Санкт-Петербургский Государственный Политехнический Университет
Кафедра «Прикладная математика»

Модуль для обучения решению задач по курсу «Дискретная математика»

Диссертация на соискание степени магистра

Выполнила студентка гр.63601/2: Лейзарович Е.В.
Научный руководитель: ст.преп. Суетова Н.Б.

Санкт-Петербург
2014

Электронное обучение

- Системы управления учебными материалами
  - Moodle
  - BlackBoard
  - Sakai

- Проверка знаний – тесты
  - Верно/неверно
  - Нет контроля хода решения

Зарегистрированные сайты	64,076
Стран	235
Курсов	7,494,446
Пользователей	71,216,446
Преподавателей	1,153,582
Записей на курсы	103,588,083
Сообщений в форумах	131,475,128
Ресуров	68,502,657
Вопросов для тестов	213,832,250

http://moodle.org 29.06.2014

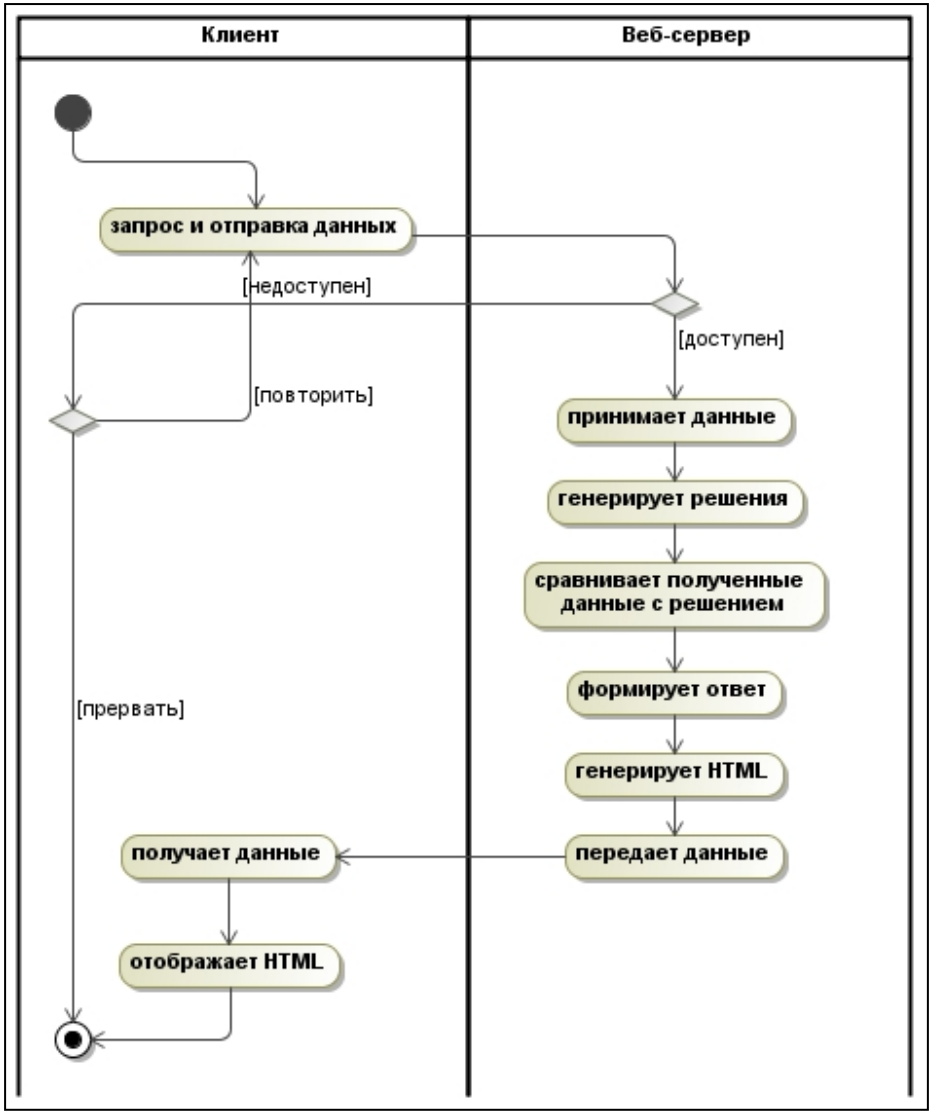
Постановка задачи

Цель: практика решения задач на построение сокращенных бинарных диаграмм решений

Разработать:

обучающее веб-приложение

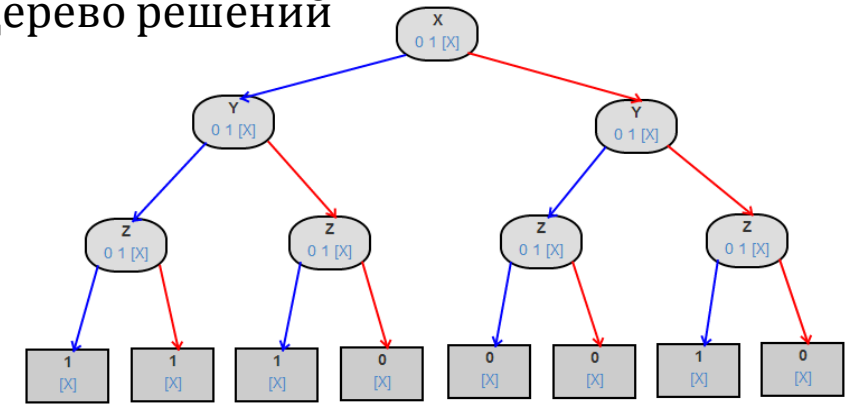
- Возможность графического ввода диаграммы
- Автоматическая проверка решения
  - Алгоритм проверки корректности с указанием оптимальная диаграмма или нет
  - Алгоритм указания ошибки



