

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по Индивидуальному домашнему заданию**  
**по дисциплине «Математическая логика и теория алгоритмов»**  
**Тема: Использование бинарных решающих диаграмм для**  
**решения логических задач**

Студент гр. 3341

Мальцев К.Л.

Преподаватель

Шошмина И.В.

Санкт-Петербург

2025

## Цель работы

Цель работы – освоить использование бинарных решающих диаграмм (BDD) для решения логических задач.

## Задание

Курсовая работа выполняется индивидуально, в соответствии со своим вариантом. Задача определяется следующими параметрами.

- Имеется  $N = 9$  объектов с  $M = 4$  свойствами. Свойства принимают  $N$  различных значений.
- Имеются индивидуальные соседские отношения, склейка, обязательное количество ограничений  $n_1, n_2, n_3, n_4$ .
- Дополнительно определена дельта  $C$  (используется для вычисления варианта).

Курсовую работу можно выполнять на разных уровнях:

1. простой — без склейки;
2. продвинутый — выполняется вариант без склейки и со склейкой;
3. сложный — то же, что на продвинутом, плюс творческая модификация.

В отчете необходимо объяснить результаты. В зависимости от выбранного вами уровня выполнения и качества исполнения, оценивается курсовая работа. Количество баллов в зависимости от уровня сложности выполнения курсовой работы приведено в Moodle. Опишем последовательность действий при варианте без склейки:

1. Определите соседей всех объектов по своему варианту без склейки.
2. Модифицируйте ограничения типа 3 и 4 задачи Эйнштейна с учетом соседских отношений вашего варианта. Задайте  $n_3$  ограничений подобных отношениям типа 3 и  $n_4$  ограничений подобных отношениям типа 4 в соответствии со своим заданием.
3. Придумайте  $n_1$  ограничений 1-ого типа и  $n_2$  ограничений 2-ого типа.

4. Постройте программу для решения вашей задачи с использованием BDD.

5. Найдите все возможные решения.

6. Придумайте физическую интерпретацию задачи.

7. Выпишите интерпретацию ограничений на естественном языке.

8. Оставьте около 12-16 решений задачи. Этого можно добиться, добавив дополнительные ограничения любого типа или модифицировав текущие. Если у вас получилось меньше решений, то часть ограничений типа 1-2 можно убрать.

9. Выберите одно из решений и запишите его интерпретацию на естественном языке.

Если вы выбираете продвинутый уровень, то повторяете предыдущую последовательность для случая, когда склейка есть. Интерпретация сохраняется одинаковой для всех уровней.

Требования к реализации, если вы сдаете на компьютере лаборатории: реализация на языке C++ с использованием библиотеки BuDDy. Компилировать под MS Visual Studio.

Для выполнения задания необходимо написать программу Реализация должна быть масштабируемой, т. е. изменение параметров N, M не должны приводить к изменению основной части программы.

Вариант

$v_1 = 19; v_2 = 1; v_3 = 2.$

## Выполнение работы

Класс Solution:

*Solution()*: инициализирует BDD, и заполняет массив p.

*solve()*: применяет все ограничения, считает и выводит решения

Классы ограничений (наследуются от *Limitation*):

*Lim1*: устанавливает, что свойство k1 конкретного объекта i1 имеет значение j1.

*Lim2*: устанавливает соответствие между двумя свойствами какого-либо объекта.

*Lim3*: устанавливает ограничения типа объект, у которого свойство k1 имеет значение j1, расположен слева|справа от объекта, у которого свойство k2 имеет значение j2.

*Lim4*: устанавливает ограничение расположения объектов – расположение «рядом» - дизъюнкция ограничений «слева» и «справа» типа 3

*Lim5*: у двух различных объектов значения любого параметра (свойства) не совпадают.

*Lim6*: устанавливает ограничение, что параметры принимают значения только из заданных множеств.

(значение свойств должно быть меньше N)

Все классы имеют метод *apply()*, который добавляет своё ограничение в BDD.

Соседские отношения (реализация *Lim3*):

Определение отношений left/right представлены на рис.1:

|    |   |   |   |
|----|---|---|---|
|    |   |   |   |
| 19 | * | 0 | * |
|    |   |   |   |

Рис. 1 Соседские отношения

У класса переопределен *operator()*:

```
bdd operator() ()
{
    bdd ret_val = bddtrue;
    for (int p = 0; p < _n; ++p)
    {
        bdd a = _p[_k1][p][_j1];
        bdd b = toSide(p);
        ret_val &= !a & !b | a & b;
    }
    return ret_val;
}
```

В методе *toSide()* вычисляются координаты соседнего объекта.

