Глава 2 Логическое программирование. Основы языка Пролог

PROLOG (programming in logic) - 1972 г., Колмероэ, Марсельский университет.

Реализации языка Пролог:

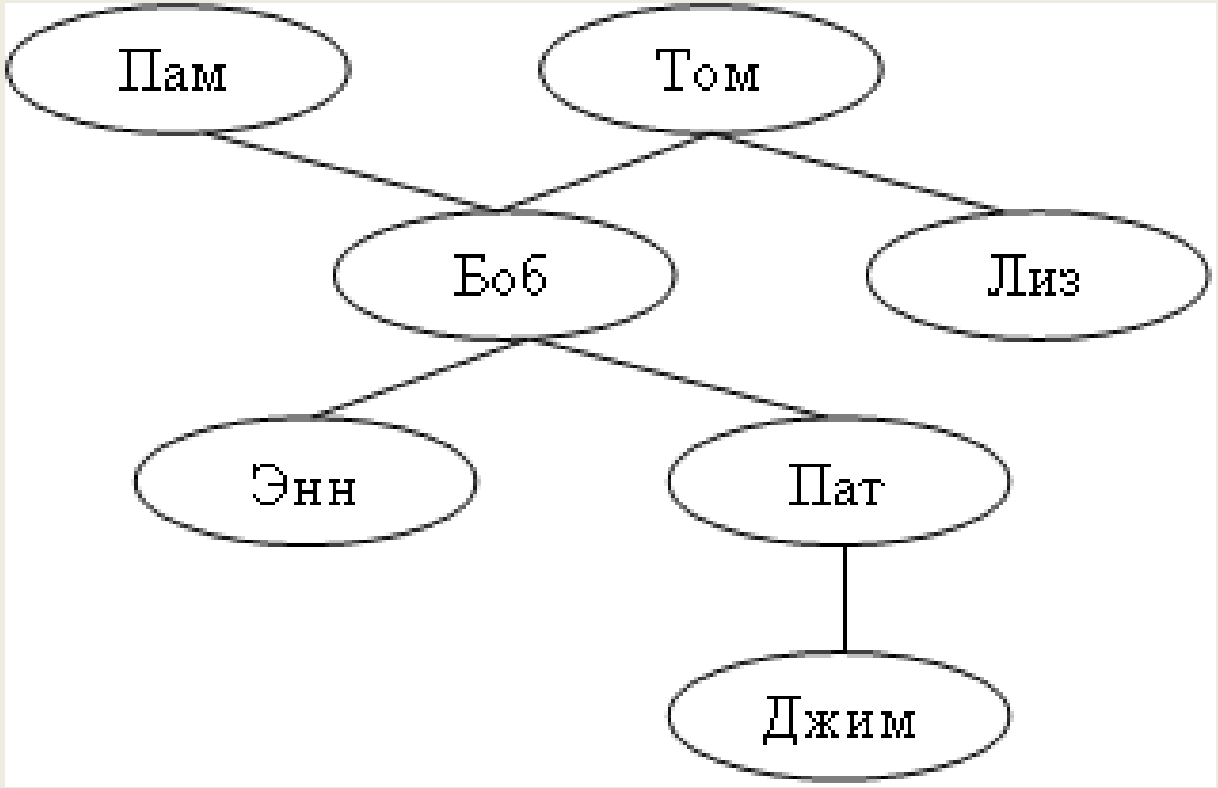
Wisdom Prolog, SWI Prolog, Turbo Prolog, Visual Prolog, Arity Prolog и т.д.

SWI-Prolog (SWI перевод с гол. социально-научная информатика) -1987 г., Ян Вьелемакер, Амстердамский университет.

SWI-Prolog Editor – среда программирования, работает с 32-битными версиями SWI-Prolog.

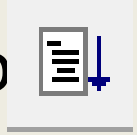
2.1 Факты и правила

Пример 1: Написать программу, описывающую дерево семейных отношений



Программа на Прологе, описывающая дерево семейных отношений

родитель(пам,боб).
родитель(том,боб).
родитель(том,лиз).
родитель(боб,энн).
родитель(боб, пат).
родитель(пат,джим).

