

# Экспертные системы. Представление знаний в экспертных системах

Муртазин Т.М.  
Кафедра автоматизации  
телекоммуникации и метрологии

# Разделы курса

1. Экспертные системы как системы, основанные на знаниях.  
Классификация, структура. Модели представления знания
2. Интеллектуальные системы поддержки и принятия решений в ключе ЭС.  
Классификация. Порядок разработки. Обзор моделей ИСППР (модели регрессии, ИНС, модели нечеткой логики, случайный лес, бустреп).
3. Методы интеллектуального анализа. Инженерия знаний. Добыча данных (Data Mining).  
Работа с данными, математический аппарат, характеристические показатели качества данных.  
Технологии извлечения знаний на основе анализа больших данных (Общие сведения  
о технологии «Большие данные», ее роль при разработке ЭС)  
Применение технологии больших данных в задаче диагностики состояния оборудования. Обзор программных средств для построения предиктивных моделей диагностики оборудования (Loginom, Statistica, Excel, Phyton)
4. Экспертные системы в управлении технологическими объектами  
Инструментальные средства разработки экспертных систем. Этапы создания экспертных систем. Примеры реализации. Автоматные модели принятия решений

# Что такое экспертные системы

Компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации (википедия).

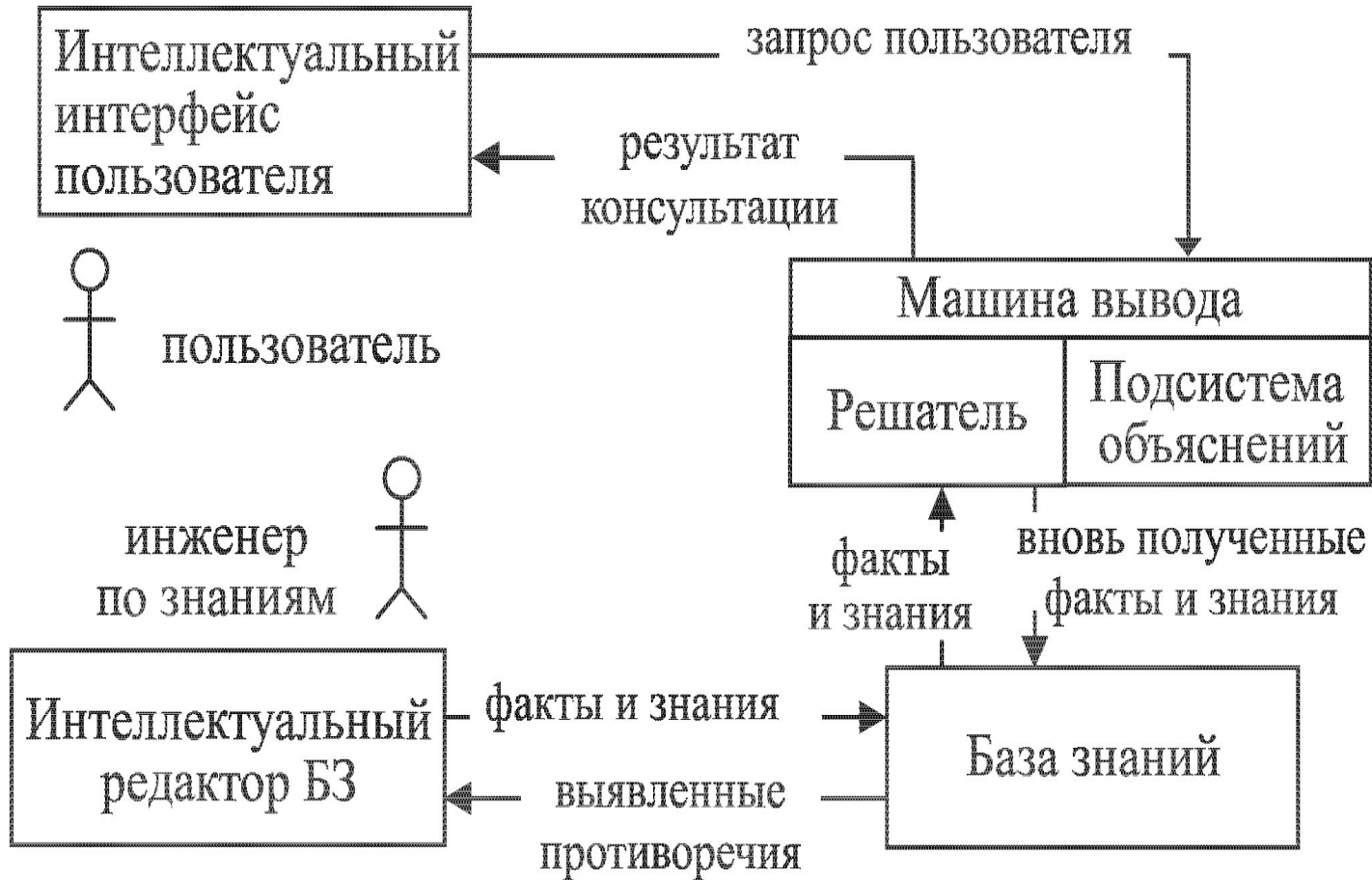
Коммерческое развитие 1980-е годы.

ЭС предметно-ориентированы.

