[**Список определений**](#_xt5inyhfjy9j) **1**

[**Список теорем**](#_ineponnyaayn) **7**

[**Список типов задач**](#_wa0nl4rhidhv) **12**

### **Список определений**

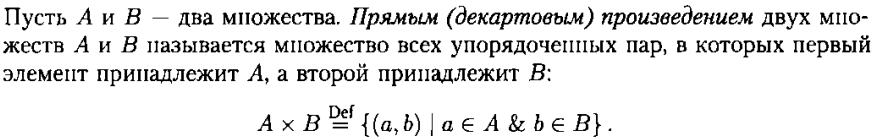
1. *Объединение множеств*

**

1. *Пересечение множеств*

**

1. *Декартово произведение множеств*

**

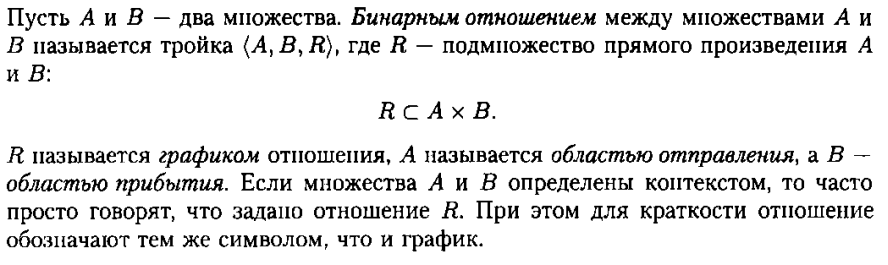
1. *Разность множеств*

**

1. *Дополнение множества до универсума*

, x ∈ U

1. *Бинарное отношение*

**

1. *Транзитивность*

**

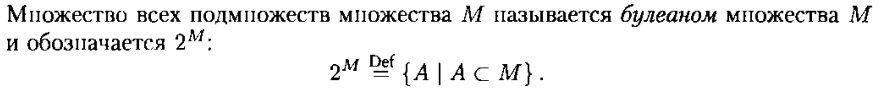
1. *Рефлексивность*

**

1. *Симметричность*

**

1. *Булеан*

**

1. *Объединение отношений*

рассмотрим отношения R и S над множеством A x B

R и S это подмножества A x B

R U S = T это отношение T, полученное объединением R и S в смысле множеств

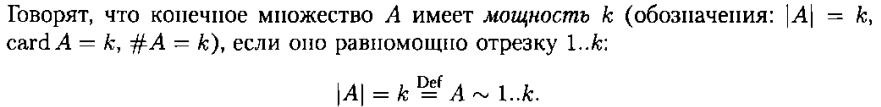
1. *Пересечение отношений*

рассмотрим отношения R и S над множеством A x B

R и S это подмножества A x B

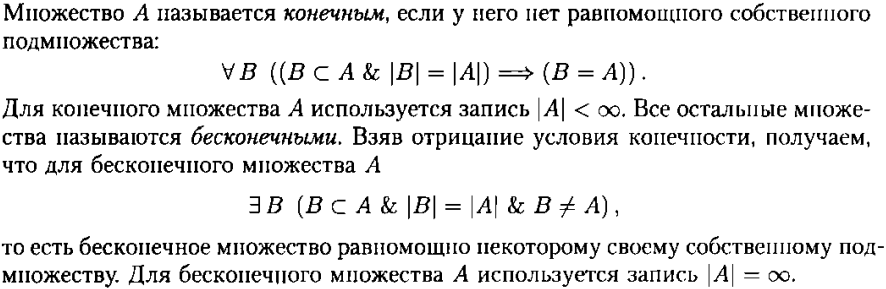
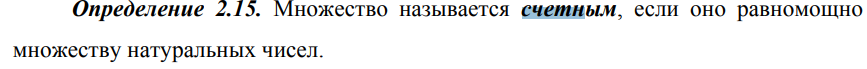
R ∩ S = T это отношение T, полученное пересечением R и S в смысле множеств

1. *Мощность множества*

**

Множество называется **бесконечным**, если его мощность не меньше мощности множества натуральных чисел

1. *Конечные, счетные, несчетные множества*

* **
* **

(бесконечное множество, элементы которого возможно пронумеровать натуральными числами.)

