БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Научная работа

Минимизация булевой функции методом Квайна-Мак-Класки

Исполнители: студенты группы 10903423 Романюк Д.

Харчук У. Н.

Руководитель: Зуенок А. Ю.

МИНСК 2023

# Введение

Минимизация булевой функции является одной из важных задач в области дискретной математики и цифровой логики. Она направлена на упрощение логического выражения, представляющего собой комбинацию логических операторов «и», «или» и «не». Одним из ключевых методов минимизации булевых функций является метод Квайна-МакКласки, который базируется на использовании алгебры логики и таблиц истинности. Этот метод позволяет найти наименьшую формулу, которая описывает данную функцию, и занимает центральное место в изучении булевых функций.

# Список сокращений

ДНФ - дизъюнктивная нормальная форма.

# Булевая функция

Булева функция (или логическая функция, или функция а́лгебры ло́гики) от *n* аргументов — в дискретной математике — отображение *Bn* → *B*, где *B* = {0,1} — булево множество. Элементы булева множества {1, 0} обычно интерпретируют как логические значения «истинно» и «ложно», хотя в общем случае они рассматриваются как формальные символы, не несущие определённого смысла. Неотрицательное целое число *n*, обозначающее количество аргументов, называется арностью или местностью функции, в случае *n* = 0 булева функция превращается в булеву константу. Элементы декартова произведения (*n*-я прямая степень) *Bn* называют булевыми векторами. Множество всех булевых функций от любого числа аргументов часто обозначается *P*2, а от *n* аргументов — *P*2(*n*). Переменные, принимающие значения из булева множества, называются булевыми переменными. Булевы функции названы по фамилии математика Джорджа Буля.

## Основные сведения об булевой функции

Каждая булева функция арности *n* полностью определяется заданием своих значений на своей области определения, то есть на всех булевых векторах длины *n*. Число таких векторов равно 2*n*. Поскольку на каждом векторе булева функция может принимать значение либо 0, либо 1, то количество всех *n*-арных булевых функций равно 2(2*n*). Поэтому в этом разделе рассматриваются только простейшие и важнейшие булевы функции.

Практически все булевы функции низших арностей (0, 1, 2 и 3) получили исторически сложившиеся имена. Если значение функции не зависит от одной из переменных (то есть, по сути, для любых двух булевых векторов, отличающихся лишь в значении этой переменной, значение функции на них совпадает), то эта переменная, не играя никакого «значения», называется фиктивной.

## Представление булевых функций

Теорема Поста открывает путь к представлению булевых функций синтаксическим способом, который в ряде случаев оказывается намного удобнее, чем таблицы истинности. Отправной точкой здесь служит нахождение некоторой полной системы функций. Тогда каждая булева функция может быть представлена некоторым термом в сигнатуре

Σ, который в данном случае называют также формулой. Относительно выбранной системы функций полезно знать ответы на следующие вопросы:

* Как построить по данной функции представляющую её формулу?
* Как проверить, что две разные формулы эквивалентны, то есть задают одну и ту же функцию?
	+ В частности: существует ли способ приведения произвольной формулы к эквивалентной ей *канонической* форме такой, что две формулы эквивалентны тогда и только тогда, когда их канонические формы совпадают?
* Как по данной функции построить представляющую её формулу с теми или иными заданными свойствами (например, наименьшего размера), и возможно ли это?

Положительные ответы на эти и другие вопросы существенно увеличивают прикладное значение выбранной системы функций.

## Дизъюнктивная нормальная форма (ДНФ)

Простой конъюнкцией или *к*онъюнктом называется конъюнкция некоторого конечного набора переменных или их отрицаний, причём каждая переменная встречается не более одного раза. Дизъюнктивной нормальной формой или ДНФ называется дизъюнкция простых конъюнкций. Элементарная конъюнкция

* **правильная**, если каждая переменная входит в неё не более одного раза (включая отрицание);
* **полная**, если каждая переменная (или её отрицание) входит в неё ровно 1 раз;
* **монотонная**, если она не содержит отрицаний переменных.

Совершенной дизъюнктивной нормальной формой или СДНФ относительно некоторого заданного конечного набора переменных называется такая ДНФ, у которой в каждую конъюнкцию входят все переменные данного набора, причём в одном и том же порядке.

