|  |  |
| --- | --- |
|  | **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»

**Лабораторная работа № 15**

|  |  |
| --- | --- |
| **Студент:** Керимов А. Ш.  **Группа:** ИУ7-64Б  **Преподаватель:** Толпинская Н. Б. |  |

Москва.

2020 г.

**Цель работы** — изучить структуру, особенности и принципы оформления программы, и способ выполнения программы на Prolog.

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

* **«Телефонный справочник»:** Фамилия, №тел, Адрес – структура (Город, Улица, №дома, №кв),
* **«Автомобили»:** Фамилия\_владельца,Марка, Цвет, Стоимость, и др.,
* **«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):

* **Строение, стоимость** и другие его характеристики;
* **Участок, стоимость** и другие его характеристики;
* **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 пункт и **одной** фамилии **составить таблицу**, отражающую конкретный порядок работы системы, с объяснениями порядка работы и особенностей использования доменов (указать конкретные Т1 и Т2 и полную подстановку на каждом шаге)

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1… | -попытка унификации: Т1=Т2 …  -результат: Успех и подстановка,  или Нет | Комментарий, вывод… |
| 2 | … | … |

# Практическая часть

*Листинг 1. Телефонный справочник, автомобили и вклады*

**domains**

name, lastName, phoneNumber, city, street,

brend, color, bank, bankAccount = **symbol**.

flat, price, ownPrice, summ, house, sumVar = **integer**.

address = adr(city, street, house, flat).

clause = autos(name, price);

building(name, price);

ship(name, price);

area(name, price).

**predicates**

phoneBook(lastName, phoneNumber, address)

investors(lastName, bank, bankAccount, summ)

owner(lastName, clause)

ownership(lastName, name, price)

clauseProperty(clause, name, price)

**clauses**

clauseProperty(autos(Name, Price),OwnName, OwnPrice) :- OwnName=Name, OwnPrice=Price.

clauseProperty(ship(Name, Price),OwnName, OwnPrice) :- OwnName=Name, OwnPrice=Price.

clauseProperty(building(Name, Price),OwnName, OwnPrice) :- OwnName=Name, OwnPrice=Price.

clauseProperty(area(Name, Price),OwnName, OwnPrice) :- OwnName=Name, OwnPrice=Price.

phoneBook(sanginov, "6", adr(moscow, mayakovskaya, 6, 26)).

phoneBook(pupkin, "5", adr(moscow, mayakovskaya, 7, 27)).

phoneBook(ivanov, "7", adr(urengoy, mayakovskaya, 8, 28)).

phoneBook(ivanov, "8", adr(moscow, tulchinskaya, 8, 28)).

investors(sanginov, sber, "1588", 20).

investors(pupkin, tinkoff, "1088", 12).

investors(ivanov, alpha, "1588", 19).

investors(ivanov, otkritie, "1588", 19).

owner(sanginov, autos(mersedesbenssclass, 55)).

owner(ivanov, ship(mssclass, 67)).

owner(ivanov, autos(mercedes, 21)).

owner(sanginov, building(moscowCity, 160)).

owner(sanginov, area(derbentskiy, 12203)).

owner(ivanov, area(moscowskiy, 120)).

owner(petrov, area(hmelnick, 510)).

ownership(Lastname, OwnName, OwnPrice) :-

owner(Lastname, CLAUSE),

clauseProperty(CLAUSE, OwnName, OwnPrice).

**goal**

ownership(petrov, OwnName, OwnPrice).

% ownership(sanginow, OwnName, OwnPrice).