* Несчетное множество — бесконечное множество, не являющееся счетным

1. *Отношение частичного порядка*

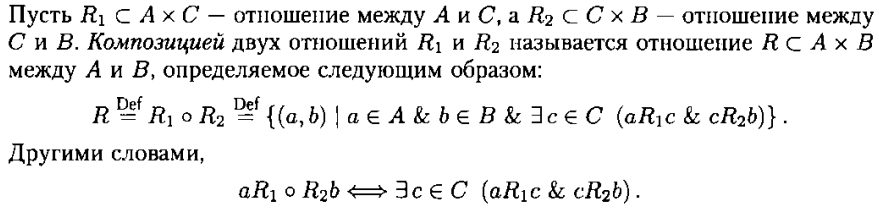
Бинарное отношение R на множестве X называется **отношением частичного порядка**, если оно обладает следующими свойствами:

* Рефлексивность 
* Антисимметричность 
* Транзитивность 

1. *Отношение линейного порядка*

Бинарное отношение R на множестве X называется **отношением линейного порядка**, если оно является отношением частичного порядка и обладает следующим свойством: 

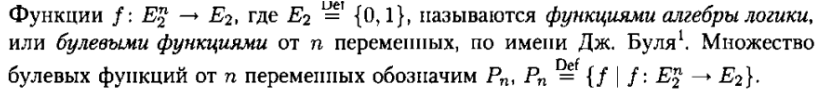
1. *Композиция отношений*

**

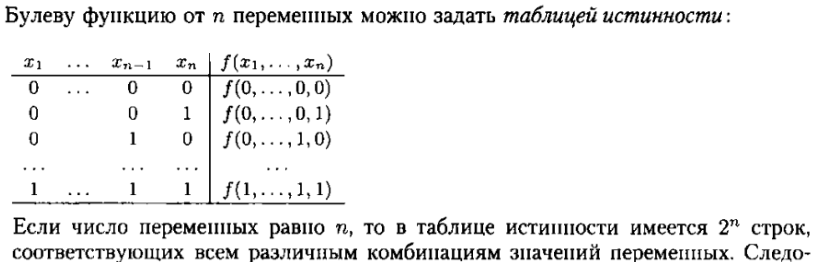
1. *Булево пространство*

пространство, состоящее из двух элементов {0;1}. математическая структура, основанная на двух значениях — обычно обозначаемых как "истина" (1) и "ложь" (0) — которая включает в себя операции, позволяющие выполнять логические вычисления.

1. *Булева функция*



Это функция, у которой несколько аргументов, принимающих значения 0 или 1, и сама функция тоже принимает значение 0 или 1

1. *Таблица истинности*

способ представления значений логического выражения для всех возможных комбинаций его переменных

1. *Бинарная функция*

*Булева функция от 2 аргументов*

1. *Тернарная функция*

*Булева функция от 3 аргументов*

1. *n-арная функция*

*Булева функция от n аргументов*

1. *Эквивалентные формулы*

Говорят, что формулы А и В логически эквивалентны (обозначается A ⇔ В или просто А = В), если они являются логическими следствиями друг друга.

1. *КНФ*

форма представления булевой функции, где формула записана как конъюнкция дизъюнктов

1. *ДНФ*

форма представления булевой функции, где формула записана как дизъюнкция конъюнктов

1. *Дизъюнкт*

дизъюнкция одной или нескольких переменных или их отрицаний, где каждая переменная встречается не более одного раза

1. *Конъюнкт*

конъюнкция одной или нескольких переменных или их отрицаний, где каждая переменная встречается не более одного раза (пример: x1 ∧ x2 ∧ ¬x3)

1. *СДНФ*

Такая ДНФ, в которой каждое слагаемое содержит все переменные или их отрицание

1. *СКНФ*

Такая КНФ, в которой каждый множитель содержит все переменные или их отрицание

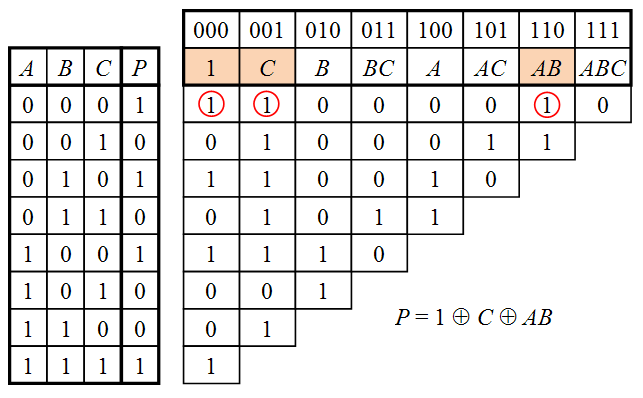
1. *Полином Жегалкина*

Полином с коэффициентами вида 0 и 1, где в качестве произведения берётся конъюнкция, а в качестве сложения — сложение по модулю {0,1, ∧, +}