Запускаем программу: F9 или ико  на панели инструментов.

Теперь можно задавать вопросы.

Вопросы

1. Боб является родителем Пат?

На Прологе: родитель(боб,пат).

Ответ Пролога: true.

2. Пат – ребенок Лиз?

На Прологе: родитель(лиз,пат).

Ответ Пролога: false.

3. Кто родители Лиз?

На Прологе: родитель(X,лиз).

Ответ Пролога: X = том.

4. Кто дети Энн?

На Прологе: родитель(Энн, X).

Ответ Пролога: false.

5. Кто дети Боба? (заметим, что их детей - двое)

На Прологе: родитель(Боб, X).

Ответ Пролога: X = энн ;

X = пат.

После найденного первого решения Пролог ждет дальнейших указаний: продолжить поиск решений (тогда нажимаем ;) или прекратить (тогда нажимаем .)

4. Есть ли дети у Пам?

На Прологе: родитель(пам, _).

Ответ Пролога: true.

Варианты ответов Пролог-системы на заданные вопросы

- true
- false
- перечисление возможных значений переменных в ответе, при которых утверждение истинно. Если решение не единственно, то Пролог ожидает дальнейших указаний по продолжению поиска решений. Точка – остановить поиск, точка с запятой – продолжить поиск.

Программа на Прологе состоит из фактов и правил.

Факт – безусловное истинное утверждение

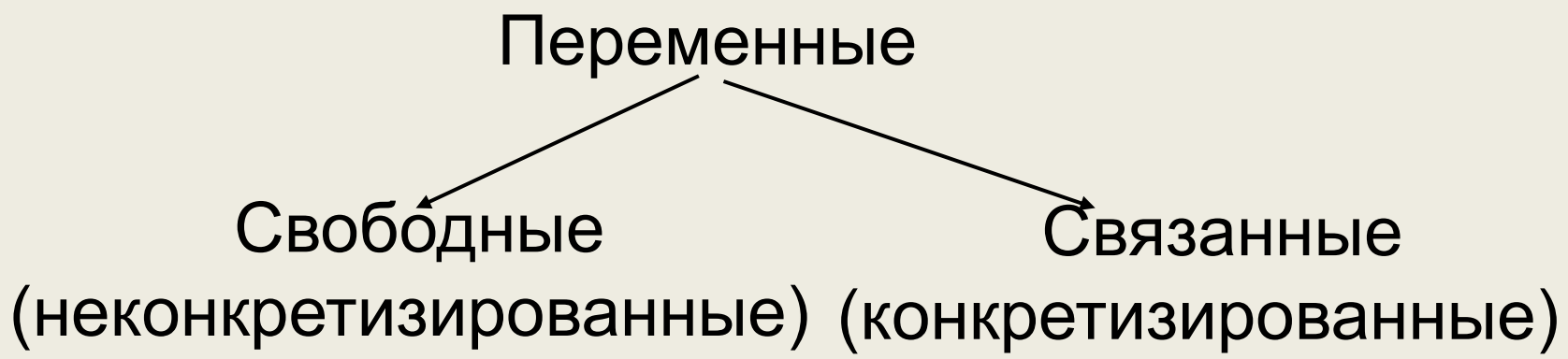
<имя предиката>(O_1,O_2,…,O_n).

O_i – конкретный объект (константа) или абстрактный объект (переменная).

Константы начинаются со строчной буквы

Переменные начинаются с прописной буквы или подчерка.

Переменная обозначает объект, а не область памяти! Поэтому не можем менять ее значение (типа X=X+1).



Область действия переменной – одно предложение! Связанная переменная не может изменяться внутри предложения.

Анонимная переменная начинается с символа подчеркивания и предписывает интерпретатору проигнорировать значение этой переменной.

Правило – утверждение, которое истинно при выполнении некоторых условий, позволяет описывать новые отношения.

