Ссылка на статью:
// Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн. 2019, № 2, с. 69-75.

Поступила: 01.11.2019
Окончательный вариант: 18.11.2019

УДК 519.7

Программная реализация булевых функций контактными схемами

Михеева Е.А., Тихоненко А.А.*

*tialexsey@gmail.com
УлГУ, Ульяновск, Россия

Работа посвящена задаче синтеза контактных схем, реализующих булевы функции. Рассмотрены алгоритмы трех методов синтеза: совершенной дизъюнктивной нормальной формы, совершенной конъюнктивной нормальной формы и метода каскадов. На основе описанных алгоритмов разработана программа на языке C#. Приведены особенности программной реализации изложенных алгоритмов, представлены результаты работы программы.

Ключевые слова: *булевы функции, контактные схемы, метод каскадов, метод совершенной дизъюнктивной нормальной формы, метод совершенной конъюнктивной нормальной формы, программная реализация.*

Введение

Данная работа относится к теме программной реализации объектов дискретной математики. Одним из интересных примеров приложения булевых функций (БФ) является теория контактных схем (КС). В теории схем, так же как в алгебре логики, мы сталкиваемся с задачами: известна функция, требуется дать (схемную) реализацию; из различных реализаций надо выбрать по возможности наиболее компактные; дать оценку индекса простоты L(n) (простоты в смысле минимума контактов). Таким образом, в теории контактных схем возникают задачи, родственные по природе задачам из алгебры логики. Отметим также, что решение вышеуказанных задач требует привлечения аппарата алгебры логики. Следовательно, теория контактных схем имеет тесную связь с алгеброй логики, как по линии постановки задач, так и в методах их решения.

В качестве объекта дискретной математики рассматриваются КС. Контактной схемой называется неориентированная сеть (т.е. неориентированный граф с выделенными вершинами - полюсами), каждому ребру которой приписана некоторая булева переменная с отрицанием или без него. Ребро вместе с прописанной ему буквой X^{σ}(σ=0,1) называют замыкающим контактом, если $\sigma = 1$, и размыкающим контактом, если σ=0. Или контакт X^{σ} считается замкнутым, когда $X^{\sigma} = 1$, и разомкнутым, когда $X^{\sigma} = 0$.

Применение алгебры логики в синтезе КС было разработано Шенноном в 1949 г. [4]. Основываясь на литературе Шеннона и Лупанова [1, 2] можно выделить следующие методы синтеза:

1) метод совершенной дизъюнктивной нормальной формы (СДНФ),

2) метод совершенной конъюнктивной нормальной формы (СКНФ),

3) метод каскадов.

В данной работе разрабатываются алгоритмы этих методов, а так же приводится их программная реализация.

1. Алгоритм реализации БФ КС методом совершенной нормальной формы

Рассмотрим булеву функцию [3]:

$$F(f_1(x, y, z), f_2(x, y, z), \dots, f_n(x, y, z)) \tag{1.1}$$

и ее векторное представление

$$F = (f_1\, f_2 \dots f_n) \tag{1.2}$$

Пример 1. Пусть функция имеет вид $F(f_1, f_2, \dots, f_8) = f_1 + f_3 + f_6 + f_8$, векторное представление F = (10100101) и табличное представление (таблица 1).

Таблица 1. Полная таблица истинности

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

При реализации программы выводится БФ. Далее составляется таблица истинности данной функции. Исходя из таблицы истинности введенной БФ, в зависимости от наименьшего числа единичных значений или нулевых значений, выбирается метод. Если число нулевых значений меньше, чем число единичных значений, выбирается метод СКНФ.

Если единичных значений меньше – метод СДНФ. Когда одинаковое количество - используются оба метода.

1.1. Представление СКНФ

Так как СКНФ есть форма разложения любой БФ по всем её переменным в виде конъюнкций дизъюнкций, выводим КС, где конъюнкция дает последовательное соединение, а любой множитель определяет параллельное соединение.

Для таблицы 1 представление СКНФ будет иметь следующий вид:

$$F(1,0,1,0,0,1,0,1) = \left(XvYv\overline{Z}\right)\&(Xv\overline{Y}v\overline{Z})\&\left(\overline{X}vYvZ\right)\&(\overline{X}v\overline{Y}vZ)$$

1.2. Представление СДНФ

Так как СДНФ есть форма разложения любой БФ по всем её переменным в виде дизъюнкций конъюнкций, выводим КС, где дизъюнкция дает параллельное соединение, а любое слагаемое определяет последовательное соединение.