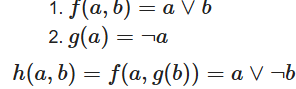
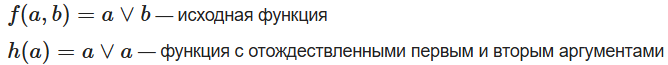
1. *Метод треугольника нахождения полинома Жегалкина*

Метод треугольника позволяет преобразовать таблицу истинности в полином Жегалкина путем построения вспомогательной треугольной таблицы в соответствии со следующими правилами:

1. Строится полная таблица истинности, в которой строки идут в порядке возрастания двоичных кодов от 000…00 до 111…11.
2. Строится вспомогательная треугольная таблица, в которой первый столбец совпадает со столбцом значений функции в таблице истинности.
3. Ячейка в каждом последующем столбце получается путем сложения по модулю 2 двух ячеек предыдущего столбца — стоящей в той же строке и строкой ниже.
4. Столбцы вспомогательной таблицы нумеруются двоичными кодами в том же порядке, что и строки таблицы истинности.
5. Каждому двоичному коду ставится в соответствие один из членов полинома Жегалкина в зависимости от позиций кода, в которых стоят единицы. Например, ячейке 111 соответствует член ABC, ячейке 101 — член AC, ячейке 010 — член B, ячейке 000 — член 1 и т.д.
6. Если в верхней строке какого-либо столбца стоит единица, то соответствующий член присутствует в полиноме Жегалкина.

**

1. *Суперпозиция*

Суперпозиция функций — это функция, полученная из некоторого множества функций путем подстановки одной функции в другую () или отождествления переменных ()

1. *Замкнутые классы*

Если любая булева функция, являющаяся суперпозицией функций некоторого множества, принадлежит этому множеству, то такое множество называют замкнутым

1. *Предикат*

Предикат — это математическое высказывание с переменными, которое может принимать различные истинностные значения в зависимости от значений переменных

1. *Разрешимость предиката*

Предикат называется разрешимым, если существует некоторый дедуктивный вывод, позволяющий определить, является ли предикат тождественно истинным или нет.

1. *Аксиома*

Аксиома — формула некоторой логической системы, считающаяся истинной без вывода

1. *Гипотеза*

Гипотеза — формула некоторой логической системы, для которой не получен вывод о ее разрешимости

1. *Теорема*

Теорема — формула некоторой логической системы, для которой получен вывод о ее разрешимости и она является тождественно истинной

1. *Теория*

Теория — совокупность аксиом, гипотез, теорем и правил вывода некоторой логической системы

1. *Аксиома исключенного третьего*

Аксиома исключенного третьего — аксиома гильбертовой математической логики: A ∨ ¬A. Истинно либо утверждение некоторого факта, либо его отрицание. Третьего не дано

1. *Интуиционистская логика*

Интуиционистская логика — формальная логика, в которой присутствуют все аксиомы классической логики за исключением аксиомы исключенного третьего

