Санкт-Петербургский государственный университет

Фундаментальная информатика и информационные технологии
Профиль Информационные технологии

Глазырин Павел Александрович

Выделение общих с точностью до имен переменных подформул из базы данных для решения задачи «Конъюнктивный булевский запрос»

Дипломная работа

Научный руководитель:
д. ф.-м. н., профессор Косовская Т. М.

Рецензент:
к. ф.-м. н., доцент Петухова Н. Д.

Санкт-Петербург
2018

Оглавление

Введение

В данной работе предлагается подход для выделения общих с точностью до имен переменных подформул из базы данных для решения задачи «Конъюнктивный булевский запрос» и описывается его реализация.

При использовании баз данных время выполнения запроса – очень важная характеристика, особенно если учитывать объемы данных, хранящихся в современных базах. «Конъюнктивный булевский запрос» – NP-полная задача, касающаяся баз данных. Далее приведена ее формулировка, как она дана в [2]:

Конъюнктивный булевский запрос

ДАНО. Домен D, набор отношений $R = \{R_1, R_2, ..., R_m\}$(где R_i есть подмножество множества D^{d_i}), конъюнктивный булевский запрос Q над R и D, т.е. запрос вида

$$(\exists y_1, y_2, ..., y_m)(A_1 \wedge A_2 \wedge ... \wedge A_r)$$

где A_i имеет вид $R_l(u_1, u_2, ..., u_{d_l})$, а переменные u принадлежат множеству $(\{y_1, y_2, ..., y_l\} \cup D)$.

ВОПРОС. Верно ли, что Q истинен при интерпретации его как утверждения об R и D?

Данную задачу можно переформулировать следующим образом:

Конъюнктивный булевский запрос

ДАНО. Множество объектов D, предикаты $p_1, p_2, ...p_m$, задающие отношения на D. Множество истинных на D постоянных атомарных формул $S(D)$. Элементарная конъюнкция Q вида A_1 & A_2 & ... & A_r, где $A_i = p_l(u_1, u_2, ..., u_{k_l})$, u_t – переменные или константы из D.

ВОПРОС. Верно ли, что Q выполняется на D?

$$S(D) \Rightarrow \exists y_1, y_2, ..., y_m \,(A_1 \,\&\, A_2 \,\&\, ... \,\&\, A_r)?$$

В этом виде задача получается весьма схожей с задачей «Выполнимость в конечной интерпретации» [4, 5], которая возникает при распознавании объектов в рамках логико-предметного подхода к распознаванию образов. Данная задача формулируется следующим образом:

Выполнимость в конечной интерпретации

ДАНО. Множество $\omega = \{\omega_1, \omega_2, ..., \omega_t\}$, набор предикатов $\{p_1, p_2, ..., p_n\}$, задающих свойства элементов из ω и отношения между ними, набор $S(\omega)$ истинных постоянных атомарных формул вида $p_i(\overline{\tau})$, где $i = 1, ..., n$, $\tau \subseteq \omega$, бескванторная формула $A(\overline{y})$, представленная в виде дизъюнкции элементарных конъюнкций атомарных формул ($\overline{y} = (y_1, ..., y_a)$ – список предметных переменных, входящих в формулу).
ВОПРОС. Существует ли набор значений для $\overline{y}$ из ω^a, для которого формула $A(\overline{y})$ истинна? Или другими словами

$$S(\omega) \Rightarrow \exists \overline{y} A(\overline{y})?$$

Алгоритм полного перебора всех возможных подстановок, как было доказано в [4], имеет число шагов

$$O(t^m ||S||),$$

где t – число элементов в ω, m – количество переменных в $A(\overline{x})$, $||S||$ – число атомарных формул в описании $S(\omega)$.
Алгоритмы, основанные на построении вывода в исчислении предикатов, имеют число шагов

$$O(\sum_{i:p_i \, is \, in \, A(\overline{x})} s_i^{a_i}),$$

где s_i и a_i – количество вхождений предиката p_i в описание $S(\omega)$ и в формулу $A(\overline{x})$ соответственно. Очевидно, те же оценки будут и у алгоритмов решения задачи «Конъюнктивный булевский запрос».

Важное различие данных задач состоит в следующем:
– база данных, как правило, меняется довольно редко или вообще по-

стоянна, а запросы, наоборот, каждый раз различные ($S(D)$ неизменна, а запрос Q сильно изменчив);
– в распознавании образов множество формул может либо не меняться совсем, либо меняться, но очень редко, а распознаваемые объекты могут появляться различные ($A(\overline{y})$ неизменна, а объект ω и его описание $S(\omega)$ часто меняются).

Для уменьшения числа шагов решения задачи выполнимости существует алгоритм построения многоуровневого описания классов, который будет рассмотрен далее. Однако тот же алгоритм не может быть эффективно использован для решения задачи «Конъюнктивный булевский запрос» из-за различий между данными задачами, в связи с чем предлагается его модификация.

Многоуровневое описание классов

Для уменьшения числа шагов можно построить [5] иерархическое многоуровневое описание классов распознаваемых объектов. Алгоритм его построения существенным образом использует понятие изоморфности элементарных конъюнкций предикатных формул.

Определение. *Две элементарные конъюнкции предикатных формул A и B называются изоморфными, если существует такая элементарная конъюнкция C и такие подстановки $\lambda_{A,C}$ и $\lambda_{B,C}$ переменных из формулы C в формулы A и B соответственно, что результаты этих подстановок $A\lambda_{A,C}$ и $B\lambda_{B,C}$ совпадают с формулой C с точностью до порядка литералов. А подстановки $\lambda_{A,C}$ и $\lambda_{B,C}$ называются унификаторами формул A и B с формулой C.*

Пусть множество всех объектов ω разбито на некоторые классы, и эти классы имеют описания, состоящие из конъюнкций атомарных предикатных формул $A_1(\overline{x}_1), ..., A_K(\overline{x}_K)$. Выделяются изоморфные одной и той же формуле подформулы $P_i^1(\overline{y})$ формул $A_k(\overline{x})$ «небольшой сложности». При этом записывается система эквивалентностей вида $p_i^1(y^1) \leftrightarrow P_i^1(\overline{y})$, где p_i^1 – новые предикаты 1-го уровня, а переменные y^1 – новые переменные 1-го уровня для списков исходных переменных.

Получаем систему предикатов 1-го уровня p_i $(i = 1, ..., n_1)$, которые определяются равносильностями

$$p_i^1(x_i^1) \leftrightarrow P_i^1(\overline{y}_i^1).$$

Заменой в $A_k(\overline{x})$ всех вхождений формул $P_i^1(\overline{y}_i^1)$ на атомарные формулы $p_i^1(x_i^1)$ (при $y_i^1 \subseteq x$) получим новые формулы, которые обозначим $A_k^1(\overline{x}^1)$, где $\overline{x}^1$ – список всех переменных данной формулы, в котором есть некоторые (может быть и все) переменные формулы $A_k(\overline{x})$, так и переменные 1-го уровня, появившиеся в формуле $A_k^1(\overline{x}^1)$. Эти формулы можно рассматривать как описания классов в терминах предикатов исходного (нулевого) и первого уровней.

Описанием объекта 1-го уровня $S^1(\omega)$ назовем множество всех атомарных формул вида $p_i^1(\omega_{ij}^1)$, для которых истинна подформула $P_i^1(\overline{\tau}_{ij}^1)$ при

$\tau^1_{ij} \subset \omega$, а объект первого уровня ω^1_{ij} представляет из себя список исходных объектов $\overline{\tau}^1_{ij}$.

Эту процедуру можно повторить уже для формул $A^1_k(\overline{x}^1)$.

В результате построение предикатов различных уровней и многоуровневого описания классов исходную систему описания классов можно записать с помощью равносильной ей многоуровневой системы:

$$\begin{cases} & \mathrm{A}^L_k(\overline{x}^L) & \\ \mathrm{p}^1_1(x^1_1) & \Leftrightarrow & \mathrm{P}^1_1(\overline{y}^1_1) \\ & \vdots & \\ \mathrm{p}^1_{n_1}(x^1_{n_1}) & \Leftrightarrow & \mathrm{P}^1_{n_1}(\overline{y}^1_{n_1}) \\ & \vdots & \\ \mathrm{p}^l_i(x^l_i) & \Leftrightarrow & \mathrm{P}^l_i(\overline{y}^l_i) \\ & \vdots & \\ \mathrm{p}^L_{n_L}(x^L_{n_L}) & \Leftrightarrow & \mathrm{P}^L_{n_L}(\overline{y}^L_{n_L}) \end{cases}$$

Для того, чтобы число шагов полного перебора уменьшилось при использовании 2-уровневого описания, достаточно, чтобы

$$n_1 \cdot t^r + t^{s_1+n_1} < t^m,$$

где r – наибольшее число переменных в формулах $P^1_i(\overline{y}^1_i)$, n_1 – количество предикатов 1-го уровня, s_1 – число атомарных формул в описании 1-го уровня, m – число переменных в описании классов.

Подобное условие для уменьшения числа шагов работы алгоритмов, основанных на исчислении предикатов, выглядит следующим образом:

$$\sum_{k=1}^{K} s^{1\,a^1_k} + \sum_{j=1}^{n_1} s^{\rho^1_j} < \sum_{k=1}^{K} s^{a_k},$$

где a_k и a^1_k – наибольшее количество атомарных формул в $A_k(\overline{x}_k)$ и $A^1_k(\overline{x}^1_k)$ соответственно, s и s_1 – число атомарных формул в $S(\omega)$ и $S^1(\omega)$ соответственно, ρ^1_j – число атомарных формул в $P^1_i(\overline{y}^1_i)$

В [3] указан алгоритм построения многоуровневого описания классов путем выделения общих пропозициональных подформул.

Определение. *Атомом называется переменная или ее отрицание.*
Определение. *Парой называется конъюнкция двух атомов.*
Определение. *Индекс пары – это количество элементарных конъюнкций в описании класса, для которых данная формула (конъюнкция двух атомов) является подформулой.*
Определение. *Адрес пары – это номер класса и номер элементарной конъюнкции в описании этого класса, в которую эта пара входит как подформула.*
Сам алгоритм выглядит следующим образом:

1. Для каждого класса из каждой элементарной конъюнкции выделяем все возможные пары входящих в нее атомов. При этом для каждой пары запоминается ее адрес и индекс.

2. Берем пару с максимальным индексом (p_{max}). Вводим переменную $(l+1)$-го уровня, определяемую этой парой. (l – максимальный уровень переменных входящих в пару).

3. Для каждого адреса p_{max} найдем пары, имеющие с ней пересечения (за счет одинакового адреса). Объединение таких пар записываем в виде новой пары с переменной нового уровня. Сохраняем также полную форму заменяемой части, как объединение полных форм 2-х заменяемых переменных. После этого вычисляем и изменяем количество вхождений результатов замен в описания классов.

4. Повторяем пункты 2 и 3 до тех пор, пока встречаются пары с индексом ≥ 2.

5. Для $l = 1, ..., L$ заменяем в описаниях классов все подформулы, определяемые переменными l-го уровня на эти переменныею.

Предлагаемый мною в дипломной работе алгоритм к выделению таких формул использует идеи данного алгоритма.
Значительное различие состоит в том, что вдобавок к адресу пар так-

же хранятся унификаторы для пары с переменными с парой литералов.

Многоуровневое описание базы данных для решения задачи «Конъюнктивный булевский запрос»

Как было ранее упомянуто, оценки числа шагов проверки логического следования вида $S(D) \Rightarrow \exists y_1, y_2, ..., y_l \, A(y_1, y_2, ..., y_l, d_1, ..., d_r)$ имеют вид

– при полном переборе: $O(t^l||S||)$, где t – количество элементов в базе данных $S(D)$, l – количество переменных в запросе

– при использовании алгоритмов, основанных на исчислении предикатов: $O(\sum_{i:p_i \, is \, in \, A(\overline{x})} s_i^{a_i})$, где s_i и a_i – число вхождений атомарных формул с предикатом p_i в базу данных $S(D)$ и в формулу $A(\overline{x})$ соответственно.

При решении задач искуственного интеллекта правая часть (дизъюнкция элементарных конъюнкций) неизменна и в [6, 7] разработан алгоритм и программа для выделения изоморфных подформул из множества элементарных конъюнкций предикатных формул. В задаче «Конъюнктивный булевский запрос» же дано множество литералов (постоянных атомарных формул), при этом отсутствуют элементарные конъюнкции, и из множества конъюнкций литералов требуется выделить «часто» встречающиеся «короткие» формулы.

Двухуровневая база данных.

1. Пусть среди множества элементов базы данных выделено N подмножеств таких, что эти подмножества попарно не пересекаются, и все конъюнкции элементов i -го подмножества $(i = 1..N)$ изоморфны друг другу и какой-то формуле $P_i^1(\overline{y}_i^1)$. При этом для каждой конъюнкции известны её унификаторы с $P_i^1(\overline{y}_i^1)$.

2. Вводим предикаты 1-го уровня $p_i^1(i = 1..N)$, определяемые равносильностями $p_i^1(y_i^1) \Leftrightarrow P_i^1(\overline{y}_i^1)$, где y_i^1 - переменная первого уровня для списка изначальных переменных $\overline{y}_i^1$.

3. Добавим к содержанию базы данных множество атомарных формул вида $p_i^1(d_i^{1,j})$, где $d_i^{1,j}$ – список констант унификатора для конъ-

юнкции из множества конъюнкций, изоморфных $P_i^1(\overline{y}_i^1)$.

В результате получаем двухуровневую базу данных.
Допустим, что мы смогли построить двухуровневую базу данных, тогда применить ее можно следующим образом:
Обозначим полные формы элементарных конъюнкций, равносильных полученным предикатам 1-го уровня $p_i^1(y_i^1)$, за $P_i^1(\overline{y}_i^1)$.
Получен конъюнктивный булевский запрос $\exists y_1, y_2, ..., y_l\, A(y_1, y_2, ..., y_l, d_1, ..., d_r)$, где $d_1, ..., d_r$ - константы из D.

Для уменьшения числа шагов будем искать в запросе подформулы, изоморфные полученным нами в базе предикатам 1-го уровня. Это можно сделать, проверяя выводимость предикатов из запроса.