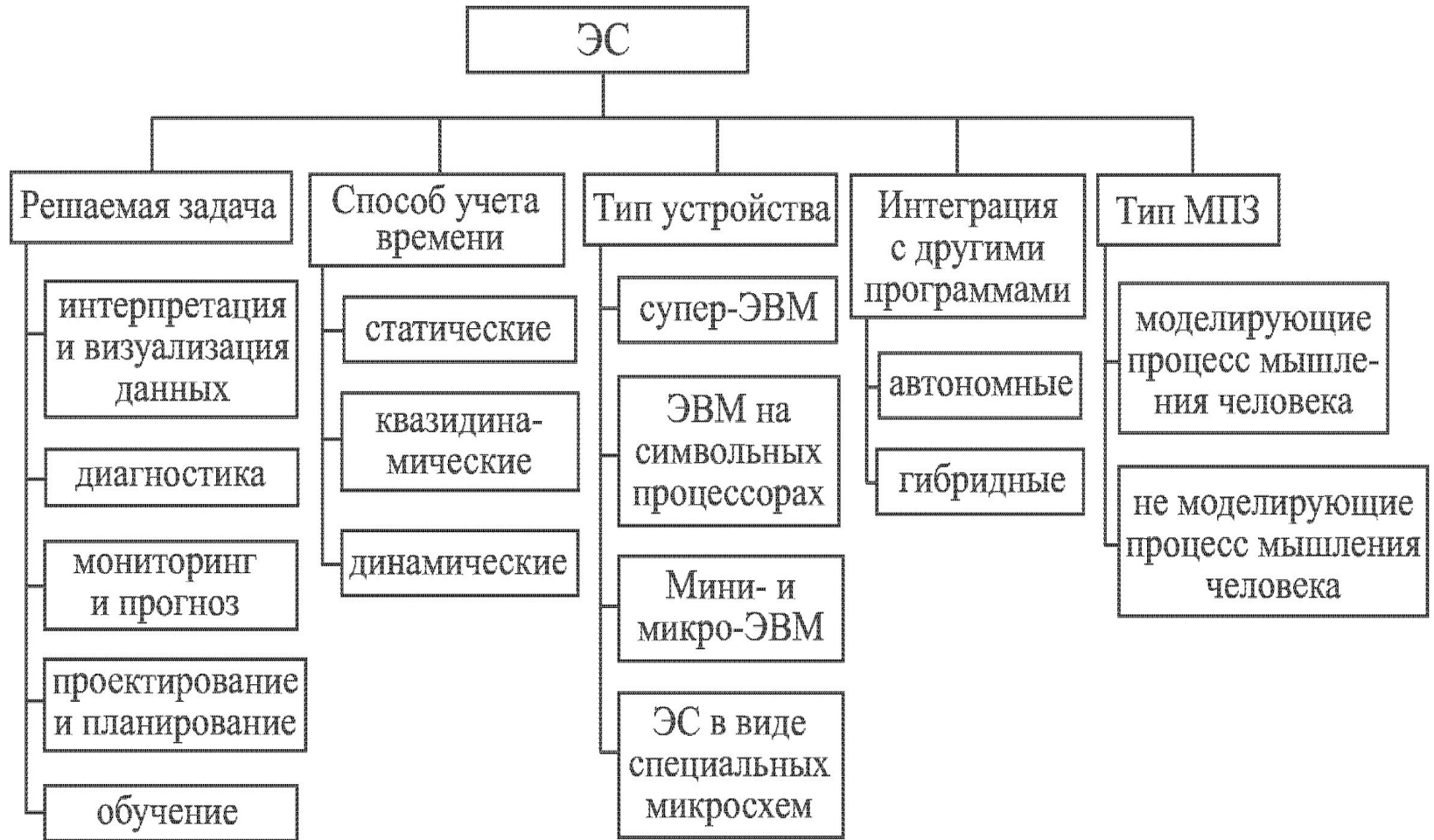
Проблемы создания:

1. Извлечения знаний
2. Формализации знаний
3. Нехватки времени экспертов
4. Недостаток ресурсов
5. Инструментальные средства