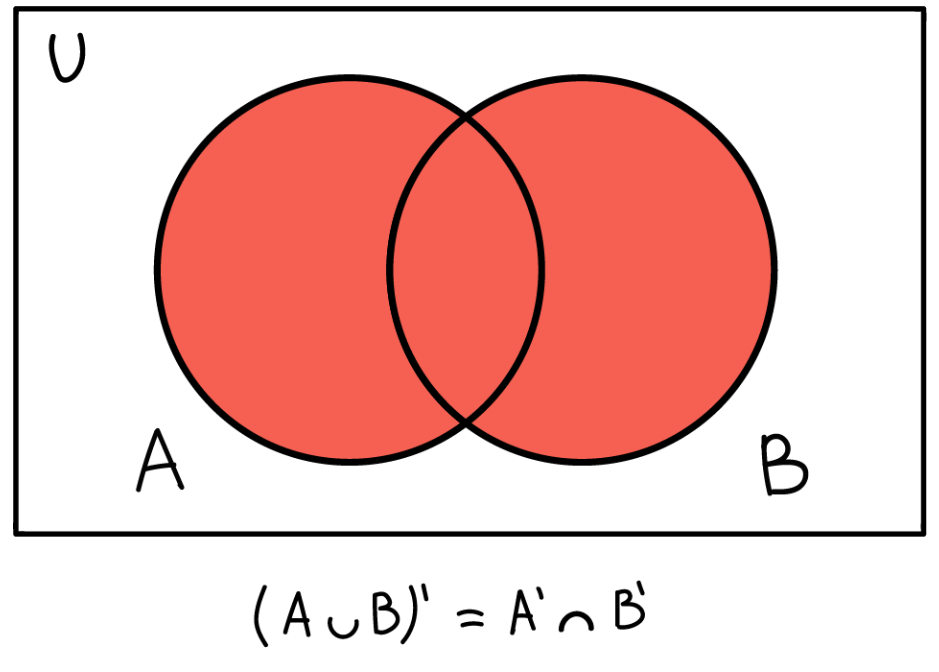
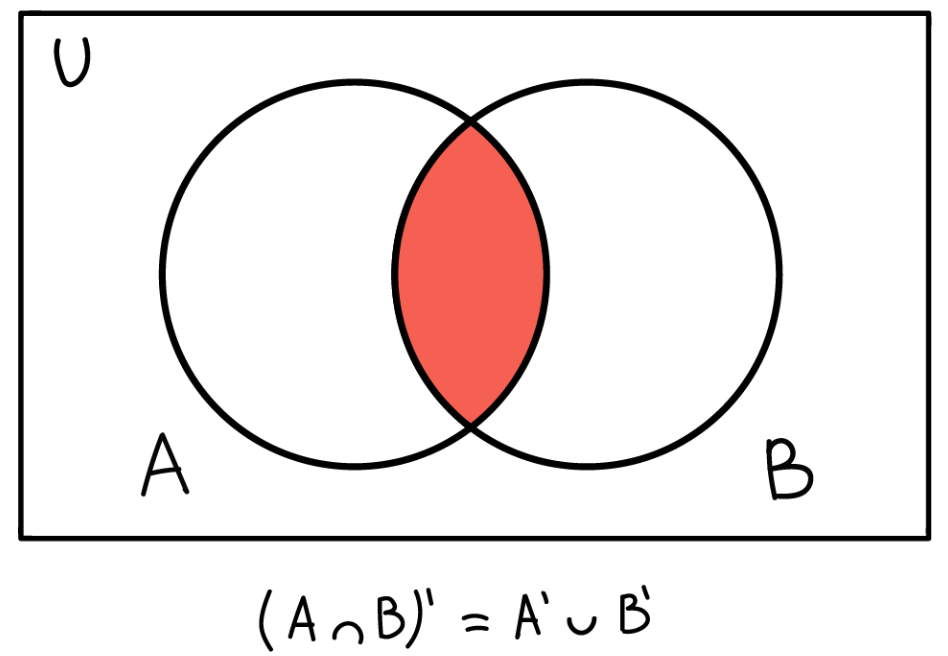
1. *Линейная логика*

Линейная (афинная) логика — формальная логика, в которой на аксиомы, гипотезы и теоремы могут быть введены точные (верхние) ограничения по количеству использований. В линейной логике ресурсы рассматриваются как ограниченные и не могут использоваться более одного раза, если это не оговорено отдельно.

### **Список теорем**

1. *Закон де Моргана для множеств*

Возможны доказательства через круги Эйлера <https://ru.hexlet.io/courses/set-theory/lessons/de_morgan/theory_unit>

