Дискретні структури ІПЗ-21-2

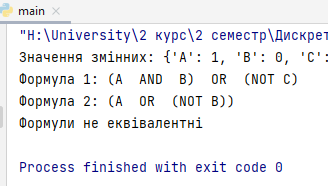
Губарєв Ростислав Вадимович

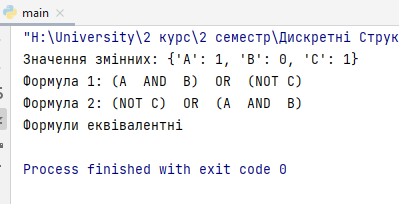
**Лабораторна робота №6.2**

Вказано дві формули з логічними операціями (див. 6.1). Перевірте їхню еквівалентність.

|  |
| --- |
| **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, - - NOT).  - 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, NOT).  """* replacements = {'\*': ' AND ', '+': ' OR ', '-': 'NOT '}  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 \* B) + (-C)" formula2 = "(-C) + (A \* 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**

****

****