



**«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 №15
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	8
Список литературы	11

Постановка задачи

Создать базу знаний «**Собственники**», дополнив базу знаний, хранящую знания (лаб. 13):

- 1) «Телефонный справочник»: Фамилия, №тел, Адрес – структура (Город, Улица, №дома, №кв);
- 2) «Автомобили»: Фамилия владельца, Марка, Цвет, Стоимость, и др;
- 3) «Вкладчики банков»: Фамилия, Банк, счет, сумма, др.

знаниями о дополнительной собственности владельца. Преобразовать знания об автомобиле к форме знаний о собственности.

Вид собственности (кроме автомобиля):

- 1) Строение, стоимость и другие его характеристики;
- 2) Участок, стоимость и другие его характеристики;
- 3) Водный_транспорт, стоимость и другие его характеристики.

Описать и использовать вариантный домен: Собственность. Владелец может иметь, но только один объект каждого вида собственности (это касается и автомобиля), или не иметь некоторых видов собственности.

Используя конъюнктивное правило и разные формы задания одного вопроса (пояснять для какого №задания – какой вопрос), **обеспечить возможность поиска:**

- 1) Названий всех объектов собственности заданного субъекта,
- 2) Названий и стоимости всех объектов собственности заданного субъекта,
- 3) * Разработать правило, позволяющее найти суммарную стоимость всех объектов собственности заданного субъекта.

Для 2-го пункта и одной фамилии составить таблицу, отражающую конкретный порядок работы системы, с объяснениями порядка работы и особенностей использования доменов (указать конкретные T1 и T2 и полную подстановку на каждом шаге).

При желании, можно усложнить свою базу знаний, введя варианты: строение: (Дом, офис, торговый центр), участок: (садовый, территория под застройку, территория под агро-работы), Водный_транспорт: варианты названий.

Листинг программы

Ниже представлен листинг программы:

```
domains

addr = ad(symbol town, symbol street, integer building, integer apartment)
property = building(symbol name, integer cost, addr);
           place(symbol name, integer cost, addr, string date);
           transport(symbol name, integer cost, string date);
           car(symbol brand, symbol color, integer cost, integer year, symbol town)

predicates

person(symbol l_name, symbol ph_number, addr, property)
get_property(symbol name, symbol property, integer cost)
get_building_cost(symbol name, integer cost)
get_place_cost(symbol name, integer cost)
get_car_cost(symbol name, integer cost)
get_transport_cost(symbol name, integer cost)
get_sum(symbol name, integer sum)

clauses

person(klyuge, "+7912571026", ad("volgograd", "varshavshkoe sh.", 52, 83),
       building(barn, 3213124, ad("volgograd", "varshavshkoe sh.", 11, 1))).
person(toporova, "+79165072034", ad("saint—petersburg", "varshavshkoe sh.", 11, 158),
       building(shed, 123124, ad("saint—petersburg", "varshavshkoe sh.", 11, 17))).
person(klyuge, "+7912571026", ad("volgograd", "varshavshkoe sh.", 52, 83),
       place(appartment, 12312312, ad("volgograd", "varshavshkoe sh.", 52, 83), "11.11.1999")).
person(zayceva, "+79138767153", ad("kasimov", "lermontovskiy pr.", 83, 578),
       car("geely", "white", 6409914, 2015, "kasimov")).
person(nikitin, "+7913336524", ad("kiev", "pushkina", 86, 973),
       car("geely", "black", 58439837, 2011, "ryazan")).
person(klyuge, "+7912571026", ad("volgograd", "varshavshkoe sh.", 52, 83),
       transport("water_bike", 213124, "21.11.2007")).
person(toporova, "+79165072034", ad("saint—petersburg", "varshavshkoe sh.", 33, 45),
       car("mitsubishi", "green", 12836628, 2018, "saint—petersburg")).

get_property(X, Y, Z) :-person(X, _, _, building (Y,Z,_)).
get_property(X, Y, Z) :-person(X, _, _, place (Y,Z,_,_)).
get_property(X, Y, Z) :-person(X, _, _, car(Y,_,Z,_,_)).
get_property(X, Y, Z) :-person(X, _, _, transport(Y,Z,_)).

get_building_cost(X, Y) :-person(X, _, _, building (_,Y,_)), !.
get_building_cost(_, 0).

get_place_cost(X, Y) :-person(X, _, _, place (_,Y,_,_)), !.
get_place_cost(_, 0).

get_car_cost(X, Y) :-person(X, _, _, car(_,_,Y,_,_)), !.
get_car_cost(_, 0).
```

```

get_transport_cost(X, Y) :-person(X, _, _, transport(_,Y,_)), !.
get_transport_cost(_, 0).

get_sum(X, Sum) :-get_building_cost(X, A), get_place_cost(X, B),
                  get_car_cost(X, C), get_transport_cost(X, D),
                  Sum = A + B + C + D.

goal

%property
get_property(Name, Property, _), Name = klyuge.

%Name=klyuge, Property=barn
%Name=klyuge, Property=apartment
%Name=klyuge, Property=water_bike
%3 Solutions

%property & cost
get_property(Name, Property, Cost), Name = klyuge.

%Name=klyuge, Property=barn, Cost=3213124
%Name=klyuge, Property=apartment, Cost=12312312
%Name=klyuge, Property=water_bike, Cost=213124
%3 Solutions

%sum of cost
get_sum(klyuge, Sum).

%Sum=15738560
%1 Solution