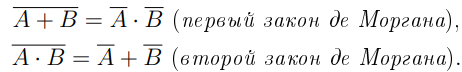


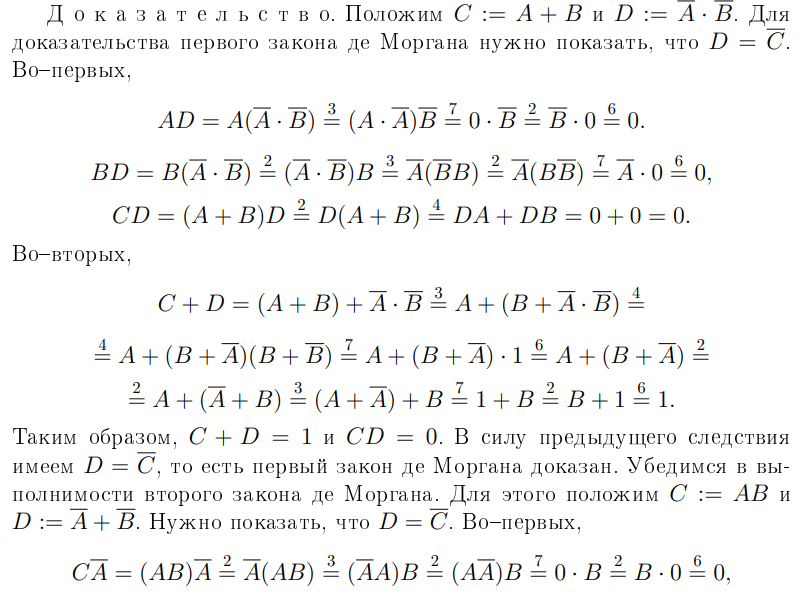
1. *Количество n-арных булевых функций*

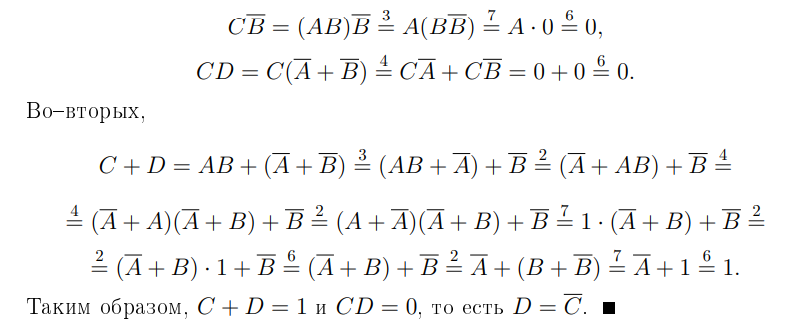
Арность - количество аргументов функции. Каждая булева функция арности n полностью определяется заданием своих значений на своей области определения, то есть на всех булевых векторах длины n. Число таких векторов равно 2^n (формула размещения с повторениями). Поскольку на каждом векторе булева функция может принимать значение либо 0, либо 1, то количество всех *n*-арных булевых функций равно .

1. *Закон де Моргана для булевых функций*

Достаточно доказать эквивалентность формул







1. *Законы поглощения*

Достаточно доказать эквивалентность формул

* х ∧ у v х = х
* (х v у) ∧ х = х

Доказательства:

* x ∨ xy = x×1 ∨ xy = x(1∨y) = x
* x(x∨y) = xx∨xy = x∨xy = x

1. *Алгоритм нахождения СКНФ*

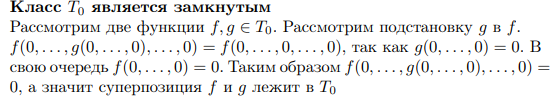
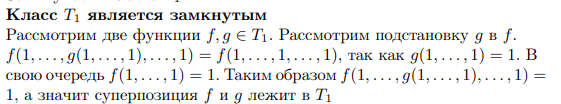
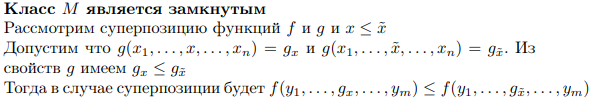
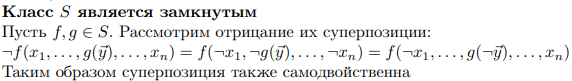
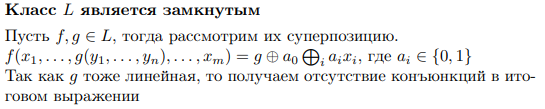
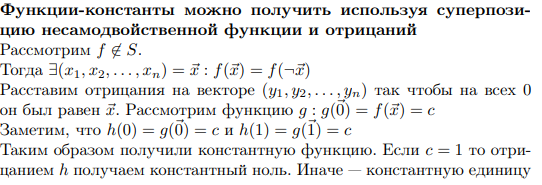
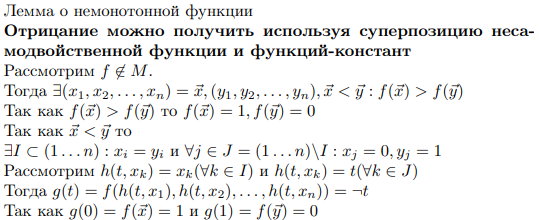
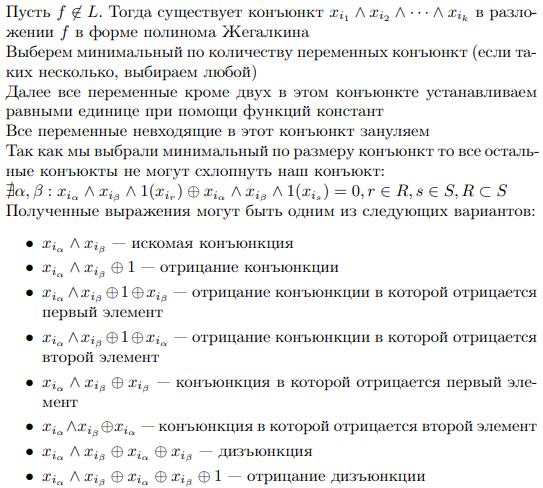
ВНИМАНИЕ! необходимо не только сформулировать алгоритм но и доказать его корректность (эквивалентность полученной СКНФ и исходной формулы)

