



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 №12

Структура программы на Prolog

Студент: Луговой Д.М.

Группа: ИУ7-61Б

Преподаватель: Толпинская Н.Б.

Москва, 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.

Теория

Программа на Prolog

Программа на Prolog представляет собой набор фактов и правил, обеспечивающих получение заключений на основе этих утверждений. Программа содержит базу знаний и вопрос. База знаний содержит истинные значения, используя которые программа выдает ответ на вопрос.

Основным элементом языка является терм. Терм – это:

1. Константа:

- Число (целое, вещественное),
- Символьный атом (комбинация символов латинского алфавита, цифр и символа подчеркивания, начинающаяся со строчной буквы),
- Строка: последовательность символов, заключенных в кавычки.

2. Переменная:

- Именованная – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания,
- Анонимная - обозначается символом подчеркивания

3. Составной терм: Это средство организации группы отдельных элементов знаний в единый объект, синтаксически представляется: $f(t_1, t_2, \dots, t_m)$, где f - функтор (отношение между объектами), t_1, t_2, \dots, t_m – термы, в том числе и составные.

База знаний состоит из предложений. Каждое предложение заканчивается точкой. Предложения бывают двух видов: факты и правила. Правило имеет вид: $A :- B_1, \dots, B_n$. A называется заголовком правила, а B_1, \dots, B_n – телом правила. Факт – это частный случай правила, в котором отсутствует тело. Заголовок содержит отдельное знание о предметной области (составной терм), а

тело содержит условия истинности этого знания. Заголовок правила синтаксически это составной терм. Тело правила может представлять собой один терм или быть последовательностью термов (конъюнкцией или дизъюнкцией). Правила называют условной истиной, а факт – безусловной.

Вопрос состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

Назначение и использование переменных

Факты, правила, и вопросы могут содержать переменные. Именованные переменные уникальны в рамках предложения, а анонимная переменная – любая уникальна. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов. В процессе выполнения программы переменные могут связываться с различными объектами – конкретизироваться. Это относится только к именованным переменным. Анонимные переменные не могут быть связаны со значением.

Также поддерживается механизм деструктивной конкретизации переменной. Т.е. используется идея реконкретизации переменной путем «отката» вычислительного процесса и отказа от выполненной ранее конкретизации. Это реализовано для возможности поиска нового значения для именованной переменной.

Структура программы

Программа на Prolog состоит из следующих разделов:

- 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 программе не обязательно должны быть все разделы.

Формирование результата

Ответ на поставленный вопрос система дает в логической форме - «Да» или «Нет». Цель системы состоит в том, чтобы на поставленный вопрос найти возможность, исходя из базы знаний, ответить «Да». Вариантов ответить «Да» на поставленный вопрос может быть несколько. В нашем случае система настроена в режим получения всех возможных вариантов ответа.

Поиск содержательного ответа на поставленный вопрос, с помощью имеющейся базы знаний, фактически заключается в поиске нужного знания. Этот поиск осуществляется с помощью механизма унификации.

Унификация – операция, которая позволяет формализовать процесс логического вывода. С практической точки зрения - это основной вычислительный шаг, с помощью которого происходит:

- Двухнаправленная передача параметров процедурам,
- Неразрушающее присваивание,
- Проверка условий (доказательство).

Упрощенно, процесс унификации можно представить как формальный процесс сравнения (сопоставления) термина вопроса с очередным термом знания. Знания по умолчанию просматриваются сверху вниз. В процессе сравнения для переменных «подбираются», исходя из базы знаний, значения (для именованных переменных). И эти подобранные для переменных значения возвращаются в качестве побочного эффекта ответа на поставленный вопрос.

В процессе работы система выполняет большое число унификаций. Попытка "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией (неудачей). В последнем случае включается механизм отката к предыдущему шагу.

