



**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 № 11
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 средой Visual Prolog, познакомиться со структурой программы: способом запуска и формой вывода результатов. Изучить среду визуальной разработки Visual Prolog. Создать проект и запустить его на выполнение.

Задачи работы: изучить принципы работы в среде Visual Prolog, возможность получения однократного и многократного результата, изучить базовые конструкции языка Prolog, структуру программы Prolog, форму ввода исходных данных и вывода результатов работы программы.

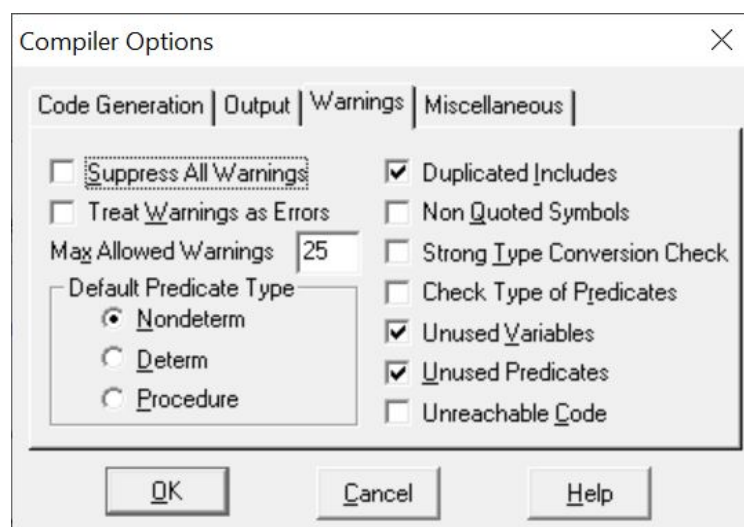
Содержимое работы:

- 1) Запустить среду Visual Prolog5.2. Настроить утилиту TestGoal.

TestGoal - это команда, которая используется для тестирования простых целей (Goals). Программа компилируется и компоуется в специальном режиме, и затем запускается соответствующий исполняемый файл. Утилита Test Goal ищет все решения для определенной в программе цели.

Настройки компилятора по умолчанию были заменены следующим образом.

Тип предикатов переключен на Nondeterm. Это нужно для того, чтобы компилятор Visual Prolog принимал по умолчанию, что все определенные пользователем предикаты — недетерминированные (могут породить более одного решения). А также сняты флажки Not Quoted Symbols, Strong Type Conversion Check и Check Type of Predicates, чтобы подавлять некоторые не интересующие нас предупреждения компилятора.



- 2) Запустить тестовую программу, проанализировать реакцию системы и множество ответов.