## Конъюнктивная нормальная форма (КНФ)

Конъюнктивная нормальная форма (КНФ) определяется двойственно к ДНФ. *Простой дизъюнкцией* или *дизъюнктом* называется дизъюнкция одной или нескольких переменных или их отрицаний, причём каждая переменная входит в неё не более одного раза. КНФ — это конъюнкция простых дизъюнкций.

*Совершенной конъюнктивной нормальной формой* (СКНФ), относительно некоторого заданного конечного набора переменных, называется такая КНФ, у которой в каждую дизъюнкцию входят все переменные данного набора, причём в одном и том же порядке. Поскольку (С)КНФ и (С)ДНФ взаимодвойственны, свойства (С)КНФ повторяют все свойства (С)ДНФ, грубо говоря, «с точностью до наоборот».

# Минимизация булевой функции

Значение минимизации булевой функции состоит в уменьшении количества логических элементов, необходимых для ее реализации, а, следовательно, и в уменьшении стоимости и сложности системы, в которой эта функция используется. Кроме того, минимизация булевых функций помогает увеличить быстродействие такой системы, уменьшить потребление энергии и улучшить надежность ее работы. Интерес к кратчайшим минимальный ДНФ основа на их оптимальности (по длине и рангу соответственно), которая положительно проявляется, по крайней мере, в следующих двух случаях. Во-первых, с оптимальным ДНФ легче оперировать, то есть вычислять значение функции, строить матрицу Грея и подставлять в другие формулы. Во-вторых, оптимальные ДНФ более предпочтительны для построения по ним схем из логических элементов: дизъюнкторов, конъюнкторов и инверторов.

**Пример.** Рассмотрим мажоритарную функцию.



Нарисуем схемы по совершенной и кратчайшей ДНФ. Логические элементы изобразим в виде соответственно обозначенных прямоугольников.



Схема, построенная по кратчайшей ДНФ (справа), оказались проще: она содержит меньше элементов.

Интерес к сокращенной ДНФ вызван тем, что она является промежуточной при построении кратчайших, минимальных и безызбыточных ДНФ.

Безызбыточные ДНФ интересны как сами по себе, так как часто оказываются близкими к оптимальным, так и тем, что среди них находятся все минимальные и простые кратчайшие ДНФ.

**Определение.** Минимизировать булеву функцию это значит построить ее кратчайшую или минимальную ДНФ, или все кратчайшие, или все минимальные ДНФ (в постановке задачи уточняется дополнительно).

Рассмотрим двухэтапный подход к минимизации булевой функции, основанных на теоремах о кратчайшей и минимальных ДНФ, утверждающих, что все минимальные и хотя бы одна из кратчайших ДНФ состоят из простых импликант:

**Первый этап:** найдем все простые импликанты функции, то есть конъюнкции ее сокращенной ДНФ.

**Второй этап:** из сокращенной ДНФ выделим конъюнкции искомых ДНФ.

# Метод Куайна-Мак-Класки

Метод Куайна-Мак\_класки - это табличный метод минимизации булевых функций, предложенный Уиллардом Куайном и усовершенствованный Эдвардом Мак-Класки. Представляет собой попытку избавиться от недостатков метода Куайна.

**Метод Куайна.** Чтобы получить сокращенную ДНФ булевой функции из ее совершенной ДНФ, надо выполнить всевозможные неполные склеивания соседних конъюнкций, а затем всевозможные поглощения конъюнкий.

Опираясь на теорему Куайна, Эдвард Мак-Класки сформулировал алгоритм, который организует построение сокращенной ДНФ более эффективно, чем предложено в теореме Куайна.

Во-первых, конъюнкции в алгоритме представляются булевыми и троичными векторами, что делает вычисление более простыми и более приспособленными для компьютерной реализации.

Во-вторых, соседние векторы отличаются по весу (числу единичных компонентов) ровно на единицу. Поэтому в алгоритме не проверяется на возможность склеивания векторы, чьи веса равны или отличаются более чем на единицу.

В-третьих, соседние вектора поглощаются результатом их склеивания. Поэтому отметка склеиваемых соседей (но не вычеркивание их, так как один и тот же вектор может быть соседом нескольким векторам) позволяет свести поглощение к выписыванию не отмеченных векторов.

## Алгоритм Куайна-Мак-Класки

Начало. Задана совершенная ДНФ булевой функции.

Шаг 1. Построить список всех точек функции (булевых векторов) и упорядочить их по убыванию числа единиц - веса.