Бинарные диаграммы решений (1)

- C. Lee, 1959; S. Akers, 1978
  - Сокращенные бинарные диаграммы решений
- R. Bryant, 1986
  - Эффективные алгоритмы

Дерево решений



БДР

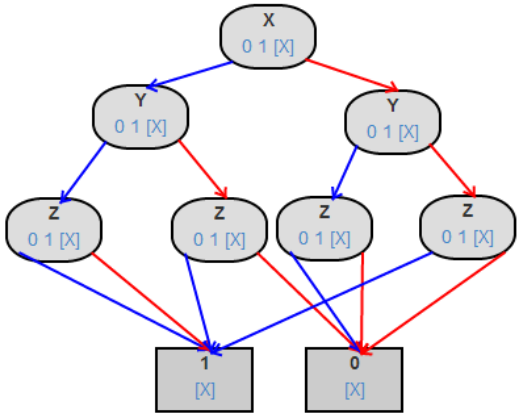


Таблица истинности

↓

Дерево решений

↓

Бинарная диаграмма решений (БДР)

↓

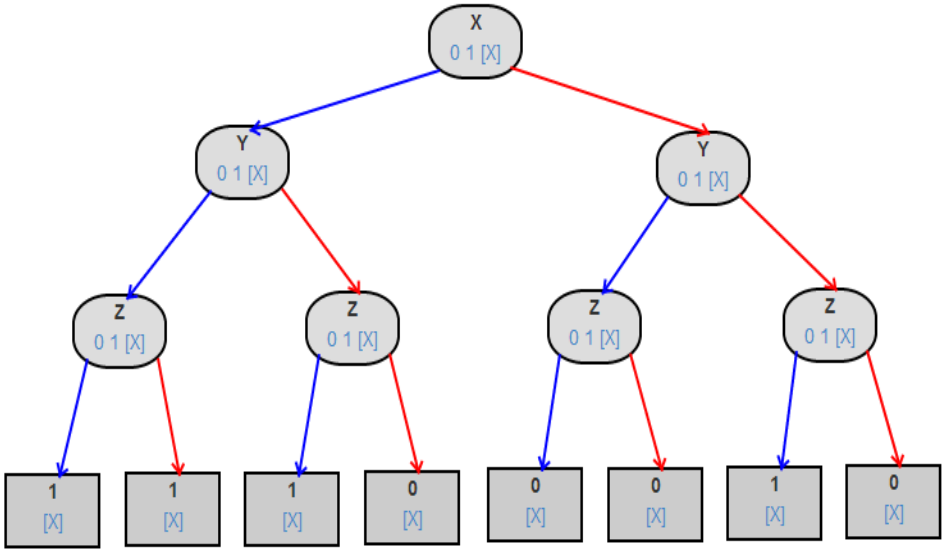
Упорядоченная БДР (УБДР)

↓

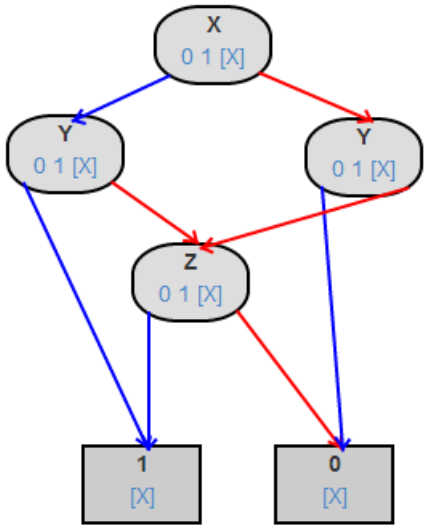
Сокращенная УБДР

Бинарные диаграммы решений (2)

- Упорядоченная БДР (OBDD) – задан линейный порядок на множестве переменных
- Сокращенная УБДР (ROBDD) – нет кратных ребер, нет изоморфных поддиаграмм
  - Правило удаления, правило слияния



