



**«Московский государственный технический
университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**

Факультет «Информатика и системы управления»
Кафедра «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №16
по курсу:
«Функциональное и Логическое программирование»

Студент группы ИУ7-63Б: Фурдик Н. О.
(Фамилия И.О.)

Преподаватель: Толпинская Н. Б., Строганов Ю. В.
(Фамилия И.О.)

Оглавление

Постановка задачи	2
Листинг программы	3
Описание порядка поиска ответов	4
Ответы на вопросы	6
Список литературы	8

Постановка задачи

Создать базу знаний: «ПРЕДКИ», позволяющую наиболее эффективным способом (за меньшее количество шагов, что обеспечивается меньшим количеством предложений БЗ - правил), используя разные варианты (примеры) одного вопроса, **определить** (указать: какой вопрос для какого варианта):

- 1) по имени субъекта определить всех его бабушек (предки 2-го колена),
- 2) по имени субъекта определить всех его дедушек (предки 2-го колена),
- 3) по имени субъекта определить всех его бабушек и дедушек (предки 2-го колена),
- 4) по имени субъекта определить его бабушку по материнской линии (предки 2-го колена),
- 5) по имени субъекта определить его бабушку и дедушку по материнской линии (предки 2-го колена).

Минимизировать количество правил и количество вариантов вопросов. Использовать конъюнктивные правила и простой вопрос. Для одного из вариантов **ВОПРОСА и конкретной БЗ** составить таблицу, отражающую конкретный порядок работы системы, с объяснениями: очередная проблема на каждом шаге и метод ее решения; каково новое текущее состояние резольвенты, как получено; какие дальнейшие действия? (Запускается ли алгоритм унификации? Каких термов? Почему этих?); вывод по результатам очередного шага и дальнейшие действия.

Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты!

Листинг программы

Ниже представлен листинг программы:

```
predicates

mom_or_dad(symbol name, symbol sex, symbol child_name)
grandparents(symbol name, symbol sex, symbol grandparent_name, symbol parent_sex)

clauses

mom_or_dad(anya, w, oleg).
mom_or_dad(vasya, m, oleg).
mom_or_dad(masha, w, anya).
mom_or_dad(anton, m, anya).
mom_or_dad(zhenya, m, vasya).

grandparents(X, Y, Z, B) :-mom_or_dad(Z, Y, A), mom_or_dad(A, B, X).

goal

%grandmom
grandparents(oleg, w, Grandmother_name, _).

%Grandmother_name=masha
%1 Solutions

%granddad
grandparents(oleg, m, Grandfather_name, _).

%Grandfather_name=anton
%Grandfather_name=zhenya
%2 Solution

%grandparents
grandparents(oleg, _, Grandparent_name, _).

%Grandparent_name=masha
%Grandparent_name=anton
%Grandparent_name=nastya
%3 Solutions

%moms moms
grandparents(oleg, w, Grandmother_name, w).

%Grandmother_name=masha
%1 Solution

%moms dads
```

```
grandparents(oleg, m, Grandfather_name, w).
```

```
%Grandfather_name=anton
```

```
%1 Solution
```

Листинг 1: Задания 1 и 2

Описание работы системы

Ниже представлен алгоритм поиска ответов на вопрос `grandparents(oleg, w, Grandmother_name, _)`. Всего должен получиться один ответ.