Задание

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 1: База знаний о студентах

```
1 domains
2   name, university, speciality, city, budget_place = string.
3   price = integer.
4 predicates
```

```

5  study(name, university, speciality, budget_place).
6  locate(university, city).
7  pays(name, price).
8  clauses
9  study("Ilya Ivanov", "BMSTU", "Software Engineer", "Budget").
10 study("Ivan Petrov", "HSE", "Designer", "Paid").
11 study("Dmitriy Vlasov", "KSU", "Historian", "Paid").
12 study("Alexander Pushkin", "ITMO", "Software Engineer", "Budget").
13 study("Mihail Knyazev", "BMSTU", "Mathematician", "Budget").
14 locate("BMSTU", "Moscow").
15 locate("HSE", "Moscow").
16 locate("ITMO", "St-Petersburg").
17 locate("KSU", "Krasnodar").
18 pays(Name, 200) :- study(Name, University, _, "Paid"),
19                    locate(University, "Moscow").
20 pays(Name, 150) :- study(Name, University, _, "Paid"),
                    locate(University, "St-Petersburg").
pays(Name, 100) :- study(Name, University, _, "Paid"),
                    locate(University, "Krasnodar").

```

Данная база знаний содержит информацию о студентах(имя, вуз, специальность, основа обучения), вузах(название, город) и плате за обучение студентами(имя, сумма).

Примеры:

goal	Результат
study(Name, "BMSTU", _, _).	Name=Ilya Ivanov Name=Mihail Knyazev 2 Solutions
study(Name, University, "Software Engineer", _).	Name=Ilya Ivanov, University=BMSTU Name=Alexander Pushkin, University=ITMO 2 Solutions
study(Name, University, _, _), locate(University, "Moscow").	Name=Ilya Ivanov, University=BMSTU Name=Ivan Petrov, University=HSE Name=Mihail Knyazev, University=BMSTU 3 Solutions
pays(Name, Price).	Name=Ivan Petrov, Price=200 Name=Dmitriy Vlasov, Price=100 2 Solutions

С помощью первого вопроса получаются все студенты, которые учатся в МГТУ. Происходит проход сверху вниз по всем фактам предиката *study(name, university, speciality, budget_place)* и осуществляется унификация с *study(Name, "BMSTU", _, _)*. Унификацию успешно проходят два факта: *study("Ilya Ivanov", "BMSTU", "Software Engineer", "Budget")* и *study("Mihail Knyazev", "BMSTU", "Mathematician", "Budget")*.

С помощью второго вопроса получаются все студенты и их вузы, которые имеют специальность программный инженер. Происходит проход по всем фак-

там предиката *study(name, university, speciality, budget_place)* и осуществляется унификация с *study(Name, University, "Software Engineer", _)*. Успешно унификацию проходят факты *study("Ilya Ivanov", "BMSTU", "Software Engineer", "Budget")* и *study("Alexander Pushkin", "ITMO", "Software Engineer", "Budget")*.

С помощью третьего вопроса находятся все студенты, университеты которых расположены в Москве. Вопрос построен с помощью конъюнкции, поэтому осуществляется проход по всем фактам предиката *study(name, university, speciality, budget_place)* и предиката *locate(university, city)* и осуществляется унификация с термом *study(Name, University, _, _)*, *locate(University, "Moscow")*. В результате унификацию успешно проходят: *study("Ilya Ivanov", "BMSTU", "Software Engineer", "Budget")*, *locate("BMSTU", "Moscow")*; *study("Ivan Petrov", "HSE", "Designer", "Paid")*, *locate("HSE", "Moscow")* и *study("Mihail Knyazev", "BMSTU", "Mathematician", "Budget")*, *locate("BMSTU", "Moscow")*.

С помощью последнего вопроса находятся все студенты, обучающиеся на платной основе, и их стоимость обучения. Для этого осуществляется проход по всем правилам предиката *pays(name, price)* и унификация с термом *pays(Name, Price)*. При унификации для каждого правила осуществляется унификация каждого терма из тела правила. Успешно унифицируются терм *study("Ivan Petrov", "HSE", "Designer", "Paid")*, *locate("HSE", "Moscow")* с телом правила *pays(Name, 200)* и терм *study("Dmitriy Vlasov", "KSU", "Historian", "Paid")*, *locate("KSU", Krasnodar)* с телом правила *pays(Name, 100)*.