if you have a set like {a, b, c } a string is a finite sequence of symbols from the set. Like aabccab

A string contains:

* connectives ^ v → ¬
* prepositional letters p, q, r… they can be infinite
* brackets ) (

Some strings are meaningless, some are meaningful. To distinguish them, there are precise instruction

* If P is a propositional variable then P is a formula
* If A, B are formulas so are (A ^ B), (A v B), (A → B), (¬ A)
* ((P v Q) → R) is a formula because it is made of (P v Q) and R, so P, Q, R. They are constructed in a precise way
* This construction is **unique**. If you don’t write down the brackets you don’t know precisely what it is saying. It is ambiguous, can be written in different ways, if you use the bracket then there is only 1 unique way

**NUMBERS OF CONNECTIVES**

C(P) = 0 if P is a propositional variable

C(A^B) = C(A) + C(B) + 1

C(A → B) = C(A) + C(B) + 1

C( ¬ A) = C(A) + 1

For continuing the number of connectives you have to split it and then add 1

**Priorities**

From the bigger to the smaller

¬ (biggest priority)

^ V

→ (smallest priority)

Example: ¬ P → Q ^ R it is the same of (( ¬ P) → (Q ^R))

No priorities between ^ v

So P ^ Q v R is illegal

But

P ^ Q ^ R is ok, have the same meaning, don’t depend on the brackets

P v Q v R is ok, have the same meaning, don’t depend on the brackets

P → Q → R is illegal

P ^ Q ^ ( R → S) → S1 v S2 is legal (((P ^ Q) ^ ( R → S)) → (S1 v S2))

**Semantics**

Semantics: Something that connects your formulas with words

The notation of a formula is true or false, the meaning of a formula is nothing absolute, depends on the circumstances

P “It is sunny” can be true or false, depends on the situation

True, T, 1

False, F, 0

Formula: is a piece of information, depends on the situations

**Assignment**

Assignment: is a function that associate to any propositional letter a truth value (can be true (1) or false (0))

L set——>{1, 0}

The assignment assigns a truth value ONLY at the propositional letters, we want to extend the assignment to all the formulas

P “It is sunny”

Q “Professor SG is in room 405”

V(P)=1 V(Q) = 1

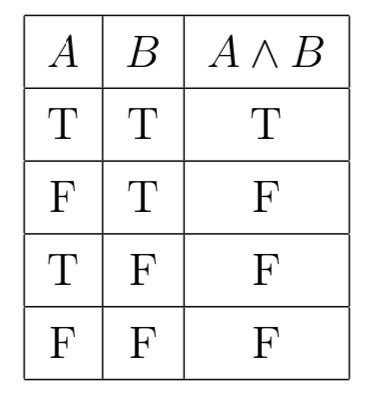
What is the valuation of P and Q?

V(P^Q) = ? It is a compositional meaning, it’s meaning it is computed starting from smaller species, there are compositional rules depending on the connectives

Compositional meaning: giving meaning to complex prepositions starting from the atomic one

**Connective ^ (and)**

(The table can be written also with 1 and 0)



The valuation of A and B:

V(A^B) = min(V(A), V(B))

T = True = 1

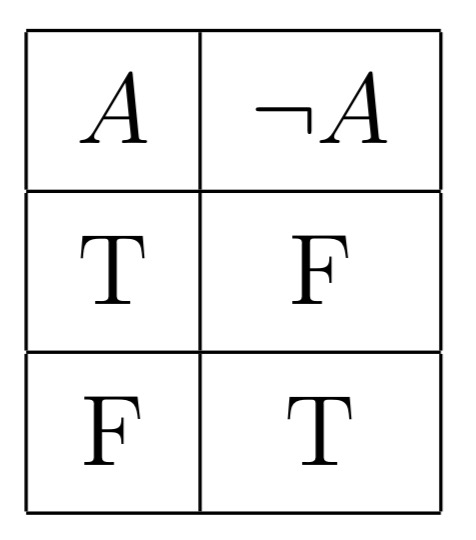
F = False = 0

If V(A) and V(B) are both True (1) the minimum is 1, so V(A ^ B) is = 1 (True)

If only one is true then the other is false and the minimum is 0 so V(A^B) = 0

If both V(A) and V(B) are false then the minimum between 0 and 0 is 0, so V(A^B)=0