<голова правила > :– <тело правила>.

Связки в теле правила:

,

;

not или \+

Порядок выполнения логических операций можно менять с помощью скобок.

Можно в теле правила использовать разветвление вида:

(<условие>-><действие 1>;<действие 2>)

Пример 2: Добавим одноместное отношение мужчина (факты). Опишем новое двуместное отношение дед в виде правила.

мужчина(том).

мужчина(боб).

мужчина(джим).

дед(X,Y):-мужчина(X), родитель(X,Z),родитель(Z,Y).

Вопрос: Кто дед Джима?

На Прологе: дед(X, джим).

Ответ Пролога: X = боб ;

false.

Вопрос: Кто внуки Тома?

На Прологе: дед(том,X).

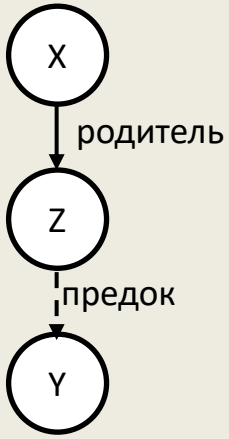
Ответ Пролога: X = энн ;

X = пат ;

false.

Пример 3: Опишем новое двуместное отношение предок.

Y - предок X, если выполнено одно из двух условий:

1. Y – родитель X
2. 

Получаем рекурсивное правило.

предок(X,Y):-родитель(X,Y);

родитель(X,Z),предок(Z,Y).

ИЛИ

предок(X,Y):-родитель(X,Y).

Получаем рекурсивное правило.

```
предок(X,Y):-родитель(X,Y);
                    родитель(X,Z),предок(Z,Y).
```

или

```
предок(X,Y):-родитель(X,Y).
предок(X,Y):-родитель(X,Z),предок(Z,Y).
```

Вопрос: Кто предок Джима?

На Прологе: предок(X, джим).

Ответ Пролога: X = пат ;

X = пам ;

X = том ;

X = боб ;

false.

2.2 Поиск решений Пролог-системой

Вопрос к системе – это последовательность, состоящая из одной или нескольких целей.

Факты и правила – аксиомы, вопрос – теорема.

При поиске ответа на поставленный вопрос находится факт или правило для содержащегося в вопросе предиката и выполняет операцию сопоставления (унификации) объектов предиката.

Операция унификации объектов успешна:

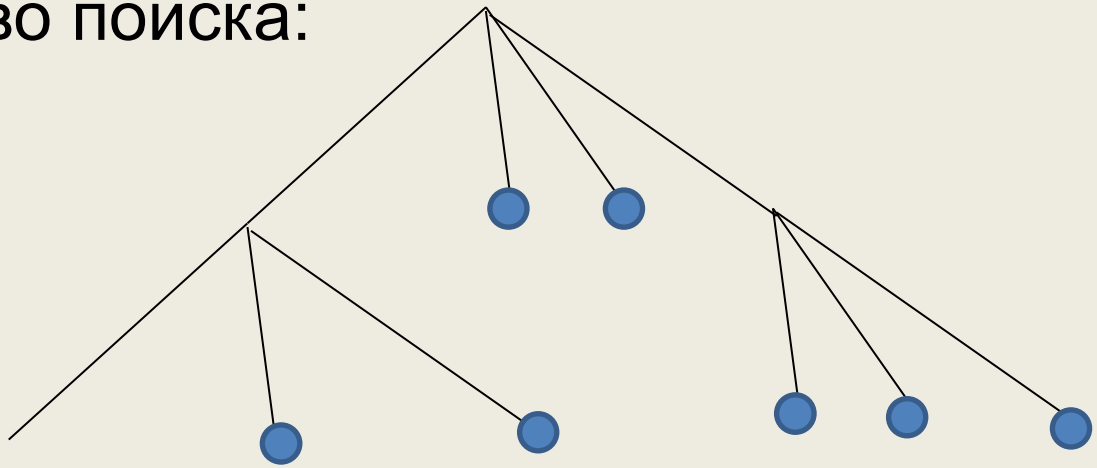
- сопоставляются две одинаковые константы;
- сопоставляется свободная переменная с константой (при этом свободная переменная становится означенной);
- сопоставляется связанная переменная с константой, равной значению переменной;
- сопоставляется свободная переменная с другой свободной переменной (переменные не получают значений, но становятся сцепленными, т.е. когда одна из них получит значение, то и вторая получит это же значение).

