Отчет по лабораторной работе №12
по курсу "ФИЛП"

Студент: Барсуков Н.М. ИУ7-56
Преподаватель: Толпинская Н.Б Строганов Ю.В.

Содержание

1 Текст задания

1) 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.

2) Составить программа - базу знаний, с помощью которой можно определить множество (живых) родственников некоего человека из города Н. Человек может быть одинок или иметь 1 родителя или быть счастливым обладателем полной ячейки общества.

2 Примеры вопросов и ответов

Листинг 1: код программы

```
1  include "lab_12.inc"
2
3  domains
4    fio, univ = string.
5
6  predicates
7    studing(fio, univ)
8
9
10 clauses
11   studing("Barsukov N.", "BMSTU").
12   studing("Barsukov N.", "Green Leaf").
13   studing("Ivanov I.", "BMSTU").
14   studing("Sidorov I.", "MGU").
15   studing("Sidorov I.", "BMSTU").
16   studing("Kolesnikov G.", "MAI").
17   studing("Ivanov I.", Where) :- studing("Kolesnikov
       G.", Where).
18
19 goal
20   /*studing(_, "BMSTU"). <-- If someone studing in
       BMSTU?*/
21   /*studing(Who, "BMSTU"). <-- Who is studing in
       BMSTU?*/
22   /*studing("Ivanov I", _). <-- Ivanov is studing?*/
23   /*studing("Ivanov I", Where). <-- Where is studing
       Ivanov?*/
```

Листинг 2: Возможные варианты вопросов к базе

```
1    studing(_, "BMSTU").
2    studing(Who, "BMSTU").
```

```
3     studing("Ivanov I", _).
4     studing("Ivanov I", Where).
```

Формулировка возможных вопросов:

1) Кто нибудь вообще учится в МГТУ?

2) Кто учится в МГТУ?

3) Иванов где нибудь учится?

4) Где Учится Иванов?

Ответы на вопросы:

1. yes

2. Who=Barsukov N; Who= Ivanov I; Who=Sidorov I; 3 Solutions

3. yes

4. Where=BMSTU, Where=MAI, 2 Solutions

3 Переменные

В прологе как почти и во всех других языках программирования есть понятие переменной. Различие заключается в том что в прологе понятие переменной ближе к математическому смыслу. parent(Parent, child) - Parent является переменной. Которое является не конкретным элементом, а элементом множества parent (может быть пустым). Переменные имеет локальную видимость внутри фактов и правил

В прологе различается двоя состояния переменной:

1) free (Свободные)

2) bound (связанные)

4 Порядок формирования результата работы программы

Процесс нахождения решения в Прологе заключается в сопоставлении предиката цели с предикатами базы знаний. Этот процесс называется унификацией. Пусть Пролог-системе предъявлена цель parent(X, kristina), parent(Y, X). Пролог выделяет из нее первую подцель parent(X, kristina) и начинает сопоставлять ее с базой знаний (проводить унификацию). База знаний повторена ниже.

lstname

```
1    parent(boris, alla).
2    parent(bedros, filipp).
3    parent(edmuntas, kristina).
4    parent(alla, kristina).
5    parent(kristina, denis).
```

Вначале сопоставляются первый предикат и подцель: parent(boris, alla) и parent(X, kristina)

Первый аргумент boris сопоставляется с переменной X. Следует иметь в виду, что в Прологе различаются состояния переменных free (свободные) и bound (связанные). Если обе переменные связаны, то при унификации происходит их сравнение. Если одна из них свободна, то происходит присвоение. Переприсвоени значений переменным не допускается. Это существенно отличает Пролог от прочих языков. Таким образом, переменной X, которая пока является свободной, присваивается значение boris. После этого унифицируются вторые аргументы, alla и kristina. Поскольку это константы, и alla не равно kristina, то унификация предиката parent(boris,alla) и подцели parent(X, kristina) заканчивается неудачей (fail). Поскольку в базе знаний несколько экземпляров предиката parent, такой предикат называется неоднозначным (non-derministic). Если предикат один, то он называется однозначным (deterministic). В случае неоднозначного предиката после неудачи выполняется откат – переход к следующему экземпляру предиката. При этом отменяется также присвоение значение переменным, если таковое имело место. Затем выполняется унификация предикатов parent(bedros, filipp) и parent(X, kristina) Очевидно, что результат унификации будет тот же, неудача (fail). При откате на следующий предикат parent(edmuntas,kristina) картина будет иная: X 10 присвоится значение edmuntas, а сопоставление вторых аргументов будет также успешным, так как kristina = kristina. Таким образом, первая подцель окажется выполненной. Пролог запоминает, какой экземпляр предиката сработал и устанавливает на следующий предикат указатель отката:

lstname

```
1    parent(boris, alla).
2    parent(bedros, filipp).
3    parent(edmuntas, kristina).
4    > parent(alla, kristina).
5    parent(kristina, denis).
```

Итак, интерпретатор Пролога автоматически выполняет поиск решения. Механизм поиска реализован с помощью отката после неудачи. Откат происходит на следующий экземпляр неоднозначного предиката. Выполнение программы на Прологе (резолюция цели) заключается в унификации цели с базой знаний.

5 Заключение

Программа на Prolog состоит из разделов. Каждый раздел начинается со своего заголовка. Структура программы:

1. Директивы компилятора - зарезервированные символьные константы
2. CONSTANTS — раздел описания констант
3. DOMAINS — раздел описания доменов
4. DATABASE — раздел описания предикатов внутренней базы данных
5. PREDICATES — раздел описания предикатов
6. CLAUSES — раздел описания предложений базы знаний
7. GOAL — раздел описания внутренней цели (вопроса).

В программе не обязательно должны быть все разделы.

Основными объектами программы на prolog являются: Предикаты, правила, цели.