# Структура ЭС



# Классификация ЭС

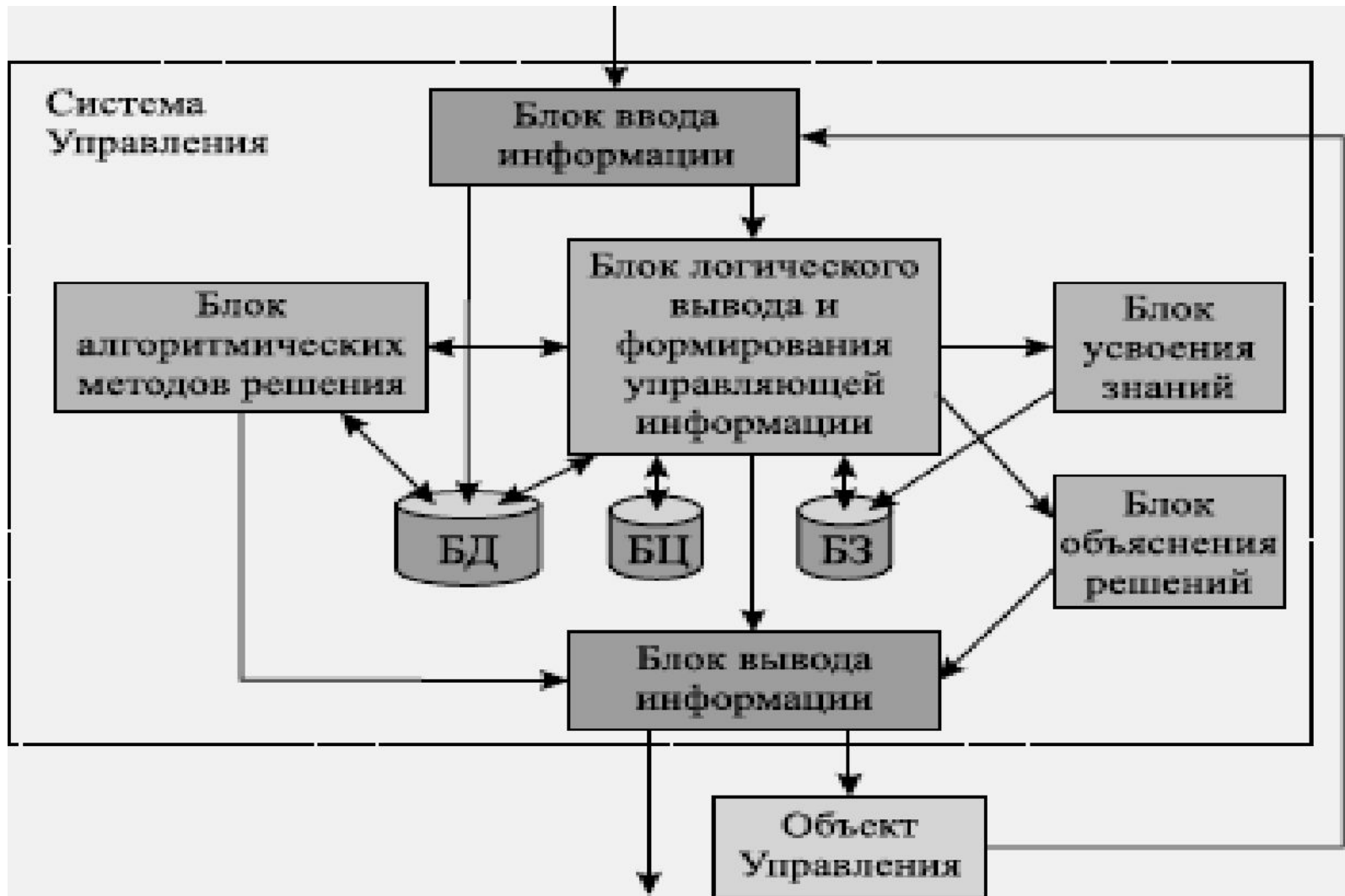


# Что такое интеллектуальное управление

## Обобщенная структура интеллектуального робота



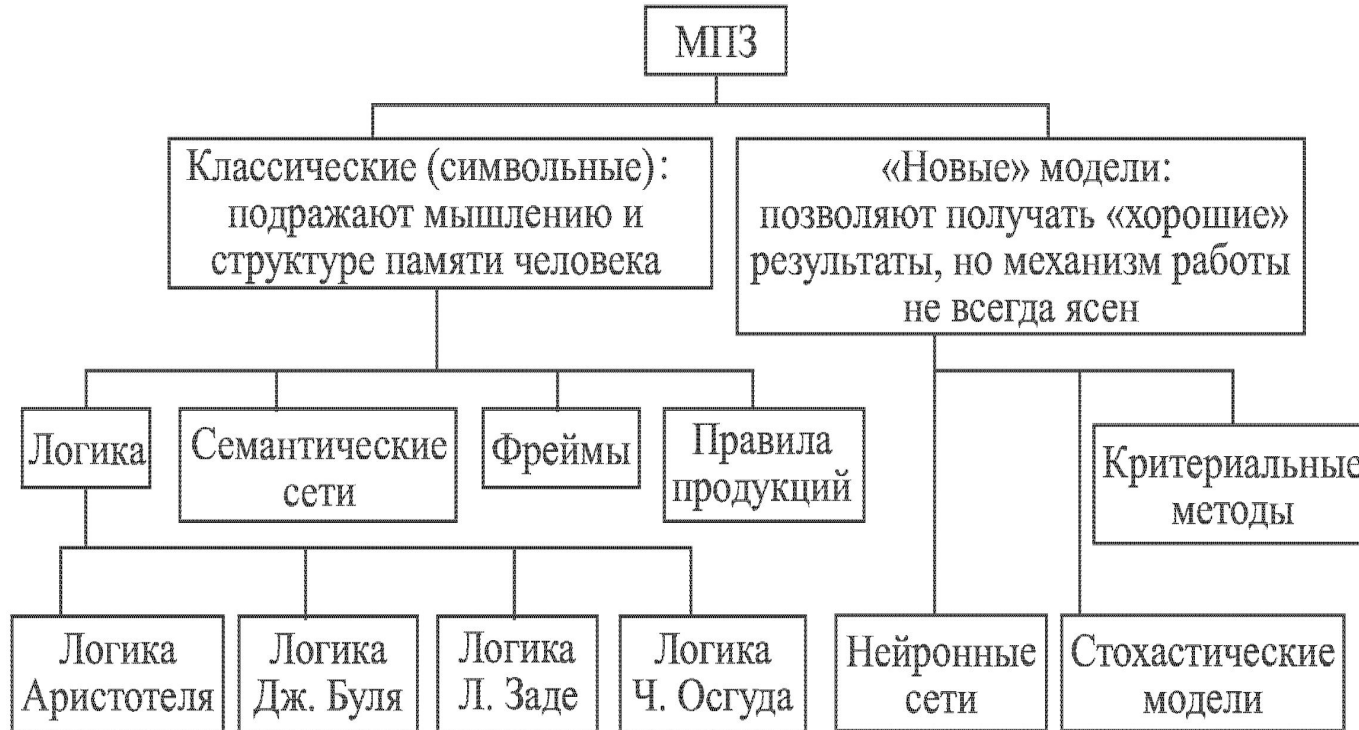
# Структура ИСУ



# Модели представления знаний

Модель представления знаний – это способ задания знаний для хранения, удобного доступа и взаимодействия с ними, который подходит для решения задач интеллектуальной системы принятия решений (ЭС).

Смолин Д.В. Введение в искусственный интеллект: конспект лекций. – М.: ФИЗМАТЛИТ, 2004. – 208 с.





# Классификация по типу МПЗ

## Классификация ЭС по типу МПЗ

Модель представления знаний	ЗНАНИЯ			
	Классификация по глубине		Классификация по жесткости	
	глубинные	поверхностные	мягкие	жесткие
Логика	—	+	—	+
Продукции	—	+	+	+
Фреймы	+	—	+	+
Семантические сети	+	—	+	+
Объектно-ориентированные языки	+	—	+	+
Стохастические модели	Эти модели не моделируют процесс мышления			
Критериальные языки выбора				
Нейронные сети				

# Модели представления знаний

Декларативные  
модели

Семантические  
сети

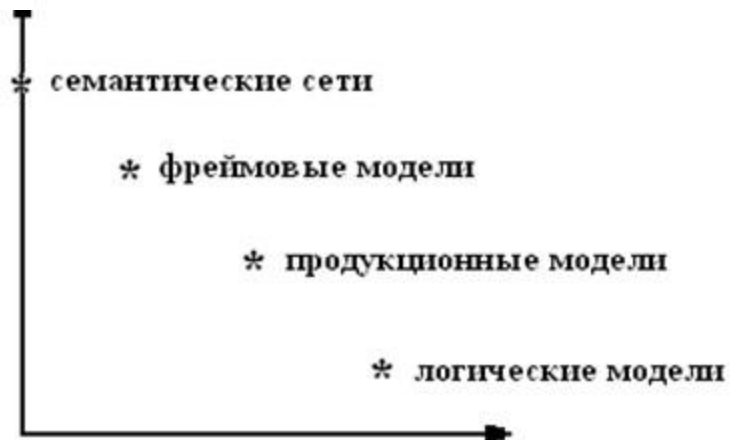
Фреймы

Процедурные  
модели

Логические  
модели

Системы  
продукций  
(продукционные  
модели)