```

Листинг 1: Задания 1 и 2

Описание работы системы

Ниже представлен алгоритм поиска ответов на вопрос `get_property(Name, Property, Cost), Name = klyuge`. Всего ответов должно получиться 3.

Таблица 1: Описание работы системы на примере вывода названия собственности и ее цены

№ ша- га	Сравниваемые термы; подста- новка, если есть	Результат	Дальнейшие действия: прямой ход или откат (к чему приво- дит?)
1	Подстановка (связывание) Name = klyuge в запрос get_property(Name, Property, Cost)	get_property(klyuge, Property, Cost)	прямой ход
2	Поиск совпадений и подста- новка (связывание) klyuge, Property и Cost в соответству- ющее правило get_property(X, Y, Z) :- person(X,_, _, building (Y,Z,_))	get_property(klyuge, Property, Cost)	прямой ход
3	Подстановка X = klyuge, Y = Property и Z = Cost в person(X, _,_, building (Y,Z,_))	Создается пример person(klyuge, _, _, building (Property,Cost,_))	прямой ход
4	Поиск person по примеру	person(klyuge, '+7912571026', ad('volgograd', 'varshavshkoe sh.', 52, 83), building(barn, 3213124, ad('volgograd', 'varshavshkoe sh.', 11, 1))), Property = building, Cost = 3213124	прямой ход
5	Общий пример найден, вывод	Name = klyuge, Property = building, Cost = 3213124	прямой ход
6	Поиск person по примеру	person(toporova, '+79165072034', ad('saint- petersburg', 'varshavshkoe sh.', 11, 158), building(shed, 123124, ad('saint-petersburg', 'varshavshkoe sh.', 11, 17))), не подходит	прямой ход
7	Поиск person по примеру	...	прямой ход
8	Поиск person по примеру	person(toporova, '+79165072034', ad('saint- petersburg', 'varshavshkoe sh.', 33, 45), car('mitsubishi', "green 12836628, 2018, 'saint- petersburg')), не подходит	откат, т.к дошли до конца БЗ (следовательно, тупик)
9	Поиск совпадений и подста- новка (связывание) klyuge, Property и Cost в соответству- ющее правило get_property(X, Y, Z) :- person(X,_, _, place (Y,Z,_,_))	get_property(klyuge, Property, Cost)	прямой ход