Для таблицы 1 представление СДНФ будет иметь следующий вид :

$$F(1,0,1,0,0,1,0,1) = \overline{XYZ}vX\overline{Y}Zv\overline{X}Y\overline{Z}vXYZ$$

2. Алгоритм реализации БФ КС методом каскадов

Этот метод является усовершенствованием метода Шеннона. Он учитывает некоторые свойства функций и благодаря этому позволяет для многих функций строить сравнительно простые схемы. Суть метода состоит в следующем.

Пусть $f(x_1, \dots, x_n)$ есть функция, подлежащая реализации. Образуем последовательность множеств функций: $G_1, G_2, \dots, G_n$ посредством следующего индуктивного процесса. Рассмотрим множество $G_1 = \{f\}$. Разложим $f(x_1, \dots, x_n)$ по первой переменной в виде: $f(x_1, \dots, x_n) = x_1 f(1, x_2, \dots, x_n) V\, \overline{x_n}\, f(0, x_2, \dots, x_n)$.

Теперь рассмотрим множество $G_2 = \{f(1, x_2, \dots, x_n), f(0, x_2, \dots, x_n)\}$ при условии, что функции $f(1, x_2, \dots, x_n)$ и $f(0, x_2, \dots, x_n)$ разные. Если они одинаковы, тогда G_2 состоит из одной функции. Продолжаем этот процесс дальше.

Пусть множество G_i уже построено. Тогда множество G_{i+1} определяется следующим образом. Разложим функцию $g \in G_i$ по переменной x_i в виде, т.е.: $g(x_i, \dots, x_n) = x_i g(1, x_{i+1}, \dots, x_n) V\, \overline{x_i} g(0, x_{i+1}, \dots, x_n)$.

Образуем множество $G_{i+1} = \{g^1(x_{i+1}, \dots, x_n)\}$. Это множество состоит из всех различных функций, зависящих от n-i переменных, которые участвуют в разложении функций из множества G_i. Этот процесс продолжается, пока не получим множество G_n, $G_n = \{g(x_n)\} \Rightarrow G_n$.Оно может состоять из не более чем четырех функций, т.е. G_n есть множество $\{x_n, \overline{x_n}, 0,1\}$ или некоторое его подмножество [2].

Из определения множеств G_n вытекает следующее свойство:

Каждая функция $g(x_i, \dots, x_n)$ из $G_i (1 \le i \le n-1)$ может быть представлена в виде $g(x_i, \dots, x_n) = x_i g'(x_{i+1}, \dots, x_n) V \overline{x_i} g''(x_{i+1}, \dots, x_n)$, где $g' \in G_{i+1}, g'' \in G_{i+1}$ (2.1)

Наконец, строится КС в соответствии с только что описанным разложением, но в обратном порядке. Сначала реализуются все функции из G_n.

Затем на основе свойства (2.1) реализуются все функции из G_{n-1}; при этом используется один инвертор (на все множество G_{n-1}), по два конъюнктора и один дизъюнктор на каждую функцию из G_{n-1} и т.д.

Алгоритм реализации БФ КС методом каскадов выглядит следующим образом (см. рис. 1). Разложим ФАЛ $f_1, f_2, \ldots, f_m$ сначала по x_1, потом по x_2 и так далее. При этом построим последовательности множеств G_i и $\widehat{G}_i$, состоящих из ФАЛ от $x_i, x_i + 1, \ldots, x_n$, где $i = 1,2, \ldots, n$ такие, что:

1. G_i состоит из всех различных ФАЛ $g(x_i, \ldots, x_n)$ вида $g = f(\sigma_1, \ldots, \sigma_{i-1}, \ldots, x_i, x_{i+1}, \ldots, x_n)$, где $1 \leq i \leq m, (\sigma_1, \ldots, \sigma_{i-1}) \in B^{i-1}$
2. $\widehat{G}_i$ состоит из всех различных функций $g, g \in G_i$, которые существенно зависят от x_i.

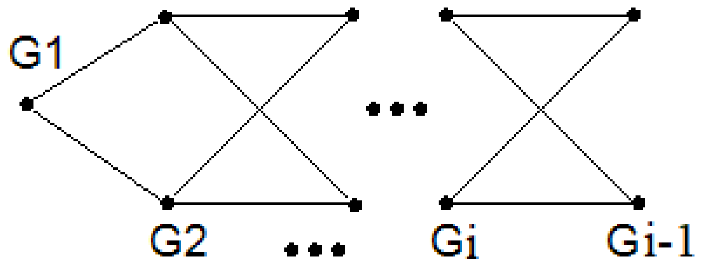

Рис. 1. Пример построения КС методом каскадов.

