

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01– «Информатика и вычислительная техника»

**Лабораторная работа
по дисциплине
«Дискретная математика и математическая логика»
на тему
«Минимизация логической функции»**

Выполнил студент гр. ИВТ-23-16
Попонин Михаил Александрович

Проверил:

(оценка)

(подпись)

(дата)

г. Пермь, 2024

Цель работы

Целью данной работы является реализация алгоритма минимизации булевых функций на основе метода склеивания. Этот метод позволяет упростить логические выражения, сократив количество используемых переменных и операторов, что важно для оптимизации логических схем и их реализации в аппаратных системах.

Задачи

1. Реализовать и вывести таблицу истинности для заданной булевой функции.
2. Построить совершенную дизъюнктивную нормальную форму (СДНФ) и совершенную конъюнктивную нормальную форму (СКНФ).
3. Построить импликантную матрицу для анализа покрытия булевой функции.
4. Найти и вывести все минимальные покрытия (короткие покрытия) для упрощённой функции.

Код программы

Программа читает вектор булевой функции из файла, строит таблицу истинности, вычисляет СКНФ и СДНФ, а также минимизирует функцию с помощью метода склеивания. Результаты включают промежуточные шаги и окончательный вид минимизированной функции, представленный в удобной форме. Сам код программы можно найти по ссылке: [Minimization of a logical function](#).

Описание функций

1. **__init__**: Конструктор класса Minimization. Инициализирует основные переменные, используемые для хранения вектора функции, термов, импликантов и промежуточных данных.
2. **read_vector_from_file(filename)**: считывает вектор булевой функции из файла. Проверяет корректность данных (длина вектора, символы '0' и '1'). Вычисляет количество переменных на основе длины вектора.
3. **create_truth_table()**: Создаёт таблицу истинности на основе числа переменных и значений функции. Разделяет термы на те, которые используются для построения СКНФ (значения 0), и СДНФ (значения 1).
4. **show_truth_table()**: Выводит таблицу истинности в удобочитаемом виде.
5. **term_to_function(term, is_cnf=False)**: Преобразует терм в строковое представление логического выражения. Используется для генерации СКНФ и СДНФ.
6. **show_nf()**: Выводит совершенные нормальные формы (СКНФ и СДНФ) в текстовом виде на основе данных таблицы истинности.
7. **can_combine(a, b)**: Проверяет возможность склеивания двух термов. Возвращает результат склеивания или индикатор невозможности склеивания.

8. **combine_terms(terms)**: выполняет склеивание термов для одной итерации. Возвращает новый список термов, учитывая использованные и неиспользованные термы.
9. **gluing()**: Основной метод для выполнения склеивания термов. Итеративно объединяет термы, пока новые термы не перестанут образовываться. Хранит результаты промежуточных этапов.
10. **build_implicant_matrix()**: Строит импликантную матрицу, показывающую покрытие исходных термов результирующими импликантами. Возвращает матрицу в виде DataFrame.
11. **minimize_cover(matrix)**: реализует процесс выбора минимального покрытия для термов. Находит минимальный набор импликантов, полностью покрывающих функцию, и возвращает минимизированную функцию в виде строки.
12. **term_to_function_string(term)**: преобразует терм в строку, представляющую логическое выражение, без учёта отрицаний.
13. **print_all_combined_terms()**: Выводит на экран все этапы склеивания термов.
14. **print_implicants()**: Выводит итоговые импликанты после завершения процесса склеивания.

Пример работы программы

Таблица истинности:

x1	x2	x3	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

СКНФ:

$$(x1 \vee \neg x2 \vee x3) * (x1 \vee \neg x2 \vee \neg x3) * (\neg x1 \vee x2 \vee x3) * (\neg x1 \vee x2 \vee \neg x3)$$

СДНФ:

$$(\neg x1 * \neg x2 * \neg x3) \vee (\neg x1 * \neg x2 * x3) \vee (x1 * x2 * \neg x3) \vee (x1 * x2 * x3)$$

Все этапы склеивания:

Шаг 1:

$$(\neg x1 \neg x2)$$

$$(x1 x2)$$

Матрица импликант (с функциями):

				$\neg x1$	$\neg x2$	$\neg x3$	$\neg x1$	$\neg x2$	$x3$	$x1$	$x2$	$\neg x3$	$x1$	$x2$	$x3$
$\neg x1$	$\neg x2$					1				1		0			0
$x1$	$x2$					0				0		1			1

Минимизированная функция:

$$(\neg x1 \wedge \neg x2) \vee (x1 \wedge x2)$$

Итоговые импликанты:

$$(\neg x1 \neg x2)$$

$$(x1 x2)$$

Рисунок 1 – пример работы программы

Таблица истинности:

x1	x2	x3	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

СКНФ:

$(\neg x1 \vee x2 \vee x3) * (\neg x1 \vee x2 \vee \neg x3) * (\neg x1 \vee \neg x2 \vee x3) * (\neg x1 \vee \neg x2 \vee \neg x3)$

СДНФ:

$(\neg x1 * \neg x2 * \neg x3) \vee (\neg x1 * \neg x2 * x3) \vee (\neg x1 * x2 * \neg x3) \vee (\neg x1 * x2 * x3)$

Все этапы склеивания:

Шаг 1:

$(\neg x1 \neg x2)$
 $(\neg x1 \neg x3)$
 $(\neg x1 x3)$
 $(\neg x1 x2)$

Шаг 2:

$(\neg x1)$
 $(\neg x1)$

Матрица импликант (с функциями):

	$\neg x1$	$\neg x2$	$\neg x3$	$\neg x1$	$\neg x2$	$x3$	$\neg x1$	$x2$	$\neg x3$	$\neg x1$	$x2$	$x3$
$\neg x1$			1			1			1			1
$\neg x1$			1			1			1			1

Минимизированная функция:

$(\neg x1)$

Итоговые импликанты:

$(\neg x1)$
 $(\neg x1)$

Рисунок 2 – пример работы программы

Таблица истинности:

x1	x2	x3	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

СКНФ:

$(x1 \vee x2 \vee \neg x3) * (\neg x1 \vee x2 \vee x3) * (\neg x1 \vee \neg x2 \vee \neg x3)$

СДНФ:

$(\neg x1 * \neg x2 * \neg x3) \vee (\neg x1 * x2 * \neg x3) \vee (\neg x1 * x2 * x3) \vee (x1 * \neg x2 * x3) \vee (x1 * x2 * \neg x3)$

Все этапы склеивания:

Шаг 1:

$(\neg x1 \neg x3)$

$(\neg x1 x2)$

$(x2 \neg x3)$

$(x1 \neg x2 x3)$

Матрица импликант (с функциями):

	$\neg x1$	$\neg x2$	$\neg x3$	$\neg x1 x2$	$\neg x3$	$\neg x1 x2 x3$	$x1 \neg x2 x3$	$x1 x2 \neg x3$
$\neg x1 \neg x3$			1		1		0	0
$\neg x1 x2$			0		1		1	0
$x2 \neg x3$			0		1		0	1
$x1 \neg x2 x3$			0		0		0	1

Минимизированная функция:

$(\neg x1 \wedge \neg x3) \vee (\neg x1 \wedge x2) \vee (x2 \wedge \neg x3) \vee (x1 \wedge \neg x2 \wedge x3)$

Итоговые импликанты:

$(\neg x1 \neg x3)$

$(\neg x1 x2)$

$(x2 \neg x3)$

$(x1 \neg x2 x3)$

Рисунок 3 – пример работы программы