Таблица 1: Описание работы системы на примере поиска бабушек

№ ша- га	Состояние резольвенты, и вы- вод: дальнейшие действия (по- чему?)	Для каких термов запускается алгоритм унификации: T1=T2 и каков результат (и подста- новка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	grandparents(oleg, w, Grandmother_name, _)		прямой ход
1	grandparents(oleg, w, Grandmother_name, _)	Находим подходящее правило grandparents(X, Y, Z, B) и связываем (подставляем) пе- ременные: X=oleg, Y=w, Z= Grandmother_name	прямой ход
2	mom_or_dad(Z, Y, A) grandparents(oleg, w, Grandmother_name, _)	Подставляем Z= Grandmother_name, Y=w в mom_or_dad(Z, Y, A). Получаем пример mom_or_dad(Grandmother_name, w, A)	прямой ход
2	mom_or_dad(Z, Y, A) grandparents(oleg, w, Grandmother_name, _)	Поиск совпадений по БЗ, на- ходим mom_or_dad(anya, w, oleg). Z = anya, A = oleg	прямой ход
3	mom_or_dad(A, B, X) grandparents(oleg, w, Grandmother_name, _)	Подставляем A = oleg, X=oleg в mom_or_dad(A, B, X). Получаем пример mom_or_dad(oleg, B, oleg)	прямой ход
3	mom_or_dad(A, B, X) grandparents(oleg, w, Grandmother_name, _)	Поиск совпадений по БЗ, ниче- го не находим	откат, снова решаем предыду- щий вопрос mom_or_dad(Z, Y, A)
4	mom_or_dad(Z, Y, A) grandparents(oleg, w, Grandmother_name, _)	Поиск совпадений по БЗ, на- ходим mom_or_dad(masha, w, anya). Z = masha, A = anya	прямой ход
4	mom_or_dad(Z, Y, A) grandparents(oleg, w, Grandmother_name, _)	Подставляем A = anya, X=oleg в mom_or_dad(A, B, X). Получаем пример mom_or_dad(anya, B, oleg)	прямой ход
5	mom_or_dad(A, B, X) grandparents(oleg, w, Grandmother_name, _)	Поиск совпадений по БЗ, на- ходим mom_or_dad(anya, w, oleg). B = w	прямой ход
6	grandparents(oleg, w, Grandmother_name, _)	Подстановка Grandmother_name=masha, ответ получен.	откат, поскольку не дошли до конца БЗ
7	mom_or_dad(Z, Y, A) grandparents(oleg, w, Grandmother_name, _)	Поиск совпадений по БЗ, ниче- го не находим	прямой ход
Вывод:	Grandmother_name=masha		конец

Ответы на вопросы

- 1) **В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)**

Система запускает унификацию в том случае, если ей был задан вопрос. Унификация вопроса и первого предложения базы знаний происходит на первом шаге работы программы.

- 2) **Каковы назначение и результат использования алгоритма унификации?**

Алгоритм унификации необходим для попытки "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией. Результат алгоритма унификации – ответ «да» или «нет».

- 3) **Какое первое состояние резольвенты?**

Изначально резольвента пуста.

- 4) **Как меняется резольвента?**

На каждом шаге имеется некоторая совокупность целей - утверждений, истинность (выводимость) которых надо доказать. Эта совокупность называется резольventой - её состояние меняется в процессе доказательства (Для хранения резольвенты система использует стек). Новая резольвента образуется в два этапа:

- в текущей резольвенте выбирается одна из подцелей (по стековому принципу - верхняя) и для неё выполняется редукция - замена подцели на тело найденного (подобранного, если удалось) правила (а как подбирается правило?),
- затем, к полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели (выбранной) и заголовок сопоставленного с ней правила.

- 5) **В каких пределах программы уникальны переменные?**

Именованные переменные уникальны в рамках одного предложения. Анонимная переменная уникальна всегда. Переменные предназначены для пе-

редачи значений «во времени и в пространстве».

6) **Как применяется подстановка, полученная с помощью алгоритма унификации?**

Пока стек не пуст – **цикл**:

- считать из стека в рабочую область очередное равенство $S=T$
 - обработать считанное по правилам:
 - если S и T несовпадающие константы, то неудача=1, и выход из цикла
 - если одинаковые константы то следующий шаг цикла
 - если S переменная и T терм содержащий S , то неудача=1, и выход из цикла
 - если S переменная и T терм НЕ содержащий S , то отыскать в стеке и в результирующей ячейке все вхождения S и заменить на T . Добавить в результирующую ячейку равенство $S=T$. Следующий шаг цикла
 - если S и T составные термы с разными функторами или разными арностями, то неудача=1, выход из цикла
 - если S и T составные термы с одинаковыми функторами и арностью:
 $S = f(s_1, s_2, \dots, s_m); T = f(t_1, t_2, \dots, t_m)$, то занести в стек равенство
 $S_1 = T_1, S_2 = T_2, \dots, S_m = T_m$.
 - очистить рабочее поле
- **конец цикла**

7) **В каком случае запускается механизм отката?**

Механизм отката запускается в 2 случаях:

- Если алгоритм попал в тупиковую ситуацию.
- Если резольвента не пуста и решение найдено, но в базе знание остались не отмеченные предложения.

Литература

1. Толпинская Н.Б. - Курс лекций по "Функциональному и Логическому программированию"[Текст], Москва 2020 год.
2. Анатолий Адаменко, Андрей Кучуков. Логическое программирование и Visual Prolog (с CD). — СПб.: БХВ-Петербург, 2003. — 990 с. — ISBN 5-94157-156-9.