Дерево решений

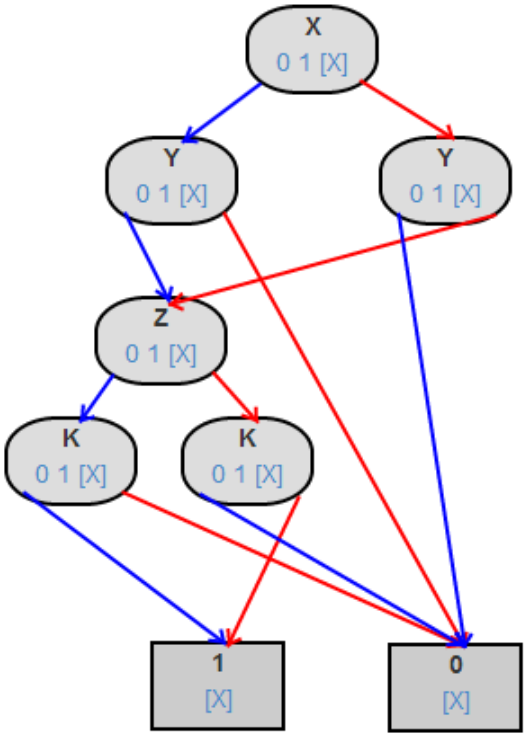


Сокращенная УБДР

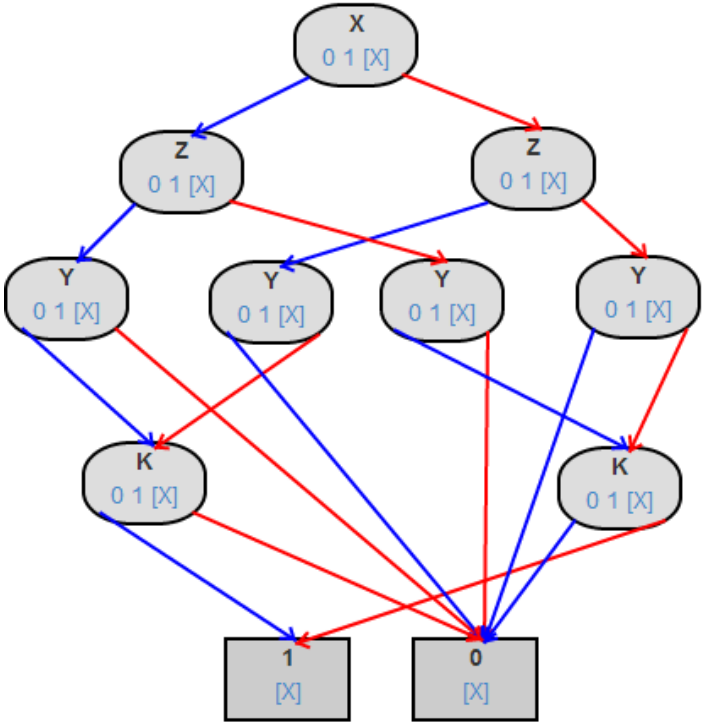
Сокращенные диаграммы решений

- Каноническое представление функции при фиксированном порядке переменных
- Зависит от порядка переменных

$$F = (x \Leftrightarrow y) \wedge (z \Leftrightarrow k)$$



$x < y < z < k$



$x < z < y < k$

Построение БДР по формуле

Проблема нахождения оптимального порядка переменных NP-трудна → требуется построить БДР для всех порядков переменных

- Порядок переменных фиксирован $x_1 < \ldots < x_n$
- Добавление строк в таблицу

  $T : u \to (i, l, h) \quad \mathrm{var}(u) = i, low(u) = l, high(u) = h$

u	var	low	$high$
0	5		
1	5		
2	4	1	0
3	4	0	1
4	3	2	3
5	2	4	0
6	2	0	4
7	1	5	6

- Функция if-then-else (ite)

  $ite(x, y, z) =$ if x then y else z
- БДР – минимальное представление функции в базисе {ite, 0,1} при фиксированном порядке переменных
- Рекурсивное применение разложения Шеннона

$$f = x_i \cdot f|_{x_i=1} + \overline{x}_i \cdot f|_{x_i=0}, \text{ т.е. } ite(x_i, f|_{x_i=1}, f|_{x_i=0})$$

Алгоритм проверки корректности

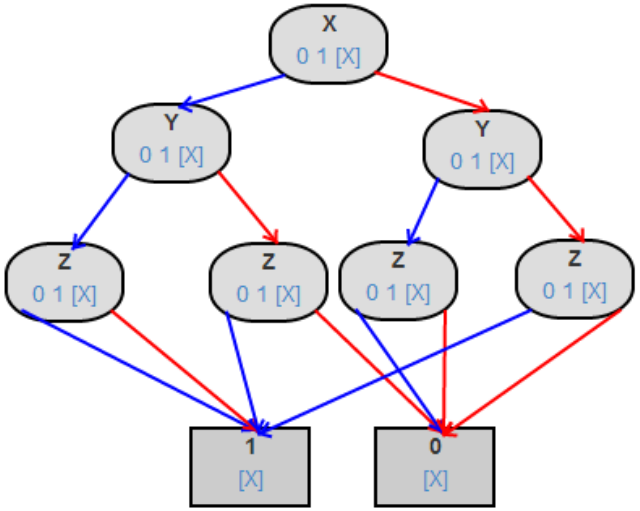
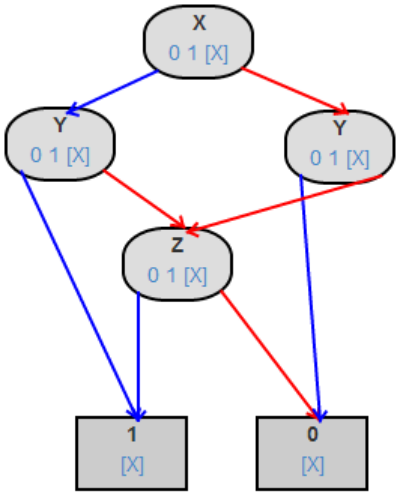


