# Лекция 8 (06.04.2022)

## Prolog

База знаний содержит факты и правила. (База знаний — это символьная форма записи)

Переменные бывают именованные и анонимные. (с прошлой лекции)

Переменные получают значения в процессе работы системы. (переменная будет конкретизирована каким-то значением, но это может быть ошибочное предположение, поэтому должна иметься возможность отказа)

Prolog включает в себя механизм вывода, который основан на сопоставлении образцов термов, например терма вопроса и терма факта. Программа на Prolog представляет собой: базу знаний и вопрос.

Язык Prolog – позволяющий решать задачи искусственного интеллекта, является безтиповым языком программирования. Основным элементом языка является терм: константа, переменная или составной терм. В некоторых случаях, можно сказать, что составной терм является предикатом.

\*(Именованная переменная уникальна в рамках одного предложения)

\*(Анонимная переменная уникальна всегда)

В заголовке мы указываем знания, а в теле условия. (Это про правила)

Факт — это частный случай правила (имеет заголовок, не имеет тела). Безусловная истина.

А правило это условная истина.

## Процедурные и декларативные особенности.

Знания находятся в заголовках. Вопрос сравнивается с заголовками в формальном порядке – сверху вниз.

Система сравнивает по два составных терма.

## Подстановки и примеры термов

Основные термы не включают переменные.

Неосновные термы включают переменные. (Например: A(X1, X2, … Xn))

Подстановкой называется множество пар вида , где это переменные а это термы не содержащие переменные

Применение подстановки заключается в замене каждого вхождения переменной на соотв.

Терм B является примером трема A, если сущ. такая подстановка , что B=A

Терм C является общим примером термов A и B, если существуют такие подстановки и , что C=A и C=B

могут быть равны

A = plus(1, 2, z)  
 plus(x, y, 3)

Father(X,Y),father(Y,Z) значит X это дед Z

## Логические программы и процедуры

Процедура это – совокупность правил, имеющих заголовки с одинаковыми главными функторами, одним количеством аргументов, обозначающих объекты одной и той же природы. (То есть процедура — это одно знание, но оно может быть зафиксировано с помощью нескольких предложений и это знание называется предикат)

\*Мой пример процедуры:

plus(1, 2, 3).  
plus(2, 1, 3).  
plus(0, 3, 3).

## Механизм логического вывода, простейшие правила лог. Вывода

Простая программа - …, один вопрос

Правило вывода это – утверждения о взаимосвязи между допущениями и заключениями, которые с позиции исчисления предикатов верны всегда.

Всего может быть 4 варианта:

1. Факты не содержат переменных, и вопрос не содержат переменных
2. Факты не содержат переменных, а вопрос содержит переменные (Правило обобщения факта)
3. Факты содержат переменных, а вопрос нет. (Правило конкретизации)
4. Факты содержат переменных, и вопрос содержат переменных (не основные) (система должна построить общий пример)

Доказательство осуществляется с помощью двух правил, сначала конкретизация, а потом обобщение.

Унификация — это операция которая, позволяет формализовать процесс вычислительного алгоритма? (Подобрать знания из БЗ)

Всё это осуществляется с помощью унификации:

1. происходит двунаправленная передача параметров процедурам
2. неразрушающее присваивание (конкретизация)
3. проверка условий

T1 = T2 – явно запускаем алгоритм унификации

Два терма унифицируются по след. правилам:

1. Если T1 и T2 константы, то только если они совпадают
2. Если T1 не конкретизированная переменная, а T2 константа или составной терм, не содержащий в качестве аргумента T1, то унификация успешна, а T1 конкретизируется значением T2
3. Если Т1 и Т2 не конкретизированные переменные, то их унификация всегда успешна, причём Т1 и Т2 становятся сцепленными, то есть двумя именами (указателями) одного и того же объекта. (если один из них конкретизируется, то моментально конкретизируется и второй)
4. Если Т1 и Т2 составные термы, то они успешно конкретизируются если:
   1. У Т1 и Т2 одинаковые главные функторы
   2. Т1 и Т2 имеют одинаковые R-ности (одинаковое кол-во аргументов)
   3. Успешно унифицируется каждая пара их соотв. компонентов

## Алгоритм унификации (на следующую лекцию)