Today is sunny and Paul is attending CL (Computational Logic) course

First sentence - P

Second sentence - Q

And ^ it is a connective

P^Q

## Connectives

^ = and

v = or

¬ not

→ if … then

P → Q IF today is sunny THAN Paul is attending CL course

& = than

| = all

~ = not

c (rotated 180’, facing left) = implication

Artificial language (for comicating to computer) is poorer than natural language (italian, english…)

There is an inverse proportion between the expressivity of the language and the computational performance. If you use an expressive language that you have to scanseate many things for a computational performance you get less performances

Poorer language gives better results. They are inversely proportional

What is an “expression” built up with connectives and atomic sentences (can’t be split)?

P is an atomic sentence

P^Q can be split in P and Q

We can’t left anything inutive, our aim is to build an artificial language that is completely precise

We have at disposal

* The connectives ^, v, ¬, →
* Atomic sentences (these are a set L)
* Brackets ) (

A string of our basic material could be:

P → Q

P^Q

PQ^→^ Meaningless!

Need to establish a rule for legal expression (that have sense) and meaningless things

A legal expression is called a formula (a formula is a string, but not all string are formulas)

DEFINITION:

Any element of L is a formula (Any P € L is a Formula). Any atomic sentence is a legal expression

if A and B are formulas so are:

* (A^B)
* (¬A)
* (A v B)
* (A → B)
* …

Parker: A program that recognises if a string is legal or not

The parker checks if the string is inside set L, if it is then is fine. If not it split it in smaller pieces, then restarted and checked again in set L. Every time it restart it checks a smaller string

EXAMPLE:

((P →Q)v(¬R)) these is legal

* it start with ( and end with )
* there is a connector (v) in the middle (in this case it is a disjunction)
* the parker restart with two smaller string
* The parker is ok if in the end it obtains atomic sentences. If it success means that it is a legal expression

((P →Q)v(¬R))

Two restart

↓ \

((P →Q) and (¬R))

restart again

/ \ ↓

P Q R

(P→→Q) is not legal

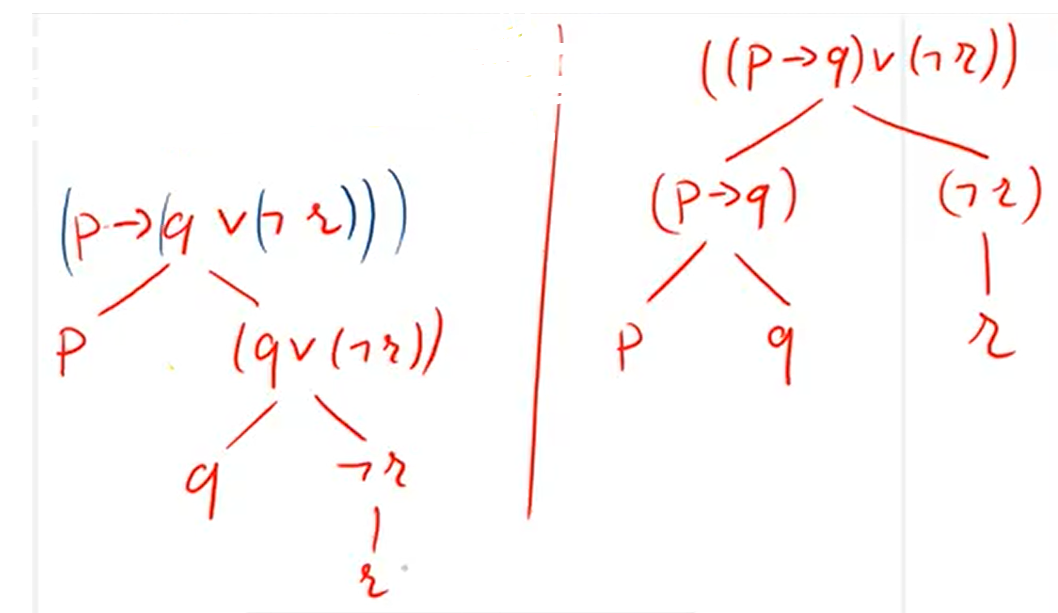
* it start with ( and end with )
* but the connective is not working, it choose one
* restart with P and →Q
* P is ok (atomic sentence)
* →Q is not

If you don't have the brackets ((P →Q)v(¬R)) can become (P →(Qv¬R))

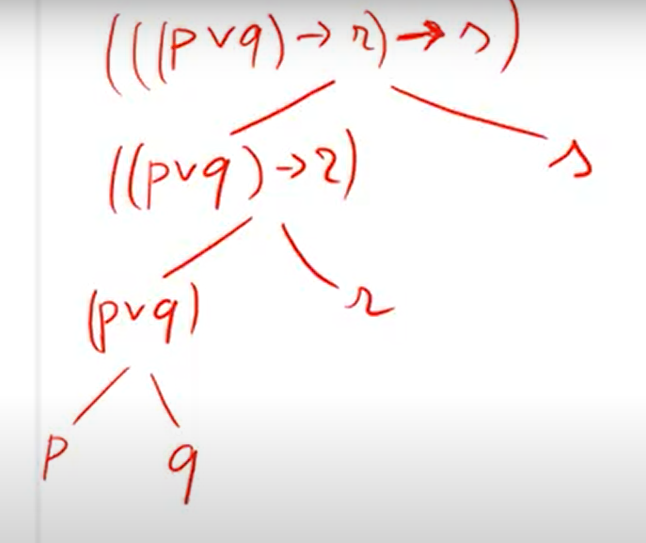
And it restart in another way

Using the bracket you create a unique way. If a formula is legal there is ONLY ONE way to be interpreted.

The brackets help you staying on the safe side, it avoid ambiguity



Example



**Uniqueness of reading**

From the definition it follows that if a string is a formula (so the string can’t be meaningless) it can build up in only 1 way

you can decompose a formula in two formulas in a unique way (that because of brackets)