

# Лекция 1.

September 7, 2020

9:58 AM

1. Системы основанные на знаниях. Для поддержки принятия решений. Примеры: системы диагностики, системы разведки, распознавания образов.
2. Знания можно обрабатывать.
3. Модели:
  - a. Продукционные
  - b. Сети
  - c. Фреймы
  - d. Не чёткая логика и нейронные сети

Продукционная модель - знания представляются в виде совокупности правил : "Если - то".

Пример:

Если двигатель не заводится и фары не горят, то сел аккумулятор

Если бензин на нуле, то двигатель не заводится

Знания основаны на опыте

Составные элементы (базы знаний?):

- База правил
- Рабочая память
- Механизм логического вывода

Механизм не нужно реализовывать. Нужно описать предметную область и в ней будет уже храниться решение.

В основе продукционных моделей лежит Резолюция. И они декларативные. Недостаток : трудно описать сложную предметную область. Для таких систем нет отладчиков.

Механизм: прямой (от фактов к заключению), обратный (от гипотезы к фактам).

Механизм обратного считается наиболее удобным, потому что рассматривает правила которые явно относятся к решаемой задаче.

Семантические сети - смысловая сеть, ориентированный граф. Вершины графа - понятия, дуги - отношения между понятиями.

Наиболее часто встречаются следующие типы связей: часть - целое, функциональные связи

Проблема поиска для семантических сетей связана с поиском подсети, подходящей для выставленных требований.

Фрейм - это абстрактный образ для представления некоторого стереотип восприятия. Возник для обозначения структуры знаний.

Фрейм - структура, - роль, - ситуация, - сценарий.

(1) список свойств

(2) может храниться присоединенная процедура

(3) свойства через наследование

## PROLOG

Язык позволяет реализовывать сети, модели.

Представляет собой парадигму логического программирования. Ключевое понятие - предикат - отношения связывающие определённый объект.

Программа строится как набор утверждений. Вычисления представляют собой доказательства для поиска ответов. Декларативен.

Прикладные задачи: разработка экспертных систем, автоматическое доказательство теорем, создание баз знаний с возможностью логического выбора.

Удобен для комбинаторных задач.

При записи предиката указывается его имя. Далее в скобках аргументы.

Минимальная единица - атом.

:- импликация

Основной механизм в основе логического поиска - поиск с возвратом.

Следующий механизм: рекурсия.

Процедурный подход здесь сложнее для анализа.

$a \leftarrow v$  - вернёт ложь для атомов

$X = a$  -  $X$  принимает значение  $a$

$A \rightarrow B$

Списки в ПРОЛОГЕ:

[a,b,c,d]

длина([], 0).

длина([X|Y], L):- длина(Y, L1), L is L1+1

список([заказ(a,100),заказ(b,80)])

запрос на подсчет кол-ва элементов  
↓  
длина([a,b], X) → 1.е нужно  
не забыть

либо а отменяется  
возвращается к правому  
Список укорачивается  
и на обратном  
ходе снимается длина.

$X = 2$

для задачи:

сумма([], 0)  
сумма([заказ(-, X)|Y], S):-  
сумма(Y, S1), S is S1+X

→ хранение в  
стек

↓ на обратном  
ходе происходит  
возвращение S

нельзя использовать правописание  
рекурсия  
В запросе:  
список(X), сумма(X, S)

База данных  
хранится в виде  
списка



