

Лабораторна робота №6.3

Задано формулу з використанням логічних операцій (див. 6.1). Перетворіть формулу на еквівалентну, в якій використовується тільки Штрих Шеффера (або Стрілка Пірса).

Код

```
def evaluate_formula(formula, variable_values):  
    """  
    Обчислює значення логічної формули з використанням заданих значень змінних.  
  
    Аргументи:  
    - formula (str): Рядок, що представляє логічну формулу з логічними операціями (*  
    - AND, + - OR, - - NAND).  
    - variable_values (dict): Словник, що містить значення змінних у форматі змінна:  
    значення.  
  
    Повертає:  
    - int: Результат обчислення формули (0 або 1).  
    """  
  
    output_queue = []  
    operator_stack = []  
  
    precedence = {'*': 2, '+': 1, '-': 1}  
  
    for char in formula:  
        if char.isalpha():  
            output_queue.append(variable_values[char])  
        elif char == '0':  
            output_queue.append(0)  
        elif char == '1':  
            output_queue.append(1)  
        elif char in ['*', '+', '-']:  
            while operator_stack and operator_stack[-1] != '(' and precedence[char]  
            <= precedence[operator_stack[-1]]:  
                output_queue.append(operator_stack.pop())  
                operator_stack.append(char)  
            elif char == '(':  
                operator_stack.append(char)  
            elif char == ')':  
                while operator_stack and operator_stack[-1] != '(':  
                    output_queue.append(operator_stack.pop())  
                operator_stack.pop()  
  
    while operator_stack:  
        output_queue.append(operator_stack.pop())  
  
    stack = []  
    for token in output_queue:  
        if token == '-':  
            operand = stack.pop()  
            stack.append(int(not operand))  
        elif token == '*':  
            operand2 = stack.pop()  
            operand1 = stack.pop()  
            stack.append(int(operand1 and operand2))  
        elif token == '+':  
            operand2 = stack.pop()
```

```

        operand1 = stack.pop()
        stack.append(int(operand1 or operand2))
    else:
        stack.append(token)

    return stack.pop()

def replace_operators(formula):
    """
    Замінює символи логічних операцій в формулі на слова.

    Аргументи:
    - formula (str): Рядок, що містить логічну формулу з символами операцій (*, +, -
    ).

    Повертає:
    - str: Рядок заміненої формули з операціями у словесному форматі (AND, OR, NAND).
    """

    replacements = {'*': ' AND ', '+': ' OR ', '-': ' NAND '}
    for old, new in replacements.items():
        formula = formula.replace(old, new)
    return formula

def check_equivalence(formula1, formula2, variable_values):
    """
    Перевіряє еквівалентність двох логічних формул з використанням заданих значень
    змінних.

    Аргументи:
    - formula1 (str): Рядок, що представляє першу логічну формулу.
    - formula2 (str): Рядок, що представляє другу логічну формулу.
    - variable_values (dict): Словник, що містить значення змінних у форматі змінна:
    значення.

    Повертає:
    - bool: True, якщо формули еквівалентні, False - якщо ні.
    """

    result1 = evaluate_formula(formula1, variable_values)
    result2 = evaluate_formula(formula2, variable_values)
    return result1 == result2

formula1 = "A - C"
formula2 = "B"
variable_values = {'A': 1, 'B': 0, 'C': 1}

converted_formula1 = replace_operators(formula1)
converted_formula2 = replace_operators(formula2)

equivalent = check_equivalence(formula1, formula2, variable_values)

print(f"Значення змінних: {variable_values}")
print(f"Формула 1: {converted_formula1}")
print(f"Формула 2: {converted_formula2}")
if equivalent:
    print("Формули еквівалентні")
else:
    print("Формули не еквівалентні")

```

Результат

```
main x
"N:\University\2 курс\2 семестр\Дискретні Структури\Лабораторія 1\main.c"
Значення змінних: {'A': 1, 'B': 0, 'C': 1}
Формула 1: A NAND C
Формула 2: B
Формули еквівалентні

Process finished with exit code 0

"N:\University\2 курс\2 семестр\Дискретні Структури\Лабораторія 1\main.c"
Значення змінних: {'A': 1, 'B': 0, 'C': 1}
Формула 1: A NAND B
Формула 2: B
Формули не еквівалентні

Process finished with exit code 0
```