10	Подстановка $X = klyuge$, $Y = \text{Property}$ и $Z = \text{Cost}$ в $\text{person}(X, _, _, \text{place}(Y, Z, _, _))$	Создается пример $\text{person}(klyuge, _, _, \text{building}(\text{Property}, \text{Cost}, _, _))$	прямой ход
11	Поиск person по примеру	$\text{person}(klyuge, '+7912571026', \text{ad}('volgograd', 'varshavshkoe sh.', 52, 83), \text{building}(\text{barn}, 3213124, \text{ad}('volgograd', 'varshavshkoe sh.', 11, 1)))$, не подходит	прямой ход
12	Поиск person по примеру	...	прямой ход
13	Поиск person по примеру	$\text{person}(klyuge, '+7912571026', \text{ad}('volgograd', 'varshavshkoe sh.', 52, 83), \text{place}(\text{apartment}, 12312312, \text{ad}('volgograd', 'varshavshkoe sh.', 52, 83), '11.11.1999')), \text{Property} = \text{apartment}, \text{Cost} = 12312312$	прямой ход
14	Общий пример найден, вывод	$\text{Name} = klyuge, \text{Property} = \text{apartment}, \text{Cost} = 12312312$	прямой ход
15	Поиск person по примеру	$\text{person}(zayceva, '+79138767153', \text{ad}('kasimov', 'lermontovskiy pr.', 83, 578), \text{car}('geely', 'white', 6409914, 2015, 'kasimov'))$, не подходит	прямой ход
16	Поиск person по примеру	...	прямой ход
17	Поиск person по примеру	$\text{person}(\text{toporova}, '+79165072034', \text{ad}('saint-petersburg', 'varshavshkoe sh.', 33, 45), \text{car}('mitsubishi', 'green', 12836628, 2018, 'saint-petersburg'))$, не подходит	откат, т.к дошли до конца БЗ (следовательно, тупик)
18	Поиск совпадений и подстановка (связывание) $klyuge, \text{Property}$ и Cost в соответствующее правило $\text{get_property}(X, Y, Z) :- \text{person}(X, _, _, \text{person}(X, _, _, \text{car}(Y, _, Z, _, _)))$	$\text{get_property}(klyuge, \text{Property}, \text{Cost})$	прямой ход
19	Подстановка $X = klyuge$, $Y = \text{Property}$ и $Z = \text{Cost}$ в $\text{person}(X, _, _, \text{car}(Y, _, Z, _, _))$	Создается пример $\text{person}(klyuge, _, _, \text{car}(\text{Property}, _, \text{Cost}, _, _))$	прямой ход

20	Поиск person по примеру	person(klyuge, '7912571026', ad('volgograd', 'varshavshkoe sh.', 52, 83), building(barn, 3213124, ad('volgograd', 'varshavshkoe sh.', 11, 1))), не подходит	прямой ход
21	Поиск person по примеру	...	прямой ход
22	Поиск person по примеру	person(toporova, '79165072034', ad('saint- petersburg', 'varshavshkoe sh.', 33, 45), car('mitsubishi', 'green', 12836628, 2018, 'saint- petersburg')), не подходит	откат, т.к дошли до конца БЗ (следовательно, тупик)
23	Поиск совпадений и подста- новка (связывание) klyuge, Property и Cost в соответству- ющее правило get_property(X, Y, Z) :- person(X,_, _ , transport(Y,Z,_))	get_property(klyuge, Property, Cost)	прямой ход
24	Подстановка X = klyuge, Y = Property и Z = Cost в person(X, _,_, transport (Y,Z,_))	Создается пример person(klyuge, _, _, building (Property,Cost,_))	прямой ход
25	Поиск person по примеру	person(klyuge, '7912571026', ad('volgograd', 'varshavshkoe sh.', 52, 83), building(barn, 3213124, ad('volgograd', 'varshavshkoe sh.', 11, 1))), не подходит	прямой ход
26	Поиск person по примеру	...	прямой ход
27	Поиск person по примеру	person(klyuge, '7912571026 ad('volgograd', 'varshavshkoe sh.', 52, 83), transport('water_bike', 213124, '21.11.2007'))., Property = 'water_bike', Cost = 213124	прямой ход
28	Общий пример найден, вывод	Name = klyuge, Property = 'water_bike', Cost = 213124	прямой ход
29	Поиск person по примеру	person(toporova, '79165072034', ad('saint- petersburg', 'varshavshkoe sh.', 33, 45), car('mitsubishi', 'green', 12836628, 2018, 'saint- petersburg')), не подходит	откат, т.к дошли до конца БЗ (следовательно, тупик)
Вывод:	подстановка	Т.к. стек пуст - успех и рез. ячейке подстановка	