3. Программная реализация

Вышеописанные алгоритмы реализованы в программе, написанной на языке высокого уровня C#. Программа была создана в среде разработки Visual Studio 2017. Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms. Также предоставляют возможность создавать веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

К преимуществам C# можно отнести объектно-ориентированный подход, большое количество библиотек и шаблонов, позволяющих оптимизировать процесс разработки, средний порог вхождения, а так же наличие в сети большого количества справочной и обучающей информации.

В качестве входных данных программы будем использовать функции, описанные в пункте 1, в векторном виде (1.2).

Главное окно программы представлено на рис. 2. Программа состоит из нескольких областей: область ввода функции, область отображения функции в табличном виде, область вывода результата, область отображения контактной схемы

Для того, чтобы выбрать метод расчета и построения КС, достаточно переключить вкладку вывода результата.

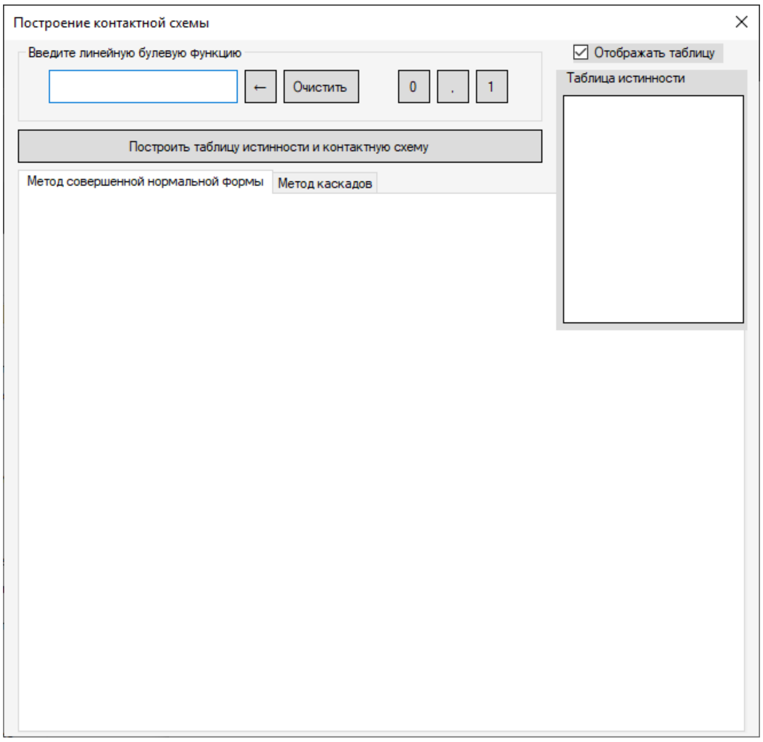

Рис. 2. Главное окно программы.

Во время выполнения программа может показать сообщения с ошибками. В таблице 2 описаны сообщения с ошибками, их возможные причины и решения:

Таблица 2. Сообщения с ошибками

Ошибка	Возможная причина	Решение
1. Не введена функция	Не введена функция в поле ввода для функции	Требуется ввести функцию в поле ввода
2. Неправильный ввод данных	В записи функции допущена ошибка, например, пропущен знак «,» или недостаточно «0» и «1»	Требуется ввести функцию корректно (недостаточная длина функции)
3. Ошибка при построении таблицы истинности функции	При заполнении таблицы истинности функции произошла ошибка. Количество строк в таблице не совпадает с длиной функции в векторном представлении	Требуется ввести функцию корректно (избыточная длина функции)
4. Функция должна состоять из '0' и '1'!	При вводе функции в векторном виде были введены другие цифры, например «2», «3» и т.д.	Функция должна состоять только из «0» и «1»

Результаты выполнения программы для функции $(f_1, f_2, \ldots, f_8) = f_1 + f_3 + f_6 + f_8 = (1,0,1,0,0,1,0,1)$, описанной в примере 1 пункта 1, представлены ниже. На рис. 3 представлен результат построения КС методом совершенной нормальной формы, на рис. 4 – для метода каскадов.

