

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 <u>«Информатика и системы управления»</u> КАФЕДРА <u>«Программное обеспечение ЭВМ и информационные</u> технологии»

Лабораторная работа № 12 Знакомство со средой Visual Prolog 5.2

Студент Лучина Е.Д

Группа ИУ7-61Б

Преподаватель Толпинская Н.Б.

Москва.

апрель 2020 г.

Цель работы – познакомиться со структурой, принципами оформления и логикой выполнения программы на Prolog

Задачи работы: приобрести навыки декларативного описания предметной области с использованием фактов и правил. Изучить способы использования фактов и правил в программе на Prolog, принципы и правила сопоставления и отождествления, принцип унификации.

Содержание работы:

Составить программу — базу знаний, с помощью которой можно определить, например, множество студентов, обучающихся в одном ВУЗе. Студент может одновременно обучаться в нескольких ВУЗах. Привести примеры возможных вариантов вопросов и варианты ответов (не менее 3-х). Описать порядок формирования вариантов ответа.

Листинг программы

```
domains
  person type = person(firstname, lastname)
  firstname, lastname = symbol
  university = string
predicates
  student(person type, university)
clauses
  student(person(james, gelse), "HSE").
  student(person(dave, smith), "BMSTU").
  student(person(dave, smith), "MGU").
  student(person(david, green), "BMSTU").
  student(person(mike, klein), "HSE").
  student(person(nate, dest), "MGU").
  student(person(nataly, green), "BMSTU").
  student(person(james, gelse), "MGU").
qoal
  student (person (Firstname, Lastname), Uni).
```

В разделе доменов для удобства чтения программы описаны

нестандартные домены firstname, lastname, university, а также составной домен person_type, представляющий собой структуру полного имени. При желании и большем объеме данных, можно расширить person_type, добавив возраст, телефон, место проживания. В данном случае имени и фамилии достаточно.

В разделе предикатов имеется только один предикат - student, аргументами которого являются определенные в домене структура person type и строка university.

В базе знаний имеем только факты об обучении конкретного человека в конкретном университете. Имеется восемь записей о трех университетах и шести студентах. Можно отметить, что dave smith и james gelse являются студентами нескольких университетов.

В качестве цели выведем все имеющиеся записи.

Результат

```
Firstname=james, Lastname=gelse, Uni=HSE
Firstname=dave, Lastname=smith, Uni=BMSTU
Firstname=dave, Lastname=smith, Uni=MGU
Firstname=david, Lastname=green, Uni=BMSTU
Firstname=mike, Lastname=klein, Uni=HSE
Firstname=nate, Lastname=dest, Uni=MGU
Firstname=nataly, Lastname=green, Uni=BMSTU
Firstname=james, Lastname=gelse, Uni=MGU
8 Solutions
```

Можно отфильтровать студентов по названию университета. Получим следующие множества студентов, обучающихся в одном ВУЗе.

<pre>write("Students of MGU:"), nl, student(X, "MGU").</pre>	<pre>Students of MGU: X=person("dave", "smith") X=person("nate", "dest") X=person("james", "gelse") 3 Solutions</pre>
<pre>write("Students of HSE:"), nl, student(X, "HSE").</pre>	Students of HSE: X=person("james","gelse") X=person("mike","klein")

	2 Solutions
<pre>write("Students of BMSTU:"), nl, student(X, "BMSTU").</pre>	Students of BMSTU: X=person("dave", "smith") X=person("david", "green") X=person("nataly", "green") 3 Solutions

Добавим в clauses правила нахождения университета по полному имени студента, нахождения университета по фамилии, нахождения студентов конкретного университета. (и соответствующие предикаты в predicates)

Примеры работы правил. (возвращают только первый подходящий факт)

```
get_uni_by_fullname(nataly, green),
get_uni_by_lastname(dest),
get_students_of("BMSTU").

nataly green studies in BMSTU.
nate dest studies in MGU.
dave smith studies in BMSTU.
yes
```

Теоретическая часть

Основным элементом языка Prolog является терм:

- 1. Константа:
 - а. Число (целое, вещественное),
 - b. Символьный атом (комбинация символов латинского алфавита, цифр и символа подчеркивания, начинающаяся со

- строчной буквы: aA, ab 2),
- с. Строка: последовательность символов, заключенных в кавычки,

2. Переменная:

- а. Именованная обозначается комбинацией символов цифр и латинского алфавита, символа подчеркивания, прописной буквы начинающейся c или символа подчеркивания (X, A21, X),
- b. Анонимная обозначается символом подчеркивания (),

3. Составной терм:

а. Это средство организации группы отдельных элементов знаний в единый объект, синтаксически представляется: f(t1, t2, ...,tm), где f - функтор (функциональный символ), t1, t2, ..., tm – термы (их называют аргументами).

Программа на Prolog состоит из разделов. (В программе не обязательно должны быть все разделы.) Каждый раздел начинается со своего заголовка.

Структура программы:

- директивы компилятора зарезервированные символьные константы
- CONSTANTS раздел описания констант
- DOMAINS раздел описания доменов
- DATABASE раздел описания предикатов внутренней базы данных
- PREDICATES раздел описания предикатов
- © CLAUSES раздел описания предложений базы знаний
- GOAL раздел описания внутренней цели (вопроса).

В разделе описания доменов объявляются любые нестандартные домены в формате: <имя домена> = <определение домена>.

Можно использовать описание доменов для сокращения имен стандартных доменов, например i = integer.

Из доменов можно конструировать составные домены:

<имя структуры> = <имя функтора> (<имя домена первой компоненты>); <имя домена последней компоненты>) [; <имя функтора>(...)]*

В разделе предикатов описываются составные термы, которые будут использованы в фактах, правилах и вопросах. Здесь задаются названия отношений между объектами.

База знаний clauses состоит из предложений (отдельных знаний или утверждений): фактов и правил.

- Правила являются обобщенной формулировкой условия истинности знания отношения между объектами предметной области (аргументами терма), которое записано в заголовке правила. Условие истинности этого отношения является телом правила. Правило завершается точкой. <правило> := <заголовок> :- <тело правила>.
- Факт это правило, в котором отсутствует тело (т.е. тело пустое). Представляют собой составные термы, с помощью которых фиксируется наличие истинностных отношений между объектами предметной области аргументами терма.

Boпрос (goal) состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами.

Цель системы состоит в том, чтобы на поставленный вопрос найти возможность, исходя из базы знаний, ответить «Да». Этот поиск осуществляется формально с помощью механизма унификации, встроенного в систему. В процессе выполнения программы именованные переменные могут связываться с различными объектами — конкретизироваться. При этом подобранное значение становится

побочным эффектом работы системы – оно возвращается как результат

работы.