**Connective ¬ (not)**



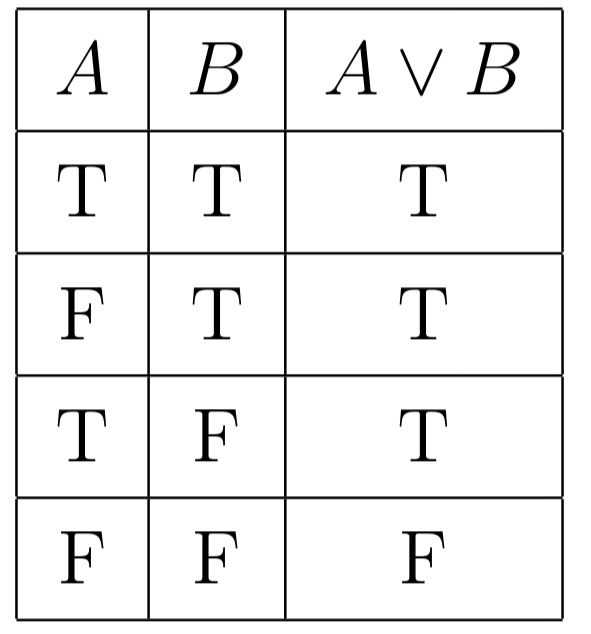
Valuation:

V ( ¬ A) = 1- V(A)

If V(A) is 0 than V( ¬A) is 1

If V(A) is 1 than V ( ¬A) is 0

**Connective v (or)**



If they are both True then A v B is True. It is conventional, in the concrete use it can be False.

Our aim is to build an artificial language. So it is conventionally decided that if both are True, so A v B is true.

If you want A v B to be false (even if A is true and B is false ) it can be written as

((A v B) ^ ¬ (A ^ B))

Valuation:

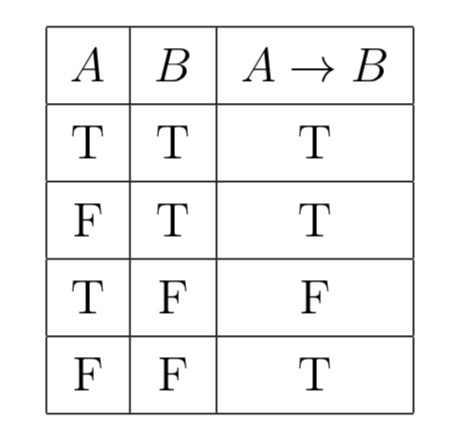
V(A v B) = max (V(A), V(B))

If both V(A) and V(B) are true then the maximum between 1 and 1 is 1 so V(A v B) = 1

If only one is true then the other is false. The maximum between 1 and 0 is 1 so V (A v B) = 1

If both are false then the maximum is 0 and V(A v B) = 0

**Connective → (implies)**



It is impossible to evaluate those sentences. It is important to limit our language in a mathematical way.

The only false possibility is when something is true and the results are false.

Valuation:

V(A → B) = max(1- V(A), V(B))

If V(A) is 1 and V(B) is 0 V(A→ B) = 0

In all the other cases the V(A → B) = 1

**Evaluate the true value of complex sentences**

Example: ¬ (P^Q)

P Q P^Q ¬ (P^Q)

1 1 1 0

1 0 0 1

0 1 0 1

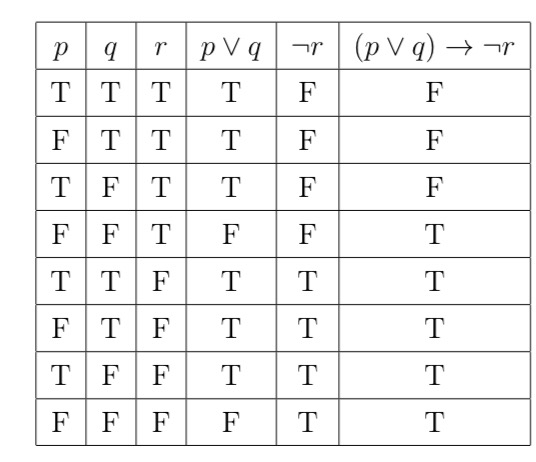
0 0 0 1

If A has n variables the table (all possible assignment ) A has **2^n** rows

The main problem is that if there are many variables it becomes impossible to create a table

Example 2: (P v Q) → ¬ R

With three variables (P,Q,R) it becomes 8 rows. 2^3 = 8



**Inference**

An inference is true if it is composed of true arguments. An inference is correct when all the premises are true the conclusion should be true

An inference is correct iff whenever its premises are true, so it is its conclusion.