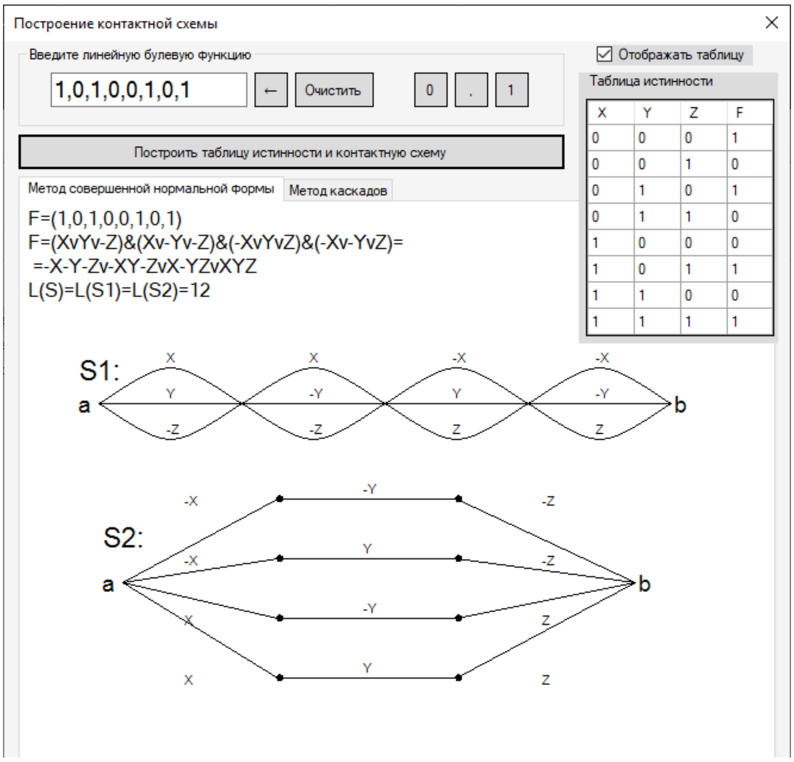

Рис. 3. Результат выполнения программы для метода СНФ.

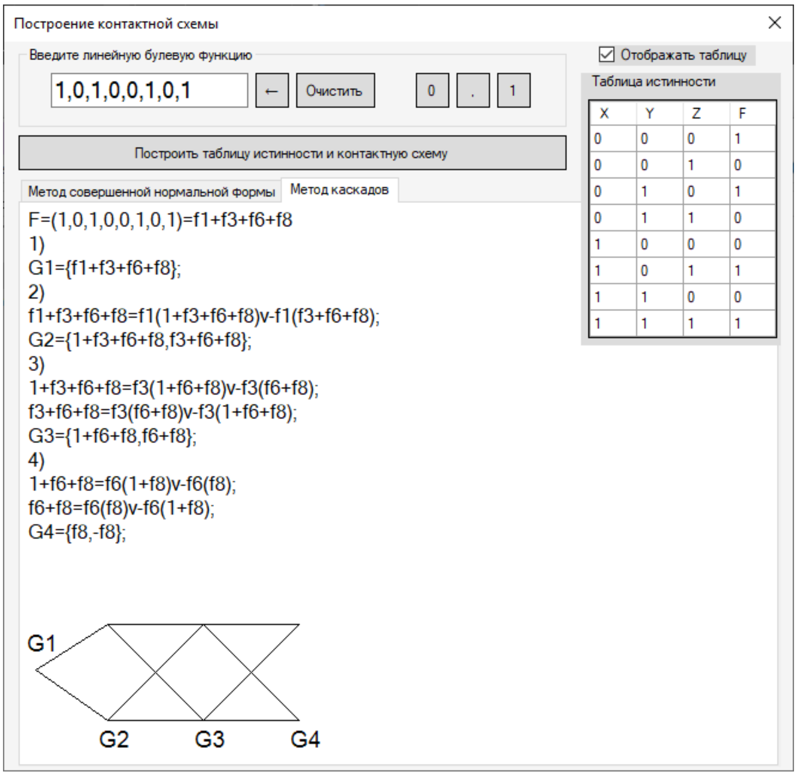

Рис. 4. Результат выполнения программы для метода каскадов.

Заключение

В настоящей статье представлены результаты исследовательской работы по описанию алгоритмов реализации булевых функций контактными схемами. На основе описанных алгоритмов разработана программа, которая может представить булеву функцию из векторного вида в табличный. После этого программа производит вычисления методами СКНФ, СДНФ и методом каскадов. В конечном результате программа строит контактную схему, реализующую данную БФ, для выбранного метода, и выдает результат проделанной работы.

Список литературы

1. Лупанов О.Б. *Асимптотические оценки сложности управляющих систем*. М. : Изд-во МГУ, 1984.
2. Шеннон К. *Работы по теории информации и кибернетике*. М.: Изд-во иностранной литературы, 1963. 830 с.
3. Яблонский С.В. *Введение в дискретную математику*. М.:Наука, 2001. 384с.
4. Shannon C. *The synthesis of two-terminal switching circuits* // BSTJ, 1949, v. 28, №1, p. 59-98.