Декларативные  
знания

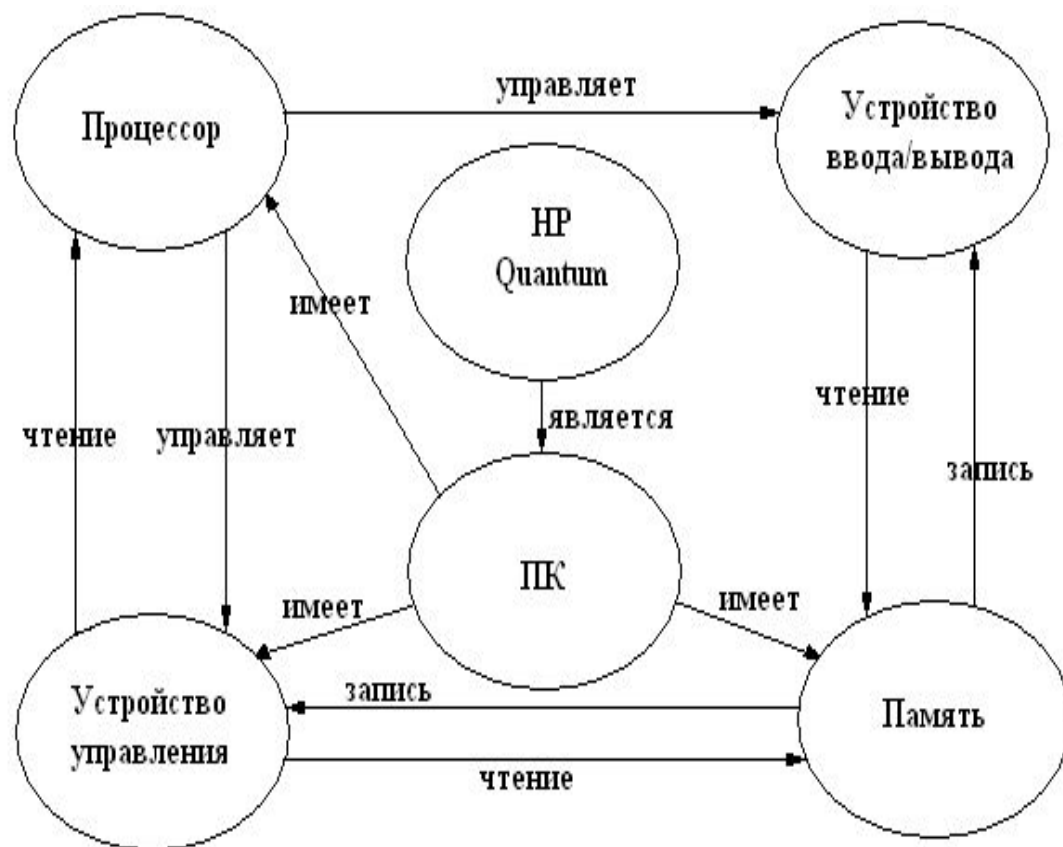


Процедурные знания

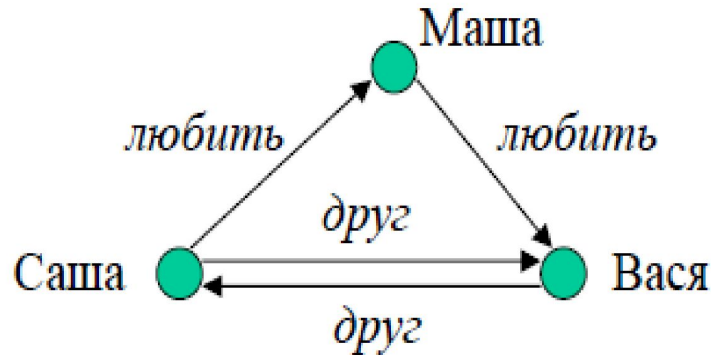
# Семантические сети

**Семантическая сеть** - это графы, вершины которого соответствуют понятиям (события, процессы, явления), а дугам графа - отношения между этими понятиями.

Вся необходимая информация может быть описана как совокупность троек  $(arb)$ , где  $a$  и  $b$  - объекты или понятия,  $r$  - бинарное отношение между ними.



# Семантические сети



Наиболее часто в семантических сетях используются следующие отношения:

- связи типа «часть - целое» («класс - подкласс», «элемент - множество» и т. п.);
- функциональные связи (определяемые обычно глаголами «производит», «влияет»,...);
- количественные (больше, меньше, равно, ...);
- пространственные (далеко от, близко от, за, под, над,...);
- временные (раньше, позже, в течение,...);
- атрибутивные связи (иметь свойство);
- логические связи (и, или, не) и др.

# Типы семантических сетей

*Классификация СС.*

**По типам отношений:**

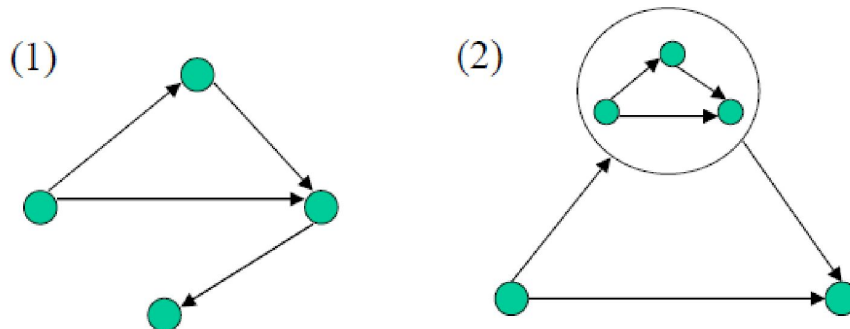
- бинарные (в которых отношения связывают два объекта);
- n-арные (в которых есть специальные отношения, связывающие более двух понятий, например «треугольник»).

**По количеству типов отношений:**

- однородные (с единственным типом отношений);
- неоднородные (с различными типами отношений).

**По типу вершин.**

Если вершины сети не обладают внутренней структурой, то такие сети называют **простыми (1)** сетями, если вершины обладают внутренней структурой, то такие сети называют **иерархическими (2)**.



# Семантические сети

- Очень полезным свойством, которым могут обладать отношения СС, является **транзитивность**.
- Пример. Рассмотрим предложения «Куин Мери является океанским лайнером» и «Каждый океанский лайнер является кораблем». Используются дуги «является». Поскольку мы знаем свойства дуг, связывающих узлы (отношение «является» - ISA – транзитивно), мы можем вывести из сети третье утверждение: «Куин Мери является кораблем», хотя оно и не было сформулировано в явном виде.
- Транзитивность: для любых  $a, b \in R$  таких, что  $a < b$  и  $b < c$ , справедливо соотношение  $a < c$ .
- Пример нарушения транзитивности:  
А сын В, С сын В, но С не сын А.

# Представление семантических сетей матрицей смежности. Формализация.

Матрица смежности – квадратная матрица  $A = \{a_{ij}\}$ ,

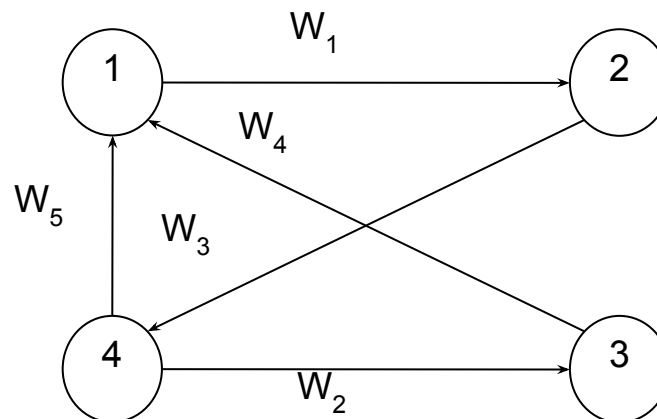
$i, j = \overline{1, m}$ , где  $m$  – число узлов, т.е.  $A_{m \times m}$ , для которой

$$a_{ij} = \begin{cases} 1, & \text{если есть дуга из узла } v_i \text{ в узел } v_j \\ 0, & \text{если дуги нет.} \end{cases}$$

Пример

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

$$A = \begin{bmatrix} 0 & W1 & 0 & 0 \\ 0 & 0 & 0 & W3 \\ W4 & 0 & 0 & 0 \\ W5 & 0 & W2 & 0 \end{bmatrix}.$$



# Практические приложения модели СС

1. Джейм Карбонел (Jaime Carbonell) создал специфичную экспертно-обучающую систему SCHOLAR, предназначенную для изучения географии Южной Америки. Из подобных систем в последствии выросли современные ГИС – географические информационные системы.

2. Хайдерн создал PLNLP (Programming Language for Natural Language Processing) – язык программирования для обработки естественного языка (задача анализа сходства двух предложений; частный случай — одно предложение является частью другого). Этот язык используется для работы с большими грамматиками. PLNLP работает по следующим правилам:

- с помощью правил декодирования производится синтаксический анализ линейной языковой цепочки и строится сеть;
- с помощью правил кодирования сканируется сеть, порождается языковая цепочка или другая трансформированная сеть.