You should not consider the meaning of the sentence but only the logical form

If it rains, I take an umbrella

I do not take an umbrella

———————

Hence it does not rain

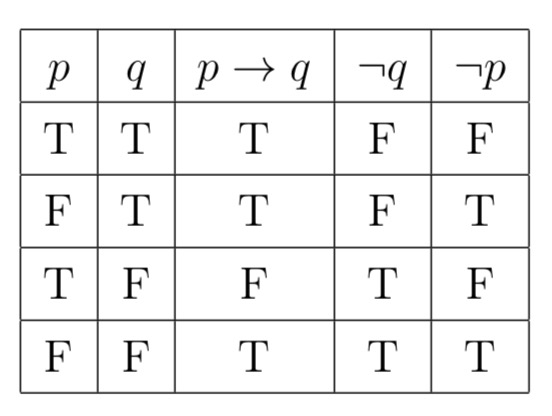
This is the same as:

P → Q

¬ Q

————

¬ P



¬ P is the conclusion

Example

P → Q

¬ P

————

¬ Q

this is wrong. If you consider all the possible cases

P Q P→ Q ¬ P ¬ Q

1 1 1. 0. 0

1 0 0. 0. 1

0 1 1. 1. 0

0 0 1. 1. 1

Consider the rows V(P→ Q) = 1

V ( ¬ P) = 1

For the inference to be correct I must have V( ¬ Q) = 1 in the last two rows

Since its not the case the inference is wrong

Example 2:

P v Q

¬ P

————-

Q

Is this correct?

Let’s write

| P | Q | P v Q | ¬ P | Q |
| --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |

Since V(P v Q) = 1 and V(¬P) = 1 the premisies are true. The conclusion Q is also true, so the inference was correct

**Tuples**

A(P1,…, Pn) when A contains at most the variables P1,…, Pn

We can associate with A function called **truth-function**

*(‘N = powered to the N)*

{0,1}’N means all possible **tuplets** of zero’s and one’s

Tuples : string of n elemets

{0,1}’3 = {<0, 0, 0,>, <0, 0, 1>, ...}

{0,1}’4 = {<0, 0, 0, 0>, <0, 0, 0, 1>, …, <1, 0, 1, 1>, ...}

{0,1}’N it is a set containing all possible tuples

To a formula A(P1, …, P2) I can associate a function

Fa: {0,1}’N → {0,1}

A tuple <t1, …, tn> is an assignment

I can assume V(P1) = t1

V(Pn) = tn

Tuples and assignments are the same

N = 3 <1, 0, 1> € {0,1}’3

↓

V(P1) = 1

V(P2) = 0

V(P3) = 1

Fa(t1,…tn) rearrange t1,tn as assignment V and take V(A)

Tuples are all the possible assignments, take the assignment and gives a results

**Functional completeness**

P Q ?

1 1 0

1 0 1

0 1 1

0 0 0

Need to find the function/formula so that the truth table of the formula is exactly this

Theorem:

For every truth function Y: {0,1}’N → {0,1} there is a formula A(P1, … , P2) such that Fa = Y

Whatever abstract “connective” you invent there is a combination of our connectives that gives it. You can imagine a connective (Y) that works. It can be realized using a combination of connectives.

Functional completeness: don’t need more connectives, using a combination of connectives that I have I can build abstract connectives. You don’t have the formula but you have the input and the output. Can be a little difficult

The abstract connective can be realised using a combination of other connectives

**Abstract connectives**

P Q ?

1 1 0

0 1 1

1 0 1

0 0 0

Steps for finding the formula:

1. focusing ONLY on the rows that give 1 as results
2. second row: ¬ P ^ Q so it gives 1 as result
3. third row: P ^ ¬ Q so it gives 1 as result

Unite the two formulas of the second and third rows with a disjunction

((¬ P^ Q) v (P ^ ¬ Q)) if you check the truth table you will get the correct result

Say that A and B are logically equivalent iff we have V(A) = V(B) for every V

(logically equivalent means that A and B have the same truth table)

Exercise: check that the following pairs of formulae are logically equivalent

*CHECK ON GOODNOTES*

I ¬ (A ^ B) ¬ A v ¬B

II ¬ (A v B) ¬ A ^ ¬ B

( ↑ Those are DeMorgans laws)

III A → B ¬ A v B

IV A ^ B ¬ ( ¬ A v ¬ B)

V A v B ¬ ( ¬ A ^ ¬ B)

VI ¬ ¬ A A

(↓ Distributive laws)

VII A ^ ( B v C) (A ^ B) v (A ^ C)

VIII A v( B ^ C) (A v B) ^ (A v C)

**Tautology**

A is a tautology iff V(A) = 1 for all V.

*(iff = if and only if)*

P v ¬ P

¬ ¬ P → P

A is contradiction iff V(A) = 0 for all V

¬ (P v ¬ P)

¬ ( ¬ ¬ P → P)

A contradiction and a tautology are exactly the opposite

**Tautology**: All the rows of the **final column** are = **1**

**Contradiction**: All the rows of the **final column** are = **0**

A is satisfiable iff V(A) = 1 for **at least one V**.

**SAT problem**

Given A decides whether A is satisfiable.

The truth table is too large, and needs a quick algorithm.