Ответы на вопросы

1) В каком фрагменте программы сформулировано знание?

Правило - это предложение, имеющее следующий вид: $A : -B_1, \dots, B_n$, где A называется заголовком правила, а B_1, \dots, B_n - телом правила. Заголовок содержит отдельное знание о предметной области (составной терм), а тело содержит условия истинности этого знания. Правило называют условной истиной, а факт, не содержащий тела - безусловной истиной.

Заголовок, как составной терм $f(t_1, t_2, \dots, t_m)$, содержит знание о том, что между аргументами: t_1, t_2, \dots, t_m существует отношение (взаимосвязь, взаимозависимость). А имя этого отношения - это f . Например, `school(judy, high_school)`

2) Что содержит тело правила?

Тело правила содержит условие истинности заголовка правила.

3) Что дает использование переменных при формулировании знаний? В чем отличие формулировки знания с помощью термов с одинаковой арностью при использовании одной переменной и при использовании нескольких переменных?

Использование переменных в формулировании знаний позволяют уточнять значения и переносить их в пространстве и времени.

Формулировка знаний с использованием переменных носит более общий характер по отношению к знанию, состоящему только лишь из констант. Например, использование знаний с одинаковой арностью при использовании одной переменной носит менее общий характер по отношению знания с использованием нескольких переменных.

4) С каким квантором переменные входят в правило, в каких пределах переменная уникальна?

Переменные входят в правило с квантором всеобщности (для любой). Именованные переменные уникальны в пределах одного предложения, анонимные уникальны все.

5) Какова семантика (смысл) предложений раздела DOMAINS? Когда, где и с какой целью используется это описание?

Домены дают возможность присваивать различным видам информации, которая в противном случае выглядела бы одинаково, отличные имена. В программе Visual Prolog объекты в отношении (аргументы предиката) принадлежат доменам; это могут быть домены стандартные или специальные, определяемые программистами.

Раздел `domains` служит двум целям. Во-первых, можно определить для доменов осмысленные имена, причем даже в том случае, если внутренне они совпадают с именами уже существующих доменов. Во-вторых, объявления специальных доменов используются для объявления структур данных, которые стандартными доменами не определяются.

Иногда целесообразно объявить домен тогда, когда возникает потребность более четкого выделения каких-либо частей раздела `predicates`. Объявление программистом своих собственных доменов помогает документировать предикаты, которые определяются путем задания в качестве типа аргумента удобного и понятного имени.

6) Какова семантика (смысл) предложений раздела PREDICATES? Когда, и где используется это описание? С какой целью?

Некоторые предикаты уже известны системе, они называются стандартными или встроенными. Если программист определяет в разделе `clauses` свой собственный предикат, то он должен объявить его в разделе `predicates`. В противном случае Visual Prolog не будет знать, о чем идет речь. Когда объявляется предикат, Прологу сообщается о том, к каким доменам принадлежат аргументы этого предиката. Предикаты определяются фактами и правилами. В разделе `predicates` перечисляется каждый предикат с указанием доменов аргументов.

7) Унификация каких термов запускается на самом первом шаге работы системы? Каковы назначение и результат использования алгоритма унификации?

На первом шаге работы происходит унификация вопроса и первого предложения базы знаний. Алгоритм унификации необходим для попытки "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией. Результат унификации – ответ «да» или

«нет».

8) 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.