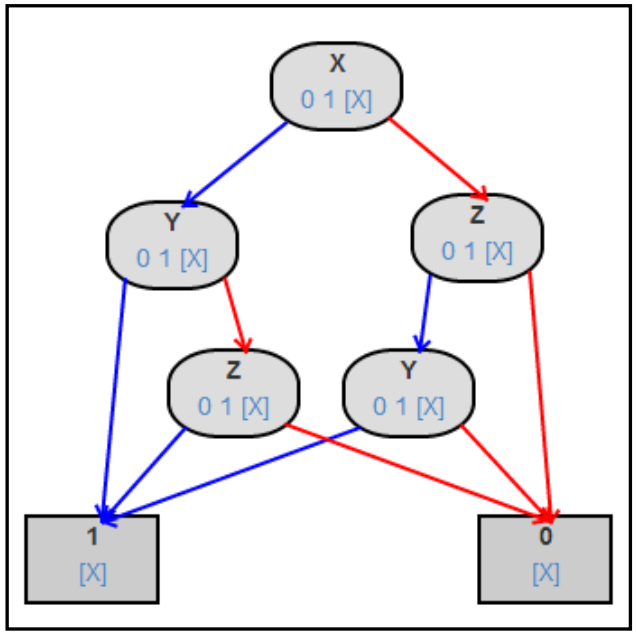
Диаграмма пользователя



Эталон

Шаг	Проверка
1	Наличие узлов, ребер
2	Наличие терминальной вершины 0 и 1
3	Наличие у нетерминальных вершин ровно двух ребер
4	Наличие одной корневой вершины
5	Ацикличность
6	Упорядоченность
7	Реализует заданную функцию
8	Сокращенная
9	Оптимальная

Проверка упорядоченности



- Построенная диаграмма должна быть упорядочена
  - при прохождении пути из корня в терминальную вершину каждая переменная встречается не более одного раза
  - всех путях переменные встречаются в одном и том же порядке
- Наличие/отсутствие несущественных переменных

