



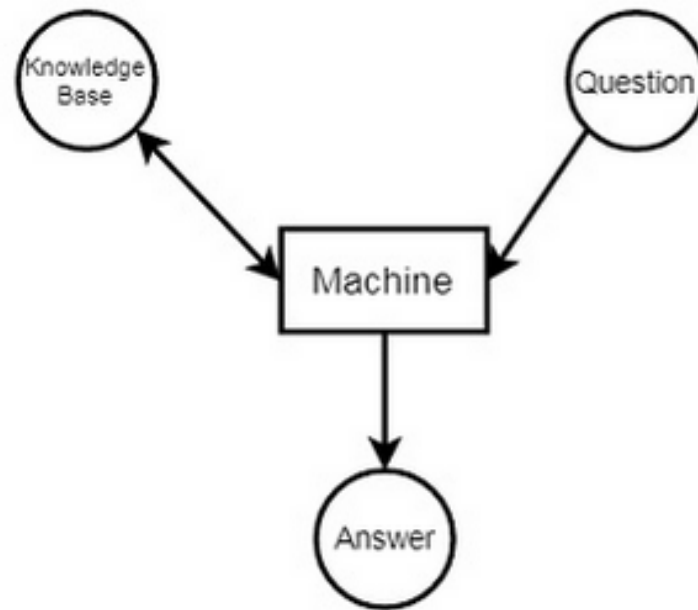
PROLOG

Осипенко Д.В. 595гр.

Что такое Prolog?

- Prolog (**PRO**gramming in **LOG**iscs) - логический и декларативный язык программирования.
 - Логическое программирование — это одна из парадигм компьютерного программирования, в которой операторы программы выражают факты и правила о различных проблемах в рамках системы формальной логики.
- Имеет важную роль в ИИ.
- Один из основных примеров языка четвертого поколения, поддерживающего парадигму декларативного программирования.
- Особенно подходит для программ, использующих символьные или нечисловые вычисления.
- играет жизненно важную роль в
 - Интеллектуальный поиск в базе данных ,
 - Понимание естественного языка,
 - Язык спецификации,
 - Машинное обучение,
 - Планирование роботов,
 - Системы автоматизации ,
 - Решение проблем

Логическое и функциональное программирование



Logical Programming

```
main(){  
  procedure1()  
  procedure2()  
  return  
}
```

Functional Programming

Функциональное vs Логическое

- Функциональное программирование

- следует архитектуре фон Неймана или использует последовательные шаги,
- синтаксис представляет собой последовательность выражений, таких как (a, s, l),
- вычисление происходит за счет последовательного выполнения выражений,
- логика и элементы управления смешаны вместе.

- Логическое программирование

- использует абстрактную модель или имеет дело с объектами и их отношениями,
- синтаксис в основном представляет собой логические формулы (предложения Хорна),
- он вычисляется путем вычитания предложений,
- логика и элементы управления могут быть разделены.

- Пролог в основном состоит из трех различных элементов:
 - Факты — это предикат, который верен, например, если мы говорим: «Том — сын Джека», то это факт.
 - Правила — это вымирание фактов, содержащих условные предложения. Для выполнения правила эти условия должны выполняться. Например, если мы определяем правило как **grandfather(X, Y) :- father(X, Z), parent(Z, Y)**, Это означает, что для того, чтобы X был дедушкой Y, X должен быть отцом Z, Z должен быть родителем Y.
 - Вопросы -- для запуска программы на прологе нам нужны некоторые вопросы, и на эти вопросы можно ответить с помощью данных фактов и правил.

ФАКТЫ

- Том - кот
- Маркиз любит есть макароны
- Волосы черные
- Максим любит играть в игры
- Данил ленивый.

Syntax:

relation(object1,object2,13243,...).

- cat(tom).
- loves_to_eat(markiz,pasta).
- of_color(hair,black).
- loves_to_play_games(maksim).
- lazy(daniel).

Правила (':-' ',' ';')

Syntax

```
rule_name(object1, object2, ...) :-  
fact/rule(object1, object2, ...).
```

% Предположим, что предложение похоже на

```
P :- Q;R.
```

% Это также можно записать как

```
P :- Q.
```

```
P :- R.
```

```
P :- Q,R;S,T,U.
```

% Это также можно записать как

```
P :- (Q,R);(S,T,U).
```

```
P :- Q,R.
```

```
P :- S,T,U.
```

Examples

% Том голоден, если он ищет еду

```
hungry(tom) :- search_for_food(tom).
```

% Джек и Били — друзья, если они оба любят играть в футбол

```
friends(jack, bili) :- lovesFootball(jack),  
lovesFootball(bili).
```

База знаний

% (KB1)

girl(pahsa).

girl(tania).

girl(dasha).

can_cook(pasha).

| ?- [kb1] % Загружаем базу знаний

.
compiling D:/TP Prolog/Sample_Codes/kb1.pl for byte code...
D:/TP Prolog/Sample_Codes/kb1.pl compiled, 3 lines read -
489 bytes written, 10 ms

yes

| ?- girl(pasha)

.

yes

| ?- girl(serega).

no

| ?- can_cook(pasha).

yes

| ?- can_cook(tania).

no

| ?-

MinMax

```
find_max(X, Y, X) :- X >= Y, !.
```

```
find_max(X, Y, Y) :- X < Y.
```

```
find_min(X, Y, X) :- X =< Y, !.
```

```
find_min(X, Y, Y) :- X > Y.
```

```
| ?- find_max(40,10,Max).
```

Max = 40

yes

```
| ?- find_min(40,10,Min).
```

Min = 10

Resistanse

```
series(R1,R2,Re) :- Re is R1 + R2.
```

```
parallel(R1,R2,Re) :- Re is ((R1 * R2) /  
(R1 + R2)).
```

```
| ?-parallel(10,40,R3),series(R3,12,R4),parallel(R4,30,R5).
```

R3 = 8.0

R4 = 20.0

R5 = 12.0

yes



СПАСИБО ЗА
ВНИМАНИЕ