Шаг 2. Разбить список на подмножества (классы) векторов одинакового веса. Обозначив через Сi класс векторов веса i.

Шаг 3. Выполнить неполные склеивания всех соседних векторов классов Сi и Сi+1, i = 0, 1, …, n - 1. Учавствующие в склеивании векторов (α и β) отметим, а полученные векторы (**γ**) занесем в новый список и приведем в нем подобые.

Шаг 4. Если новый список векторов не пуст, возвратимся с ним на шаг 2 (заметим, что троичные векторы списка оказываются уже упорядоченные по числу единиц).

Конец. Выписан из всех списков неотмеченные векторы, получим множество всех максимальных интервалов, оно задает сокращенную ДНФ исходной функции.

**Пример.** Продемонстрируем выполнение алгоритма, для наглядности сопровождая матрицами Грея.

Начало. Задача совершенная ДНФ булевой функции:



Шаг 1, 2. Совершенную ДНФ представляем списком точек, упорядочиваем их по весу и разбиваем на классы.



Шаг 3. Выполняя неполные склеивания векторов из классов С2 и С3, а затем С3 и С4, и отмечая в упорядоченном списке 0 склеиваем векторы символом \*, получаем список 1 - список интервалов, состоящий из двух точек. Справа в новом списке указаны номера векторов - участников склеивания. Интервалы списка изображены на двух матрицах Грея. 

Шаг 4. Список 1 не пуст, поэтому идем на шаг 2 (так как склеивания производились в строгом порядке “сверху вниз”, векторы в новом списке уже упорядочены по весу).

Шаг 2, 3. Разбиваем полученный список на классы С2 и С3. Выполняя склеивание векторов из классов С2 и С3, получаем список интервалов, состоящих из четырех точек (список 2). Приводим в нем подобные.



Шаг 4. Список 2 не пуст, но дальнейшее склеивание невозможно, поэтому список 3 окажется пустым, идем на конец.

Конец. Выписываем из всех списков неотмеченные векторы. Они задают сокращенную ДНФ:



# Особенности метода Куайна-Мак-Класки.

Специфика метода Куайна — Мак-Класки по сравнению с методом Куайна в сокращении количества попарных сравнений на предмет их склеивания. Сокращение достигается за счёт исходного разбиения термов на группы с равным количеством единиц (нулей). Это позволяет исключить сравнения, заведомо не дающие склеивания.

Несмотря на большую возможность практического применения чем у карт Карно когда речь идёт о более чем четыре переменных, метод Куайна—Мак-Класки тоже имеет ограничения области применения, так как время работы метода растёт экспоненциально с увеличением входных данных. Можно показать, что для функции от *n* переменных верхняя граница количества основных импликант 3*n*/*n*. Если *n* = 32 их может быть больше чем 6.5 \* 1015. См. также Трансвычисленная задача.

Функции с большим количеством переменных должны быть минимизированы с помощью потенциально не оптимального эвристического алгоритма. На сегодня эвристический алгоритм минимизации [эспрессо](https://ru.wikipedia.org/w/index.php?title=%D0%AD%D1%81%D0%BF%D1%80%D0%B5%D1%81%D1%81%D0%BE_(%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0)&action=edit&redlink=1) является фактическим мировым стандартом.

# Применение минимизации к схемотехнике

Принцип минимизации булевых функций широко применяется в схемотехнике для оптимизации работы цифровых схем. Минимизация позволяет сократить количество логических элементов, а следовательно, уменьшить размер и сложность схемы, а также повысить ее производительность.

Один из основных методов минимизации – метод Карно, который позволяет представить булевую функцию в виде таблицы истинности и затем выделить поддиаграммы, имеющие одинаковые значения. После этого можно упростить функцию, заменяя каждую поддиаграмму на единственный максимальный произведение сумм.

После минимизации булевой функции, полученные результаты могут быть использованы для создания логической схемы. На практике, минимизацию часто применяют при проектировании процессоров, микроконтроллеров и других цифровых устройств.

#

# Преимущества применения минимизации

Преимуществами являются:

Уменьшение количества логических элементов;

* Снижение сложности схемы;
* Экономия затрат на производство и монтаж;
* Более надежная работа устройства;
* Увеличение скорости работы цифровой схемы;
* Сокращение потребления энергии;
* Улучшение общей производительности устройства.

Применение минимизации к схемотехнике является эффективным и востребованным подходом, который позволяет создавать более компактные, быстрые и надежные цифровые устройства.

