**Код**

def вычислить\_полином\_жегалкина(таблица\_истинности):

"""

Вычисляет полином Жегалкина для булевой функции, заданной таблицей истинности.

:param таблица\_истинности: Список значений таблицы истинности (0 и 1).

:return: Строка, представляющая полином Жегалкина.

"""

n = len(таблица\_истинности) # Длина таблицы истинности

коэффициенты = таблица\_истинности[:] # Копируем таблицу истинности для вычислений

# Метод конечных разностей

for i in range(1, n): # Итерация по строкам таблицы

for j in range(n - i): # Пересчет коэффициентов

коэффициенты[j] ^= коэффициенты[j + 1] # Операция XOR для обновления коэффициентов

# Определение переменных

количество\_переменных = len(bin(n - 1)) - 2 # Количество переменных (2^k = n)

переменные = [f'x{i + 1}' for i in range(количество\_переменных)]

полином = []

# Построение полинома на основе коэффициентов

for i, c in enumerate(коэффициенты):

if c == 1: # Если коэффициент равен 1, добавляем член в полином

член = []

for j in range(количество\_переменных):

if i & (1 << j): # Проверяем, входит ли переменная в член

член.append(переменные[j])

полином.append(' & '.join(член) if член else '1') # "1" для константного члена

return ' ^ '.join(полином) if полином else '0' # XOR представлен как ^

# Пример использования

if \_\_name\_\_ == "\_\_main\_\_":

# Таблица истинности для функции x1 XOR x2

таблица\_истинности = [0, 1, 1, 0]

print("Таблица истинности:", таблица\_истинности)

print("Полином Жегалкина:", вычислить\_полином\_жегалкина(таблица\_истинности))

**Объяснение кода и теории**

1. **def вычислить\_полином\_жегалкина(таблица\_истинности):**
   * Функция принимает на вход таблицу истинности (список значений 0 и 1).
   * Таблица истинности задает булеву функцию, для которой вычисляется полином.
2. **коэффициенты = таблица\_истинности[:]**
   * Создается копия таблицы истинности для работы с конечными разностями.
3. **Метод конечных разностей:**
4. for i in range(1, n):
5. for j in range(n - i):
6. коэффициенты[j] ^= коэффициенты[j + 1]
   * Здесь обновляются коэффициенты полинома Жегалкина.
   * Каждый новый коэффициент получается как XOR предыдущих значений.
7. **количество\_переменных = len(bin(n - 1)) - 2**
   * Вычисляем количество переменных в булевой функции, исходя из длины таблицы истинности.
8. **переменные = [f'x{i + 1}' for i in range(количество\_переменных)]**
   * Создаем список имен переменных: x1,x2,…x\_1, x\_2, \dots.
9. **Построение полинома:**
10. for i, c in enumerate(коэффициенты):
11. if c == 1:
12. член = []
13. for j in range(количество\_переменных):
14. if i & (1 << j):
15. член.append(переменные[j])
16. полином.append(' & '.join(член) if член else '1')
    * Для каждого ненулевого коэффициента формируем соответствующий член полинома.
    * Проверяем, какие переменные участвуют в данном члене.
17. **return ' ^ '.join(полином) if полином else '0'**
    * Формируем строку, представляющую полином Жегалкина, используя XOR (^) между членами.
18. **Пример использования:**
19. таблица\_истинности = [0, 1, 1, 0]
20. print("Полином Жегалкина:", вычислить\_полином\_жегалкина(таблица\_истинности))
    * Таблица истинности задает функцию x1⊕x2x\_1 \oplus x\_2.
    * Программа выводит соответствующий полином: x1 ^ x2.