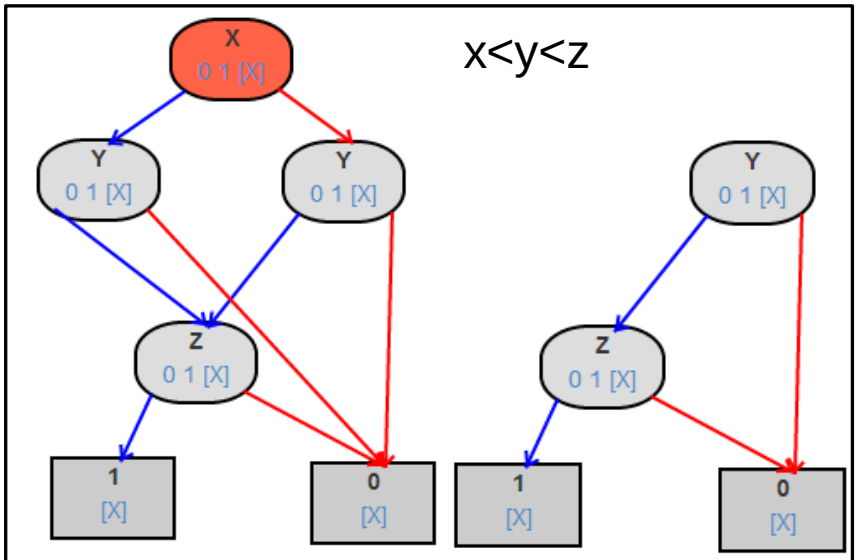


Диаграмма пользователя Эталон

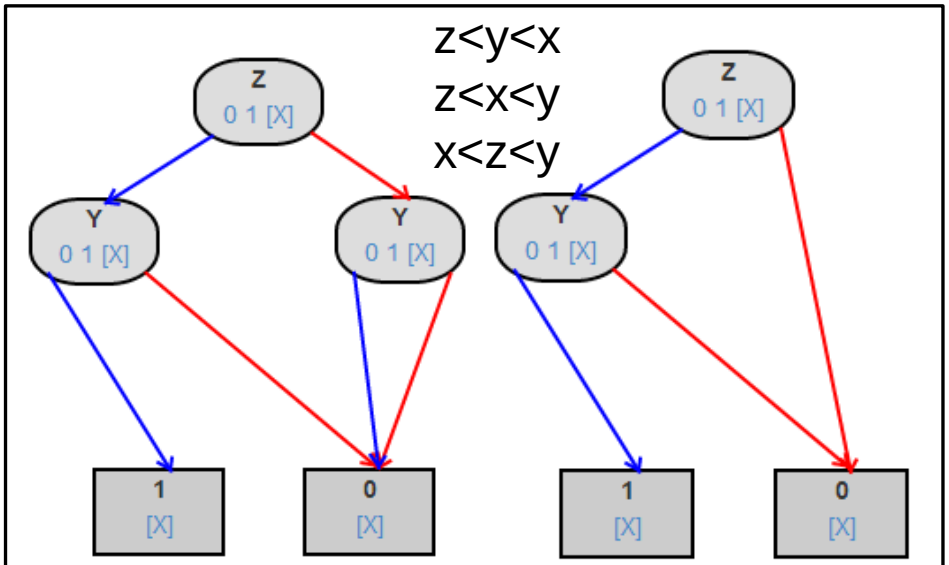


Диаграмма пользователя Эталон

Алгоритм проверки корректности

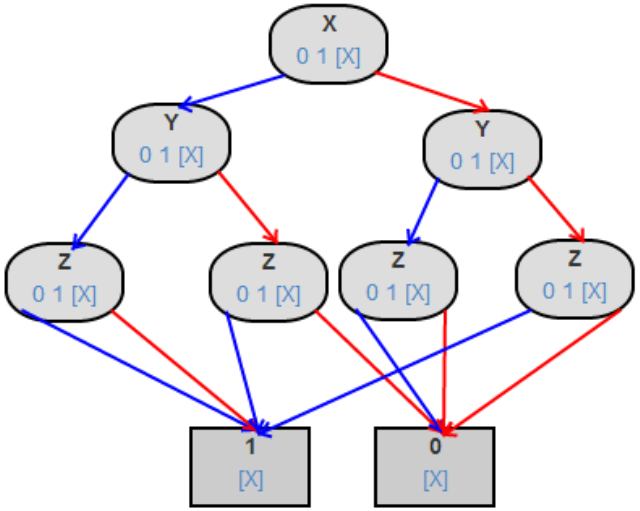
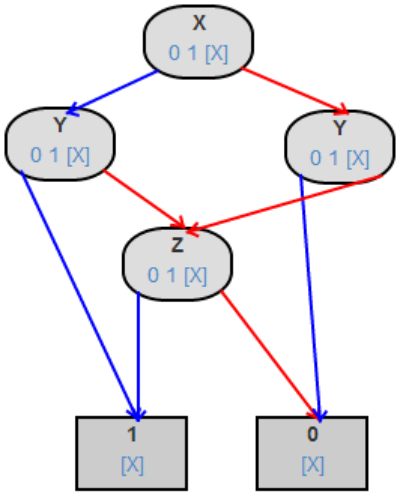


Диаграмма пользователя



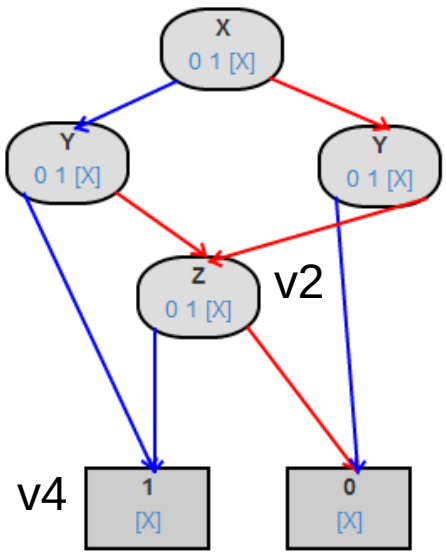
Эталон

Шаг	Проверка
1	Наличие узлов, ребер
2	Наличие терминальной вершины 0 и 1
3	Наличие у нетерминальных вершин ровно двух ребер
4	Наличие одной корневой вершины
5	Ацикличность
6	Упорядоченность
7	Реализует заданную функцию
8	Сокращенная
9	Оптимальная

Алгоритм указания ошибки (1)

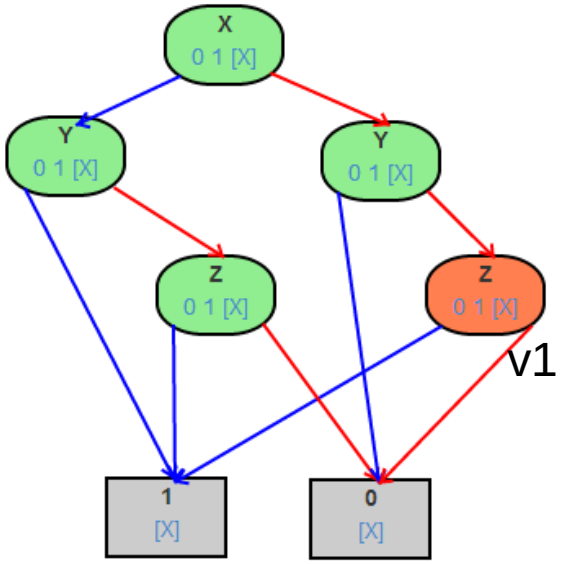
G1 (граф пользователя), G2 (эталонный граф) – корневые ориентированные ациклические помеченные графы

- Дано: указатели T1, T2 на корень G1 и G2 соответственно
- Требуется: найти максимальный общий подграф G1 и G2, корень которого соответствует корню G1
- Метод: обход в глубину со сравнением статуса «посещения» и сравнением меток