1. Проверка логического следования

$$A(y_1, y_2, ..., y_l, d_1, ..., d_r) \Rightarrow \exists \overline{y}_i^1 P_i^1(\overline{y}_i^1)$$

в случае, если оно выполняется, позволит найти все подформулы запроса, изоморфные P_i^1 и их унификаторы.
Хотя эта задача и NP-трудная, ее исходные данные имеют небольшую длину и оценка числа шагов имеет в показателе степени параметры формулы $P_i^1(\overline{y}_i^1)$ (количество переменных или количество атомарных формулов с одним и тем же предикатом). И эти параметры меньше таковых формулы $A(y_1, y_2, ..., y_l, d_1, ..., d_r)$.
Оценка числа шагов имеет вид
– для полного перебора:

$$O(\sum_{i=1}^{n_1} (l+r)^{||\overline{y}_i^1||}),$$

где $||\overline{y}_i^1||$ – количество параметров в $P_i^1(\overline{y}_i^1)$;
– для алгоритмов на основе исчисления предикатов:

$$O(\sum_{i=1}^{n_1} \sum_{j: p_j\, is\, in\, P_i^1(\overline{y}_i^1)} a_j^{\alpha_j}),$$

где a_j и α_j – количество атомарных формул с предикатом p_j в $A(y_1, y_2, ..., y_l, d_1, ..., d_r)$ и $P_i^1(\overline{y}_i^1)$ соответственно.

2. Заменим в A все найденные подформулы, изоморфные P_i^1, на $p_i^1(y_i^{1,j})$, где $y_i^{1,j}$ – аргумент для списка исходных аргументов этой подформулы. Это можно сделать, так как для каждой подформулы элементарной конъюнкции существует унификатор с ней. В результате получим элементарную конъюнкцию $A^1(\overline{x}^1)$, где $\overline{x}^1$ – список переменных (как исходных, так и 1-го уровня) и констант. На это уйдет линейное от длины записи A числа шагов. Заметим, что количество атомарных формул в $A^1(\overline{x}^1)$ не превосходит такового в $A(y_1, y_2, ..., y_l, d_1, ..., d_r)$ (и строго меньше, если хотя бы одна подформула была заменена на атомарную формулу 1-го уровня).

3. При проверке $S^1(D) \Rightarrow \exists \overline{x}^1 A^1(\overline{x}^1)$ в первую очередь проверяем это логическое следования для атомарных формул с предикатами 1-го уровня.

   При полном переборе потребуется $O(t_1^{l_1}||S||)$, где t_1 – количество элементов в базе $S^1(D)$ с предикатами 1-го уровня, l_1 – количество переменных 1-го уровня в запросе, полученном в п.2. А при использовании алгоритмов на основе исчисления предикатов потребуется $O(\sum_{i:p_i^1 \, is \, in \, A^1(\overline{x}^1)} s_i^{a_i})$, где s_i и a_i – число вхождений атомарных формул с предикатом p_i в базу данных $S(D)$ и в формулу $A(\overline{x})$ соответственно.

   Число шагов для проверки атомарных формул с исходными предикатами и поиска значений для исходных переменных составляет
   – для полного перебора: $O(t^{l-l_1})$ (l_1 – количество переменных 1-го уровня в $A^1(\overline{x}^1)$);
   – для алгоритмов на основе исчисления предикатов:
   $O(\sum_{i:p_i \, is \, in \, A^1(\overline{x}^1)} s_i^{a_i - a_i^1})$

Если просуммировать оценки для каждого этапа, то получим[1] полную оценку
– для полного перебора: $O(\sum_{i=1}^{n_1}(l + r)^{||\overline{y}_i^1||}) + O(t_1^{l_1}) + O(t^{l-l_1}) = O(t_1^{l_1} + t^{l-l_1})$;

– для алгоритмов на основе исчисления предикатов: $O(\sum_{i=1}^{n_1} \sum_{j:p_j\, is\, in\, P_i^1(\overline{y}_i^1)} a_j^{\alpha_j}) + O(\sum_{i:p_i^1\, is\, in\, A^1(\overline{x}^1)} s_i^{a_i}) + O(\sum_{i:p_i\, is\, in\, A^1(\overline{x}^1)} s_i^{a_i - a_i^1})$
$= O(\sum_{i:p_i^1\, is\, in\, A^1(\overline{x}^1)} s_i^{a_i}) + O(\sum_{i:p_i\, is\, in\, A^1(\overline{x}^1)} s_i^{a_i - a_i^1})$
Отсюда легко увидеть, что если хотя бы одна подформула была заменена на атомарную формулу 1-го уровня, то оценка числа шагов проверки запроса в двухуровневой базе данных меньше, чем таковая для изначальной базы данных.

Алгоритм выделения общих подформул

Определение. *Пара – это конъюнкция литералов.*
Теорема. *Если для двух пар литералов выполняются условия:*

- *их предикатные символы совпадают,*
- *каждому аргументу любой пары, который входит в нее более одного раза, соответствует один аргумент другой пары,*

то эти пары изоморфны.
Доказательство. Необходимость первого условия очевидна. Пусть выполняется второе условие. Рассмотрим две пары литералов: $A = p_i(a_1, ..., a_k)$ & $p_j(a_{k+1}, ..., a_m)$ и $B = p_i(b_1, ..., b_k)$ & $p_j(b_{k+1}, ..., b_m)$.
Покажем, что эти пары изоморфны, то есть существует элементарная конъюнкция C и подстановки $\lambda_{A,C}$ и $\lambda_{B,C}$ такие, что результаты этих подстановок совпадают с C с точностью до порядка литералов. Для того, чтобы найти C, пройдем по аргументам первой пары и заменим их на переменные x_i, при этом если аргумент входит в пару более одного раза, то ему будет соответствовать только один x_i. Существование $\lambda_{A,C}$ очевидно.
Найдем теперь $\lambda_{B,C}$. Рассмотрим $\lambda_{A,C} = (x_1 \to a_{i_1}, ..., x_l \to a_{i_l})$. Поскольку выполняется второе условие теоремы, то аргументы второй пары, соответствующие аргументам первой пары, попарно неравны между собой, потому что, в противном случае, если существуют такие i_1, i_2, что $a_{i_1} \neq a_{i_2}$, а при этом $b_{i_1} = b_{i_2}$, это противоречит второму условию. Следовательно, заменив в $\lambda_{A,C}$ a_{i_1} на b_{i_1}, мы получим $\lambda_{B,C}$, то есть A и B изоморфны, что и требовалось доказать.
Определение. *Индекс пары – это минимальное количество литералов в базе данных, конъюнкция которых изоморфна данной паре.*
Определение. *Адрес пары – это список адресов в базе данных входящих в нее литералов.*
Алгоритм выделения общих подформул

0. $l := 0$

1. Создадим список всевозможных конъюнкций пар литералов. Для этого попарно перебираем все литералы из базы данных и храним вместе с каждой парой ее адрес (список из двух чисел – номеров литералов в базе данных). Литералы в паре считаем упорядоченными по возрастанию номеров предикатных символов.

2. Для каждых двух непомеченных пар с одними и теми же предикатными символами проверяем их на изоморфность. Для этого сравниваем списки их аргументов.

   - Если списки разной длины, то пары не изоморфны
   - Если найдется аргумент в одной из пар, который входит в пару более одного раза, при этом ему соответствуют разные аргументы в другой паре, то пары не изоморфны. Иначе пары изоморфны.
   - Если предикатные символы у литералов одинаковые, повторяем процедуру, поменяв литералы одной из пар местами

3. Найденные изоморфные пары помечаем как проверенные, создаем пару с переменными вместо аргументов, заносим адреса пар с литералами в список адресов пары с переменными, добавляем унификаторы к списку подстановок.

4. Перебираем остальные пары в поисках изоморфных паре с переменными, применяя пункты 2-3.

5. Повторяем пункты 2-4, игнорируя уже проверенные пары.

6. Удаляем проверенные пары, сортируем список по убыванию индекса.

7. Берем пару с максимальным индексом(P_{max}) и вводим новый предикатный символ $(l+1)$-го уровня (где l – максимальный уровень предикатных символов, входящих в пару), определяемый этой парой. Вместе с символом храним полную форму пары.

8. Ищем пары, имеющие пересечения с P_{max}. Для этого рассмотрим отдельно каждую пару, имеющую с ней пересекающиеся адреса(P_{int_i}).

9. Поочередно подставляя в P_{int_i} соответствующие общим адресам унификаторы, смотрим, имеются ли у полученных конъюнкций общие литералы.

10. Если да, то добавляем конъюнкцию этих пар(3 литерала предыдущих уровней) к списку в виде пары с предикатным символом наибольшего из уровней исходных пар + 1, при этом сохраняем полную форму конъюнкции.

11. Выделяем изоморфные пары среди только что добавленных в п.10.

12. Обновляем индексы пар (если с учетом унификаторов пара является подформулой ранее полученного объединения, удаляем адрес из списка ее адресов), удаляем пары с нулевым индексом.

13. Повторяем процедуру с пункта 7, пока остаются пары с индексом ≥ 2

14. Добавляем новые предикатные символы и пары, содержащие их, в базу данных.

Алгоритм реализован на языке C++.

Оценка числа шагов работы алгоритма.

Пусть

– N – количество литералов в базе данных;

– L – максимально достижимый уровень пар(или количество литералов в конъюнкции);

– $a(p)$ – количество аргументов предикатного символа p;

– $p(l)$ – предикатный символ литерала l;

– P_i – i-я пара литералов;

– $C(P)$ – индекс P;

– l_{i_1} – 1-й литерал P_i;
– l_{i_2} – 2-й литерал P_i;
– $I(P)$ – количество пар, пересекающихся с P;
– c_i – количество «проверенных» пар на i-й итерации п.п. 2-4.

1. Поскольку первая стадия представляет из себя полный перебор, то число ее шагов составляет A_N^2.

2. В проверке на изоморфность двух пар используется конкатенация списков аргументов пар, длина которых не превосходит числа аргументов пар, т.е. $a(p(l_{i_1})) + a(p(l_{i_2}))$. В процессе проверки попарно сравниваются аргументы обеих пар, поэтому число шагов составляет $O((a(p(l_{i_1})) + a(p(l_{i_2}))^2)$, поскольку длина списков аргументов должна быть одинаковой, чтобы выполнялся данный этап.

3. Создание пары с переменными, занесение в нее адресов и унификаторов требует $O(a(p(l_{i_1})) + a(p(l_{i_2})))$ (замена констант на переменные).

4. Проверка остальных пар на изоморфность с парой с переменными займет $O((A_N^2 - 2) \cdot ((a(p(l_{i_1})) + a(p(l_{i_2}))^2))$ (поскольку пары с отличающимися предикатными символами сразу же отвергаются).

5. На этапах 2-4 помечаются проверенными c_i пар (которые после этого не нужно будет проверять на следующих этапах), поэтому итоговое число шагов для этапов 2-5 можно записать следующим образом: $O(\sum_{i=1}^{A_N^2}(A_N^2 - \sum_{j=1}^{i} c_j) \cdot (a(p(l_{i_1})) + a(p(l_{i_2}))^2)$.

6. Сортировка списка требует $O((A_N^2) \cdot log(A_N^2))$ шагов.
   Пусть K– число пар в списке по окончанию выполнения пункта 5.
   $K = A_N^2 - \sum_{i=1}^{A_N^2} c_i$.

7. Формирование нового предикатного символа не превосходит длине пары (т.е. числу ее аргументов). Поэтому число шагов данного этапа составляет $O(a(p(l_{1_1})) + a(p(l_{1_2})))$.

8. Поиск пар, имеющих с P_{max} общие адреса, занимает $O(\sum_{i=1}^{K}(C(P_{max})(P_i)))$.

9. Для проверки того, что эти пары имеют общие литералы с P_{max}, требуется подставить унификаторы, соответствующие общим адресам, после чего сравнить полученные литералы, что не превосходит произведения длин строк. Получим $O(\sum_{i=1}^{I(P)}(a(p(l_{max_1})) + a(p(l_{max_2})) + a(p(l_{i_1})) + a(p(l_{i_2}))) + (a(p(l_{max_1})) + a(p(l_{max_2}))(a(p(l_{i_1})) + a(p(l_{i_2})))) = O(\sum_{i=1}^{I(P)}(a(p(l_{max_1})) + a(p(l_{max_2}))) \cdot (a(p(l_{i_1})) + a(p(l_{i_2}))))$.

10. Формирование конъюнкции таких пар и добавление их к списку требует $O(a(p(l_{max_1})) + a(p(l_{max_2})) + a(p(l_{i_1}))$.

11. Выделение изоморфных пар среди полученных конъюнкций займет $O(\sum_{i=1}^{I(P)} I(P) \cdot (a(p(l_{i_1})) + a(p(l_{i_2})))^2)$.

12. Обновление индексов требует $O(\sum_{i=1}^{K} \sum_{j=1}^{I(P)} C(P_i) \cdot C(P_j) \cdot ((a(p_{i_1}) + a(p_{i_2})) \cdot (a(p(l_{j_1})) + a(p(l_{j_2}))))$.

13. Число пар 0-го уровня, обработанных во время выполнения пунктов 7-13, не превосходит A_K^2. Число полученных в результате выполнение пунктов 6-13 пар 1-го уровня не превзойдет A_K^2. Обозначим $K_k = A_{K_{k-1}}^2$. Тогда, число обработанных пар 1-го уровня будет не больше $A_{K_1}^2$. Поскольку максимально достижимый уровень пар равен L, то по индукции для оценки шагов получаем $O(A_N^2 \cdot \sum_{k=1}^{L} A_{K_k}^2 \cdot (a(p(l_{1_1})) + a(p(l_{1_2})) + \sum_{i=1}^{K_k}(C(P_{max})(P_i)))$.

Итоговое число шагов получается $A_N^2 + O(\sum_{i=1}^{A_N^2}((A_N^2 - \sum_{j=1}^{i} c_j) \cdot (a(p(l_{i_1})) + a(p(l_{i_2}))^2) + A_N^2 \cdot \sum_{k=1}^{L}(A_{K_k}^2 \cdot (a(p(l_{1_1})) + a(p(l_{1_2})) + \sum_{i=1}^{K_k}(C(P_{max}) \cdot C(P_i)))$

Список литературы

[1] Kosovskaya Tatyana. Conjunctive Boolean Query as a logic-objective recognition problem // International Journal on Information Theories Applications. — 2017. — Vol. 24, no. 3. — P. 72–78.

[2] Гэри М., Джонсон Д. Труднорешаемые задачи. — М. : Мир, 1982.

[3] Коба Д.А. Алгоритм построения многоуровневого описания в задачах распознавания образов и оценка числа шагов его работы // Материалы XVII Международной школы-семинара «Синтез и сложность управляющих систем» им. акад. О.Б. Лупанова (Новосибирск, 27 октября – 1 ноября 2008 г.). — P. 55–59.

[4] Косовская Т.М. Доказательства оценок числа шагов решения некоторых задач распознавания образов, имеющих логические описания // Вестник Санкт-Петербургского университета. Сер. 1. 2007. Вып. 4. — P. 82–90.

[5] Косовская Т.М. Многоуровневые описания классов для уменьшения числа шагов решения задач распознавания образов, описываемых формулами исчисления предикатов // Вестник Санкт-Петербургского университета. Сер. 10. 2008. Вып. 1. — P. 64–72.

