# Logic programming

## Логические выводы

1. Прямой логический вывод
2. Обратный логический вывод

Прямой логический вывод - это метод формирования рассуждений от ФАКТОВ к ЗАКЛЮЧЕНИЯМ, которые следуют из фактов.

IF идёт на улице идёт дождь -> факт

THEN нужно взять зонтик ----> заключение

Обратный логический вывод - это метод формирования рассуждений от ГИПОТЕЗЫ к ФАКТАМ, которые подтверждают гипотезу.

Гипотеза - это потенциальное заключение, которое должно быть доказано.

IF кто-то вошел в дом с влажными ботинками и зонтиком -> факт

THEN идёт дождь -> приняли гипотезу

Для подтверждения гипотезы нужно спросить у человека, идет ли дождь.

Если да, то гипотеза истинная, то она становится фактом.

## 

## Хорновские выражения

(a1 ^ a2 ^ a3 ^ … ^ an) -> b, где ai и b eA. А - это множество всех булевых переменных.

Хорновское выражение утверждает, что из истинности всех условий набора {ai} следует истинность заключения b.

Пример:

“Условиями нормальной эксплуатации автомобиля является наличие бензина, масла и надутых шин”

Хорновское выражение:

х является исправным автомобилем и может нормально эксплуатироваться  
 IF x снабжен бензином и  
 IF х снабжен маслом и  
 IF х имеет надуты шины

Проблема определения, пригоден ли авто, сведена к 3 подброблемам, более простым (подцелям).

## Язык Prolog

Язык Prolog создан для реализации принципа логического программирования на основе обратного логического вывода с использованием хорновских выражений.

Prolog - это командный интерпретатор состоящий из таких компонентов:

1. интерпретатор или машина логического вывода;
2. база данных (факты и правила);
3. определённая форма сопоставления с шаблонами - **унификация**;
4. есть механизм перебора с возвратами: переход к исследованию альтернативных подцелей если попытка поиска для выполнения цели оказалась неудачной.

Семантическая сеть - это помеченный ориентированный граф.

Семантические сети легко преобразовываются в программу на языке PROLOG.

Операторы на прологе являются **предикативными выражениями** (или **предикатами**).

Программы на прологе состоят из фактов и правил! Они заданы в общей форме целей: p:- p\_1, p\_2,..., p\_N, где p - голова правила, а pk - это подцели.

## Обеспечения поиска в Prolog

В центре внимания находится интерпретатор. Интерпретатор принимает запросы и возвращает ответы.

Если запрос - это ФАКТ, то он находится в базе знаний и ответом будет: yes или no.

Если запрос - это правило, то выполняется поиск в глубину и интепретатор пытается выполнить подцели правила.

Поиск в глубину происходит от корня дерева и идёт слева направо.

В прологе применяются три типа хорновских выражений:

1. Факты (parent(susan, john))
2. Правила - способ задания предикатов: (ancestor(X,Y):-parent(X,Y))
3. Запросы (:- ancestor(ann, susan))

#### Пример поиска

Правила

1. ancestor(X,Y):- parent(X,Y).  
2. ancestor(X,Y):- ancestor(X,Z), ancestor(Z,Y).

Факты

3. parent(ann, mary).  
4. parent(ann, susan).

5. parent(mary, bob).  
6. parent(susan, john).

Запрос  
 :- ancestor(ann, susan).

Выполняется поиск выражения, голова которого согласуется с входным шаблоном ancestor(ann, susan) - **сопоставление с шаблоном**.

Начинаем с 1 оператора и двигаемся к 6у.

Согласуем шаблон с правилом 1 (сначала голову согласуем): переменная X согласуется с ann, Y - susan.

Итак, голова предиката согласована. Теперь пролог пытается согласовать тело правила 1 -> в итоге создаётся подцель parent(ann,susan).

После этого пролог берет подцель parent(ann,susan) и пытается согласовать её с другими выражаниеми. В итоге согласует с фактом 4.

Все цели подлежащие согласованию исчерпываются и пролог говорит: yes!

Важные особенности пролога:

1. Перебор с возвратами
2. Сопоставление с шаблоном
3. Имеет декларативный характер - исполняемая программа является спецификацией программы.