3. Создан специальный сетевой язык NET.

4. Созданы широко известные экспертные системы, использующие семантические сети в качестве языка представления знаний – PROSPECTOR, CASNET, TORUS.



# Приложения

- TheBrain помогает вам организовать все ваши веб-страницы, контакты, документы, электронные письма и файлы в виде визуализируемой семантической сети.
- Элементы связаны в визуальной сети, которая разработана по принципу имитации работы головного мозга.
- Интерфейс PersonalBrain - это графическая карта, похожая на ментальную, но позволяющая любому предмету стать центральной темой.
- Продукт позволяет организовать информацию в сети, преодолевая ограничения иерархического хранения и мышления.

Gleb's Brain THE BRAIN

Search

Research subjects ... \_Things to reme... Skills I want to dev... 🤔 R

May 30, 2017

- Barba.js
- Javascript Frameworks
- vuejs.org
- deepstreamHub
- Basis.today telegram bot
- Tech to try
- Basis.today**
- Themes from the Songbook (ECM 2188) – between...
- Giya Kancheli
- Sobaka Pavlova
- Olga Pavlova

• Hemorrhagic stroke  
 • Documentaries  
 • A new way to study the bra...  
 • The Best Marijuana Docu...  
 • Optimize Your Brain: The S...  
 • Amygdala  
 • Ultra-fast Fear Responses ...  
 • The Mescaline experimen...  
 • The mind isn't locked in th...  
 • Reconstructing through altered states – Mind Hacks

61 Электростимуляция моз...  
 Transcranial direct current stim...  
 Point-of-view shot  
 David Lynch  
 Netflix  
 • 'I felt as if I had become fear itself': life after a stroke at 34

**My Beautiful Broken Brain**

Bessel van d...  
 Want to 'train y...  
 Your brain doe...  
 Grey Matter  
 Brain Developme...  
 Synaptic pruni...  
 My Beautiful Br...

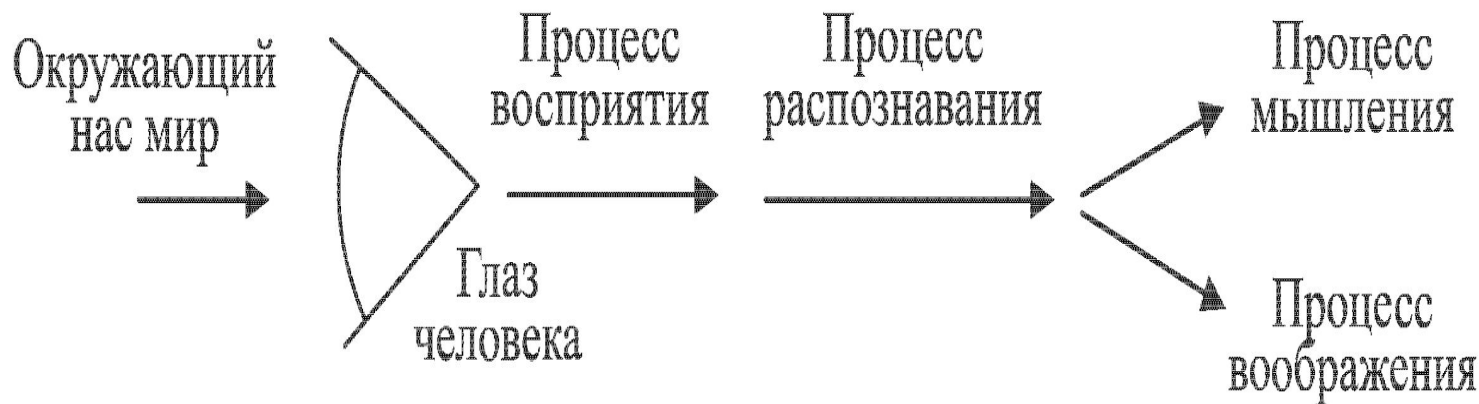
# Аналоги The Brain

- FreeMind
- Xmind
- Freeplane
- **И др. см.**  
<https://ruprogi.ru/software/freemind>

# Фреймы

Марвин Мински (1979г). Фрейм с англ. переводится как «каркас» или «рама».

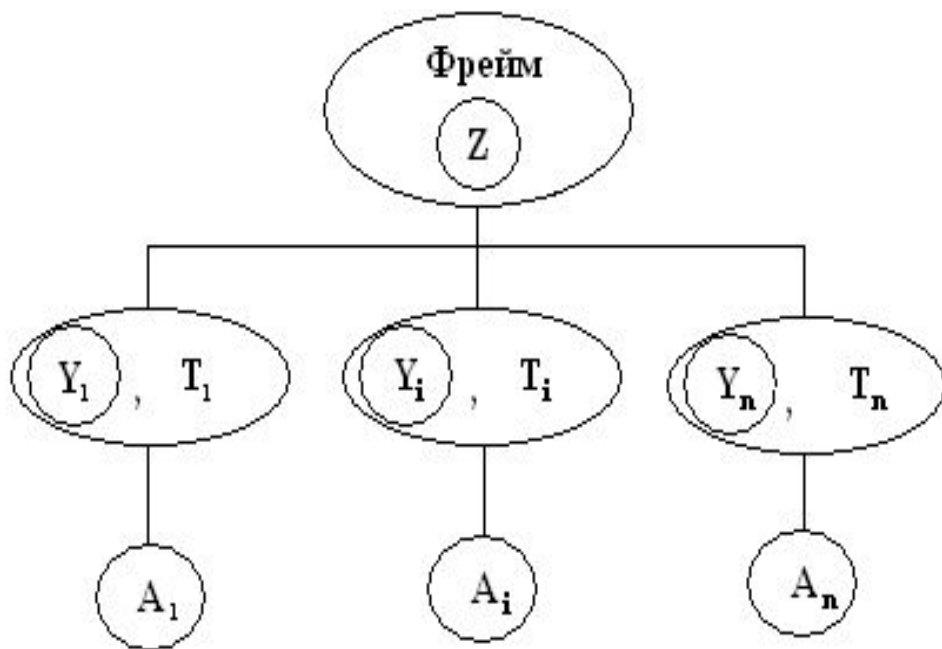
Предложена как структура знаний для восприятия пространственных сцен. Идея: сконцентрировать все знания о конкретном классе объектов или событий в единой структуре.



**Фрейм** - некоторое минимальное описание объекта, т.е. набор таких его признаков (атрибутов), без любого из которых описать этот объект невозможно.

# Структура фрейма

- Структура фрейма в виде списка слотов
- $Z[\langle Y_1, T_1 \rangle, \langle Y_2, T_2 \rangle, \dots, \langle Y_k, T_k \rangle]$
- где  $Z$  – имя фрейма; пара  $\langle Y_i, T_i \rangle$  –  $i$ -ый слот,  $Y_i$  – имя слота и  $T_i$  – его значение.



