

Міністерство освіти та науки України  
Львівський національний університет імені Івана Франка

Факультет електроніки та  
Комп'ютерних технологій

# Звіт

Про виконання лабораторної роботи №6  
“Логічні операції над нечіткими висловлюваннями”

Виконав:  
Студент групи ФеІ-44  
Сапанюк М.І.  
Перевірила:  
Притула М.

Львів 2022

## Мета:

ознайомитися з основними поняттями логіки нечітких висловлювань та операціями над ними.

## Хід роботи:

1. Запрограмувати реалізацію функцій, що відповідають логічним зв'язкам: “заперечення” у вигляді функції NOT(a); “кон’юнкція” у вигляді функції AND(a,b); “диз’юнкція” у вигляді функції OR(a,b); “імплікація” у вигляді функції IMP(a,b); “еквівалентність” у вигляді функції EQU(a,b).

| A | B | NOT A | A AND B | A OR B | A IMP B | A EQU B |
|---|---|-------|---------|--------|---------|---------|
| 0 | 0 | 1     | 0       | 0      | 1       | 1       |
| 1 | 0 | 0     | 0       | 1      | 0       | 0       |
| 0 | 1 | 1     | 0       | 1      | 1       | 0       |
| 1 | 1 | 0     | 1       | 1      | 1       | 1       |

2. Реалізувати алгоритм знаходження ступеня істинності складного висловлювання, заданого викладачем у вигляді логічної функції, з використанням запрограмованих логічних операцій над нечіткими висловлюваннями.

3. Для перевірки результатів побудувати таблицю істинності заданого складного висловлювання для десяти різних значень аргументів логічної функції

$$f_3 = [(A \vee B) \rightarrow C] \equiv [(A \rightarrow C) \vee (\neg C)]$$

| A | B | C | A OR B | (A OR B) IMP C | A IMP C | NOT C | (A IMP C) OR (NOT C) | ((A OR B) IMP C) EQL ((A IMP C) OR (NOT C)) |
|---|---|---|--------|----------------|---------|-------|----------------------|---|
| 0 | 0 | 0 | 0      | 1              | 1       | 1     | 1                    | 1   |
| 1 | 0 | 0 | 1      | 0              | 0       | 1     | 1                    | 0   |
| 0 | 1 | 0 | 1      | 0              | 1       | 1     | 1                    | 0   |
| 0 | 0 | 1 | 0      | 1              | 1       | 0     | 1                    | 1   |
| 1 | 1 | 0 | 1      | 0              | 0       | 1     | 1                    | 0   |
| 0 | 1 | 1 | 1      | 1              | 1       | 0     | 1                    | 1   |
| 1 | 0 | 1 | 1      | 0              | 0       | 0     | 0                    | 0   |
| 1 | 1 | 1 | 1      | 1              | 1       | 0     | 1                    | 1   |
| 1 | 0 | 1 | 1      | 0              | 0       | 0     | 0                    | 0   |
| 1 | 1 | 1 | 1      | 1              | 1       | 0     | 1                    | 1   |

## Висновок:

Виконавши лабораторну роботу я ознайомився з основними поняттями логіки нечітких висловлювань та операціями над ними.

## Додаток:

```
def NOT(a: int) -> int:
    return 1 - a
def AND(a: int, b: int) -> int:
    return min(a, b)
def OR(a: int, b: int) -> int:
    return max(a, b)
def IMP(a: int, b: int) -> int:
    return max(min(a, b), 1 - a)
def EQL(a: int, b: int) -> int:
    return min(max(NOT(a), b), max(a, NOT(b)))
def default() -> None:
    a_array = [0, 1, 0, 1]
    b_array = [0, 0, 1, 1]
    print('A | B | NOT A | A AND B | A OR B | A IMP B | A EQU B |')
```

```

for i in range(0, len(a_array)):
    a = a_array[i]
    b = b_array[i]
    print(f' {a} | {b} | {NOT(a)} | {AND(a, b)} | {OR(a, b)} | {IMP(a, b)} |'
          f' {EQL(a, b)} |')
def f3() -> None:
    a_array = [0, 1, 0, 0, 1, 0, 1, 1]
    b_array = [0, 0, 1, 0, 1, 1, 0, 1]
    c_array = [0, 0, 0, 1, 0, 1, 1, 1]
    print(' A |'
          ' B |'
          ' C |'
          ' A OR B |'
          ' (A OR B) IMP C |'
          ' A IMP C |'
          ' NOT C |'
          ' (A IMP C) OR (NOT C) |'
          ' ((A OR B) IMP C) EQL ((A IMP C) OR (NOT C)) |')
    for i in range(0, len(a_array)):
        a = a_array[i]
        b = b_array[i]
        c = c_array[i]
        AorB = OR(a, b)
        AorBimpC = IMP(AorB, c)
        AimpC = IMP(a, c)
        notC = NOT(c)
        AimpCnotC = OR(AimpC, notC)
        eql = EQL(AorBimpC, AimpCnotC)
        print(f' {a} |'
              f' {b} |'
              f' {c} |'
              f' {AorB} |'
              f' {AorBimpC} |'
              f' {AimpC} |'
              f' {notC} |'
              f' {AimpCnotC} |'
              f' {eql} |')
def main() -> int:
    default()
    print("")
    f3()
    return 0
if __name__ == '__main__':
    main()

```