[6] Косовская Т.М., Петров Д.А. Выделение наибольшей общей подформулы предикатных формул для решения ряда задач искусственного интеллекта // Вестник СПбГУ. Прикладная математика. Информатика. Процессы управления. — 2017. — Vol. 13, no. 3. — P. 250–263.

[7] Петров Д.А. Алгоритмы выделения наибольшей общей с точностью до имён переменных формул исчисления предикатов и их реализация // Материалы 9-й конференции «Информационные технологии в управлении (ИТУ – 2016)». СПб. — 2016. — P. 97–102.

Приложение А: Код программы

Pairs.h

```
#pragma once
class Pairs
{
public:
int Level;
std::vector<std::string> Arguments;
std::string First;
std::string Second;
std::string ShortForm;
std::string FullForm;
std::vector<std::vector<std::string>> Substitutions;
std::vector<std::vector<std::vector<int>>> Addresses;
friend std::ostream& operator<<(std::ostream& os,
const Pairs& pair);
Pairs() : Level(0), ShortForm(""), FullForm(""),
Substitutions(0), Addresses(0) {};
Pairs(int level, const std::vector<std::string>& arg,
const char* first, const char* second,
const char* shortform, const char* fullform,
const std::vector<std::vector<std::string>>& subst,
const std::vector<std::vector<
std::vector<int>>>& addresses);
~Pairs();
std::string ApplySubstitutionShort(int i);
std::string ApplySubstitutionFull(int i);
};
```

Pairs.cpp

```cpp
#pragma once
#include <string>
#include "stdafx.h"
#include <vector>
#include "Pairs.h"
Pairs::Pairs(int level,
const std::vector<std::string>& arg,
const char* first, const char* second,
const char* shortform, const char* fullform,
const std::vector<std::vector<std::string>>& subst,
const std::vector<std::vector<std::vector<int>>>& addresses)
{
Level = level;
Arguments = arg;
First = std::string(first);
Second = std::string(second);
ShortForm = std::string(shortform);
FullForm = std::string(fullform);
Substitutions = std::vector<std::vector<std::string>>(subst)
Addresses = std::vector<std::vector<
std::vector<int>>>(addresses);
}
Pairs::~Pairs()
{
}
std::ostream& operator<<(std::ostream& os,
const Pairs& pair)
{
os << pair.ShortForm << " " << "<-> "
<< pair.FullForm << " (";
for (int i = 0; i < pair.Substitutions.size(); i++) {
```

```
os << "(";
int j;
for (j = 0; j < pair.Substitutions[i].size() - 1; j++)
os << pair.Substitutions[i][j] << ",";
os << pair.Substitutions[i][j] << "))" << " ";
}
for (int i = 0; i < pair.Addresses.size(); i++) {
int j;
os << "(";
for (j = 0; j < pair.Addresses[i][0].size(); j++)
os << pair.Addresses[i][0][j] << ",";
for (j = 0; j < pair.Addresses[i][1].size() - 1; j++)
os << pair.Addresses[i][0][j] << ",";
os << pair.Addresses[i][1][j] << ") ";
}
os << std::endl;
return os;
}
std::string Pairs::ApplySubstitutionShort(int i)
{
std::string res = ShortForm;
for (int j = 0; j < Arguments.size(); j++) {
size_t index;
while ((index = res.find(Arguments[j]))
!= std::string::npos)
res.replace(index, Arguments[j].length(),
Substitutions[i][j]);
}
return res;
}
std::string Pairs::ApplySubstitutionFull(int i)
{
std::string res = FullForm;
```

```
for (int j = 0; j < Arguments.size(); j++) {
size_t index;
while ((index = res.find(Arguments[j]))
!= std::string::npos)
res.replace(index, Arguments[j].length(),
Substitutions[i][j]);
}
return res;
}
```

main.cpp

```
#include "stdafx.h"
#include "Pairs.h"
//Vector with raw data
std::vector<std::string> DataBase(0);
std::vector<Pairs> StartPairs(0);
std::vector<Pairs> PairsAfter(0);
std::vector<Pairs> Predicates(0);
#pragma region
template<typename Out>
void split(const std::string &s,
char delim, Out result) {
std::stringstream ss(s);
std::string item;
while (std::getline(ss, item, delim)) {
*(result++) = item;
}
}
std::vector<std::string> split(
const std::string &s, char delim) {
std::vector<std::string> elems;
split(s, delim, std::back_inserter(elems));
return elems;
}
#pragma endregion Function for splitting string
#pragma region
template <typename T>
void remove_dups_from_vector(std::vector<T> &v)
{
std::set<T> unique_elements;
v.erase(std::remove_if(v.begin(), v.end(),
[&](const T &value)
{ return !unique_elements.insert(value).second; }), v.end()
```

```
}
#pragma endregion Removing duplicate words from a string
#pragma region
bool areIsomorphic(std::vector<std::string> arg1_1,
std::vector<std::string> arg1_2, std::string arg2,
std::vector<std::vector<std::string>> &substitutions,
std::vector<std::string> &args)
{
arg1_1.insert(arg1_1.end(), arg1_2.begin(), arg1_2.end());
int last = arg2.find_last_of('(');
auto arg_2_1 = arg2.substr(0, arg2.find(')')).substr(
arg2.find('(') + 1, arg2.length());
auto arg_2_2 = arg2.substr(last + 1,
arg2.find_last_of(')') - last - 1);
auto sarg = split(arg_2_1, ',');
auto sarg2 = split(arg_2_2, ',');
sarg.insert(sarg.end(), std::make_move_iterator(
sarg2.begin()),std::make_move_iterator(sarg2.end()));
int c1, c2;
for (int k = 0; k < arg1_1.size(); k++) {
for (int l = k + 1; l < arg1_1.size(); l++) {
c1 = arg1_1[k].compare(arg1_1[l]);
c2 = sarg[k].compare(sarg[l]);
if ((c1 == 0 && c2 != 0) || (c1 != 0 && c2 == 0)) {
return false;
}
}
}
remove_dups_from_vector(arg1_1);
remove_dups_from_vector(sarg);
substitutions.push_back(arg1_1);
args = arg1_1;
substitutions.push_back(sarg);
```

```
return true;
}
#pragma endregion Function for comparing lists of arguments
#pragma region
static bool paircompare(Pairs a, Pairs b)
{
        return a.Addresses.size() > b.Addresses.size();
}
#pragma endregion Comparator for pairs based on indexes
#pragma region
static bool paircomparelvl(Pairs a, Pairs b)
{
        if (a.Level > b.Level) {
                return true;
        } else if (b.Level > a.Level) {
                return false;
        } else {
                return paircompare(a, b);
        }
}
#pragma endregion Comparator for pairs based on their level
//Constructing list of all pairs
void ExtractAllPairs()
{
std::vector<std::vector<std::string>> substitute(0);
for (int i = 0; i < DataBase.size(); i++) {
std::string fname =
DataBase[i].substr(0, DataBase[i].find('('));
for (int j = i + 1; j < DataBase.size(); j++) {
std::string sname =
DataBase[j].substr(0, DataBase[j].find('('));
std::vector<std::vector<std::vector<int>>> address(1);
address[0] = std::vector<std::vector<int>>(0);
```

```
std::string fullform = DataBase[i];
if (std::stoi(fname.substr(1, fname.length()))
<= std::stoi(sname.substr(1, sname.length()))) {
fullform += ''&'' + DataBase[j];
address[0].push_back({ i });
address[0].push_back({ j });
StartPairs.push_back(Pairs(0, std::vector<std::string>(0),
fname.c_str(), sname.c_str(), fullform.c_str(),
fullform.c_str(), substitute, address));
}
else {
fullform = DataBase[j] + ''&'' + fullform;
address[0].push_back({ j });
address[0].push_back({ i });
StartPairs.push_back(Pairs(0, std::vector<std::string>(0),
sname.c_str(), fname.c_str(), fullform.c_str(),
fullform.c_str(), substitute, address));
}
}
}
}
//Grouping together isomoprhic pairs
std::vector<Pairs> FindIsomoprhicPairs(
std::vector<Pairs> input)
{
std::vector<Pairs> result(0);
std::vector<bool> checked(input.size());
std::fill(checked.begin(), checked.end(), false);
for (int i = 0; i < checked.size(); i++) {
std::vector<std::vector<std::string>> subst(0);
std::string ff;
std::vector<std::vector<std::vector<int>>> addr;
std::vector<std::string> args(0);
```

```
if (!checked[i]) {
std::string buf = input[i].ShortForm;
std::string arg_1 = buf.substr(0,
buf.find(')')).substr(buf.find('(') + 1, buf.length());
int last = buf.find_last_of('(');
std::string arg_2 = buf.substr(last + 1,
buf.find_last_of(')') - last - 1);
std::vector<std::string> farg = split(arg_1, ',');
std::vector<std::string> farg1;
int p = farg.size();
std::vector<std::string> arg2 = split(arg_2, ',');
for (int j = i + 1; j < checked.size(); j++) {
//if pairs have different predicate symbols they
//obviously are not isomorphic
if ((input[i].First.compare(input[j].First) != 0) ||
(input[i].Second.compare(input[j].Second) != 0) ||
checked[j])
continue;
//checking if argument lists can be transformed
//into each other
if (areIsomorphic(farg, arg2, input[j].ShortForm, subst, fa
(input[i].First.compare(input[i].Second) == 0 &&
areIsomorphic(arg2, farg,
input[j].FullForm, subst, farg1))) {
if (!checked[i]) {
addr.push_back(input[i].Addresses[0]);
ff = input[i].FullForm;
if (args.size() == 0) {
for (int k = 0; k < farg1.size(); k++) {
args.push_back("x" + std::to_string(k + 1));
size_t index;
while ((index = ff.find(farg1[k])) != std::string::npos)
ff.replace(index, farg1[k].length(),
```

```
"x" + std::to_string(k + 1));
}
}
checked[i] = true;
}
else {
subst.erase(subst.end() - 2);
}
addr.push_back(input[j].Addresses[0]);
checked[j] = true;
}
}
}
if (args.size() != 0) {
input.push_back(Pairs(input[i].Level, args,
input[i].First.c_str(), input[i].Second.c_str(),
ff.c_str(), ff.c_str(), subst, addr));
}
}
for (int i = 0; i < checked.size(); i++) {
if (!checked[i])
result.push_back(input[i]);
}
for (int i = checked.size(); i < input.size(); i++)
result.push_back(input[i]);
return result;
}
//Update pairs indexes
void UpdateIndexes(std::vector<Pairs> &input,
std::vector<Pairs> &intersect)
{
std::vector<std::vector<int>> addresses(0);
std::vector<int> address(0);
```

```
for (int i = 0; i < input.size();) {
bool flag1 = false;
for (int j = 0; j < addresses.size(); j++) {
bool flag = true;
for (int k = 0; k < input[i].Addresses[0][0].size(); k++) {
flag = flag && std::find(addresses[j].begin(),
addresses[j].end(), input[i].Addresses[0][0][k])
!= addresses[j].end();
if (!flag)
break;
}
if (flag) {
for (int k = 0; k < input[i].Addresses[0][1].size(); k++) {
flag = flag && std::find(addresses[j].begin(),
addresses[j].end(), input[i].Addresses[0][1][k])
!= addresses[j].end();
if (!flag)
break;
}
}
flag1 = flag1 || flag;
}
if (!flag1) {
for (int j = 0; j < intersect.size(); j++) {
bool flag = true;
for (int k = 0; k < input[i].Addresses[0][0].size(); k++) {
flag = flag && (std::find(
intersect[j].Addresses[0][0].begin(),
intersect[j].Addresses[0][0].end(),
input[i].Addresses[0][0][k])
!= intersect[j].Addresses[0][0].end() ||
std::find(intersect[j].Addresses[0][1].begin(),
intersect[j].Addresses[0][1].end(),
```

```
input[i].Addresses[0][0][k])
!= intersect[j].Addresses[0][1].end());
if (!flag)
break;
}
if (flag) {
for (int k = 0; k < input[i].Addresses[0][1].size(); k++) {
flag = flag && (std::find(
intersect[j].Addresses[0][0].begin(),
intersect[j].Addresses[0][0].end(),
input[i].Addresses[0][1][k]
!= intersect[j].Addresses[0][0].end() ||
std::find(intersect[j].Addresses[0][1].begin(),
intersect[j].Addresses[0][1].end(),
input[i].Addresses[0][1][k])
!= intersect[j].Addresses[0][1].end());
if (!flag)
break;
}
}
flag1 = flag1 || flag;
}
}
if (flag1) {
input.erase(input.begin() + i);
}
else {
for (int k = 0; k < input[i].Addresses[0][0].size(); k++) {
address.push_back(input[i].Addresses[0][0][k]);
}
for (int k = 0; k < input[i].Addresses[0][1].size(); k++) {
address.push_back(input[i].Addresses[0][1][k]);
}
```

```
i++;
addresses.push_back(address);
}
address.clear();
}
for (int i = 0; i < PairsAfter.size();) {
for (int l = 0; l < addresses.size(); l++) {
for (int j = 0; j < PairsAfter[i].Addresses.size();) {
bool flag = true;
for (int k = 0;
k < PairsAfter[i].Addresses[j][0].size(); k++) {
flag = flag && std::find(addresses[l].begin(),
addresses[l].end(),
PairsAfter[i].Addresses[j][0][k]) != addresses[l].end();
if (!flag)
break;
}
if (flag) {
for (int k = 0;
k < PairsAfter[i].Addresses[j][1].size(); k++) {
flag = flag && std::find(addresses[l].begin(),
addresses[l].end(),
PairsAfter[i].Addresses[j][1][k]) != addresses[l].end();
if (!flag)
break;
}
}
if (flag) {
PairsAfter[i].Addresses.erase(
PairsAfter[i].Addresses.begin() + j);
if (PairsAfter[i].Substitutions.size() != 0)
PairsAfter[i].Substitutions.erase(
PairsAfter[i].Substitutions.begin() + j);
```

```
} else {
j++;
}
}
}
if (PairsAfter[i].Addresses.size() == 0) {
PairsAfter.erase(PairsAfter.begin() + i);
} else {
if (PairsAfter[i].Addresses.size() == 1
&& PairsAfter[i].Substitutions.size() != 0) {
PairsAfter[i].ShortForm =
PairsAfter[i].ApplySubstitutionShort(0);
PairsAfter[i].FullForm =
PairsAfter[i].ApplySubstitutionFull(0);
PairsAfter[i].Substitutions.clear();
PairsAfter[i].Arguments.clear();
}
i++;
}
}
}
//Constructing multi-level database
void CreateNewStructure()
{
while (PairsAfter[0].Addresses.size() > 1) {
Pairs p_max = Pairs();
p_max.Level = PairsAfter[0].Level + 1;
p_max.Addresses = PairsAfter[0].Addresses;
p_max.Substitutions = PairsAfter[0].Substitutions;
p_max.FullForm = PairsAfter[0].FullForm;
p_max.Arguments = PairsAfter[0].Arguments;
std::string shortform;
int number;
```

```
auto it = std::find_if(Predicates.rbegin(),
Predicates.rend(), [&p_max](Pairs p)
{ return p.Level == p_max.Level; });
if (it != Predicates.rend()) {
std::string lvlmaxname = (*it).ShortForm;
number = std::stoi(lvlmaxname.substr(0,
lvlmaxname.find('^')).substr(1, lvlmaxname.length())) + 1;
}
else {
number = 1;
}
shortform = "p" + std::to_string(number) + "^" + s
td::to_string(p_max.Level) + "(";
auto args = split(PairsAfter[0].ShortForm, '&');
std::string arg;
int i;
for (i = 0; i < args.size() - 1; i++) {
arg = args[i].substr(
args[i].find('(') + 1, args[i].length());
arg.erase(arg.end() - 1);
shortform += arg + ",";
}
arg = args[i].substr(
args[i].find('(') + 1, args[i].length());
arg.erase(arg.end() - 1);
shortform += arg + ")";
p_max.ShortForm = shortform;
p_max.First = p_max.Second =
p_max.ShortForm.substr(0, p_max.ShortForm.find('('));
Predicates.push_back(p_max);
PairsAfter.erase(PairsAfter.begin());
std::cout << p_max;
std::vector<Pairs> intersections(0);
```

```
args.clear();
std::vector<std::vector<std::string>> subst(0);
for (int j = 0; j < PairsAfter.size();) {
Pairs p_cur = PairsAfter[j];
while (j < PairsAfter.size()
&& PairsAfter[j].Level >= p_max.Level) {
j++;
}
std::vector<Pairs> intersections_temp(0);
for (int k = 0; k < p_max.Addresses.size(); k++) {
if (j < PairsAfter.size()
&& PairsAfter[j].Level < p_max.Level) {
std::vector<int> address_1(0);
for (int m = 0; m < p_max.Addresses[k][0].size(); m++) {
address_1.push_back(p_max.Addresses[k][0][m]);
}
for (int m = 0; m < p_max.Addresses[k][1].size(); m++) {
address_1.push_back(p_max.Addresses[k][1][m]);
}
std::vector<std::vector<std::vector<int>>> address(0);
if (PairsAfter[j].Addresses.size() == 1) {
std::vector<int> address_2(0);
std::string s = p_max.ApplySubstitutionShort(k) + ''&'';
if (PairsAfter[j].Addresses[0][0] == p_max.Addresses[k][0]
|| PairsAfter[j].Addresses[0][0]==p_max.Addresses[k][1]){
s += split(PairsAfter[j].ShortForm, '&')[1];
for (int m = 0;
m < PairsAfter[j].Addresses[0][1].size(); m++) {
address_2.push_back(PairsAfter[j].Addresses[0][1][m]);
}
address = { { address_1, address_2 } };
intersections_temp.push_back(Pairs(p_max.Level, args,
p_max.First.c_str(), PairsAfter[j].Second.c_str(),
```

```
s.c_str(), s.c_str(), subst, address));
}
else if (PairsAfter[j].Addresses[0][1]
== p_max.Addresses[k][0]
|| PairsAfter[j].Addresses[0][1] == p_max.Addresses[k][1])
s += split(PairsAfter[j].ShortForm, '&')[0];
for (int m = 0; m < PairsAfter[j].Addresses[0][0].size(); m
address_2.push_back(PairsAfter[j].Addresses[0][0][m]);
}
address = { { address_1, address_2 } };
intersections_temp.push_back(Pairs(p_max.Level, args,
p_max.First.c_str(), PairsAfter[j].First.c_str(),
s.c_str(), s.c_str(), subst, address));
}
}
else {
for (int l = 0; l < PairsAfter[j].Addresses.size(); l++) {
std::vector<int> address_2(0);
std::string s = p_max.ApplySubstitutionShort(k) + "&";
if (PairsAfter[j].Addresses[l][0] == p_max.Addresses[k][0]
|| PairsAfter[j].Addresses[l][0] == p_max.Addresses[k][1])
s +=split(PairsAfter[j].ApplySubstitutionShort(l), '&')[1];
for (int m = 0; m < PairsAfter[j].Addresses[l][1].size(); m
address_2.push_back(PairsAfter[j].Addresses[l][1][m]);
}
address = { { address_1, address_2 } };
intersections_temp.push_back(Pairs(p_max.Level, args,
p_max.First.c_str(), PairsAfter[j].Second.c_str(),
s.c_str(), s.c_str(), subst, address));
}
else if (PairsAfter[j].Addresses[l][1]
== p_max.Addresses[k][0]
|| PairsAfter[j].Addresses[l][1] == p_max.Addresses[k][1])
```

```
s +=split(PairsAfter[j].ApplySubstitutionShort(1), '&')[0];
for (int m = 0;
m < PairsAfter[j].Addresses[1][0].size(); m++) {
address_2.push_back(PairsAfter[j].Addresses[1][0][m]);
}
address = { { address_1, address_2 } };
intersections_temp.push_back(Pairs(p_max.Level, args,
p_max.First.c_str(), PairsAfter[j].First.c_str(),
s.c_str(), s.c_str(), subst, address));
}
}
if (PairsAfter[j].Addresses.size() == 1) {
PairsAfter[j].ShortForm =
PairsAfter[j].ApplySubstitutionShort(0);
PairsAfter[j].FullForm =
PairsAfter[j].ApplySubstitutionFull(0);
PairsAfter[j].Substitutions.clear();
PairsAfter[j].Arguments.clear();
}
}
}
}
UpdateIndexes(intersections_temp, intersections);
intersections.insert(intersections.end(),
intersections_temp.begin(), intersections_temp.end());
if (j < PairsAfter.size()
&& p_cur.ShortForm.compare(PairsAfter[j].ShortForm) == 0) {
j++;
}
}
std::vector<Pairs> groupedIntersect =
FindIsomoprhicPairs(intersections);
PairsAfter.insert(PairsAfter.end(),
```

```
groupedIntersect.begin(), groupedIntersect.end());
std::sort(PairsAfter.begin(),
PairsAfter.end(), paircompare);
}
std::sort(PairsAfter.begin(),
PairsAfter.end(), paircomparelvl);
}
int main()
{
//Filling the vector with predicates from file
std::ifstream dbfile("database1.txt");
std::string line;
if (!dbfile)
{
std::cout << "Error␣opening␣output␣file" << std::endl;
system("pause");
return -1;
}
while (std::getline(dbfile, line))
DataBase.push_back(line);
//Handle possible errors
try {
std::cout << DataBase[0] << std::endl << std::endl;
}
catch (...) {
std::cout << "Data␣is␣unavailable" << std::endl;
}
ExtractAllPairs();
PairsAfter = FindIsomoprhicPairs(StartPairs);
std::sort(PairsAfter.begin(),
PairsAfter.end(), paircompare);
CreateNewStructure();
for (int i = 0; i < Predicates.size(); i++) {
```

```
std::cout << Predicates[i];
}
std::cout << std::endl;
for (int i = 0; i < PairsAfter.size(); i++) {
std::cout << PairsAfter[i];
}
return 0;
}
```

