

Системы логического вывода. Основы языка Prolog.

Название языка Prolog происходит из двух слов: PROgramming in LOGic , т. е. «Логическое программирование».

Язык Prolog является логическим декларативным.

Основной целью данной лекции является не столько изучение языка Prolog, сколько рассмотрение принципа логического вывода на примере этого языка.

В 1981 году Японское министерство международной торговли и промышленности объявило о создании исследовательской организации — Института по разработке компьютеров нового поколения. Главной целью организации было создание систем обработки информации, основанной на знаниях, в предположении последующей простоты управления за счет общения с пользователем на естественном языке.

В качестве основной методологии разработки программных средств для ЭВМ пятого поколения было избрано логическое программирование, ярким представителем является Prolog.

На сегодняшний день существует довольно много реализаций языка Prolog. Наиболее известные: BinProlog, AMZI-Prolog, Arity Prolog, Cprolog, GNU Prolog, Micro Prolog, МПролог, Prolog-2, Quintus Prolog, SICTUS Prolog, Silogic Knowledge Workbench, Strawberry Prolog, SWI Prolog, UNSW Prolog и др.

Классический пример:

%Сэм является дедом, Джон — его сын
%предикат говорит, что Сэм является родителем Джона
parent(sam, john).

%София является бабушкой, Джон — ее сын
%предикат говорит, что София является родителем Джона
parent(sofia, john).

%У Джона есть сын Ник и дочь Сара
%Джон является родителем Ника
parent(john, nick).
%Джон является родителем Сары
parent(john, sarah).

%определяем понятие «ребенок» через понятие «родитель»
child(Y, X):-parent(X, Y).

%вводим предикат, указывающий пол каждого члена семьи
male(sam).
male(john).
male(nick).
female(sofia).
female(sarah).

%матерью является родитель женского пола
mother(X, Y):-parent(X, Y), female(X).
% X является бабушкой для Y, если она является матерью некоего Z, являющегося родителем Y
grandma(X, Y):-mother(X, Z), parent(Z, Y).
grandpa(X, Y):-male(X), parent(X, Z), parent(Z, Y).

Основными элементами в программе на Prolog можно выделить:

- ◆ Объекты (термы, константы, переменные)
- ◆ Предикаты (наиболее распространенные — условия)

Предикаты бывают унарные, бинарные, ... n -арные — по количеству параметров-объектов. Например:

```
male(sam).  
male(john).  
parent(john, nick).
```

Переменные записываются с большой буквы:

```
bad(Mark):-Mark < 61.  
child(Y, X):-parent(X, Y).
```

Если значение переменной в предикате не важно для выражения, то его можно заменить анонимом «_»:

```
%имя ребенка не имеет смыслового значения для данного выражения  
%выражение означает «некто X имеет ребенка» и определяется через известный предикат «родитель»  
has_child(X):-parent(X,_).
```

Наиболее часто употребляемые **логические операции** при описании предикатов и правил:

- ◆ «:-» - логическое «Если», Например: «A:-B,C.» читается: «Если B и C истины, то A ИСТИННО»
- ◆ «,» (запятая) — конъюнкция, логическое «И».
- ◆ «;» (точка с запятой) — дизъюнкция, логическое «ИЛИ».
- ◆ «<» и «>» («>=» и «<=») - сравнение числовых данных (больше или меньше)
- ◆ «=» и «\=» - равно и не равно

Арифметические операции:

- ◆ «+», «-», «*», «/»,
- ◆ «**mod**» - получение остатка от целочисленного деления

f(X,Y,Z):-Z is X+Y.

?-f(1,2,X). %вопрос

X = 3. %ответ

Пример работами с числами и интервалами на примере баллов и оценок :

%предикат, выражающий находится ли число Mark между Left и Right

between(Mark, Left, Right):-Left < Mark, Mark < Right.

%оценка «удовлетворительно» если Mark между 60 и 76

average(Mark):-between(Mark, 60, 76).

%оценка «хорошо», если Mark между 75 и 91

good(Mark):-between(Mark, 75, 91).

%отлично, если Mark между 90 и 101

excellent(Mark):-between(Mark,90,101).

%плохо, если оценка Mark менее 61 балла

bad(Mark):-Mark < 61.

%устанавливаем соответствие констант полученным интервалам баллов

mark(Mark, excellent):-excellent(Mark).

mark(Mark, good):-good(Mark).

mark(Mark, average):-average(Mark).

mark(Mark, bad):-bad(Mark).

Пример выбора типа автомобиля в зависимости от места поездки:

%указываем предикат, выражающий связь типа авто с качеством дороги

car_for_road(coupe, asphalt).

car_for_road(sedan, asphalt).

car_for_road(universal, asphalt).

car_for_road(sedan, dirt).

car_for_road(universal, dirt).

car_for_road(jeep, stones).

%вводим предикат, выражающий связь местности и типа дороги

road_place(city, asphalt).

road_place(country, dirt).

road_place(beach, dirt).

road_place(forest, stones).

road_place(mountains, stones).

%формулируем правило вывода зависимости типа авто от типа местности

car_for_place(X, Y):- road_place(Y, Z), car_for_road(X, Z).

Пример выполнения:

?-car_for_place(X, city).

X = coupe

X = sedan

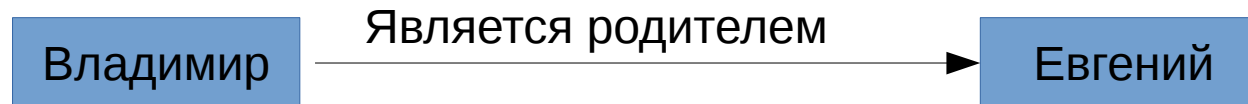
X = universal

При описании правил в виде предикатов важно понимать:

- ♦ В качестве предиката чаще выступает свойство различных объектов

```
red(wine).  
red(apple).  
red(dress_3)  
green(cucumber).  
green(dress_1).  
green(jacket).  
age(mary, 22).  
age(tanya, 21).  
age(alex, 24).
```

- ♦ В качестве предиката чаще выступает отношение между объектами:



```
parent(john, nick).  
parent(john, sarah).
```


Онлайн интерпретаторы языка Prolog:

- ◆ SWI Prolog: <http://swish.swi-prolog.org>
- ◆ CompileOnline.com: http://www.tutorialspoint.com/execute_prolog_online.php