Имя фрейма (Z)	Имя слота (Y)	Значение слота (T)
ПК HP	Процессор	Int P
	Монитор	22'
	Жест.диск	500 Гб
	ОЗУ	4 ГБ
	ОС	Win 8

$A_i$  - имя присоединенной процедуры.

**Фрейм-прототип** – фрейм с незаполненными слотами

**Фрейм-экземпляр** – фрейм с заполненными слотами

# Структура фреймов

Имя слота	значение слота	способ получения значения	присоединённая процедура

**Имя фрейма.** Оно служит для идентификации фрейма в системе и должно быть уникальным. Число слотов в каждом фрейме устанавливается проектировщиком системы, при этом часть слотов определяется самой системой для выполнения специфических функций (системные слоты), примерами которых являются: слот-указатель родителя данного фрейма (**IS-A**), слот-указатель дочерних фреймов, слот для ввода имени пользователя, слот для ввода даты определения фрейма, слот для ввода даты изменения фрейма и т.д.

**Имя слота.** Оно должно быть уникальным в пределах фрейма. Обычно имя слота представляет собой идентификатор, который наделен определенной семантикой. В качестве имени слота может выступать произвольный текст.

Например, <Имя слота> = староста группы БУС-20, <Значение слота>= ФИО. Имена системных слотов обычно зарезервированы, в различных системах они могут иметь различные значения.

Примеры имен системных слотов: *IS-A*, *HASPART*, *RELATIONS* и т.д. Системные слоты служат для редактирования базы знаний и управления выводом во фреймовой системе.

# Атрибуты фрейма

В таблице дополнительные столбцы предназначены для описания способа получения слотом его значения и возможного присоединения к тому или иному слоту специальных процедур, что допускается в теории фреймов. В качестве значения слота может выступать имя другого фрейма; так образуют сети фреймов.

Модель фрейма позволяет отобразить все многообразие знаний о мире:

- через фреймы-структуры, для обозначения объектов и понятий (университет, семья, автомобиль);
- через фреймы-роли (руководитель, кассир, клиент);
- через фреймы-сценарии (банкротство, рабочее совещание, празднование именин);
- через фреймы-ситуации (тревога, авария, рабочий режим устройства) и др.

# Иерархическая система фреймов

- Между фреймами могут существовать аналогии
- Сложные объекты представляются комбинацией нескольких фреймов (вложенными фреймами)
- Фрейм может наследовать информацию от предшественников (АКО-связи (от англ. A Kind Of - «разновидность»). Слот с именем АКО указывает на имя фрейма более высокого уровня иерархии.

Имя фрейма (Z)	Имя слота (Y)	Значение слота (T)
Легковой автомобиль	<i>IS-A</i>	Автомобили или
	Двигатель	Бензиновый
	Колес	4
	Кузов	Седан

Надфрейм: автомобили

Подфреймы: двигатель, кузов и т.

Имя фрейма (Z)	Имя слота (Y)	Значение слота (T)
Двигатель	<i>IS-A</i>	Легковой автомобиль
	Объем	1,5 л.
	Мощность	250 л.с.
	Цилиндров	4

Надфрейм: легковой автомобиль



# Фреймы. Преимущества. недостатки

Основным преимуществом **фреймов** как **модели представления знаний** является то, что она отражает концептуальную основу организации памяти человека, а также ее гибкость и наглядность.

К недостаткам фреймовых систем относят их относительно высокую сложность, что проявляется в снижении скорости работы механизма вывода и увеличения трудоемкости внесения изменений в родовую иерархию. Поэтому при разработке фреймовых систем уделяют наглядным способам отображения и эффективным средствам редактирования фреймовых структур.

Специальные языки представления знаний в сетях фреймов FRL (Frame Representation Language), KRL (Knowledge Representation Language), **фреймовая** оболочка Кappa, PILOT/2 и другие программные средства позволяют эффективно строить промышленные системы.

В последние годы термин «фреймовый» часто заменяют термином «объектно-ориентированный». Этот подход является развитием **фреймового представления**.

# Логические модели.

**Предика́т** — это утверждение, высказанное об объекте

Базовое понятие: предметная область и термы.

$P(x)$  – одномерный (одноместный) предикат или переменная высказывания.

Может быть «истина» или «ложь».

$P$ -предикат

$x$  – предметная переменная

**Лингвистическая переменная** – переменная в терминах естественного языка

**Терм** – всякая константа лингвистической переменной

$P(x)$  – одномерный (одноместный) предикат (стол деревянный)

$Q(x,y)$  – бинарный предикат (Маша помогает Саше, Маша выше Саши, температура на улице выше чем в помещении)

$F(x_1, x_2, \dots, x_n)$  –  $n$ -мерный предикат

# Исчисление предикатов

Исчисление предикатов – некоторая аксиоматическая система, предназначенная для моделирования некоторой среды и проверки каких-либо гипотез относительно свойств этой среды при помощи разработанной модели.

Базовые понятия:

1. Предметная область и термы.
2. Переменные высказывания и предикаты

Пример.

Предикат 1-го порядка: ОТЕЦ (Петр) истина (И), если у Петра есть дети, ложь (Л), если у него нет детей.

$P(x)$  – утверждение  $\exists x=5$ ,  $x$  – переменная.

Предикат 2-го порядка ОТЕЦ(Петр, Мария), И, если Петр является отцом Марии, Л, в обратном случае.

$R(x,y)$  – утверждение  $x+y>0$ ,  $x, y$  – переменные.

Предикат 3-го порядка  $Q(x,y,z)$  – утверждение  $x+y>x$ ,  $x, y, z$  – переменные

# Исчисление предикатов

## 3. Элементарные (атомарные) формулы.

Элементарные (атомарные) высказывания — простые высказывания, например, «светает», «дом белый» и т. п.

## 4. Логические операции.

Пример.

$\neg P(x)$  – отрицание утверждения  $3+x=5$ .

## 5. Скобки и кванторы.

Квантор – некоторый способ приписать наличие каких-либо свойств целому множеству объектов:  $\forall$  (квантор общности),  $\exists$  (квантор существования).

«Каждый человек смертен. Конфуций – человек. Следовательно, Конфуций смертен» в формулу.

Обозначим « $x$  есть человек» через  $ЧЕЛОВЕК(x)$  и « $x$  смертен» через  $СМЕРТЕН(x)$ .

«каждый человек смертен» -  $(\forall x) (ЧЕЛОВЕК(x) \rightarrow СМЕРТЕН(x))$ ,

«Конфуций – человек» –  $ЧЕЛОВЕК(Конфуций)$

«Конфуций смертен» –  $СМЕРТЕН(Конфуций)$ .

Утверждение в целом

$(\forall x) (ЧЕЛОВЕК(x) \rightarrow СМЕРТЕН(x)) \wedge ЧЕЛОВЕК(Конфуций) \rightarrow СМЕРТЕН$

28 (Конфуций).