Определим, что объекты проиндексированы следующим образом:

```
0 1 2
3 4 5
6 7 8
```

Будем переводить индекс объекта в двумерный индекс *i,j* и вычислять соседа используя новую индексацию. Далее переведем индекс обратно в одномерный:

```
// индекс текущего объекта - p; индекс соседа - q
int p_y = p / sqrt_n;
int p_x = p % sqrt_n;
// ...
int q_y = (Side::kLeft == _side) ? p_y : p_y;
int q_x = (Side::kLeft == _side) ? p_x - 1 : p_x + 1;
// ...
int q = q_y * sqrt_n + q_x;
```

Реализация склейки:

```
// using positive modulo
switch (_splice)
{
    case Splice::kHorizontal: q_x = mod(q_x, sqrt_n); break;
    case Splice::kVertical: q_y = mod(q_y, sqrt_n); break;
    default: break;
}
```

При использовании исходного количества ограничений (7 4 2 6)

Получим: Solutions: 138240

```
k maltsev@mint:~/repo/LETI/MLTA/cw$ g++ main.cpp print.cpp -lbdd
k maltsev@mint:~/repo/LETI/MLTA/cw$ ./a.out
Solutions: 138240
k maltsev@mint:~/repo/LETI/MLTA/cw$
```

Рис. 2 Запуск программы

Введем дополнительные ограничения (+2, +11, +1, 0)

Получим: Solutions: 16

Интерпретация на естественном языке:

У Оушена есть 9 друзей, которые проиндексированы от 0 до 8 и стоят по сетке 3x3. У каждого друга есть 4 свойства (любимый цвет, напиток, жанр музыки и игра), которые также проиндексированы от 0 до 8 и могут быть представлены в виде таблицы:

| Значение | Цвет       | Напиток  | Жанр        | Игра      |
|----------|------------|----------|-------------|-----------|
| 0        | Красный    | Кофе     | Рок         | Шахматы   |
| 1        | Синий      | Чай      | Поп         | Карты     |
| 2        | Зеленый    | Сок      | Джаз        | Домино    |
| 3        | Желтый     | Вода     | Классика    | Мафия     |
| 4        | Фиолетовый | Пиво     | Хип-хоп     | Крокодил  |
| 5        | Оранжевый  | Вино     | Электроника | Монополия |
| 6        | Черный     | Коктейль | Метал       | Нарды     |
| 7        | Белый      | Лимонад  | Кантри      | Скрэббл   |
| 8        | Розовый    | Молоко   | Регги       | Маджонг   |

Рассмотрим одно из решений:

```
0: 0 0 0 6
1: 1 1 1 7
2: 2 2 2 8
3: 3 3 3 3
4: 4 4 4 4
5: 5 5 5 5
6: 6 6 6 2
7: 7 7 7 1
8: 8 8 8 0
```

Интерпретация:

| Друг | Цвет       | Напиток  | Жанр        | Игра      |
|------|------------|----------|-------------|-----------|
| 0    | Красный    | Кофе     | Рок         | Нарды     |
| 1    | Синий      | Чай      | Поп         | Скрэббл   |
| 2    | Зеленый    | Сок      | Джаз        | Маджонг   |
| 3    | Желтый     | Вода     | Классика    | Мафия     |
| 4    | Фиолетовый | Пиво     | Хип-хоп     | Крокодил  |
| 5    | Оранжевый  | Вино     | Электроника | Монополия |
| 6    | Черный     | Коктейль | Метал       | Домино    |
| 7    | Белый      | Лимонад  | Кантри      | Карты     |
| 8    | Розовый    | Молоко   | Регги       | Шахматы   |

## **Вывод**

Реализована программная система для решения логической задачи размещения 9 объектов с 4 свойствами с использованием BDD.

**ПРИЛОЖЕНИЕ А**  
**ИСХОДНЫЙ КОД ПРОГРАММЫ**

<https://github.com/kmaltsev2006/LETI/tree/MLTA/MLTA/cw>