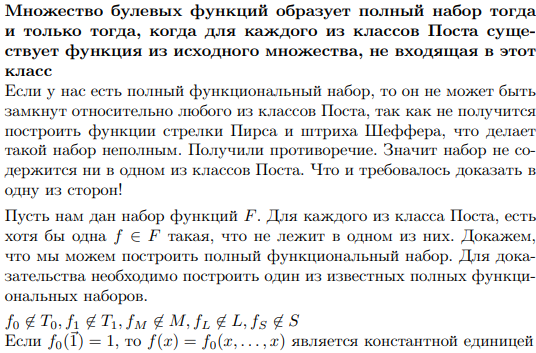
* Метод преобразований
* Метод через таблицу истинности:
  + Построить таблицу истинности функции
  + Если значение строки = 0, смотрим, какая комбинация дизъюнкций х, у, z и их отрицаний даст 0
  + Все полученные комбинации перемножаем конъюнкцией
* Проверить эквивалентность можно по таблице истинности
* СКНФ равна 0 там, где функция равна 0
* Каждая дизъюнкция в СКНФ соответствует набору, где функция равна 0. Если функция равна 0 на каком-либо наборе, то хотя бы одна дизъюнкция в СКНФ будет равна 0, что сделает всю конъюнкцию равной 0
* СКНФ единственна по определению, ведь дизъюнкция равна 1, если все перестановки равны 1

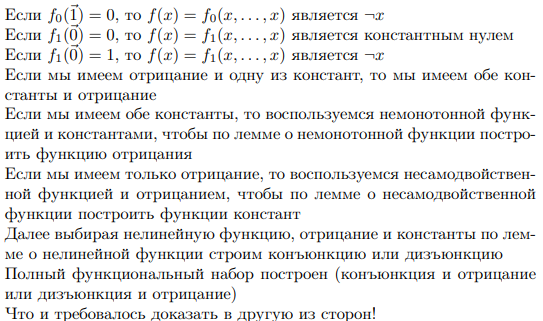
1. *Алгоритм нахождения СДНФ*

ВНИМАНИЕ! необходимо не только сформулировать алгоритм но и доказать его корректность (эквивалентность полученной СДНФ и исходной формулы)

* Метод преобразований
* Метод через таблицу истинности:
  + Построить таблицу истинности функции
  + Если значение строки = 1, смотрим, какая комбинация конъюнкций х, у, z и их отрицаний даст 1
  + Все полученные комбинации складываем дизъюнкцией
* Проверить эквивалентность можно по таблице истинности. В случае, если это затруднительно, применяем теорему: Для любой булевой функции , не равной тождественному нулю, существует СДНФ, ее задающая.
* СДНФ покрывает все случаи, где функция равна 1
* Каждая конъюнкция в СДНФ соответствует одному набору, на котором функция равна 1. Следовательно, на этих наборах СДНФ совпадает с функцией.
* СДНФ равна 0 там, где функция равна 0
* Если на наборе переменных функция равна 0, то ни одна конъюнкция в СДНФ не будет истинной. В результате значение СДНФ на этом наборе тоже будет равно 0.

1. *Замкнутость класса T0*
2. *Замкнутость класса T1*
3. *Замкнутость класса M*
4. *Замкнутость класса S*
5. *Замкнутость класса L*
6. *Лемма о несамодвойственной функции*
7. *Лемма о немонотонной функции*
8. *Лемма о нелинейной функции*
9. *Теорема Поста*

**

**

### **Список типов задач**

1. Поиск нормальных форм формул булевых алгебр
2. Поиск классов Поста формул булевых алгебр
3. Подсчет алгебраических выражений конечных множеств
4. Нахождение свойств бинарных отношений