Лабораторная работа №2: Разработка и отладка простой программы на языке Visual Prolog

Цель работы

Получение необходимых навыков для разработки и отладки простейшей программы на языке Visual Prolog.

Основные теоретические положения

Пролог (англ. Prolog) – язык и система логического программирования, основанные на языке предикатов математической логики дизъюнктов Хорна, представляющей собой подмножество логики предикатов первого порядка.

Prolog является декларативным языком программирования: логика программы выражается в терминах отношений, представленных в виде фактов и правил. Для того чтобы инициировать вычисления, выполняется специальный запрос к базе знаний, на которые система логического программирования генерирует ответы «истина» и «ложь». Для обобщённых запросов с переменными в качестве аргументов созданная система Пролог выводит конкретные данные в подтверждение истинности обобщённых сведений и правил вывода.

Иначе говоря, предикат можно определить как функцию, отображающую множество произвольной природы в множество булевых значений {ложно, истинно}. Задача пролог-программы заключается в том, чтобы доказать, является ли заданное целевое утверждение следствием из имеющихся фактов и правил.

Язык программирования, реализованный в Visual (а ранее в Turbo) Prolog'е отличается от классического пролога тем, что он основан на строгой статической типизации. В него также добавлены средства объектно-ориентированного программирования, анонимные предикаты (лямбда-предикаты), факты-переменные и разрушающее присваивание для них, аргументы-домены (`Generic Interfaces and Classes`) и параметрический полиморфизм, мониторы (`Monitors with guards`), императивные конструкции (`foreach`, `if...then...else`), коллекторы списков (`[...||...]`) и пр.

Постановка задачи

Разработать программу на языке Visual Prolog, описывающие родственные связи между собаками, а также реализовать генеалогическое древо Вашей семьи.

Порядок выполнения работы

1. Загрузите систему программирования Visual Prolog.
2. Установите в опциях среды шрифт с русской раскладкой **Options/Global/Environment**:
    1. На карточке **Fonts** поменяйте шрифт окна редактора **Editor Windows** на **Courier** с раскладкой Кириллица.
    2. Либо воспользуйтесь кнопкой **F** (*Font*) на панели инструментов.
3. Выполните команду **Project/New Project**. Определите в качестве *Project Name* имя *DOG* и укажите главный каталог (*Base Directory*) в личной папке на диске. Нажмите кнопку **Create**.
   В созданном проекте удалите модуль *Vpitools.pro*. Откройте модуль *DOG.PRO*, удалите все содержимое и наберите текст программы:

```
domains
  name=string
predicates
  dog(name)
  parent(name, name)
clauses
  dog(X) :- parent(X, Y), dog(Y).
  dog("Rex").
  parent("Jack", "Rex").
goal
  dog(Who).
```

4. Установите опции компилятора **Options/Project/Compiler Options**:
    1. на карточке **Output** установите **Generate Debug Information** и **Disable Optimizations**,
    2. на карточке **Warnings** установите **Nondeterm**, снимите **Unused Variables** и **Unused Predicates**.
5. Запустите программу на выполнение, нажав кнопку **Test Goal** на панели инструментов. Объясните полученный результат.
6. Выполните компиляцию (**Compile Module**), построение (**Build**) и запустите отладчик **Debug**.
7. В среде отладчика включите окна просмотра переменных (**Local Variables**) и стека вызовов (**Call Stack**). Произведите пошаговую прокрутку программы, нажимая на кнопку **Trace Into** на панели инструментов.
8. Запишите и проанализируйте трассу выполнения программы по содержимому окон **Call Stack** и **Variables For Current Clause** для одного решения.
9. Напишите на языке Visual Prolog программу *DOG1*, описывающую родственные связи собак в соответствии с вариантом задания. Программа должна использовать предикаты `dog` и `parent` из программы *DOG.PRO* и содержать не менее двух правил.
10. Произведите отладку программы в системе Visual Prolog на следующих запросах:
    1. Кто является собакой?
    2. Кто является родителем?
    3. Кто является внуком (внучкой)?
    4. Balto – собака?
    5. Кто родитель собаки Balto?
    6. Кому Balto является родителем?

11. Постройте трассу выполнения программы для каждого запроса.
12. ***Задание для самостоятельной работы.*** Используя предикаты `parent(symbol, symbol)`, `man(symbol)`, `woman(symbol)`, `married(symbol, symbol)`, записать факты, описывающие Вашу семью.
    Написать на языке Visual Prolog программу *FAMILY*, содержащую не менее 8 правил вывода для любых родственных отношений в Вашей семье (например: мать, отец, сестра, брат, племянница, племянник, тетя, дядя, внучка, внук, бабушка, дедушка, двоюродная сестра, двоюродный брат и т.д.).
    Отладить программу на 5-6 различных запросах.

Варианты заданий

Варианты к лабораторной работе №2

Содержание отчёта

- Цель работы.
- Краткое изложение основных теоретических понятий.
- Постановка задачи с кратким описанием порядка выполнения работы.
- Объяснение результатов выполнения программы *DOG.PRO*.
- Трасса выполнения программы *DOG.PRO* и пояснения к ней.
- Трассы выполнения запросов программы *DOG1.PRO* и объяснение результатов их выполнения.
- Описание родственных связей в семье (в виде дерева).
- Результаты работы программы *FAMILY.PRO* для 5-6 разных запросов.
- Общий вывод по проделанной работе.
- Коды программ.

From:
http://se.moevm.info/ - **se.moevm.info**

Permanent link:
http://se.moevm.info/doku.php/courses:knowledge_base_and_expert_system:lab2

Last update: **2022/12/10 09:08**