Приложение B: Пример работы алгоритма

Рассмотрим следующий пример, реализованный программой. Пусть содержимое базы данных выглядит следующим образом:

0. $p_1(a, b, c)$
1. $p_3(a, d)$
2. $p_1(a, b, a)$
3. $p_2(c, d, d)$
4. $p_2(a, a, a)$
5. $p_1(r, b, t)$
6. $p_3(r, m)$
7. $p_3(u, l)$
8. $p_1(r, w, r)$
9. $p_2(m, i, i)$

В соответствии с первым пунктом алгоритма выделим всевозможные пары литералов, указывая адрес в виде списка номеров литералов в базе данных. Результат выглядит следующим образом:

Номер	**Пара**	**Адрес**
1	$p_1(a, b, c)$ & $p_3(a, d)$	(0,1)
2	$p_1(a, b, c)$ & $p_1(a, b, a)$	(0,2)
3	$p_1(a, b, c)$ & $p_2(c, d, d)$	(0,3)
4	$p_1(a, b, c)$ & $p_2(a, a, a)$	(0,4)
5	$p_1(a, b, c)$ & $p_1(r, b, t)$	(0,5)
6	$p_1(a, b, c)$ & $p_3(r, m)$	(0,6)
7	$p_1(a, b, c)$ & $p_3(u, l)$	(0,7)

8	$p_1(a,b,c)$ & $p_1(r,w,r)$	(0,8)
9	$p_1(a,b,c)$ & $p_2(m,i,i)$	(0,9)
10	$p_1(a,b,a)$ & $p_3(a,d)$	(2,1)
11	$p_2(c,d,d)$ & $p_3(a,d)$	(3,1)
12	$p_2(a,a,a)$ & $p_3(a,d)$	(4,1)
13	$p_1(r,b,t)$ & $p_3(a,d)$	(5,1)
14	$p_3(a,d)$ & $p_3(r,m)$	(1,6)
15	$p_3(a,d)$ & $p_3(u,l)$	(1,7)
16	$p_1(r,w,r)$ & $p_3(a,d)$	(8,1)
17	$p_2(m,i,i)$ & $p_3(a,d)$	(9,1)
18	$p_1(a,b,a)$ & $p_2(c,d,d)$	(2,3)
19	$p_1(a,b,a)$ & $p_2(a,a,a)$	(2,4)
20	$p_1(a,b,a)$ & $p_1(r,b,t)$	(2,5)
21	$p_1(a,b,a)$ & $p_3(r,m)$	(2,6)
22	$p_1(a,b,a)$ & $p_3(u,l)$	(2,7)
23	$p_1(a,b,a)$ & $p_1(r,w,r)$	(2,8)
24	$p_1(a,b,a)$ & $p_2(m,i,i)$	(2,9)
25	$p_2(c,d,d)$ & $p_2(a,a,a)$	(3,4)
26	$p_1(r,b,t)$ & $p_2(c,d,d)$	(5,3)
27	$p_2(c,d,d)$ & $p_3(r,m)$	(3,6)
28	$p_2(c,d,d)$ & $p_3(u,l)$	(3,7)
29	$p_1(r,w,r)$ & $p_2(c,d,d)$	(8,3)
30	$p_2(c,d,d)$ & $p_2(m,i,i)$	(3,9)
31	$p_1(r,b,t)$ & $p_2(a,a,a)$	(5,4)
32	$p_2(a,a,a)$ & $p_3(r,m)$	(4,6)
33	$p_2(a,a,a)$ & $p_3(u,l)$	(4,7)
34	$p_1(r,w,r)$ & $p_2(a,a,a)$	(8,4)
35	$p_2(a,a,a)$ & $p_2(m,i,i)$	(4,9)
36	$p_1(r,b,t)$ & $p_3(r,m)$	(5,6)
37	$p_1(r,b,t)$ & $p_3(u,l)$	(5,7)
38	$p_1(r,b,t)$ & $p_1(r,w,r)$	(5,8)

39	$p_1(r, b, t)$ & $p_2(m, i, i)$	(5,9)
40	$p_3(r, m)$ & $p_3(u, l)$	(6,7)
41	$p_1(r, w, r)$ & $p_3(r, m)$	(8,6)
42	$p_2(m, i, i)$ & $p_3(r, m)$	(9,6)
43	$p_1(r, w, r)$ & $p_3(u, l)$	(8,7)
44	$p_2(m, i, i)$ & $p_3(u, l)$	(9,7)
45	$p_1(r, w, r)$ & $p_2(m, i, i)$	(8,9)

Проверяем пары из списка на изоморфность: берем пару $p_1(a, b, c) \& p_3(a, d)$ и проходим по списку в поисках изоморфных ей. Пары под номером 2-5 не изоморфны ей, поскольку предикатные символы не совпадают. Рассмотрим пару $p_1(a, b, c) \,\&\, p_3(r, m)$. Списки аргументов этих пар выглядят следующим образом: (a, b, c, d) и (a, b, c, r, m). Списки аргументов разной длины, поэтому пары не изоморфны. По той же схеме отбрасываются последующие пары, пока не дойдем до пары $p_1(r, b, t) \,\&\, p_3(r, m)$. Списки аргументов получаются следующими: (a, b, c, d) и (r, b, t, m). Легко устанавливается однозначное соответствие между аргументами двух пар: $(a \leftrightarrow r, b \leftrightarrow b, c \leftrightarrow t, d \leftrightarrow m)$. Значит, эти пары изоморфны.

Создаем пару с переменными: $p_1(x_1, x_2, x_3) \& p_3(x_1, x_4)$. В ее список адресов заносятся адреса пар с литералами – $(0, 1)$ и $(5, 6)$. В список подстановок добавляем $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ и $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to m)$. Пары с литералами помечаем как проверенные (символом $+$).

Проверкой оставшихся пар получаем, что других пар, изоморфных данной, нет.