# Элементы логики высказываний.

Обозначение	Название
$\neg$	отрицание, логическое НЕ
$\wedge$	конъюнкция, логическое И, логическое умножение
$\vee$	дизъюнкция, логическое ИЛИ, логическое сложение
$\supset$ или $\rightarrow$	импликация, ЕСЛИ ... ТО...
$\equiv$ или $\leftrightarrow$	эквивалентность
$\neg \equiv$	неравнозначность

Таблицы истинности основных логических операций

X	Y	$\neg X$	$Xg\wedge Y$	$Xg\vee Y$	$X\rightarrow Y$	$X\leftrightarrow Y$
Л	Л	И	Л	Л	И	И
Л	И	И	Л	И	И	Л
И	Л	Л	Л	И	Л	Л
И	И	Л	И	И	И	И

$$\neg P(x) \wedge [Q(g(y)) \vee R(x, y)]$$

# Логический вывод. Метод резолюций

В основе этого метода лежит идея доказательства от противного.

**Определение.** Резольвента.

$C1: P$

$C2: \neg P \vee Q.$

Тогда,  $C3: Q$  есть *резольвента* дизъюнктов  $C1$  и  $C2$ , полученная как дизъюнкция  $C1 \vee C2$ , из которой вычеркнуты контрарные пары ( $P$  и  $\neg P$ ).

$$\begin{array}{l} 1) P \vee R \\ \neg P \vee Q \\ \hline \end{array}$$

$$R \vee Q$$

$$\begin{array}{l} 2) \neg P \vee Q \vee R \\ \neg Q \vee S \\ \hline \end{array}$$

$$\neg P \vee R \vee S$$

$$\begin{array}{l} 3) \neg P \vee Q \\ \neg P \vee R \\ \hline \end{array}$$

нет резольвенты

# Формальная (Аристотелева) логика



Отношение «отец» не транзитивно



# Продукционная модель

- Основана на правилах продукции, позволяет представлять знания в виде предложений типа

**ЕСЛИ** (*условие*), **ТО** (*действие*)

Общий вид продукции

$$i: Q; P; A \rightarrow B; N$$

- $i$  – имя (порядковый номер) продукции
- $Q$  – сфера применения продукции
- $P$  – условие применимости ядра продукции
- $A \rightarrow B$  – ядро продукции
- $N$  – постусловие продукции



# Ядро продукции

Возможности А и В могут рассматриваться как сложное высказывание:

$$A_1 \wedge A_2 \wedge \dots \wedge A_n \quad \text{ТО} \quad B_1 \wedge B_2 \wedge \dots \wedge B_n$$



# Стратегии управления выполнением

## Принцип "стопки книг"

- наиболее часто используемая продукция является наиболее полезной
- порядок определяется накопленной частотой исполнения продукций в прошлом

## Принцип наиболее длинного условия

- выбирается та, у которой стало истинным наиболее длинное условие выполнимости ядра

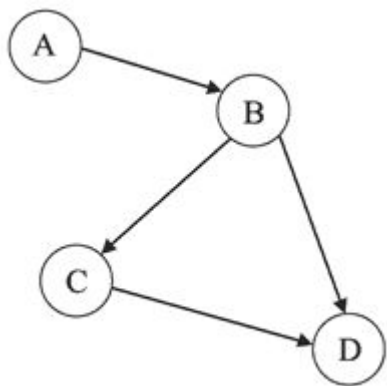
## Принцип приоритетного выбора

- введение статических или динамических приоритетов на продукции

## Принцип "классной доски"

- выделяется специальная рабочая область памяти
- выделяются специальные поля для формирования условий применимости ядер продукций
- специальные поля для записи результатов

# БЗ



## Раздел «Факты»:

- дорога (A, B);
- дорога (B, C);
- дорога (B, D);
- дорога (C, D).

## Раздел «Правила»:

1. прямая дорога (X, Y), если дорога (X, Y) или дорога (Y, X);
2. транзитная дорога (X, Y), если дорога {X, Z} и дорога (Z, Y);
3. есть дорога (X, Y), если прямая дорога (X, Y) или транзитная дорога (X, Y).

Обратный вывод «Установить, есть ли дорога между пунктами A и D».

«Есть дорога (A,D)»	«Прямая дорога (A,D)»	«Дорога»	«Транзитная дорога (A,D)»
	«Есть дорога (A,D)»	«Прямая дорога (A,D)»	«Есть дорога (A,D)»
		«Есть дорога (A,D)»	
Шаг 1	Шаг 3	Шаг 4	Шаг 6

# МПЗ

## **Достоинства:**

Самая распространенная МПЗ.

Наглядность

Высокая модульность

Легкость внесения дополнений и изменений

Простотой механизма логического вывода.

Имеется большое число программных средств, позволяющих реализовать продукционный подход

## **Недостатки:**

При большом числе продукций становится сложной проверка непротиворечивости системы продукций.

Системе присуща недетерминированность (неоднозначность выбора выполняемой продукции из фронта активизируемых продукций).

Считается, что если в ИС число продукций достигнет тысячи (по другим источникам, полутора тысяч), то мало шансов, что система продукций во всех случаях будет правильно функционировать

# Применение МПЗ

## Оболочки ЭС – «Пустые» проблемно-независимые ЭС с незаполненной базой знаний ЭС

- Несмотря на обилие оболочек ЭС — CxPERT, Exsys, GoldWorks, Guru, KDS3, KnowledgePro, Nexpert, Rule Master, VP Expert — можно выделить ограниченное число методов построения механизмов вывода ЭС: **применение правил продукций, использование механизма исчисления предикатов, символьная обработка знаний, представленных в текстовой форме, использование вероятностного подхода к обработке знаний.**
- Известно множество работ с описанием и сравнением различных МПЗ и их использование для решения разного рода задач. В системах поддержки и принятия решений наибольшее распространение получили продукционные правила и их системы.
- Это связано с тем, что при управлении технологическими объектами нефтепереработки и нефтехимии управление ведется по правилам-инструкциям. В технологических регламентах описываются различные нарушения и действия по их устранению. То есть имеет место, так называемый, ситуационный подход в выработке решений.

# Дескриптивная логика

Семейство языков представления знаний, позволяющих описывать понятия предметной области в недвусмысленном, формализованном виде.

Изначально ДЛ зародились как расширение фреймовых структур и семантических сетей механизмами формальной логики.

ДЛ оперируют понятиями концепт («одноместный предикат» или множество, класс) и роль («двуместный предикат» (или бинарное отношение))

Концепты используются для описания классов некоторых объектов, например, «Люди», «Женщины», «Машины».

Роли используются для описания двуместных отношений между объектами, например, на множестве людей имеется двуместное отношение «X есть\\_родитель\\_для Y»

В ДЛ набор утверждений общего вида или терминологии называется **ТВох** (terminological box - понятия, отношения), набор утверждений частного вида — **АВох** (assertional box - набор утверждений об индивидах), а вместе они составляют так называемую **базу знаний** или **онтологию**.