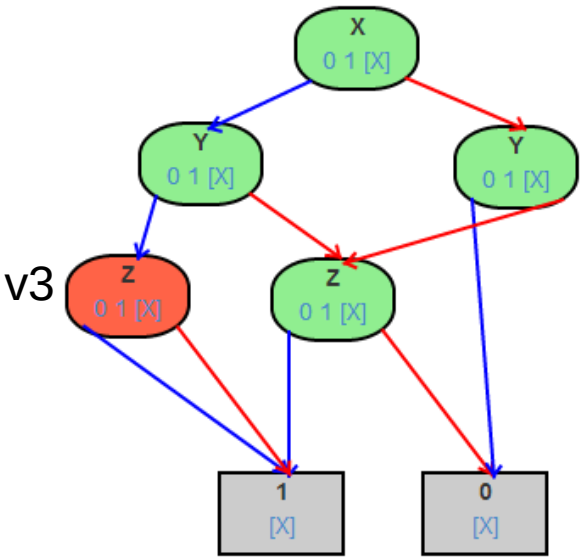
Эталон G2

v1.notVisited && v2.isVisited



Граф G1 (вариант 1)

v3.var != v4.var



Граф G1 (вариант 2)

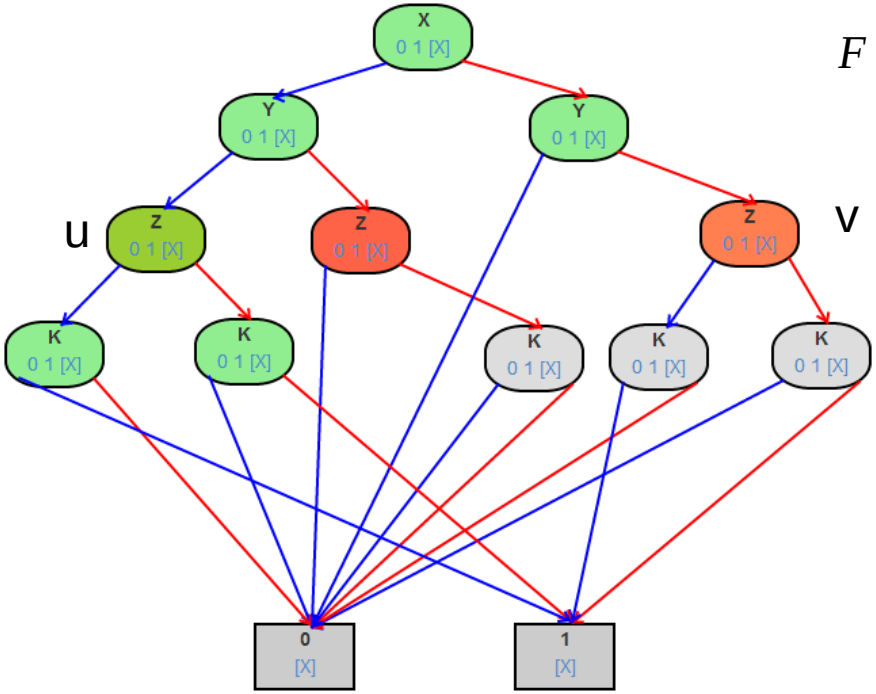
Алгоритм указания ошибки (2)

G1 (граф пользователя)

- Дано: вершина *v* границы без кратных ребер ∈ G1, граф G1
- Найти: корень подграфа, изоморфного подграфу с корнем в v
- Метод:

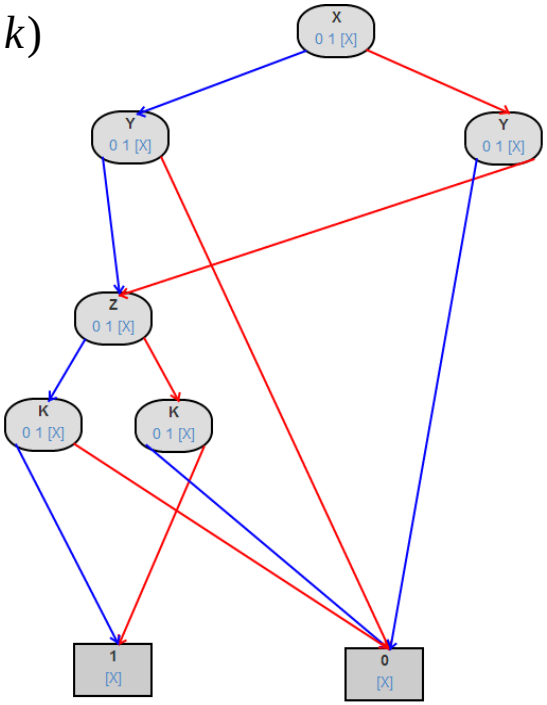
  Для каждой вершины *u* с такой же меткой ∈ G1

  поиск максимального общего подграфа(u ,v) мощности равной мощности подграфа с корнем в *v*;