Получаем следующую таблицу:

Номер	**Пара**	**Адрес**	**Проверена**
1	$p_1(a, b, c)$ & $p_3(a, d)$	(0,1)	$+$
2	$p_1(a, b, c)$ & $p_1(a, b, a)$	(0,2)	
3	$p_1(a, b, c)$ & $p_2(c, d, d)$	(0,3)	

4	$p_1(a,b,c)$ & $p_2(a,a,a)$	(0,4)	
5	$p_1(a,b,c)$ & $p_1(r,b,t)$	(0,5)	
6	$p_1(a,b,c)$ & $p_3(r,m)$	(0,6)	
7	$p_1(a,b,c)$ & $p_3(u,l)$	(0,7)	
8	$p_1(a,b,c)$ & $p_1(r,w,r)$	(0,8)	
9	$p_1(a,b,c)$ & $p_2(m,i,i)$	(0,9)	
10	$p_1(a,b,a)$ & $p_3(a,d)$	(2,1)	
11	$p_2(c,d,d)$ & $p_3(a,d)$	(3,1)	
12	$p_2(a,a,a)$ & $p_3(a,d)$	(4,1)	
13	$p_1(r,b,t)$ & $p_3(a,d)$	(5,1)	
14	$p_3(a,d)$ & $p_3(r,m)$	(1,6)	
15	$p_3(a,d)$ & $p_3(u,l)$	(1,7)	
16	$p_1(r,w,r)$ & $p_3(a,d)$	(8,1)	
17	$p_2(m,i,i)$ & $p_3(a,d)$	(9,1)	
18	$p_1(a,b,a)$ & $p_2(c,d,d)$	(2,3)	
19	$p_1(a,b,a)$ & $p_2(a,a,a)$	(2,4)	
20	$p_1(a,b,a)$ & $p_1(r,b,t)$	(2,5)	
21	$p_1(a,b,a)$ & $p_3(r,m)$	(2,6)	
22	$p_1(a,b,a)$ & $p_3(u,l)$	(2,7)	
23	$p_1(a,b,a)$ & $p_1(r,w,r)$	(2,8)	
24	$p_1(a,b,a)$ & $p_2(m,i,i)$	(2,9)	
25	$p_2(c,d,d)$ & $p_2(a,a,a)$	(3,4)	
26	$p_1(r,b,t)$ & $p_2(c,d,d)$	(5,3)	
27	$p_2(c,d,d)$ & $p_3(r,m)$	(3,6)	
28	$p_2(c,d,d)$ & $p_3(u,l)$	(3,7)	
29	$p_1(r,w,r)$ & $p_2(c,d,d)$	(8,3)	
30	$p_2(c,d,d)$ & $p_2(m,i,i)$	(3,9)	
31	$p_1(r,b,t)$ & $p_2(a,a,a)$	(5,4)	
32	$p_2(a,a,a)$ & $p_3(r,m)$	(4,6)	
33	$p_2(a,a,a)$ & $p_3(u,l)$	(4,7)	
34	$p_1(r,w,r)$ & $p_2(a,a,a)$	(8,4)	

35	$p_2(a,a,a)$ & $p_2(m,i,i)$	(4,9)	
36	$p_1(r,b,t)$ & $p_3(r,m)$	(5,6)	+
37	$p_1(r,b,t)$ & $p_3(u,l)$	(5,7)	
38	$p_1(r,b,t)$ & $p_1(r,w,r)$	(5,8)	
39	$p_1(r,b,t)$ & $p_2(m,i,i)$	(5,9)	
40	$p_3(r,m)$ & $p_3(u,l)$	(6,7)	
41	$p_1(r,w,r)$ & $p_3(r,m)$	(8,6)	
42	$p_2(m,i,i)$ & $p_3(r,m)$	(9,6)	
43	$p_1(r,w,r)$ & $p_3(u,l)$	(8,7)	
44	$p_2(m,i,i)$ & $p_3(u,l)$	(9,7)	
45	$p_1(r,w,r)$ & $p_2(m,i,i)$	(8,9)	

Номер	**Пара**	**Адреса**	**Подстановки**
46	$p_1(x_1,x_2,x_3)$ & $p_3(x_1,x_4)$	(0,1); (5,6)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to m)$ $(x_1 \to c, x_2 \to s, x_3 \to d, x_4 \to p)$ $(x_1 \to e, x_2 \to f, x_3 \to d, x_4 \to k)$

Повторяя данную процедуру для непроверенных пар, получаем в итоге следующий результат:

Номер	**Пара**	**Адрес**	**Проверена**
1	$p_1(a,b,c)$ & $p_3(a,d)$	(0,1)	+
2	$p_1(a,b,c)$ & $p_1(a,b,a)$	(0,2)	
3	$p_1(a,b,c)$ & $p_2(c,d,d)$	(0,3)	
4	$p_1(a,b,c)$ & $p_2(a,a,a)$	(0,4)	
5	$p_1(a,b,c)$ & $p_1(r,b,t)$	(0,5)	

6	$p_1(a,b,c)$ & $p_3(r,m)$	(0,6)	$+$
7	$p_1(a,b,c)$ & $p_3(u,l)$	(0,7)	$+$
8	$p_1(a,b,c)$ & $p_1(r,w,r)$	(0,8)	
9	$p_1(a,b,c)$ & $p_2(m,i,i)$	(0,9)	$+$
10	$p_1(a,b,a)$ & $p_3(a,d)$	(2,1)	$+$
11	$p_2(c,d,d)$ & $p_3(a,d)$	(3,1)	
12	$p_2(a,a,a)$ & $p_3(a,d)$	(4,1)	
13	$p_1(r,b,t)$ & $p_3(a,d)$	(5,1)	$+$
14	$p_3(a,d)$ & $p_3(r,m)$	(1,6)	$+$
15	$p_3(a,d)$ & $p_3(u,l)$	(1,7)	$+$
16	$p_1(r,w,r)$ & $p_3(a,d)$	(8,1)	$+$
17	$p_2(m,i,i)$ & $p_3(a,d)$	(9,1)	$+$
18	$p_1(a,b,a)$ & $p_2(c,d,d)$	(2,3)	$+$
19	$p_1(a,b,a)$ & $p_2(a,a,a)$	(2,4)	
20	$p_1(a,b,a)$ & $p_1(r,b,t)$	(2,5)	
21	$p_1(a,b,a)$ & $p_3(r,m)$	(2,6)	$+$
22	$p_1(a,b,a)$ & $p_3(u,l)$	(2,7)	$+$
23	$p_1(a,b,a)$ & $p_1(r,w,r)$	(2,8)	
24	$p_1(a,b,a)$ & $p_2(m,i,i)$	(2,9)	$+$
25	$p_2(c,d,d)$ & $p_2(a,a,a)$	(3,4)	
26	$p_1(r,b,t)$ & $p_2(c,d,d)$	(5,3)	$+$
27	$p_2(c,d,d)$ & $p_3(r,m)$	(3,6)	$+$
28	$p_2(c,d,d)$ & $p_3(u,l)$	(3,7)	$+$
29	$p_1(r,w,r)$ & $p_2(c,d,d)$	(8,3)	$+$
30	$p_2(c,d,d)$ & $p_2(m,i,i)$	(3,9)	
31	$p_1(r,b,t)$ & $p_2(a,a,a)$	(5,4)	
32	$p_2(a,a,a)$ & $p_3(r,m)$	(4,6)	$+$
33	$p_2(a,a,a)$ & $p_3(u,l)$	(4,7)	$+$
34	$p_1(r,w,r)$ & $p_2(a,a,a)$	(8,4)	$+$
35	$p_2(a,a,a)$ & $p_2(m,i,i)$	(4,9)	$+$
36	$p_1(r,b,t)$ & $p_3(r,m)$	(5,6)	$+$

37	$p_1(r,b,t)$ & $p_3(u,l)$	(5,7)	+
38	$p_1(r,b,t)$ & $p_1(r,w,r)$	(5,8)	
39	$p_1(r,b,t)$ & $p_2(m,i,i)$	(5,9)	+
40	$p_3(r,m)$ & $p_3(u,l)$	(6,7)	+
41	$p_1(r,w,r)$ & $p_3(r,m)$	(8,6)	+
42	$p_2(m,i,i)$ & $p_3(r,m)$	(9,6)	
43	$p_1(r,w,r)$ & $p_3(u,l)$	(8,7)	+
44	$p_2(m,i,i)$ & $p_3(u,l)$	(9,7)	+
45	$p_1(r,w,r)$ & $p_2(m,i,i)$	(8,9)	+

Номер	**Пара**	**Адреса**	**Подстановки**
46	$p_1(x_1,x_2,x_3)$ & $p_3(x_1,x_4)$	(0,1) (5,6)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to m)$ $(x_1 \to c, x_2 \to s, x_3 \to d, x_4 \to p)$ $(x_1 \to e, x_2 \to f, x_3 \to d, x_4 \to k)$
47	$p_1(x_1,x_2,x_3)$ & $p_3(x_4,x_5)$	(0,6) (0,7) (5,1) (5,7)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l)$

48	$p_1(x_1, x_2, x_3)$ & $p_2(x4, x5, x5)$	(0,9) (5,3) (5,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to m, x_5 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to c, x_5 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to m, x_5 \to i)$
49	$p_1(x_1, x_2, x_1)$ & $p_3(x_1, x_3)$	(2,1) (8,6)	$(x_1 \to a, x_2 \to b, x_3 \to d)$ $(x_1 \to r, x_2 \to w, x_3 \to m)$
50	$p_3(x_1, x_2)$ & $p_3(x_3, x_4)$	(1,6) (1,7) (6,7)	$(x_1 \to a, x_2 \to d, x_3 \to r, x_4 \to m)$ $(x_1 \to a, x_2 \to d, x_3 \to u, x_4 \to l)$ $(x_1 \to r, x_2 \to m, x_3 \to u, x_4 \to l)$
51	$p_1(x_1, x_2, x_1)$ & $p_3(x_3, x_4)$	(8,1) (2,6) (2,7) (8,7)	$(x_1 \to r, x_2 \to w, x_3 \to a, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to l, x_4 \to u)$ $(x_1 \to a, x_2 \to b, x_3 \to u, x_4 \to l)$ $(x_1 \to r, x_2 \to w, x_3 \to u, x_4 \to l)$
52	$p_2(x_1, x_2, x_2)$ & $p_3(x_3, x_4)$	(9,1) (3,6) (3,7) (9,7)	$(x_1 \to m, x_2 \to i, x_3 \to a, x_4 \to d)$ $(x_1 \to c, x_2 \to d, x_3 \to r, x_4 \to m)$ $(x_1 \to c, x_2 \to d, x_3 \to u, x_4 \to l)$ $(x_1 \to m, x_2 \to i, x_3 \to u, x_4 \to l)$

53	$p_1(x_1, x_2, x_1)$ & $p_2(x_3, x_4, x_4)$	(2,3) (2,9) (8,3) (8,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$ $(x_1 \to r, x_2 \to w, x_3 \to c, x_4 \to d)$ $(x_1 \to r, x_2 \to w, x_3 \to m, x_4 \to i)$
54	$p_2(x_1, x_1, x_1)$ & $p_2(x_2, x_3, x_3)$	(4,3) (4,9)	$(x_1 \to a, x_2 \to c, x_3 \to d)$ $(x_1 \to a, x_2 \to m, x_3 \to i)$
55	$p_2(x_1, x_1, x_1)$ & $p_3(x_2, x_3)$	(4,6) (4,7)	$(x_1 \to a, x_2 \to r, x_3 \to m)$ $(x_1 \to a, x_2 \to u, x_3 \to l)$

Упорядочив список по убыванию индекса, мы получаем следующую таблицу:

Номер	**Пара**	**Адреса**	**Подстановки**
1	$p_1(x_1, x_2, x_3)$ & $p_3(x_4, x_5)$	(0,6) (0,7) (5,1) (5,7)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l)$

2	$p_1(x_1, x_2, x_1)$ & $p_3(x_3, x_4)$	(8,1) (2,6) (2,7) (8,7)	$(x_1 \to r, x_2 \to w, x_3 \to a, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to l, x_4 \to u)$ $(x_1 \to a, x_2 \to b, x_3 \to u, x_4 \to l)$ $(x_1 \to r, x_2 \to w, x_3 \to u, x_4 \to l)$
3	$p_2(x_1, x_2, x_2)$ & $p_3(x_3, x_4)$	(9,1) (3,6) (3,7) (9,7)	$(x_1 \to m, x_2 \to i, x_3 \to a, x_4 \to d)$ $(x_1 \to c, x_2 \to d, x_3 \to r, x_4 \to m)$ $(x_1 \to c, x_2 \to d, x_3 \to u, x_4 \to l)$ $(x_1 \to m, x_2 \to i, x_3 \to u, x_4 \to l)$
4	$p_1(x_1, x_2, x_1)$ & $p_2(x_3, x_4, x_4)$	(2,3) (2,9) (8,3) (8,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$ $(x_1 \to r, x_2 \to w, x_3 \to c, x_4 \to d)$ $(x_1 \to r, x_2 \to w, x_3 \to m, x_4 \to i)$
5	$p_1(x_1, x_2, x_3)$ & $p_2(x4, x5, x5)$	(0,9) (5,3) (5,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to m, x_5 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to c, x_5 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to m, x_5 \to i)$

6	$p_3(x_1, x_2)$ & $p_3(x_3, x_4)$	(1,6) (1,7) (6,7)	$(x_1 \to a, x_2 \to d, x_3 \to r, x_4 \to m)$ $(x_1 \to a, x_2 \to d, x_3 \to u, x_4 \to l)$ $(x_1 \to r, x_2 \to m, x_3 \to u, x_4 \to l)$
7	$p_1(x_1, x_2, x_3)$ & $p_3(x_1, x_4)$	(0,1) (5,6)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to m)$ $(x_1 \to c, x_2 \to s, x_3 \to d, x_4 \to p)$ $(x_1 \to e, x_2 \to f, x_3 \to d, x_4 \to k)$
8	$p_1(x_1, x_2, x_1)$ & $p_3(x_1, x_3)$	(2,1) (8,6)	$(x_1 \to a, x_2 \to b, x_3 \to d)$ $(x_1 \to r, x_2 \to w, x_3 \to m)$
9	$p_2(x_1, x_1, x_1)$ & $p_2(x_2, x_3, x_3)$	(4,3) (4,9)	$(x_1 \to a, x_2 \to c, x_3 \to d)$ $(x_1 \to a, x_2 \to m, x_3 \to i)$
10	$p_2(x_1, x_1, x_1)$ & $p_3(x_2, x_3)$	(4,6) (4,7)	$(x_1 \to a, x_2 \to r, x_3 \to m)$ $(x_1 \to a, x_2 \to u, x_3 \to l)$
11	$p_1(a, b, c)$ & $p_1(a, b, a)$	(0,2)	
12	$p_1(a, b, c)$ & $p_2(c, d, d)$	(0,3)	
13	$p_1(a, b, c)$ & $p_2(a, a, a)$	(0,4)	
14	$p_1(a, b, c)$ & $p_1(r, b, t)$	(0,5)	
15	$p_1(a, b, c)$ & $p_1(r, w, r)$	(0,8)	
16	$p_2(c, d, d)$ & $p_3(a, d)$	(3,1)	
17	$p_2(a, a, a)$ & $p_3(a, d)$	(4,1)	
18	$p_1(a, b, a)$ & $p_2(a, a, a)$	(2,4)	
19	$p_1(a, b, a)$ & $p_1(r, b, t)$	(2,5)	
20	$p_1(a, b, a)$ & $p_1(r, w, r)$	(2,8)	
21	$p_2(c, d, d)$ & $p_2(a, a, a)$	(3,4)	
22	$p_2(c, d, d)$ & $p_2(m, i, i)$	(3,9)	

23	$p_1(r,b,t)$ & $p_2(a,a,a)$	(5,4)	
24	$p_1(r,b,t)$ & $p_1(r,w,r)$	(5,8)	
25	$p_2(m,i,i)$ & $p_3(r,m)$	(9,6)	

Возьмем первую пару из списка и введем новый предикатный символ 1-го уровня, соответствующий ей:

$$p_1^1(x_1, x_2, x_3, x_4, x_5) \leftrightarrow p_1(x_1, x_2, x_3) \& p_3(x_4, x_5)$$

Проходим по списку пар в поисках имеющих пересечения с p_1^1. В результате получаем следующее:

1. $p_1(x_1, x_2, x_1)$ & $p_3(x_3, x_4)$; (8,1);(2,6);(2,7);(8,7)
2. $p_2(x_1, x_2, x_2)$ & $p_3(x_3, x_4)$; (9,1);(3,6);(3,7);(9,7)
3. $p_1(x_1, x_2, x_3)$ & $p_2(x4, x5, x5)$; (0,9)
4. $p_3(x_1, x_2)$ & $p_3(x_3, x_4)$; (1,6);(1,7);(6,7)
5. $p_1(x_1, x_2, x_3)$ & $p_3(x_1, x_4)$; (0,1);(5,6)
6. $p_1(x_1, x_2, x_1)$ & $p_3(x_1, x_3)$; (2,1);(8,6)
7. $p_2(x_1, x_1, x_1)$ & $p_3(x_2, x_3)$; (4,6);(4,7)
8. $p_1(a,b,c)$ & $p_1(a,b,a)$; (0,2)
9. $p_1(a,b,c)$ & $p_2(c,d,d)$; (0,3)
10. $p_1(a,b,c)$ & $p_2(a,a,a)$; (0,4)
11. $p_1(a,b,c)$ & $p_1(r,b,t)$; (0,5)
12. $p_1(a,b,c)$ & $p_1(r,w,r)$; (0,8)
13. $p_2(c,d,d)$ & $p_3(a,d)$; (3,1)
14. $p_2(a,a,a)$ & $p_3(a,d)$; (4,1)

15. $p_2(m, i, i)$ & $p_3(r, m)$; (9,6)

Возьмем первую пару и подставим соответствующие первому общему адресу унификаторы: $p_1(r, b, t)$ & $p_3(a, d)$ и $p_1(r, w, r)$ & $p_3(a, d)$. У них есть общий литерал $p_3(a, d)$, поэтому добавляем к временному списку конъюнкцию $p_1^1(r, b, t, a, d)$ & $p_1(r, w, r)$. Пройдя по всем адресам первой пары получим следующий временный список:

1. $p_1^1(r, b, t, a, d)$ & $p_1(r, w, r)$; (5,1,8)
2. $p_1^1(a, b, c, r, m)$ & $p_1(a, b, a)$; (0,6,2)
3. $p_1^1(a, b, c, u, l)$ & $p_1(a, b, a)$; (0,7,2)
4. $p_1^1(r, b, t, u, l)$ & $p_1(a, b, a)$; (5,7,2)
5. $p_1^1(a, b, c, u, l)$ & $p_1(r, w, r)$; (0,7,8)
6. $p_1^1(r, b, t, u, l)$ & $p_1(r, w, r)$; (5,7,8)

Обновляем индексы исходного списка пар: удаляется пара под номером 2, поскольку все ее адреса уже содержается в ранее полученных 6 парах. Вдобавок к ней удаляются пары под номерами 11,15,24 – $p_1(a, b, c)$ & $p_1(a, b, a)$, $p_1(a, b, c)$ & $p_1(r, w, r)$ и $p_1(r, b, t)$ & $p_1(r, w, r)$. Продолжая таким же образом перебирать последующие пары и обновляя индексы пар после добавления очередного множества конъюнкций пересекающихся пар, мы получаем следующий временный список:

1. $p_1^1(r, b, t, a, d)$ & $p_1(r, w, r)$; (5,1,8)
2. $p_1^1(a, b, c, r, m)$ & $p_1(a, b, a)$; (0,6,2)
3. $p_1^1(a, b, c, u, l)$ & $p_1(a, b, a)$; (0,7,2)
4. $p_1^1(r, b, t, u, l)$ & $p_1(a, b, a)$; (5,7,2)
5. $p_1^1(a, b, c, u, l)$ & $p_1(r, w, r)$; (0,7,8)
6. $p_1^1(r, b, t, u, l)$ & $p_1(r, w, r)$; (5,7,8)

7. $p_1^1(r,b,t,a,d)$ & $p_2(m,i,i)$; (5,1,9)

8. $p_1^1(a,b,c,r,m)$ & $p_2(c,d,d)$; (0,6,3)

9. $p_1^1(a,b,c,u,l)$ & $p_2(c,d,d)$; (0,7,3)

10. $p_1^1(r,b,t,u,l)$ & $p_2(c,d,d)$; (5,7,3)

11. $p_1^1(a,b,c,u,l)$ & $p_2(m,i,i)$; (0,7,9)

12. $p_1^1(r,b,t,u,l)$ & $p_2(m,i,i)$; (5,7,9)

13. $p_1^1(r,b,t,a,d)$ & $p_3(r,m)$; (5,1,6)

14. $p_1^1(a,b,c,r,m)$ & $p_3(a,d)$; (0,6,1)

15. $p_1^1(r,b,t,a,d)$ & $p_3(u,l)$; (5,1,7)

16. $p_1^1(a,b,c,u,l)$ & $p_3(a,d)$; (0,7,1)

17. $p_1^1(a,b,c,r,m)$ & $p_3(u,l)$; (0,6,7)

18. $p_1^1(r,b,t,r,m)$ & $p_3(u,l)$; (5,6,7)

19. $p_1^1(r,b,t,a,d)$ & $p_1(a,b,a)$; (5,1,2)

20. $p_1^1(a,b,c,r,m)$ & $p_1(r,w,r)$; (0,6,8)

21. $p_1^1(a,b,c,r,m)$ & $p_2(a,a,a)$; (0,6,4)

22. $p_1^1(a,b,c,u,l)$ & $p_2(a,a,a)$; (0,7,4)

23. $p_1^1(r,b,t,u,l)$ & $p_2(a,a,a)$; (5,7,4)

24. $p_1^1(r,b,t,a,d)$ & $p_2(c,d,d)$; (5,1,3)

25. $p_1^1(r,b,t,a,d)$ & $p_2(a,a,a)$; (5,1,4)

26. $p_1^1(a,b,c,r,m)$ & $p_2(m,i,i)$; (0,6,9)

Выделив из временного списка всевозможные изоморфные пары и добавив результат к исходному списку, проведя сортировку, получаем следующую таблицу:

Номер	Пара	Адреса	Подстановки
1	$p_1(x_1,x_2,x_1)$ & $p_2(x_3,x_4,x_4)$	(2,3) (2,9) (8,3) (8,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$ $(x_1 \to r, x_2 \to w, x_3 \to c, x_4 \to d)$ $(x_1 \to r, x_2 \to w, x_3 \to m, x_4 \to i)$
2	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_2(x_6,x_7,x_7)$	(0,7,9) (5,1,9) (5,7,3) (5,7,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i)$
3	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_3(x_1,x_6)$	(0,6,1) (0,7,1) (5,1,6) (5,7,6)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m, x_6 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l, x_6 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to y, x_4 \to a, x_5 \to d, x_6 \to m)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m)$
4	$p_2(x_1,x_1,x_1)$ & $p_2(x_2,x_3,x_3)$	(4,3) (4,9)	$(x_1 \to a, x_2 \to c, x_3 \to d)$ $(x_1 \to a, x_2 \to m, x_3 \to i)$

5	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_1(x_1,x_2,x_1)$	(0,6,2) (0,7,2)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l)$
6	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_1(x_1,x_6,x_1)$	(5,1,8) (5,7,8)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to w)$
7	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_2(x_3,x_6,x_6)$	(0,6,3) (0,7,3)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m, x_6 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l, x_6 \to d)$
8	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_3(x_1,x_6)$	(0,6,7) (5,1,7)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m, x_6 \to u, x_7 \to l)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to u, x_7 \to l)$
9	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_2(x_1,x_1,x_1)$	(0,6,4) (0,7,4)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m), (x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l)$
10	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_1(x_4,x_2,x_6)$	(0,6,5) (5,1,0)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m, x_6 \to t)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to c)$
11	$p_1(a,b,a)$ & $p_2(a,a,a)$	(2,4)	
12	$p_1(a,b,a)$ & $p_1(r,w,r)$	(2,8)	
13	$p_2(c,d,d)$ & $p_2(m,i,i)$	(3,9)	
14	$p_1(r,w,r)$ & $p_2(a,a,a)$	(8,4)	
15	$p_1^1(a,b,c,u,l)$ & $p_1(r,w,r)$	(0,7,8)	

16	$p_1^1(r,b,t,u,l)$ & $p_1(a,b,a)$	(5,7,2)	
17	$p_1^1(a,b,c,r,m)$ & $p_1(r,w,r)$	(0,6,8)	
18	$p_1^1(r,b,t,a,d)$ & $p_1(a,b,a)$	(5,1,2)	
19	$p_1^1(r,b,t,u,l)$ & $p_2(a,a,a)$	(5,7,4)	
20	$p_1^1(a,b,c,u,l)$ & $p_1(r,b,t)$	(0,7,5)	
21	$p_1^1(r,b,t,a,d)$ & $p_2(c,d,d)$	(5,1,3)	
22	$p_1^1(r,b,t,a,d)$ & $p_2(a,a,a)$	(5,1,4)	
23	$p_1^1(a,b,c,r,m)$ & $p_2(m,i,i)$	(0,6,9)	

Пара с максимальным индексом – $p_1(x_1,x_2,x_1)$ & $p_2(x_3,x_4,x_4)$. Вводим новый предикатный символ 1-го уровня

$$p_2^1(x_1,x_2,x_1,x_3,x_4,x_4) \leftrightarrow p_1(x_1,x_2,x_1)\,\&\,p_2(x_3,x_4,x_4)$$

Список пар, имеющих с p_2^1 пересечение за счет адреса, выглядит следующим образом:

1. $p_2(x_1,x_1,x_1)$ & $p_2(x_2,x_3,x_3)$; (4,3);(4,9)
2. $p_1(a,b,a)$ & $p_2(a,a,a)$; (2,4)
3. $p_1(a,b,a)$ & $p_1(r,w,r)$; (2,8)
4. $p_2(c,d,d)$ & $p_2(m,i,i)$; (3,9)
5. $p_1(r,w,r)$ & $p_2(a,a,a)$; (8,4)

Подставляя унификаторы, соответствующие общим адресам, и проверив получившиеся пары на обладание общим литералом, получаем следующий временный список:

1. $p_2^1(a,b,a,c,d,d)$ & $p_2(a,a,a)$; (2,3,4)
2. $p_2^1(r,w,r,c,d,d)$ & $p_2(a,a,a)$; (8,3,4)

3. $p_2^1(a,b,a,m,i,i)$ & $p_2(a,a,a)$; (2,9,4)

4. $p_2^1(r,w,r,m,i,i)$ & $p_2(a,a,a)$; (8,9,4)

5. $p_2^1(a,b,a,c,d,d)$ & $p_1(r,w,r)$; (2,3,8)

6. $p_2^1(a,b,a,m,i,i)$ & $p_1(r,w,r)$; (2,9,8)

7. $p_2^1(a,b,a,c,d,d)$ & $p_2(m,i,i)$; (2,3,9)

8. $p_2^1(r,w,r,c,d,d)$ & $p_2(m,i,i)$; (8,3,9)

Повторяем предыдущую процедуру:

Номер	**Пара**	**Адреса**	**Подстановки**
1	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_2(x_6,x_7,x_7)$	(0,7,9) (5,1,9) (5,7,3) (5,7,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i)$

2	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_3(x_1,x_6)$	(0,6,1) (0,7,1) (5,1,6) (5,7,6)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m, x_6 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l, x_6 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to y, x_4 \to a, x_5 \to d, x_6 \to m)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m)$
3	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_1(x_1,x_2,x_1)$	(0,6,2) (0,7,2)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l)$
4	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_1(x_1,x_6,x_1)$	(5,1,8) (5,7,8)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to w)$
5	$p_1^1(x_1,x_2,x_3,x_4,x_5)$ & $p_2(x_3,x_6,x_6)$	(0,6,3) (0,7,3)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m, x_6 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l, x_6 \to d)$

6	$p_1^1(x_1, x_2, x_3, x_4, x_5)$ & $p_3(x_1, x_6)$	(0,6,7) (5,1,7)	$(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow c, x_4 \rightarrow r, x_5 \rightarrow m, x_6 \rightarrow u, x_7 \rightarrow l)$ $(x_1 \rightarrow r, x_2 \rightarrow b, x_3 \rightarrow t, x_4 \rightarrow a, x_5 \rightarrow d, x_6 \rightarrow u, x_7 \rightarrow l)$
7	$p_1^1(x_1, x_2, x_3, x_4, x_5)$ & $p_2(x_1, x_1, x_1)$	(0,6,4) (0,7,4)	$(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow c, x_4 \rightarrow r, x_5 \rightarrow m), (x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow c, x_4 \rightarrow u, x_5 \rightarrow l)$
8	$p_1^1(x_1, x_2, x_3, x_4, x_5)$ & $p_1(x_4, x_2, x_6)$	(0,6,5) (5,1,0)	$(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow c, x_4 \rightarrow r, x_5 \rightarrow m, x_6 \rightarrow t)$ $(x_1 \rightarrow r, x_2 \rightarrow b, x_3 \rightarrow t, x_4 \rightarrow a, x_5 \rightarrow d, x_6 \rightarrow c)$
9	$p_2^1(x_1, x_2, x_1, x_3, x_4, x_4)$ & $p_1(x_5, x_6, x_5)$	(2,3,8) (2,9,8)	$(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow c, x_4 \rightarrow d, x_5 \rightarrow r, x_6 \rightarrow w)$ $(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow m, x_4 \rightarrow i, x_5 \rightarrow r, x_6 \rightarrow w)$
10	$p_2^1(x_1, x_2, x_1, x_3, x_4, x_4)$ & $p_2(x_5, x_6, x_6)$	(2,3,9) (8,3,9)	$(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow c, x_4 \rightarrow d, x_5 \rightarrow m, x_6 \rightarrow i)$ $(x_1 \rightarrow r, x_2 \rightarrow w, x_3 \rightarrow c, x_4 \rightarrow d, x_5 \rightarrow m, x_6 \rightarrow i)$

11	$p_2^1(x_1, x_2, x_1, x_3, x_4, x_4)$ & $p_2(x_1, x_1, x_1)$	(2,3,4) (2,9,4)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$
12	$p_2^1(x_1, x_2, x_1, x_3, x_4, x_4)$ & $p_2(x_5, x_5, x_5)$	(8,3,4) (8,9,4)	$(x_1 \to r, x_2 \to w, x_3 \to c, x_4 \to d, x_5 \to a)$ $(x_1 \to r, x_2 \to w, x_3 \to m, x_4 \to i, x_5 \to a)$
13	$p_1^1(a, b, c, u, l)$ & $p_1(r, w, r)$	(0,7,8)	
14	$p_1^1(r, b, t, u, l)$ & $p_1(a, b, a)$	(5,7,2)	
15	$p_1^1(a, b, c, r, m)$ & $p_1(r, w, r)$	(0,6,8)	
16	$p_1^1(r, b, t, a, d)$ & $p_1(a, b, a)$	(5,1,2)	
17	$p_1^1(r, b, t, u, l)$ & $p_2(a, a, a)$	(5,7,4)	
18	$p_1^1(a, b, c, u, l)$ & $p_1(r, b, t)$	(0,7,5)	
19	$p_1^1(r, b, t, a, d)$ & $p_2(c, d, d)$	(5,1,3)	
20	$p_1^1(r, b, t, a, d)$ & $p_2(a, a, a)$	(5,1,4)	
21	$p_1^1(a, b, c, r, m)$ & $p_2(m, i, i)$	(0,6,9)	