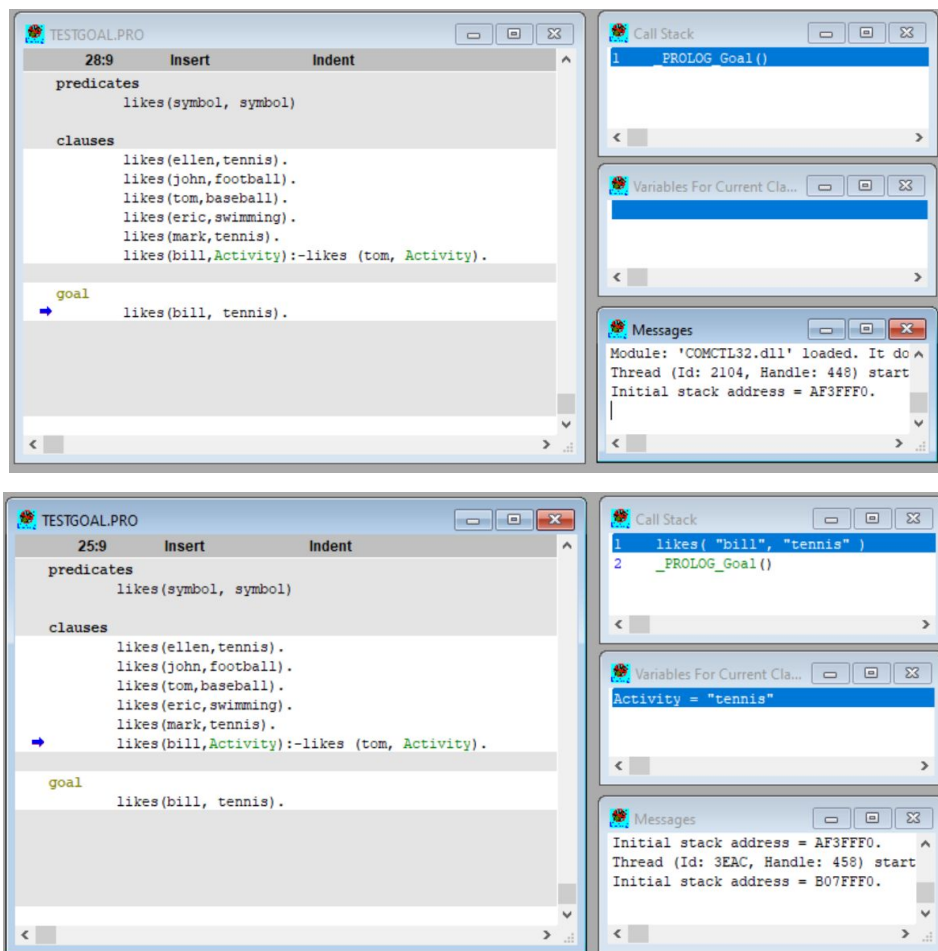
```
include "testgoal.inc"

predicates
    likes(symbol, symbol)

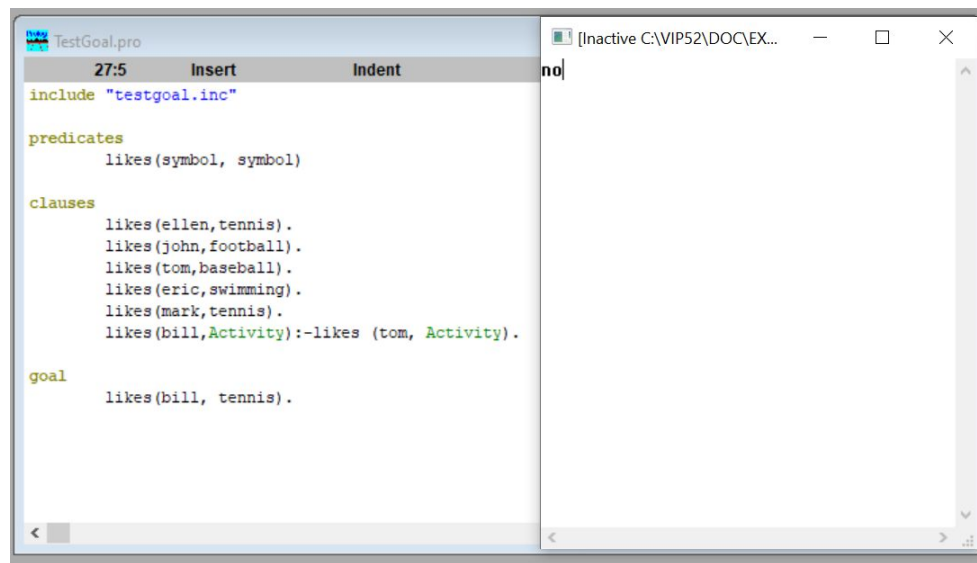
clauses
    likes(ellen,tennis).
    likes(john,football).
    likes(tom,baseball).
    likes(eric,swimming).
    likes(mark,tennis).
    likes(bill,Activity):-likes (tom, Activity).

goal
    likes(bill, tennis).
```

Ниже приведены скриншоты некоторых окон при работе дебаггера.



Результаты работы программы:



The screenshot shows the TestGoal.pro program interface. The left pane displays the Prolog code, and the right pane shows the output. The code defines a set of facts and a goal. The output shows the result of the query.

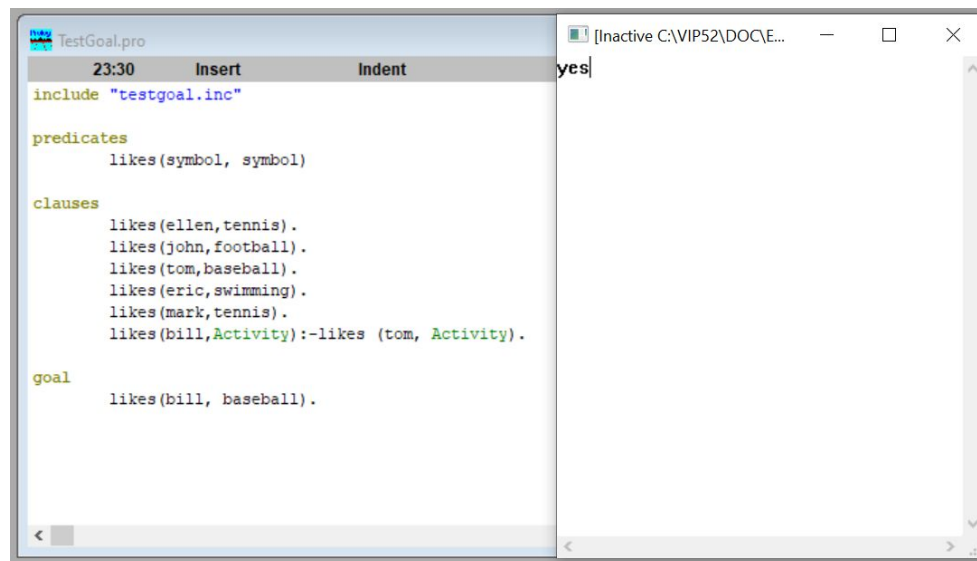
```
TestGoal.pro
27:5 Insert Indent
include "testgoal.inc"

predicates
    likes(symbol, symbol)

clauses
    likes(ellen,tennis).
    likes(john,football).
    likes(tom,baseball).
    likes(eric,swimming).
    likes(mark,tennis).
    likes(bill,Activity):-likes (tom, Activity).

goal
    likes(bill, tennis).
```