Граф пользователя G1

$$F = (x \Leftrightarrow y) \wedge (z \Leftrightarrow k)$$

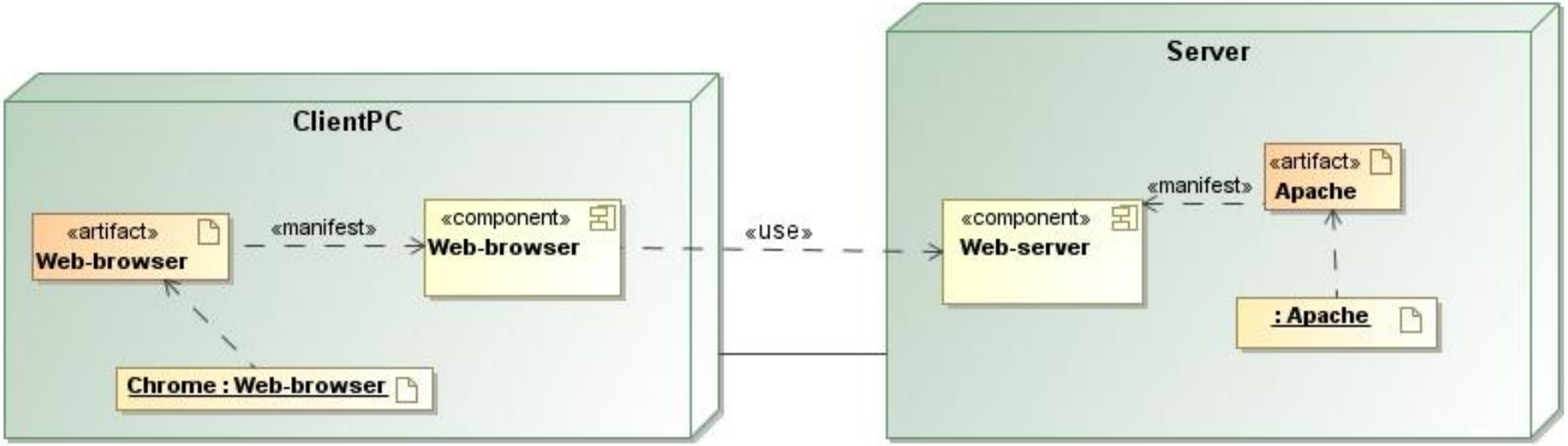


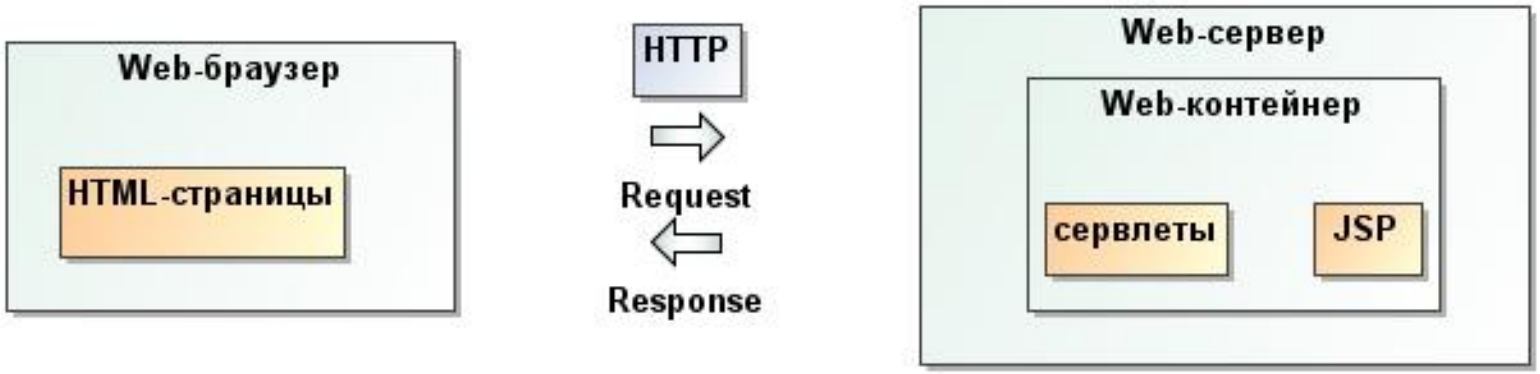
Эталон

Псевдокод алгоритма

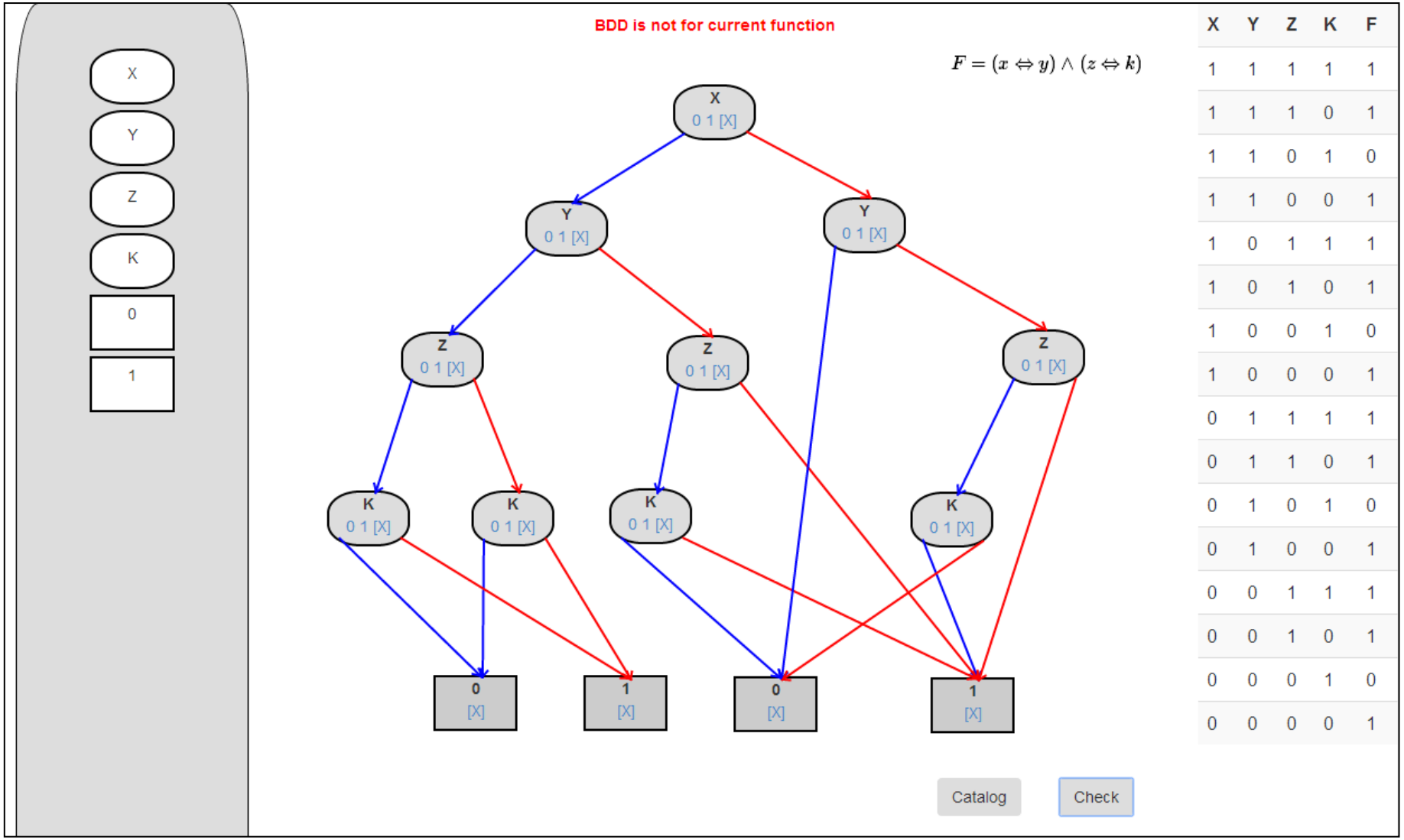
```
similarityDFS(T1 ,T2)
1   Если T1=T2  и они терминальные, то
2           return;
3   Если T1.notVisited && T2.isVisited, то
4           Отметить T1 неверной;
5           Если (T1.low != T1.high)
6                   Найти изоморфный подграф в G1;
7                   return;
8  Если T1.var != T2.var, то
9           строки 4-7;
10 similarityDFS(T1.low, T2.low);
11 similarityDFS(T1.high, T2.high);
12 Отметить T1 посещенной;
13 Отметить T2 посещенной;
14 Если T1 не отмечена неверной, то
15          Добавить T1 в общий подграф;
```

Архитектура приложения





Пример



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

Итоги

- Реализовано интерактивное обучающее веб-приложение, в рамках которого имеются
  - модуль, позволяющий преподавателю в текстовом виде задать библиотеку булевых функций и затем отображающий эти функции в графическом представлении
  - модуль, позволяющий студенту самостоятельно рисовать диаграммы выбранной функции и проверяющий их на корректность
  - модуль, позволяющий в формально правильной, но возможно неоптимальной диаграмме, построенной студентом, показать ему, где он допустил ошибку
- Реализован алгоритм, который позволяет найти ошибку пользователя в построении диаграммы и указать, с какого шага ему следует продолжить вносить изменения в свою диаграмму

Спасибо за внимание!

Псевдокод алгоритма

