- ¿Cuál es la velocidad máxima permitida?

En lógica, cada proposición (la oración completa) se representa con una letra como p, q, r, etc.

Las proposiciones lógicas son muy usadas en python, por ejemplo, luego de un if, un elif o un while, como una **condición que debe cumplirse para ejecutar un bloque de código.**

1

Conectivos lógicos (operaciones lógicas)

Podemos conectar varias proposiciones para formar nuevas proposiciones "más grandes". Algunos conectivos son: "y", "o", "no", "implica", etc. Ejemplos:

Si tenemos las proposiciones

p: Hay conexiones disponibles

q: El led indicador de señal está apagado.

Podemos formar la nueva proposición $p \wedge q$ (que se lee "p y q"):

$p \wedge q$: Hay conexiones disponibles ***y*** el led indicador de señal está apagado.

Que es lo mismo que decir "Hay conexiones disponibles ***pero*** el led indicador de señal está apagado". (Es exactamente lo mismo, el "pero" lógicamente juega el papel de la "y").
$p \wedge q$ es **una** nueva proposición, ya que es una oración que, en conjunto (es decir toda la oración), puede ser falsa o verdadera, pero no ambas cosas al mismo tiempo. En estos casos, las proposiciones "elementales" que forman la proposición más grande suelen llamarse **premisas**, es decir, en este ejemplo p y q se llaman **premisas** (sin dejar de ser proposiciones).

Tablas de verdad

Cada proposición tiene un valor de verdad: o es verdadera o es falsa. Una proposición compuesta por varias premisas va a tener un valor de verdad que depende del valor de verdad de las premisas que la componen. Consideremos el ejemplo de antes:

p: Hay conexiones disponibles

q: El led indicador de señal está apagado.

$p \wedge q$: Hay conexiones disponibles **y** el led indicador de señal está apagado.

El valor de verdad de $p \wedge q$ depende del valor de verdad de p y de q.

2

Si p fuera falso (es decir que no hubiera conexiones disponibles) y q fuera verdadero (el led realmente está apagado), entonces p ∧ q sería falso (es decir, la oración, considerada como un todo sería falsa). Y así, dependiendo de si p y q son verdaderas o falsas, la proposición p ∧ q será verdadera o será falsa. Para visualizar todas las posibilidades, se suele hacer lo que se llama una **tabla de verdad**. En las primeras columnas se representan las premisas y todas las combinaciones de V y F entre ellas. En las siguientes columnas se van combinando premisas usando conectivos lógicos (operaciones lógicas) y acá el valor de verdad se va calculando usando el valor de verdad de las premisas y conociendo cómo trabaja cada conectivo. En el ejemplo anterior empezaríamos así:

p	q	p ∧ q
V	V	
V	F	
F	V	
F	F	

A partir de acá se calcula lo de la tercera columna, sabiendo que para que p ∧ q sea verdadero, tanto p como q deben ser verdaderas:

p	q	p ∧ q
V	V	V
V	F	F
F	V	F
F	F	F

Negación

p	$\neg p$
V	F
F	V

Se trata simplemente de negar una proposición, y el valor de verdad resultante será contrario al original.

Si la premisa es p: Boby es un perro, la negación se escribe ¬p y se lee "no p", o bien "Boby no es un perro", si lo primero es verdadero, lo segundo será falso y viceversa.

En python, se utiliza la palabra reservada `not` para este operador lógico.

Por ejemplo, `print(not 2==2)` imprimirá `False`.

Ejemplo: p: Todos los perros son blancos.

¿Cuál sería la negación de p? Puede haber diferentes maneras de expresarla, pero en todo caso debe ser una proposición que **siempre** tenga el valor de verdad contrario a p, nunca podría pasar que las dos sean verdaderas o que las dos sean falsas.

"Todos los perros son negros" no está bien, ya que podría haber algunos blancos y otros negros, y en ese caso las dos expresiones serían falsas.

"Ningún perro es blanco" tampoco estaría bien, por la misma razón.

"Algunos perros son blancos y otros no" no está bien, ya que si todos los perros fueran marrones, las dos proposiciones serían falsas al mismo tiempo.

"No todos los perros son blancos" es una manera directa de negar p, y es correcta, aunque cotidianamente es común que uno entienda que se está asumiendo que hay perros blancos, pero formalmente "No todos los perros son blancos" no afirma que haya perros blancos.

"Algún perro no es blanco" también es una negación correcta de p. Esta expresión contempla el caso de que un solo perro no sea blanco, que algunos sean blancos y otros no, y el caso en que ninguno fuera blanco.

Conjunción

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Es el "y" lógico que vimos y ejemplificamos antes. La conjunción será verdadera sólo si las dos premisas lo son. Si una de las premisas es falsa, la conjunción también es falsa. En python se usa la palabra reservada `and` para este operador.

Disyunción inclusiva (o simplemente "disyunción")

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

Es el "o" lógico. La disyunción será verdadera si alguna de las premisas lo es, sino será falsa (es decir, si las dos premisas son falsas).

En matemática se usa el símbolo ∨ (que se lee "o"), mientras que en python se usa la palabra reservada `or`.

Si tenemos las proposiciones p: Boby es un perro y q: Boby puede volar, entonces p ∨ q : Boby es un perro o puede volar.

Boby es un perro	Boby puede volar	Boby es un perro o puede volar
V	V	V
V	F	V
F	V	V
F	F	F

Disyunción exclusiva

p	q	$p \veebar q$
V	V	F
V	F	V
F	V	V
F	F	F

La disyunción exclusiva es verdadera cuando alguna de las premisas lo es, pero no ambas al mismo tiempo.
En matemática se utiliza el símbolo ⊻. En otros contextos se usa la palabra xor, pero no es una palabra reservada de python para esta operación, sino que en python se usa el símbolo ^ (¡no confundir con el símbolo matemático ∧ para la conjunción!).

Boby es un perro	Boby puede volar	O Boby es un perro o Boby puede volar (pero no las dos cosas)
V	V	F
V	F	V
F	V	V
F	F	F

7

Lógica proposicional

Para comenzar esta unidad, les presento dos videos para que tomen nota de los conceptos fundamentales que en el se trabajan.

Todas las dudas pueden consultarse en la próxima clase presencial

La actividad propuesta es, como escribí más arriba, que tomen nota de todos los conceptos trabajados en los videos

Saludos