

## Самостоятельная работа №2 (Семинар3)

### Тема : Унификация и поиск с возвратом.

Выполнить ручную трассировку программ в тетради по образцу из пособия. Составить таблицу, состоящую из трех столбцов: первый столбец- протокол целей; второй столбец - соответствующие каждой цели(подцели) предложения программы; третий столбец-протокол означивания переменных. Отмечать точки отката. После выполнения ручной трассировки проверить полученные результаты на ПК. Цель – внутренняя и внешняя.

#### Программа 1

*trace*

*predicates*

*type(symbol,symbol)*

*is\_a(symbol,symbol)*

*lives(symbol,symbol)*

*can\_swim(symbol)*

*goal*

*can\_swim(What),*

*write("A ",What," can swim").*

*clauses*

*type(ungulate,animal).*

*type(fish,animal).*

*is\_a(zebra,ungulate).*

*is\_a(herring,fish).*

*is\_a(shark,fish).*

*lives(zebra,on\_land).*

*lives(frog,on\_land).*

*lives(frog,in\_water).*

*lives(shark,in\_water).*

*can\_swim(Y):-*

*type(X,animal),*

*is\_a(Y,X),*

*lives(Y,in\_water).*

#### Программа 2

*trace*

*predicates*

*likes(symbol,symbol)*

*tastes(symbol,symbol)*

*food(symbol)*

*clauses*

*likes(bill,X):-*

*food(X),*

*tastes(X,good).*

*tastes(pizza,good).*

*tastes(brussels\_spouts,bad).*

*food(brussels\_spouts).*

*food(pizza).*

Цель внешняя- *likes(bill,What).*

### Программа 3

```
trace
domains
  person=symbol
predicates
  mother(person,person)
  sister(person,person)
  aunt(person,person)
clauses
  mother(ann,kat).
  mother(ann,pat).
  sister(kat,pat).
  sister(ann,olga).
  aunt((X,Y):-
    sister(X,Z),
    mother(Z,Y).
```

Цель внешняя – aunt(olga,W)

### Программа 4

```
trace
domains
  name,thing=symbol
predicates
  likes(name,thing)
  reads(name)
  is_inquisitive(name)
clauses
  likes(john,wine).
  likes(ann,skiing).
  likes(Z,books):-
    reads(Z),
    is_inquisitive(Z).
  likes(ann,books).
  likes(ann,film).

  reads(john).
  is_inquisitive(john).
```

Цель внешняя – likes(X,wine),likes(X,books)

### Примечание

- Пролог не имеет оператора присваивания. Переменные связываются (инициализируются) при сопоставлении с константами в фактах и правилах. До связывания переменная считается свободной. После присвоения ей значения переменная становится связанной и остается такой, пока не будет получено решение по запросу. Затем система освобождает переменную и ищет другое решение, если оно есть. Переменные не есть хранилище данных (не часть памяти). Переменная используется только на этапе поиска решения. При рассмотрении и решении задач унификации обратите на это особое внимание.

- В процессе решения задачи Пролог выставляет указатели (точки отката) в тех местах программы, которые могут дать альтернативные решения. Не

забудьте эти точки в программе указать, потому что поиск другого решения начнется именно с этих точек. Откат – механизм, который используется для нахождения дополнительных фактов и правил, если текущая попытка вычисления цели оказалась неудачной.

### Самостоятельная работа №3 (семинар 4)

**ТЕМА: Управление поиском решения. Встроенные предикаты –fail и cut.**

1 Возьмите задачу о странах.

Эта программа показывает, как используется поиск с возвратом для выполнения повторяющихся операций. Поставьте внутреннюю цель – print\_countries и получите все ответы. Посмотрите и объясните, для чего используется в конце программы факт print\_countries.

```
predicates
  country(symbol)
  print_countries
clauses
  country(england).
  country(france).
  country(germany).
  country(denmark).
print_countries:-
  country(X),
  write(X),
  nl,
  fail.
print_countries.
```

a) Модифицируйте предикат country(name, population).

Выведите на печать только те страны, в которых население больше 10 миллионов(1e+7).

b) Измените программу о странах таким образом, чтобы ее результатами была выдача на экран последовательности целых чисел, например, 99,234,44,17. (это будет уже программа о числах, а не о странах) Каждое число должно находиться – на отдельной строке; \_ в виде одной строки, но числа должны быть разделены несколькими пробелами.

2 Возьмите программу с деревом родственных отношений.

a) Поставьте внутреннюю цель – parent(X,bob),write(X).

Заставьте программу с внутренней целью выдать все возможные решения,

b) Поставьте перед программой цель(внутреннюю) вывести на экран только женские имена членов семьи – show\_woman\_all ,а в программе это целевое утверждение опишите как правило, в котором будет использован предикат fail. Убедитесь, что такой способ упрощает процедуру запросов.

c) Поставьте перед программой внешнюю цель вывести на экран только часть женских имен, например, все имена до liz включительно.-show\_some\_of\_woman.

Используйте предикат cut для отсеечения лишних имен.

3. Представьте на языке Prolog утверждение "Аня любит все уроки , кроме английского". Как это можно выразить на языке Prolog в виде одного предложения, используя предикаты `cut(!)` и `fail`? А теперь переделайте это предложение, используя отношение `not`.

**4. Составьте программу, которая , используя предикат `not` , позволяет выявить успевающего по информатике студента , у которого средний балл за прошлый семестр не ниже 60 и который в текущем семестре пока не имеет долгов .**