Процесс унификации похож на использование оператора =.

A=B может интерпретироваться как присваивание слева направо, справа налево, как сравнение.

Для достижения цели используется механизм отката

Дерево поиска:



Выделены точки отката, которые Пролог запоминает для поиска альтернативных путей решения.

Если цель была неуспешна, то происходит откат к ближайшему указателю отката.

Если цель достигнута, но использовались не все указатели отката, то будет продолжен поиск решений.

Трассировка

Включение трассировки: trace. или 

Отказ от трассировки: notrace. или 

Слова, появляющиеся в окне трассировки:

Call Далее указывается текущая цель

Exit Указывается цель, которая успешна.

Redo Возврат в отмеченную точку возврата для поиска альтернативного решения.

Fail Указанная цель не была достигнута.

В круглых скобках указывается глубина в дереве поиска, нумерация начинается с 8.

?- предок(том,энн).

2.3 Графический отладчик

guitracer. или 

или через меню Тест - GUI-Tracer.

затем

trace, <предикат>.

Откроется дополнительное окно графического отладчика. Оно разделено на 3 части: верхнее левое окно показывает текущие значения переменных, верхнее правое окно показывает дерево вызовов, а нижнее – текст программы. Для пошагового выполнения следует нажимать значок стрелки, направленной вправо или пробел.

2.4 Некоторые операции в SWI-Prolog

=	Унификация (присваивание значения несвязанной переменной)
<, =<, >=, >	Арифметические (только для чисел) операции сравнения
=:=	Арифметическое равенство
=\=	Арифметическое неравенство
is	Вычисление арифметического выражения
@<, @=<, @>=, @>	Операции сравнения для констант и переменных любого типа (чисел, строк, списков и т.д.)
==	Равенство констант и переменных любого типа
\==	Неравенство констант и переменных любого типа

2.5 Предикаты ввода-вывода

read(A)	Чтение значения с клавиатуры в переменную A
write(A)	Вывод значения A на экран без перевода строки
writeln(A)	Вывод значения A на экран с переводом курсора в начало следующей строки
nl	Перевод курсора в начало следующей строки
format(‘<строка~w>’,X)	<строка> <значение X>
format(‘<строка~w~w>’, [X,Y])	<строка> <значение X> <значениеY>
format(‘<строка1~w\n строка2~w>’,[X,Y])	<строка1> <значение X> <строка2> <значениеY>

Некоторые полезные сведения

При ожидании ввода выводится приглашение |:

Комментарии заключаются между /* и */ или % комментируется строка.

pwd. – текущая папка

ls. – содержимое текущей папки

cd(‘<путь к папке, слэши дублируются’>). – сменить текущую рабочую папку

2.6 Рекурсия

Рекурсия: алгоритмическая или по данным.

Возвращаемое значение – в аргументе.

Передача параметров по значению.

Пример 1:

Определим одноместный предикат, который заданное количество раз выводит на экран строку **********.

```
w(0).
w(N):-N>0,writeln('**********'),N1 is N-1,w(N1).
```

Обращение к предикату: w(5).

Ответ Пролога:

```
**********
**********
**********
**********
**********
true.
```

Пример 2:

Определим одноместный предикат, вычисляющий n!. Ввод n с клавиатуры и вывод результата будет осуществлять предикат goal.

```
goal:-write('N=?'),read(N),f(N,P),format('~w!=~w',[N,P]).
f(0,1).
f(N,P):-N1 is N-1,f(N1,P1),P is P1*N.
```

Обращение к предикату: goal.

Ответ Пролога: N=? 6.

6!=720

true.