Пара с максимальным индексом – $p_1^1(x_1, x_2, x_3, x_4, x_5)$ & $p_2(x_6, x_7, x_7)$. Вводим новый предикатный символ уже 2-го уровня

$$p_1^2(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7) \leftrightarrow p_1^1(x_1, x_2, x_3, x_4, x_5) \,\&\, p_2(x_6, x_7, x_7)$$

Список пар, имеющих с p_1^2 пересечение за счет адреса, выглядит следующим образом:

1. $p_1^1(x_1, x_2, x_3, x_4, x_5)$ & $p_3(x_1, x_6)$; (0,7,1)(5,1,6);(5,7,6)

2. $p_1^1(a, b, c, u, l)$ & $p_1(a, b, a)$; (0,7,2)

3. $p_1^1(x_1, x_2, x_3, x_4, x_5)$ & $p_1(x_1, x_6, x_1)$; (5,1,8);(5,7,8)

4. $p_2^1(x_1, x_2, x_1, x_3, x_4, x_4)$ & $p_2(x_5, x_6, x_6)$; (2,3,9);(8,3,9)

5. $p_1^1(a, b, c, u, l)$ & $p_2(c, d, d)$; (0,7,3)

6. $p_1^1(r,b,t,a,d)$ & $p_3(r,m)$; (5,1,7)
7. $p_1^1(a,b,c,u,l)$ & $p_2(a,a,a)$; (0,7,4)
8. $p_1^1(r,b,t,a,d)$ & $p_1(a,b,c)$; (5,1,0)
9. $p_1^1(a,b,c,u,l)$ & $p_1(r,w,r)$; (0,7,8)
10. $p_1^1(r,b,t,u,l)$ & $p_1(a,b,a)$; (5,7,2)
11. $p_1^1(r,b,t,a,d)$ & $p_1(a,b,a)$; (5,1,2)
12. $p_1^1(r,b,t,u,l)$ & $p_2(a,a,a)$; (5,7,4)
13. $p_1^1(a,b,c,u,l)$ & $p_1(r,b,t)$; (0,7,5)
14. $p_1^1(r,b,t,a,d)$ & $p_2(c,d,d)$; (5,1,3)
15. $p_1^1(r,b,t,a,d)$ & $p_2(a,a,a)$; (5,1,4)

Подставляя унификаторы, соответствующие общим адресам, и проверив получившиеся пары на обладание общим литералом, получаем следующий временный список:

1. $p_1^2(a,b,c,u,l,m,i,i)$ & $p_3(a,d)$; (0,7,9,1)
2. $p_1^2(r,b,t,a,d,m,i,i)$ & $p_3(r,m)$; (5,1,9,6)
3. $p_1^2(r,b,t,u,l,c,d,d)$ & $p_3(r,m)$; (5,7,3,6)
4. $p_1^2(r,b,t,u,l,m,i,i)$ & $p_3(r,m)$; (5,7,9,6)
5. $p_1^2(a,b,c,u,l,m,i,i)$ & $p_3(r,m)$; (0,7,9,6)
6. $p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(a,b,a)$; (0,7,9,2)
7. $p_1^2(r,b,t,a,d,m,i,i)$ & $p_1(r,w,r)$; (5,1,9,8)
8. $p_1^2(r,b,t,u,l,m,i,i)$ & $p_1(r,w,r)$; (5,7,9,8)
9. $p_1^2(r,b,t,u,l,c,d,d)$ & $p_1(r,w,r)$; (5,7,3,8)

10. $p_1^2(a, b, c, u, l, m, i, i)$ & $p_2(c, d, d)$; (0,7,9,3)

11. $p_1^2(r, b, t, a, d, m, i, i)$ & $p_3(r, m)$; (5,1,9,7)

12. $p_1^2(a, b, c, u, l, m, i, i)$ & $p_2(a, a, a)$; (0,7,9,4)

13. $p_1^2(r, b, t, a, d, m, i, i)$ & $p_1(a, b, c)$; (5,1,9,0)

14. $p_1^2(a, b, c, u, l, m, i, i)$ & $p_1(r, w, r)$; (0,7,9,8)

15. $p_1^2(r, b, t, u, l, c, d, d)$ & $p_1(a, b, a)$; (5,7,3,2)

16. $p_1^2(r, b, t, u, l, m, i, i)$ & $p_1(a, b, a)$; (5,7,9,2)

17. $p_1^2(r, b, t, a, d, m, i, i)$ & $p_1(a, b, a)$; (5,1,9,2)

18. $p_1^2(r, b, t, u, l, c, d, d)$ & $p_2(a, a, a)$; (5,7,3,4)

19. $p_1^2(r, b, t, u, l, m, i, i)$ & $p_2(a, a, a)$; (5,7,9,4)

20. $p_1^2(a, b, c, u, l, m, i, i)$ & $p_1(r, b, t)$; (0,7,9,5)

21. $p_1^2(r, b, t, a, d, m, i, i)$ & $p_2(c, d, d)$; (5,1,9,3)

22. $p_1^2(r, b, t, a, d, m, i, i)$ & $p_2(a, a, a)$; (5,1,9,4)

Выделяем изоморфные пары, обновляем индексы: Повторяем процедуру

Номер	**Пара**	**Адреса**	**Подстановки**
1	$p_1^2(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7)$ & $p_1(x_1, x_8, x_1)$	(5,1,9,8) (5,7,3,8) (5,7,9,8)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to w)$

2	$p_2^1(x_1,x_2,x_1,x_3,x_4,x_4)$ & $p_1(x_5,x_6,x_5)$	(2,3,8) (2,9,8)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d, x_5 \to r, x_6 \to w)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i, x_5 \to r, x_6 \to w)$
3	$p_2^1(x_1,x_2,x_1,x_3,x_4,x_4)$ & $p_2(x_5,x_6,x_6)$	(2,3,9) (8,3,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d, x_5 \to m, x_6 \to i)$ $(x_1 \to r, x_2 \to w, x_3 \to c, x_4 \to d, x_5 \to m, x_6 \to i)$
4	$p_2^1(x_1,x_2,x_1,x_3,x_4,x_4)$ & $p_2(x_1,x_1,x_1)$	(2,3,4) (2,9,4)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$
5	$p_2^1(x_1,x_2,x_1,x_3,x_4,x_4)$ & $p_2(x_5,x_5,x_5)$	(8,3,4) (8,9,4)	$(x_1 \to r, x_2 \to w, x_3 \to c, x_4 \to d, x_5 \to a)$ $(x_1 \to r, x_2 \to w, x_3 \to m, x_4 \to i, x_5 \to a)$
6	$p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_3(x_1,x_8)$	(0,7,9,1) (5,7,3,6)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d, x_8 \to m)$
7	$p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_3(x_1,x_6)$	(5,1,9,6) (5,7,9,6)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i)$

8	$p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_1(x_8,x_2,x_8)$	(5,7,3,2) (5,7,9,2)	$x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d, x_8 \to a)$ $x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to a)$
9	$p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_2(x_8,x_8,x_8)$	(5,7,3,4) (5,7,9,4)	$x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d, x_8 \to a)$ $x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to a)$
11	$p_1^1(a,b,c,r,m)$ & $p_1(r,w,r)$	(0,6,8)	
12	$p_1^1(a,b,c,r,m)$ & $p_2(m,i,i)$	(0,6,9)	
13	$p_1^1(a,b,c,r,m)$ & $p_3(a,d)$	(0,6,1)	
14	$p_1^1(a,b,c,r,m)$ & $p_1(a,b,a)$	(0,6,2)	
15	$p_1^1(a,b,c,r,m)$ & $p_2(c,d,d)$	(0,6,3)	
16	$p_1^1(a,b,c,r,m)$ & $p_2(a,a,a)$	(0,6,4)	
17	$p_1^1(a,b,c,r,m)$ & $p_1(r,b,t)$	(0,6,5)	
18	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_3(r,m)$	(0,7,9,6)	
19	$p_1^2(r,b,t,a,d,m,i,i)$ & $p_3(r,m)$	(5,1,9,7)	
20	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(a,b,a)$	(0,7,9,2)	
21	$p_1^2(r,b,t,a,d,m,i,i)$ & $p_1(a,b,a)$	(5,1,9,2)	
22	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(r,w,r)$	(0,7,9,8)	
23	$p_1^2(r,b,t,a,d,m,i,i)$ & $p_1(a,b,c)$	(5,1,9,0)	

24	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(r,b,t)$	(0,7,9,5)	
25	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_2(c,d,d)$	(0,7,9,3)	
26	$p_1^2(r,b,t,a,d,m,i,i)$ & $p_2(c,d,d)$	(5,1,9,3)	
27	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_2(a,a,a)$	(0,7,9,4)	
28	$p_1^2(r,b,t,a,d,m,i,i)$ & $p_2(a,a,a)$	(5,1,9,4)	

Пара с максимальным индексом – $p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_1(x_1,x_8,x_1)$. Вводим новый предикатный символ 3-го уровня

$$p_1^3(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7,x_1,x_8,x_1) \leftrightarrow p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7) \,\&\, p_1(x_1,x_8,x$$

Список пар, имеющих с p_1^3 пересечение за счет адреса, выглядит следующим образом:

1. $p_1^2(r,b,t,u,l,c,d,d)$ & $p_3(r,m)$; (5,7,3,6)

2. $p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_3(x_1,x_6)$; (5,1,9,6);(5,7,9,6)

3. $p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_1(x_8,x_2,x_8)$; (5,7,3,2);(5,7,9,2)

4. $p_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_2(x_8,x_8,x_8)$; (5,7,3,4);(5,7,9,4)

5. $p_1^2(r,b,t,a,d,m,i,i)$ & $p_3(r,m)$; (5,1,9,7)

6. $p_1^2(r,b,t,a,d,m,i,i)$ & $p_1(a,b,a)$; (5,1,9,2)

7. $p_1^2(r,b,t,a,d,m,i,i)$ & $p_1(a,b,c)$; (5,1,9,0)

8. $p_1^2(r,b,t,a,d,m,i,i)$ & $p_2(c,d,d)$; (5,1,9,3)

9. $p_1^2(r,b,t,a,d,m,i,i)$ & $p_2(a,a,a)$; (5,1,9,4)

Подставляя унификаторы, соответствующие общим адресам, и проверив получившиеся пары на обладание общим литералом, получаем следующий временный список:

1. $p_1^3(r, b, t, u, l, c, d, d, r, w, r)$ & $p_3(r, m)$; (5,7,3,8,6)
2. $p_1^3(r, b, t, a, d, m, i, i, r, w, r)$ & $p_3(r, m)$; (5,1,9,8,6)
3. $p_1^3(r, b, t, u, l, m, i, i, r, w, r)$ & $p_3(r, m)$; (5,7,9,8,6)
4. $p_1^3(r, b, t, u, l, c, d, d, r, w, r)$ & $p_1(a, b, a)$; (5,7,3,8,2)
5. $p_1^3(r, b, t, u, l, m, i, i, r, w, r)$ & $p_1(a, b, a)$; (5,7,9,8,2)
6. $p_1^3(r, b, t, u, l, c, d, d, r, w, r)$ & $p_2(a, a, a)$; (5,7,3,8,4)
7. $p_1^3(r, b, t, u, l, m, i, i, r, w, r)$ & $p_2(a, a, a)$; (5,7,9,8,4)
8. $p_1^3(r, b, t, a, d, m, i, i, r, w, r)$ & $p_3(u, l)$; (5,1,9,8,7)
9. $p_1^3(r, b, t, a, d, m, i, i, r, w, r)$ & $p_1(a, b, a)$; (5,1,9,8,2)
10. $p_1^3(r, b, t, a, d, m, i, i, r, w, r)$ & $p_1(a, b, c)$; (5,1,9,8,0)
11. $p_1^3(r, b, t, a, d, m, i, i, r, w, r)$ & $p_2(c, d, d)$; (5,1,9,8,3)
12. $p_1^3(r, b, t, a, d, m, i, i, r, w, r)$ & $p_2(a, a, a)$; (5,1,9,8,4)

Выделяем изоморфные пары, обновляем индексы:

Номер	**Пара**	**Адреса**	**Подстановки**
1	$p_2^1(x_1, x_2, x_1, x_3, x_4, x_4)$ & $p_2(x_1, x_1, x_1)$	(2,3,4) (2,9,4)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$

2	$p_1^3(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7,$ $x_1,x_8,x_1)$ & $p_3(x_1,x_6)$	(5,1,9,8,6) (5,7,9,8,6)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to w)$
3	$p_1^3(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7,$ $x_1,x_8,x_1)$ & $p_1(x_9,x_2,x_9)$	(5,7,3,8,2) (5,7,9,8,2)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d, x_8 \to w, x_9 \to a)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to w, x_9 \to a)$
4	$p_1^3(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7,$ $x_1,x_8,x_1)$ & $p_2(x_9,x_9,x_9)$	(5,7,3,8,4) (5,7,9,8,4)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d, x_8 \to w, x_9 \to a)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to w, x_9 \to a)$
5	$p_1^1(a,b,c,r,m)$ & $p_1(r,w,r)$	(0,6,8)	
6	$p_1^1(a,b,c,r,m)$ & $p_2(m,i,i)$	(0,6,9)	
7	$p_1^1(a,b,c,r,m)$ & $p_3(a,d)$	(0,6,1)	
8	$p_1^1(a,b,c,r,m)$ & $p_1(a,b,a)$	(0,6,2)	
9	$p_1^1(a,b,c,r,m)$ & $p_2(c,d,d)$	(0,6,3)	
10	$p_1^1(a,b,c,r,m)$ & $p_2(a,a,a)$	(0,6,4)	
11	$p_1^1(a,b,c,r,m)$ & $p_1(r,b,t)$	(0,6,5)	

12	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_3(r,m)$	(0,7,9,6)	
13	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(a,b,a)$	(0,7,9,2)	
14	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(r,w,r)$	(0,7,9,8)	
15	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(r,b,t)$	(0,7,9,5)	
16	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_2(c,d,d)$	(0,7,9,3)	
17	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_2(a,a,a)$	(0,7,9,4)	
18	$p_2^1(a,b,a,c,d,d)$ & $p_2(m,i,i)$	(2,3,9)	
19	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_3(a,d)$	(0,7,9,1)	
20	$p_1^3(r,b,t,u,l,c,d,d,r,w,r)$ & $p_3(r,m)$	(5,7,3,8,6)	
21	$p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_3(u,l)$	(5,1,9,8,7)	
22	$p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_1(a,b,a)$	(5,1,9,8,2)	
23	$p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_1(a,b,c)$	(5,1,9,8,0)	
24	$p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_2(c,d,d)$	(5,1,9,8,3)	
25	$p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_2(a,a,a)$	(5,1,9,8,4)	

Пара с максимальным индексом – $p_2^1(x_1,x_2,x_1,x_3,x_4,x_4)$ & $p_2(x_1,x_1,x_1)$. Вводим новый предикатный символ 2-го уровня

$$p_2^2(x_1,x_2,x_1,x_3,x_4,x_4,x_1,x_1,x_1) \leftrightarrow p_2^1(x_1,x_2,x_1,x_3,x_4,x_4)\,\&\,p_2(x_4,x_4,x_4)$$

С p_2^2 имеет пересечение только пара $p_2^1(a,b,a,c,d,d)$ & $p_2(m,i,i)$. Следовательно, единственное изменение будет заключаться в замене пары $p_2^1(a,b,a,c,d,d)$ & $p_2(m,i,i)$ на $p_2^2(a,b,a,c,d,d,a,a,a)$ & $p_2(m,i,i)$.