# Реализация метода на языке программирования Python

import re

from itertools import product

def find\_core(x: dict) -> set:

 res = []

 for i in x:

 if len(x[i]) == 1:

 res.append(x[i][0])

 return set(res)

def implicate\_to\_var(x: str) -> str:

 res = []

 for i in range(len(x)):

 if x[i] == '0':

 res.append(f"!x{i + 1}")

 elif x[i] == '1':

 res.append(f"x{i + 1}")

 return "\*".join(res)

def flatten(x: dict):

 flattened\_items = []

 for i in x:

 flattened\_items.extend(x[i])

 return flattened\_items

def create\_mask(x, y):

 m = list(x)

 mismatches = 0

 for i in range(len(x)):

 if x[i] != y[i]:

 m[i] = "\*"

 mismatches += 1

 return "".join(m) if mismatches == 1 else None

def find\_outcore(chart, core):

 for i in core:

 for j in chart.copy():

 if i in chart[j]:

 del chart[j]

 return chart

def cover(input\_lst):

 combinations = list(product(\*(input\_lst.values())))

 combination = min([set(combination) for combination in combinations], key=len)

 return combination

def main():

 minterms = "0011000001111110" # input("F=")

 size = len(bin(len(minterms) - 1)[2:])

 minterms = sorted([bin(int(i))[2:].zfill(size) for i in range(len(minterms)) if minterms[i] == "1"])

 print(minterms)

 groups = {}

 primal\_implicates = set()

 for minterm in minterms:

 if groups.get(minterm.count("1")):

 groups[minterm.count("1")].append(minterm)

 else:

 groups[minterm.count("1")] = [minterm]

 while True:

 tmp = groups.copy()

 groups = {}

 stop = True

 masked = set()

 for group\_n in tmp.keys():

 if group\_n == list(tmp.keys())[-1]:

 continue

 for j in tmp[group\_n]:

 for k in tmp[group\_n + 1]:

 res = create\_mask(j, k)

 if res:

 if groups.get(group\_n):

 groups[group\_n].append(res) if res not in groups[group\_n] else None

 else:

 groups[group\_n] = [res]

 stop = False

 masked.add(j)

 masked.add(k)

 unmasked = set(flatten(tmp)).difference(masked)

 primal\_implicates = primal\_implicates.union(unmasked)

 if stop:

 # print(', '.join(primal\_implicates))

 break

 chart = {}

 for implicate in primal\_implicates:

 for minterm in minterms:

 if re.match(implicate.replace("\*", "."), minterm):

 if chart.get(minterm):

 chart[minterm].append(implicate)

 else:

 chart[minterm] = [implicate]

 core = find\_core(chart)

 unused = find\_outcore(chart, core)

 result = sorted(list(core.union(cover(unused))))

 if len(result) == 1 and result[0].count('\*') == len(result[0]):

 print("No solution")

 return

 print(' ∨ '.join(implicate\_to\_var(i) for i in result))

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

 main()

# Заключение

В предложенной научной работе мы изучили метод минимизации булевых функций Куйна-Мак-Класки. Привели примеры, которые описывают его алгоритм, а также воссоздали предложенный метод в программе на языке программирования Python.

# Список литературы

1. [https://ru.wikipedia.org/wiki/Метод\_Куайна\_—\_Мак-Класки](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%9A%D1%83%D0%B0%D0%B9%D0%BD%D0%B0_%E2%80%94_%D0%9C%D0%B0%D0%BA-%D0%9A%D0%BB%D0%B0%D1%81%D0%BA%D0%B8)
2. <https://ido.tsu.ru/iop_res/bulevfunc/text/g11_1_1.html>
3. <https://ido.tsu.ru/iop_res/bulevfunc/text/g11.html#:~:text=%D0%9C%D0%B8%D0%BD%D0%B8%D0%BC%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D1%82%D1%8C%20%D0%B1%D1%83%D0%BB%D0%B5%D0%B2%D1%83%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8E%20%D1%8D%D1%82%D0%BE%20%D0%B7%D0%BD%D0%B0%D1%87%D0%B8%D1%82,(%D0%BF%D0%BE%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8%20%D1%83%D1%82%D0%BE%D1%87%D0%BD%D1%8F%D0%B5%D1%82%D1%81%D1%8F%20%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE)>.
4. https://obzorposudy.ru/polezno/cto-takoe-minimizaciya-bulevoi-funkcii