

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Лабораторна робота №3
Дисципліна: Комп'ютерна дискретна математика

Виконав
Студент групи ПЗПІ-23-3
Харченко Федір Олександрович

Перевірив
Старший викладач каф. ПІ
Терещенко Гліб Юрійович

Тема: "Advanced Propositional Logic and Computational Logic"

Мета: Understand and implement the Advanced Propositional Logic operations and Computational Logic operations.

Код програми:

<https://github.com/rintaro129/CDM/tree/main/LAB3>

```

1  import re
2  from tabulate import tabulate
3
4  class BooleanOperations:
5      @staticmethod
6      def expressionEvaluator(expression : str, dictionary : dict) -> bool:
7          expression = re.sub('AND', 'and', expression, flags=re.IGNORECASE)
8          expression = re.sub('OR', 'or', expression, flags=re.IGNORECASE)
9          expression = re.sub('NOT', 'not', expression, flags=re.IGNORECASE)
10         expression = re.sub('XOR', '^', expression, flags=re.IGNORECASE)
11
12         for i in dictionary:
13             expression = re.sub(i, str(dictionary[i]), expression)
14
15         return eval(expression)
16
17     @staticmethod
18     def truthTableGenerator(expression : str) -> str:
19         expression2 = expression.replace('(', ' ')
20         expression2 = expression2.replace(')', ' ')
21         words = expression2.split()
22
23         headers = []
24         for word in words:
25             if word.upper() not in {'AND', 'OR', 'NOT', 'XOR'}:
26                 headers.append(word)
27
28         headers.sort()
29         headers.append(expression)
30
31         data = []
32         for bitmask in range(2 ** (len(headers) - 1)):
33             temp = bitmask
34             row = [False] * len(headers)
35             for i in range(len(headers) - 1, -1, -1):
36                 row[i] = bool(temp % 2)
37                 temp = temp // 2
38
39             dictionary = dict()
40             for i in range(len(headers) - 1):
41                 dictionary[headers[i]] = row[i]
42
43             row[-1] = BooleanOperations.expressionEvaluator(expression, dictionary)
44             data.append(row)
45
46         table = tabulate(data, headers, tablefmt="github")
47
48         return table

```

Висновок: Such functions as Boolean expression evaluator and truth table generator were implemented. Supported operations in expressions are: and, or, not, xor. Truth table generator supports any number of variables which it extracts from the expression string itself.