

LAB 01

FORMULA:

$$((\neg A \rightarrow \neg C) \wedge ((\neg B \vee (\neg B \rightarrow \neg C)) \rightarrow \neg D) \vee A) \wedge (\neg A \wedge (\neg D \rightarrow C))$$

1 Numărul de sub-formule (formula privită ca șir)

O **sub-formulă** este orice parte care este ea însăși o formulă corectă.

Le enumerăm (fără repetiții):

1. AAA
2. BBB
3. CCC
4. DDD
5. $\neg A$
6. $\neg B$
7. $\neg C$
8. $\neg D$
9. $(\neg A \rightarrow \neg C)$
10. $(\neg B \rightarrow \neg C)$
11. $(\neg B \vee (\neg B \rightarrow \neg C))$
12. $((\neg B \vee (\neg B \rightarrow \neg C)) \rightarrow \neg D)$
13. $((\neg B \vee (\neg B \rightarrow \neg C)) \rightarrow \neg D) \vee A$
14. $(\neg D \rightarrow C)$
15. $(\neg A \wedge (\neg D \rightarrow C))$

16. $((\neg A \rightarrow \neg C) \sim (((\neg B \vee (\neg B \rightarrow \neg C)) \rightarrow \neg D) \vee A))((\neg A \wedge \neg C) \sim ((\neg B \wedge \neg B \wedge \neg C) \rightarrow \neg D) \vee A))((\neg A \rightarrow \neg C) \sim (((\neg B \vee (\neg B \rightarrow \neg C)) \rightarrow \neg D) \vee A))$

17. Formula întreagă

✓ Număr sub-formule: 17

2 Numărul de variabile propoziționale distincte

Variabilele care apar sunt:

- A, B, C, D, A, B, C, D, A, B, C, D

✓ 4 variabile distincte

3 Numărul de apariții pentru fiecare variabilă

Variabilă	Apariții
A	3
B	2
C	3
D	2

4 Ordinul formulei (lungimea logică)

Ordinul = numărul de conectori logici

Conectori folosiți: \neg , \rightarrow , \vee , \sim

Numărăm:

- negații \neg : 7
- implicații \rightarrow : 4
- disjunții \vee : 2

- echivalențe \sim : 3

$$7+4+2+3=16 \quad 7 + 4 + 2 + 3 = 16 \quad 7+4+2+3=16$$

✓ **Ordinul formulei = 16**

5 Tabela de adevăr

Variabile: A,B,C,D $\Rightarrow 2^4 = 16$ A,B,C,D $\Rightarrow 2^4 = 16$ linii.

Notăm formula prin **F**.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

6 Program

```
import itertools

def imp(p, q):
    return (not p) or q

def equiv(p, q):
    return p == q

def formula(A, B, C, D):
    left = equiv(
        imp(not A, not C),
        imp((not B) or imp(not B, not C), not D) or A
    )
    right = equiv(not A, imp(not D, C))
    return equiv(left, right)

for A, B, C, D in itertools.product([False, True], repeat=4):
    print(A, B, C, D, formula(A, B, C, D))
```