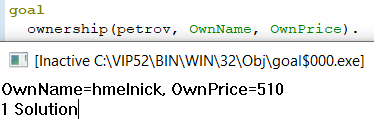
****

Рисунок – Владелец Петров

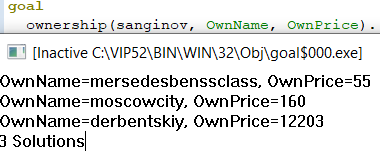
****

Рисунок – Владелец Сангинов

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия |
| 1 | Сравниваются  ownership(petrov, OwnName, OwnPrice) и clauseProperty(autos(Name, Price), OwnName, OwnPrice) Неудача. Термы не унифицируемы, имеют разные функторы. | Прямой ход. Запуск новой унификации для очередного предложения из бз. |
| … | … | … |
| 22 | Сравниваются ownership(petrov, OwnName, OwnPrice) и ownership(Lastname, OwnName, OwnPrice). Термы унифицируемы. Подстановка:  {Lastname=petrov,  OwnName=OwnName,  OwnPrice=OwnPrice}. | Прямой ход. Применение подстановки к резольвенте. Замена цели в резольвенте телом найденного правила. Выбор подцели из резольвенты и запуск унификации для нее. |
| 23 | Сравниваются owner(petrov, CLAUSE) и clauseProperty(autos(Name, Price),OwnName, OwnPrice). Термы не унифицируемы, имеют разные функторы. Неудача. | Прямой ход. Запуск новой унификации для очередного предложения из бз. |
| … | … | … |
| 35 | Сравниваются owner(petrov, CLAUSE) и owner(sanginov, autos(mersedesbenssclass, 55)). Petrov и sanginov не унифицируемы. | Прямой ход. Запуск новой унификации для очередного предложения из бз. |
| … | … | … |
| 41 | Сравниваются  owner(petrov, CLAUSE)  и owner(petrov, area(hmelnick, 510)). Термы унифицируемы. Подстановка:  {petrov=petrov,  CLAUSE=area(hmelnick, 510))} | Прямой ход. Применение подстановки к резольвенте. Замена цели в резольвенте телом найденного правила. Выбор подцели из резольвенты и запуск унификации для нее. |
| 42 | Сравниваются  clauseProperty(area(hmelnick, 510)), OwnName, OwnPrice) и  clauseProperty(autos(Name, Price),OwnName, OwnPrice).  Термы не унифицируемы, имеют разные функторы. Неудача. | Прямой ход. Запуск новой унификации для очередного предложения из бз. |
| … | … | … |
| 45 | Сравниваются  clauseProperty(area(hmelnick, 510)), OwnName, OwnPrice)  и clauseProperty(area(Name, Price),OwnName, OwnPrice)  Термы унифицируемы. Подстановка:  {Name=hmelnick,  Price=510,  OwnName=OwnName,  OwnPrice=OwnPrice} | Прямой ход. Применение подстановки к резольвенте. Замена цели в резольвенте телом найденного правила. Выбор подцели из резольвенты и запуск унификации для нее. |
| 46 | Запуск унификации с помощью = :  Подстановка: { OwnName=hmelnick } | Прямой ход. Применение подстановки к резольвенте. Замена цели в резольвенте телом найденного правила. Выбор подцели из резольвенты и запуск унификации для нее. |
| 47 | Запуск унификации с помощью = :  Подстановка: { OwnPrice=510 } | Прямой ход. Применение подстановки к резольвенте. Замена цели в резольвенте телом найденного правила. Выбор подцели из резольвенты и запуск унификации для нее. |
| 48 | Результат: Наибольший общий унификатор {petrov=petrov, OwnName=hmelnick, OwnPrice=510} | Содержимое резольвенты пусто. Все переменные связаны. Откат. Удаление значения OwnPrice. |
| 49 | Сравниваются  clauseProperty(area(hmelnick, 510)), OwnName, OwnPrice) и  phoneBook(sanginov, "6", adr(moscow, mayakovskaya, 6, 26)).  Термы не унифицируемы, имеют разные функторы. Неудача. | Прямой ход. Запуск новой унификации для очередного предложения из бз. |
| … | … | … |
| 55 | Сравниваются clauseProperty(area(hmelnick, 510)), OwnName, OwnPrice) и ownership(Lastname, OwnName, OwnPrice) Термы не унифицируемы, имеют разные функторы. Неудача. | База знаний пройдена для clauseProperty. Откат.  Значение переменной CLAUSE удаляется. |
| 56 | Сравниваются owner(petrov, CLAUSE) и ownership(Lastname, OwnName, OwnPrice)Термы не унифицируемы, имеют разные функторы. Неудача. | База знаний пройдена для owner. Откат.  Значение переменной Lastname удаляется. |
| 57 | База знаний пройдена, альтернативных решений не найдено. |  |

# Теоретическая часть

## В каком фрагменте программы сформулировано знание? Это знание о чём на формальном уровне?

Правило имеет вид . называется заголовком правила, а — телом правила. Заголовок содержит знание. Это знание формально описывает отношение между его аргументами.

## Что содержит тело правила?

Тело правила содержит условие истинности заголовка правила.

## Что даёт использование переменных при формулировании знаний? В чём 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.

## 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.

## Какова семантика (смысл) предложений раздела DOMAINS? Когда, и где используется это описание?

Предложения в разделе DOMAINS используются для объявления используемых доменов, не являющимися стандартными доменами в Prolog. Раздел доменов используется для описания структур (вариантных доменов).

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

В разделе PREDICATES описываются предикаты, их арность (местность) и домены (типы и природа аргументов). С помощью описанных предикатов, можно создавать предложения в базе знаний. Предикаты используются для представления, как фактов, так и правил.

## 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».

## В каком случае запускается механизм отката?

Механизм отката запускается в 2 случаях:

1. Если алгоритм попал в тупиковую ситуацию.
2. Если резольвента не пуста и решение найден, но в базе знание остались не отмеченные предложения.

# Замечания по ЛР13:

|  |
| --- |
| **ИУ7-64Б Керимов А. Ш.** |
| **В таблице:Сравнение phoneBook(Surname, 2222222222”, \_) и**  phoneBook("Kerimov", "1111111111", address("Lyubertsy", "Yubileynaya", "17", "2")).  **Функторы равны, термы “2222222222” и “1111111111” не равны** | |

А что, если функторы не равны, система **сравнивает** аргументы?

Если функторы не равны, система не сравнивает аргументы.

Сравнение find\_car\_by\_phone(“2222222222”, Surname, Model, Cost) и

find\_car\_by\_phone(Phone, Surname, Model, Cost) :- phoneBook(Surname, Phone, \_), car(Surname, Model, \_, Cost); Phone конкретизируется “2222222222”. Сцепление Surname к Surname, Model к Model, Cost к Cost. Сравниваемые, как вы утверждаете, фрагменты имеют разную структуру!!! Их сравнить нельзя! ГДЕ подстановки?

Осознал, подстановки буду писать

# Замечания по ЛР14:

|  |
| --- |
| **ИУ7-64Б Керимов А. Ш.** |

Знание содержится в заголовке правила. **Это знание о чём, с формальной точки зрения?**

Это знание формально описывает отношение между его аргументами.

**Что такое процедура?** Процедура описывает совокупность правил отношения. НЕТ

**Назначение и результат работы алгоритма унификации. Т.е. зачем** он запускается системой? А система заранее **знает для каких термов его запускать?**

**Процедурой** называется совокупность правил, заголовки которых имеют одно и то же имя и одну и ту же арность (местность), т.е. это совокупность правил, описывающих одно определенное отношение.

Для поиска ответа на вопрос система должна найти подходящее знание.

Знание зафиксировано в заголовке правила.

Система должна подобрать подходящее правило (заголовок).

В этом и состоит назначение алгоритма унификации — в попарном сопоставлении термов и попытке построить для них общий пример.