



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчёт по лабораторной работе №13 по курсу «Функциональное и логическое программирование»

Тема Структура программы на Prolog и ее реализация

Студент Прянишников А.Н.

Группа ИУ7-65Б

Оценка (баллы) _____

Преподаватели Строганов Ю. В., Толпинская Н. Б.

Практическая часть

Создать базу знаний «Собственники», дополнив (и минимально изменив) базу знаний, хранящую знания:

1. **«Телефонный справочник»:** фамилия, номер телефона, адрес – структура (город, улица, № дома, № кв).
2. **«Автомобили»:** фамилия владельца, марка, цвет, стоимость.
3. **«Вкладчики банков»:** Фамилия владельца, банк, счет, сумма.

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.

Преобразовать знания об автомобиле к форме знаний о собственности. Виды собственности (кроме автомобиля):

1. **Строение:** стоимость.
2. **Участок:** стоимость.
3. **Водный транспорт:** стоимость.

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).

На листинге 1 представлен код программы:

```
1 domains
2     surname = string
3     phoneNumber = integer
4     city, street, carColor = string
5     house, room = integer
6     address = address(city, street, house, room)
7     bankName, account = string
8     sum = integer
9     price = integer
10    name = string
11    ownerSubject = building(name, price); region(name, price); transport(name, price);
12    car(name, carColor, price).
13    int_list = integer*
14 predicates
15     nondeterm phoneRecord(surname, phoneNumber, address)
16     nondeterm owner(surname, ownerSubject)
17     nondeterm bankRecord(surname, bankName, account, sum)
18     nondeterm allPersonObj(surname, name)
19     nondeterm allPersonObjWithPrice(surname, name, price)
20     nondeterm allPersonSumObj(surname, price)
21     nondeterm sumlist(int_list, price)
22     nondeterm merge_list(int_list, int_list, int_list)
23 clauses
24     phoneRecord("Priianishnikov", 890801, address("Moscow", "Izmailovky", 73, 628)).
25     phoneRecord("Bogachenko", 898551, address("Kishinev", "Izmailovky", 73, 628)).
26     phoneRecord("Shelia", 891810, address("Sochi", "Tulskaya", 21, 1)).
27     phoneRecord("Solntceva", 896056, address("Vysokie Polyany", "Rimskaya", 123, 422)).
28     phoneRecord("Priianishnikova", 812312, address("Krasnoyarsk", "Mate Zalki", 6, 49)).
29     phoneRecord("Grimberg", 890298, address("Krasnoyarsk", "Yastynskaya", 8, 223)).
30     phoneRecord("Tonkoshtan", 896245, address("Stavropol", "Akadem", 29, 229)).
31     phoneRecord("Serova", 898577, address("Moscow", "Bauman", 13, 56)).
32     phoneRecord("Priianishnikov", 899999, address("Krasnoyarsk", "Mate Zalki", 18, 49)).
33
34     bankRecord("Priianishnikov", "Tinkoff", "12345", 91234).
35     bankRecord("Bogachenko", "Tinkoff", "76543", 1).
36     bankRecord("Serova", "Alpha", "65439", 666).
37     bankRecord("Solntceva", "Tinkoff", "00001", 150000).
38     bankRecord("Priianishnikova", "Raiffazen", "11111", 1000).
39     bankRecord("Priianishnikov", "Sberbank", "33333", 91234).
40
41     owner("Bogachenko", car("Lada", "black", 1500)).
42     owner("Priianishnikov", car("BMW", "white", 6666)).
43     owner("Grimberg", car("Toyota", "green", 777)).
44     owner("Shelia", car("Tesla", "georgia14", 420)).
45     owner("Serova", car("BMW", "white", 9999)).
46
```

```

47 owner("Pranishnikov", building("Kremlin", 1000)).
48 owner("Grimberg", building("PentHouse", 10000)).
49 owner("Solntceva", building("Avito", 3333)).
50
51 owner("Shelia", region("Sochi", 2014)).
52 owner("Tonkoshtan", region("Stavropol", 9)).
53 owner("Bogachenko", region("Moldova", 333)).
54
55 owner("Serova", transport("yacht", 7777)).
56 owner("Pranishnikova", transport("yacht", 7878)).
57 % First ex
58 allPersonObj(Surname, Name) :- owner(Surname, building(Name, _)).
59 allPersonObj(Surname, Name) :- owner(Surname, car(Name, _, _)).
60 allPersonObj(Surname, Name) :- owner(Surname, region(Name, _)).
61 allPersonObj(Surname, Name) :- owner(Surname, transport(Name, _)).
62 % Second ex
63 allPersonObjWithPrice(Surname, Name, Price) :- owner(Surname, building(Name, Price)).
64 allPersonObjWithPrice(Surname, Name, Price) :- owner(Surname, car(Name, _, Price)).
65 allPersonObjWithPrice(Surname, Name, Price) :- owner(Surname, region(Name, Price)).
66 allPersonObjWithPrice(Surname, Name, Price) :- owner(Surname, transport(Name,
    Price)).
67 % Third ex
68 sumlist([], 0).
69 sumlist([X|Xs], Sum) :- sumlist(Xs, SumTail), Sum = X + SumTail.
70
71 merge_list([], L, L).
72 merge_list([H|T], L, [H|M]) :- merge_list(T, L, M).
73
74 allPersonSumObj(Surname, Sum) :- findall(Price, owner(Surname, building(_, Price)),
    AllPriceBuild), findall(Price, owner(Surname, car(_, _, Price)), AllPriceCar),
    findall(Price, owner(Surname, region(_, Price)), AllPriceRegion), findall(Price,
    owner(Surname, transport(_, Price)), AllPriceTransport),
    merge_list(AllPriceBuild, AllPriceCar, All1), merge_list(All1, AllPriceRegion,
    All2), merge_list(All2, AllPriceTransport, All), sumlist(All, Sum).
75
76 goal
77 %allPersonObj("Pranishnikov", Name).
78 %allPersonObjWithPrice("Pranishnikov", Name, Price).
79 allPersonSumObj("Pranishnikov", Sum).