no|



The screenshot shows the TestGoal.pro program interface. The left pane displays the Prolog code, and the right pane shows the output. The code defines a set of facts and a goal. The output shows the result of the query.

```
TestGoal.pro
23:30 Insert Indent
include "testgoal.inc"

predicates
    likes(symbol, symbol)

clauses
    likes(ellen,tennis).
    likes(john,football).
    likes(tom,baseball).
    likes(eric,swimming).
    likes(mark,tennis).
    likes(bill,Activity):-likes (tom, Activity).

goal
    likes(bill, baseball).
```

yes|

Истинность вопроса `likes(bill, baseball)` может быть достигнута из правила `likes(bill, Activity) :- likes(tom, Activity)`, содержащего переменную `Activity`. Утверждение `likes(bill, Activity)` будет истинно, если истинно `likes(tom, Activity)`. В случае, когда `Activity` связана с константой `baseball`. Программа возвращает положительный результат `yes`, так как в базе данных содержится факт `likes(tom, baseball)`. Если `Activity` соответствует иное значение, программа возвращает `no`, так как процесс унификации (формальный процесс сравнения терма вопроса с очередным термом знания) завершается неудачей.

- 3) Разработать свою программу - «Телефонный справочник» и протестировать ее работу.

```
include "testgoal.inc"
predicates
    phone(symbol, string)
    address(symbol, string, integer)
    get_info_by_name(symbol)

clauses
    phone(ellen, "202-555-0170").
    phone(john, "202-555-0144").
    phone(tom, "202-555-0120").
    phone(eric, "202-555-0119").
    phone(mark, "202-555-0197").
    phone(kate, "202-555-0199").

    address(ellen, "Baker street" , 2).
    address(john, "Smith street", 5).
    address(tom, "Wall street", 12).
    address(eric, "Green street", 9).
    address(mark, "Gardenfield", 41).
    address(kate, "Marta street", 28).

    get_info_by_name(Name):-
        phone(Name, Number),
        address(Name, Street, House),
        write(Name),
        write("'s phone number is "),
        write(Number), nl,
        write(Name),
        write(" lives on "),
        write(Street),
        write(" ,"),
        write(House), nl.

goal
    phone(mark, Marks_phone_number);
    address(eric, Street, House);
    phone(Name, "202-555-0120");
    address(jim, Street, House);
    get_info_by_name(ellen).
```

Результат

```
Marks_phone_number=202-555-0197
Street=Green street, House=9
Name=tom
ellen's phone number is 202-555-0170
ellen lives on Baker street ,2
yes
```

Тестирование

goal	результат
phone(mark, _).	yes
phone(lena, _).	no
phone(mark, Phone_number).	Phone_number=202-555-0197 1 Solution
phone(lena, Phone_number).	No Solution
phone(Name, Phone_number).	Name=ellen, Phone_number=202-555-0170 Name=john, Phone_number=202-555-0144 Name=tom, Phone_number=202-555-0120 Name=eric, Phone_number=202-555-0119 Name=mark, Phone_number=202-555-0197 Name=kate, Phone_number=202-555-0199 6 Solutions

- 4) Что собой представляет программа на Prolog, какова ее структура. Как она реализуется, как формируются результаты работы программы?

Программа на Prolog представляет собой базу знаний и вопрос. Используя базу знаний, система Prolog будет делать логические выводы, отвечая на вопросы.

База знаний состоит из предложений - CLAUSES (отдельных знаний или утверждений): фактов и правил. Каждое предложение заканчивается точкой.

- Правило имеет вид: $A :- B_1, \dots, B_n$. A называется заголовком, а B_1, \dots, B_n – телом правила, причем все они являются термами.
- Факт – это правило, в котором отсутствует тело (т.е. тело пустое).

Вопрос состоит только из тела – составного термина (или нескольких составных терминов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами.

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». 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.