```
similarityDFS(T1 ,T2)
1   Если T1=T2  и они терминальные, то
2           return;
3   Если T1.notVisited && T2.isVisited, то
4           Отметить T1 неверной;
5           Если (T1.low != T1.high)
6                   Найти изоморфный подграф в G1;
7                   return;
8  Если T1.var != T2.var, то
9           строки 4-7;
10 similarityDFS(T1.low, T2.low);
11 similarityDFS(T1.high, T2.high);
12 Отметить T1 посещенной;
13 Отметить T2 посещенной;
14 Если T1 не отмечена неверной, то
15          Добавить T1 в общий подграф;
```

Оценка сложности: $O(N^3)$, где N – количество вершин в графе G1.

Обход в глубину O(N), поиск наиб изоморфного подграфа O(N), количество вершин для поиска O(N).

Дальнейшее развитие

- Интеграция в СДО Moodle
- Добавление других задач
- Сбор статистики и улучшение алгоритма

Применения BDD

- Решение логических и комбинаторных задач
  - Компрессия изображений
  - Информационный поиск
  - Задача о расстановке ферзей
- Анализ и синтез логических схем
- Верификация

Библиотеки и приложения

- On-line визуализаторы
  - Строят BDD
  - Отображают конечную BDD
  - Только основные операции
  - Только по введенной формуле (не отображают таблицу истинности)
- Библиотеки
  - BuDDy
    - перестраивает порядок используя эвристики
    - C++
  - CuDD
    - C++
  - JavaBDD
    - Java-интерфейс к BuDDy

Сложность

- BDD для большинства функций имеет линейную сложность
- Некоторые классы функций имеют экспоненциальную сложность при любом порядке переменных
  - Функция, выдающая средний бит результата произведения A x B n-разрядных переменных A и B