```

В приложении 1 приведены таблицы для описания порядка ответа на вопрос, как выбираются знания.

Теоретические вопросы

1. В каком фрагменте программы сформулировано знание? Это знание о чем на формальном уровне?

Знания сформулированы в базе знаний. Знания на формальном уровне о предметной области.

2. Что содержит тело правила?

Правило — это предложение, истинность которого зависит от истинности одного или нескольких предложений. Обычно правило содержит несколько хвостовых целей, которые должны быть истинными для того, чтобы правило было истинным.

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

Переменная в Прологе, в отличие от алгоритмических языков программирования, обозначает объект, а не некоторую область памяти. Пролог не поддерживает механизм деструктивного присваивания, позволяющий изменять значение инициализированной переменной, как императивные языки.

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

Имя переменной в Прологе может состоять из букв латинского алфавита, цифр, знаков подчеркивания и должно начинаться с прописной буквы или знака подчеркивания. При этом переменные в теле правила неявно связаны квантором всеобщности.

Областью действия переменной в Прологе является одно предложение. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов.

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

Раздел описания доменов является аналогом раздела описания типов в обычных императивных языках программирования и начинается с ключевого слова DOMAINS.

Если природа или структура объектов, обозначенных аргументами, между которыми устанавливается отношение в заголовке правил процедуры, не может быть определена с помощью стандартных доменов, то требуется определить имена и семантику – смысл (структуру) таких доменов в разделе DOMAINS.

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

В разделе, озаглавленном зарезервированным словом PREDICATES, содержатся описания определяемых пользователем предикатов. В традиционных языках программирования подобными разделами являются разделы описания заголовков процедур и функций.

Если природа или структура объектов, обозначенных аргументами, между которыми устанавливается отношение в заголовке правил процедуры, важна во время работы системы, то она должна быть указана в разделе PREDICATES с помощью соответствующих доменов.

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

На первом шаге работы запускается унификация вопроса и первого знания в базе знаний.

Алгоритм унификации – основной шаг с помощью которого система отвечает на вопросы унификации. Для нахождения всех решений реализуется механизм возврата.

Унификация – необходима для того, чтобы определить дальнейший путь поиска решений. На первом шаге: терм вопроса унифицируется с заголовком. Унификация заканчивается конкретизацией части переменных.

Алгоритм унификации принимает на вход два терма, и возвращает флаг успешности подстановки, и, если успешно, то подстановку.

8. В каком случае запускается механизм отката?

Механизм отката, который осуществляет откат программы к той точке, в которой выбирался унифицирующийся с последней подцелью дизъюнкт. Для этого точка, где выбирался один из возможных унифицируемых с подцелью дизъюнктов, запоминается в специальном стеке, для последующего возврата к ней и выбора альтернативы в случае неудачи. При откате все переменные, которые были означены в результате унификации после этой точки, опять становятся свободными.