Следующая пара с максимальным индексом – $p_1^3(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7, \quad x_1,x_8,x_1)$ & $p_3(x_1,x_6)$. Вводим новый предикатный символ 4-го уровня

$$p_1^4(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7,x_1,x_8,x_1,x_1,x_6) \leftrightarrow$$
$$p_1^3(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7,x_1,x_8,x_1) \,\&\, p_3(x_1,x_6)$$

Список пар, имеющих с p_1^4 пересечение за счет адреса, выглядит следующим образом:

1. $p_1^3(r,b,t,u,l,m,i,i,r,w,r)$ & $p_1(a,b,a)$; (5,7,9,8,2)

2. $p_1^3(r,b,t,u,l,m,i,i,r,w,r)$ & $p_2(a,a,a)$; (5,7,9,8,4)

3. $p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_3(u,l)$; (5,1,9,8,7)

4. $p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_1(a,b,a)$; (5,1,9,8,2)

5. $p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_1(a,b,c)$; (5,1,9,8,0)

6. $p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_2(c,d,d)$; (5,1,9,8,3)

7. $p_1^3(r,b,t,a,d,m,i,i,r,w,r)$ & $p_2(a,a,a)$; (5,1,9,8,4)

Подставляя унификаторы, соответствующие общим адресам, и проверив получившиеся пары на обладание общим литералом, получаем следующий временный список:

1. $p_1^4(r,b,t,u,l,m,i,i,r,w,r,r,m)$ & $p_1(a,b,a)$; (5,7,9,8,6,2)

2. $p_1^4(r,b,t,u,l,m,i,i,r,w,r,r,m)$ & $p_2(a,a,a)$; (5,7,9,8,6,4)

3. $p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_3(u,l)$; (5,1,9,8,6,7)

4. $p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_1(a,b,a)$; (5,1,9,8,6,2)

5. $p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_1(a,b,c)$; (5,1,9,8,6,0)

6. $p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_2(c,d,d)$; (5,1,9,8,6,3)

7. $p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_2(a,a,a)$; (5,1,9,8,6,4)

Выделяем изоморфные пары, обновляем индексы:

Номер	**Пара**	**Адреса**	**Подстановки**
1	$p_1^1(a,b,c,r,m)$ & $p_1(a,b,a)$	(0,6,2)	
2	$p_1^1(a,b,c,r,m)$ & $p_2(c,d,d)$	(0,6,3)	
3	$p_1^1(a,b,c,r,m)$ & $p_2(a,a,a)$	(0,6,4)	
4	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_3(r,m)$	(0,7,9,6)	
5	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(a,b,a)$	(0,7,9,2)	
6	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(r,w,r)$	(0,7,9,8)	
7	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_1(r,b,t)$	(0,7,9,5)	
8	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_2(c,d,d)$	(0,7,9,3)	
9	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_2(a,a,a)$	(0,7,9,4)	
10	$p_1^2(a,b,c,u,l,m,i,i)$ & $p_3(a,d)$	(0,7,9,1)	
11	$p_2^2(a,b,a,c,d,d,a,a,a)$ & $p_2(m,i,i)$	(2,3,4,9)	
12	$p_1^3(r,b,t,u,l,c,d,d,r,w,r)$ & $p_1(a,b,a)$	(5,7,3,8,2)	
13	$p_1^3(r,b,t,u,l,c,d,d,r,w,r)$ & $p_2(a,a,a)$	(5,7,3,8,4)	
14	$p_1^3(r,b,t,u,l,c,d,d,r,w,r)$ & $p_3(r,m)$	(5,7,3,8,6)	
15	$p_1^4(r,b,t,u,l,m,i,i,r,w,r,r,m)$ & $p_1(a,b,a)$	(5,7,9,8,6,2)	
16	$p_1^4(r,b,t,u,l,m,i,i,r,w,r,r,m)$ & $p_2(a,a,a)$	(5,7,9,8,6,4)	
17	$p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_3(u,l)$	(5,1,9,8,6,7)	
18	$p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_1(a,b,a)$	(5,1,9,8,6,2)	
19	$p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_1(a,b,c)$	(5,1,9,8,6,0)	

20	$p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_2(c,d,d)$	(5,1,9,8,6,3)	
21	$p_1^4(r,b,t,a,d,m,i,i,r,w,r,r,m)$ & $p_2(a,a,a)$	(5,1,9,8,6,4)	

В итоге получаем следующую систему равносильностей

$\mathrm{p}_1^1(x_1,x_2,x_3,x_4,x_5)$ (0,6) (0,7) (5,1) (5,7)	$\leftrightarrow$	$\mathrm{p}_1(x_1,x_2,x_3)\,\&\,p_3(x_4,x_5)$ $(\mathrm{x}_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l)$
$\mathrm{p}_2^1(x_1,x_2,x_1,x_3,x_4,x_4)$ (2,3) (2,9) (8,3) (8,9)	$\leftrightarrow$	$\mathrm{p}_1(x_1,x_2,x_1)\,\&\,p_2(x_3,x_4,x_4)$ $(\mathrm{x}_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$ $(x_1 \to r, x_2 \to w, x_3 \to c, x_4 \to d)$ $(x_1 \to r, x_2 \to w, x_3 \to m, x_4 \to i)$
$\mathrm{p}_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$	$\leftrightarrow$	$\mathrm{p}_1^1(x_1,x_2,x_3,x_4,x_5)$ $\&\,p_2(x_6,x_7,x_7)$

(0,7,9) (5,1,9) (5,7,3) (5,7,9)		$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i)$
$\mathrm{p}_1^3(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7, x_1, x_8, x_1)$	$\leftrightarrow$	$\mathrm{p}_1^2(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7)$ $\& \, p_1(x_1, x_8, x_1)$
(5,1,9,8) (5,7,3,8) (5,7,9,8)		$(\mathrm{x}_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to w)$
$\mathrm{p}_2^2(x_1, x_2, x_1, x_3, x_4, x_4, x_1, x_1, x_1)$	$\leftrightarrow$	$\mathrm{p}_2^1(x_1, x_2, x_1, x_3, x_4, x_4)$ $\& \, p_2(x_4, x_4, x_4)$
(2,3,4) (2,9,4)		$(\mathrm{x}_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$
$\mathrm{p}_1^4(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7, x_1, x_8, x_1, x_1, x_6)$	$\leftrightarrow$	$\mathrm{p}_1^3(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7, x_1, x_8, x_1) \, \& \, p_3(x_1, x_6)$
(5,1,9,8,6) (5,7,9,8,6)		$(\mathrm{x}_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to w)$

- $p_1(a, b, c)$
- $p_3(a, d)$
- $p_1(a, b, a)$
- $p_2(c, d, d)$
- $p_2(a, a, a)$
- $p_1(r, b, t)$
- $p_3(r, m)$
- $p_3(u, l)$
- $p_1(r, w, r)$
- $p_2(m, i, i)$

Содержимое базы 0-го уровня

Добавляя полученные предикатные символы и пары к содержанию базы данных, получаем:

Пара/Предикат	**Адреса**	**Подстановки**
$p_1^1(x_1,x_2,x_3,x_4,x_5) \leftrightarrow$ $\mathrm{p}_1(x_1,x_2,x_3)$ & $p_3(x_4,x_5)$	(0,6) (0,7) (5,1) (5,7)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to r, x_5 \to m)$ $(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to u, x_5 \to l)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l)$
$p_2^1(x_1,x_2,x_1,x_3,x_4,x_4) \leftrightarrow$ $\mathrm{p}_1(x_1,x_2,x_1)$ & $p_2(x_3,x_4,x_4)$	(2,3) (2,9) (8,3) (8,9)	$(x_1 \to a, x_2 \to b, x_3 \to c, x_4 \to d)$ $(x_1 \to a, x_2 \to b, x_3 \to m, x_4 \to i)$ $(x_1 \to r, x_2 \to w, x_3 \to c, x_4 \to d)$ $(x_1 \to r, x_2 \to w, x_3 \to m, x_4 \to i)$
$p_1^1(a,b,c,r,m)$ & $p_1(a,b,a)$	(0,6,2)	
$p_1^1(a,b,c,r,m)$ & $p_2(c,d,d)$	(0,6,3)	
$p_1^1(a,b,c,r,m)$ & $p_2(a,a,a)$	(0,6,4)	

Содержимое базы 1-го уровня

Пара/Предикат	**Адреса**	**Подстановки**
$p_1^2(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7) \leftrightarrow \mathrm{p}_1^1(x_1, x_2, x_3, x_4, x_5) \ \ \& \ \ p_2(x_6, x_7, x_7)$	(0,7,9) (5,1,9) (5,7,3) (5,7,9)	$(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow c, x_4 \rightarrow u, x_5 \rightarrow l, x_6 \rightarrow m, x_7 \rightarrow i)$ $(x_1 \rightarrow r, x_2 \rightarrow b, x_3 \rightarrow t, x_4 \rightarrow a, x_5 \rightarrow d, x_6 \rightarrow m, x_7 \rightarrow i)$ $(x_1 \rightarrow r, x_2 \rightarrow b, x_3 \rightarrow t, x_4 \rightarrow u, x_5 \rightarrow l, x_6 \rightarrow c, x_7 \rightarrow d)$ $(x_1 \rightarrow r, x_2 \rightarrow b, x_3 \rightarrow t, x_4 \rightarrow u, x_5 \rightarrow l, x_6 \rightarrow m, x_7 \rightarrow i)$
$p_2^2(x_1, x_2, x_1, x_3, x_4, x_4, x_1, x_1, x_1) \leftrightarrow \mathrm{p}_2^1(x_1, x_2, x_1, x_3, x_4, x_4) \ \ \& \ \ p_2(x_4, x_4, x_4)$	(2,3,4) (2,9,4)	$(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow c, x_4 \rightarrow d)$ $(x_1 \rightarrow a, x_2 \rightarrow b, x_3 \rightarrow m, x_4 \rightarrow i)$
$p_1^2(a, b, c, u, l, m, i, i)$ & $p_3(r, m)$	(0,7,9,6)	
$p_1^2(a, b, c, u, l, m, i, i)$ & $p_1(a, b, a)$	(0,7,9,2)	
$p_1^2(a, b, c, u, l, m, i, i)$ & $p_1(r, w, r)$	(0,7,9,8)	
$p_1^2(a, b, c, u, l, m, i, i)$ & $p_1(r, b, t)$	(0,7,9,5)	
$p_1^2(a, b, c, u, l, m, i, i)$ & $p_2(c, d, d)$	(0,7,9,3)	
$p_1^2(a, b, c, u, l, m, i, i)$ & $p_2(a, a, a)$	(0,7,9,4)	
$p_1^2(a, b, c, u, l, m, i, i)$ & $p_3(a, d)$	(0,7,9,1)	
$p_2^2(a, b, a, c, d, d, a, a, a)$ & $p_2(m, i, i)$	(2,3,4,9)	

Содержимое базы 2-го уровня

Пара/Предикат	**Адреса**	**Подстановки**
$p_1^3(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7,x_1,x_8,x_1)$ $\leftrightarrow \mathrm{p}_1^2(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_7)$ & $p_1(x_1,x_8,x_1)$	(5,1,9,8) (5,7,3,8) (5,7,9,8)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to c, x_7 \to d, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to w)$
$p_1^3(r,b,t,u,l,c,d,d,r,w,r)$ & $p_1(a,b,a)$	(5,7,3,8,2)	
$p_1^3(r,b,t,u,l,c,d,d,r,w,r)$ & $p_2(a,a,a)$	(5,7,3,8,4)	
$p_1^3(r,b,t,u,l,c,d,d,r,w,r)$ & $p_3(r,m)$	(5,7,3,8,6)	

Содержимое базы 3-го уровня

Пара/Предикат	**Адреса**	**Подстановки**
$p_1^4(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7, x_1, x_8, x_1, x_1, x_6) \leftrightarrow \mathrm{p}_1^3(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_7, x_1, x_8, x_1)$ & $p_3(x_1, x_6)$	(5,1,9,8,6) (5,7,9,8,6)	$(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to a, x_5 \to d, x_6 \to m, x_7 \to i, x_8 \to w)$ $(x_1 \to r, x_2 \to b, x_3 \to t, x_4 \to u, x_5 \to l, x_6 \to m, x_7 \to i, x_8 \to w)$
$p_1^4(r, b, t, u, l, m, i, i, r, w, r, r, m)$ & $p_1(a, b, a)$	(5,7,9,8,6,2)	
$p_1^4(r, b, t, u, l, m, i, i, r, w, r, r, m)$ & $p_2(a, a, a)$	(5,7,9,8,6,4)	
$p_1^4(r, b, t, a, d, m, i, i, r, w, r, r, m)$ & $p_3(u, l)$	(5,1,9,8,6,7)	
$p_1^4(r, b, t, a, d, m, i, i, r, w, r, r, m)$ & $p_1(a, b, a)$	(5,1,9,8,6,2)	
$p_1^4(r, b, t, a, d, m, i, i, r, w, r, r, m)$ & $p_1(a, b, c)$	(5,1,9,8,6,0)	
$p_1^4(r, b, t, a, d, m, i, i, r, w, r, r, m)$ & $p_2(c, d, d)$	(5,1,9,8,6,3)	
$p_1^4(r, b, t, a, d, m, i, i, r, w, r, r, m)$ & $p_2(a, a, a)$	(5,1,9,8,6,4)	

Содержимое базы 4-го уровня