Section 2.1: Intro to Logic

Monday, January 22, 2024 10:23 AM

logical proposition: a statement that is either true or false

examples:

1. Java is a computer language.
2. Bill Gates cofounded Microsoft
3. The number seven is an even number.

non-examples:

4. Please put your books away.
5. Where is Sarytai's office?
6. He is six feet tall.

who is he?

- if the statement contains a variable (he) and that variable is undefined, then not a proposition

but "Paul is Pat Wren's neighbour and he is six feet tall."

is a proposition

notation: use lower-case letters

p, q, r (and s, t)

example: let p = "Pat drinks coffee"

operators:

"not" - negation

for p, the negation can be written as

we will use this one

$\sim p$

$\neg p$

used in symbolic logic - advantage of using ~ (tilde) is that it appears on a standard keyboard

$!p$ often used in computing (pronounced "bang")

$\bar{p}$ we'll see this later in Boolean algebra

p' pronounced "p prime"

when are two propositions negations of each other?

- when exactly one of them is true at a time

- there is no overlap and the two statements between them cover all possibilities

---

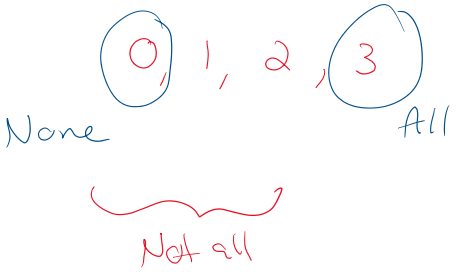
the negation of "All are ..." or "Everyone is ..."

is "At least one person was not ..."

13 "At least one person was not ..."

example: suppose I have three coins

how many are quarters?



None ≠ Not all

because "not all" includes the cases of 1 quarter and 2 quarters

while "none" doesn't

---

logical connectives:

"and" (conjunction) joins two propositions

notation: $p \wedge q$

$p \wedge q$ is true when both p and q are true

$p \wedge q$ is false when at least one of p and q is false

(one or other or both)

other notations:

in computing, will see p && q

ampersand

See this in C++/Java/Javascript

in English, there are other words that we use that are logically equivalent to "and" but sound more natural in sentences.

Sanjay ate dinner and not dessert.

Sanjay ate dinner but not dessert.

Section 2.1: cont'd 2024/01/23

"or" (inclusive disjunction)

notation: $p \vee q$

$p \vee q$ is true when at least one of them (p, q) is true (one or the other or both)

$p \vee q$ is false when both are false

(in computing, a common notation is $p \| q$,

and unfortunately Java uses $\wedge$ for "exclusive or")

"exclusive or" (exclusive disjunction)

notation: $p \oplus q$ ← I will use

p XOR q

(also $p \veebar q$)

$p \oplus q$ is true when one or the other but not both are true

$p \oplus q$ is false when p and q are both true or both false

problem: in English, the word "or" can mean either the "inclusive or" or the "exclusive or" and we tell the difference by context

---

order of operations (OoO)

when you are doing arithmetic, to evaluate the expression

$2 + 3 \cdot 4$

you need to know the "order of operations": which comes first, adding (+) or multiplying (·)?

in the same way, there is an order of operations in logic

NOT is done first
AND is next
OR is last (as well as XOR)

you can think of NOT like exponents
AND multiplication
OR addition
(more on this later)

so $p \vee q \wedge r$ is the same as $p \vee (q \wedge r)$

and $\sim p \vee q$ is the same as $(\sim p) \vee q$

note: if you wanted to negate $p \vee q$
you'd use brackets

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

so brackets are used to override the default order of operations, just like in arithmetic

$$(2+3) \cdot 4$$

↑ brackets mean addition first

examples: what is the order of operations

a) $\sim p \vee q$ NOT, then OR

b) $\sim(p \vee q)$ OR, then NOT

c) $p \vee \sim q \wedge r$ NOT, then AND, then OR

d) $(p \vee \sim q) \wedge r$ NOT, then OR, then AND

e) $p \vee \sim(q \wedge r)$ AND, then NOT, then OR