**Онтология** — это множество сущностей (понятий), характерных определенной предметной области, и отношений между ними

## предикатов

ДЛ, можно рассматривать как фрагменты логики предикатов при «естественном» переводе концептов в предикатные формулы.

### Логика высказываний

Это, по сути, декларативное предложение, имеющее значение истинности

$$(a + b)^2 = a^2 + 2ab + b^2$$

Если  $x$  реально, то  $x^2 \geq 0$

Если  $x$  реально, то  $x^2 < 0$

Солнце восходит на востоке.

Солнце восходит на западе.

### Логика предикатов

Это свойства, дополнительная информация, позволяющая лучше выразить предмет предложения

Предикат  $P(x) : x > 5$ ,  $x$  - подлежащее или переменная, а ' $>5$ ' - предикат.

$P(7) : 7 > 5$  - это предложение (высказывание), в котором мы присваиваем значения переменной  $x$ , и она имеет значение истинности, то есть True

Набор значений, которые могут принимать переменные предиката, называется Универсом, или Областью дискурса, или Доменом предиката.

# Дескриптивная логика. Связь с логикой

## предикатов

<https://www.geeksforgeeks.org/difference-between-propositional-logic-and-predicate-logic/>

	Дескриптивная логика	Логика предикатов
1	это логика, которая имеет дело с набором декларативных утверждений, которые имеют значение истинности, true или false.	это выражение, состоящее из переменных с заданной областью действия. Оно состоит из объектов, отношений и функций между объектами.
2	Это базовая и наиболее широко используемая логика. Также известная как булева логика.	Это расширение дескриптивной логики, охватывающее предикаты и кванторы.
3	Предложение имеет определенное значение истинности, либо true, либо false.	Истинностное значение предиката зависит от значения переменных.
4	Анализ области видимости в логике высказываний не выполняется.	Логика предикатов помогает анализировать область действия субъекта над предикатом. Существует три квантора: Универсальный квантор ( $\forall$ ) изображает для всех, Экзистенциальный квантор ( $\exists$ ), изображающий, что некоторые существуют, и квантор уникальности ( $\exists!$ ), изображающий ровно один.
5	Предложения объединяются с логическими операторами или логическими связками, такими как Отрицание ( $\neg$ ), Дизъюнкция ( $\vee$ ), Конъюнкция ( $\wedge$ ), Исключающее ИЛИ ( $\oplus$ ), Импликация ( $\Rightarrow$ ), Дву-условная или Двойная импликация ( $\Leftrightarrow$ ).	Логика предикатов дополняет существующее предложение, вводя кванторы.
6	Это более обобщенное представление.	Это более специализированное представление.
7	Она не может иметь дело с наборами сущностей.	Она может иметь дело с множеством сущностей с помощью кванторов.

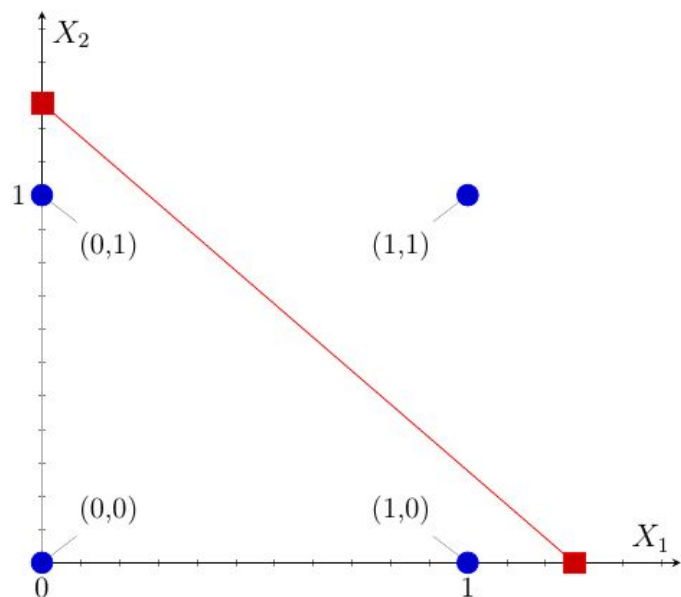


# Неклассические МПЗ

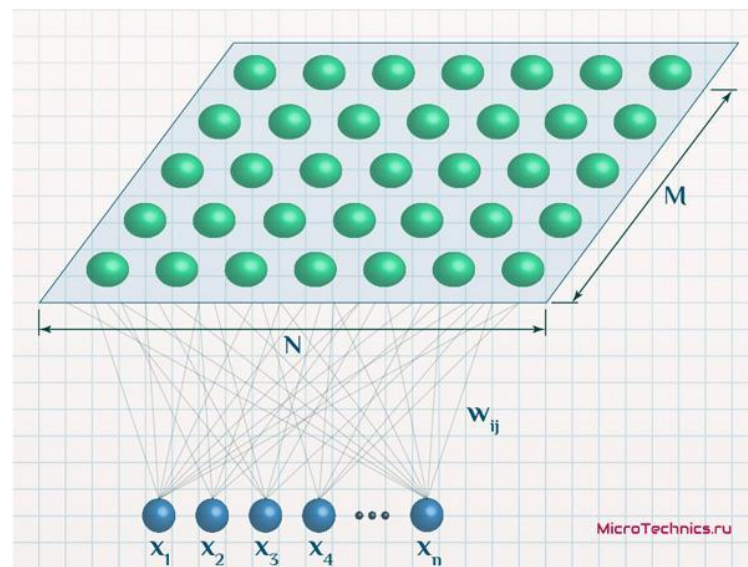
Искусственные нейронные сети, особенно для задач классификации:

- Многослойные персептроны в задачах классификации
- Сети Кохонена

## Логическая И



## Сети Кохонена



# Неклассические МПЗ

- Модели нечеткой логики

Правила работы нагревателей:

1. Если  $T \geq 20^{\circ}\text{C}$  то Н1 и Н2 выключаются,  $H1 = H2 = \langle\langle 0 \rangle\rangle$ .
2. Если  $10^{\circ}\text{C} < T < 20^{\circ}\text{C}$  то включается Н1, Н2 – выключен,  $H1 = \langle\langle 1 \rangle\rangle$ ;  $H2 = \langle\langle 0 \rangle\rangle$ .
3. Если  $T \leq 10^{\circ}\text{C}$  то включаются Н1, Н2,  $H1 = H2 = \langle\langle 1 \rangle\rangle$ .

$$X1 = \begin{cases} 1, & \text{если } T \leq 10^{\circ} \\ 0, & \text{иначе} \end{cases}$$

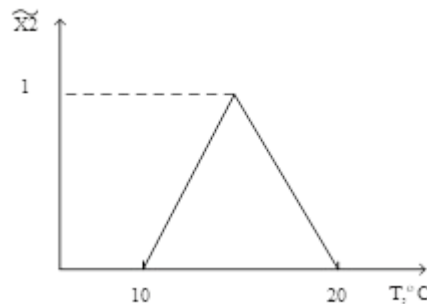
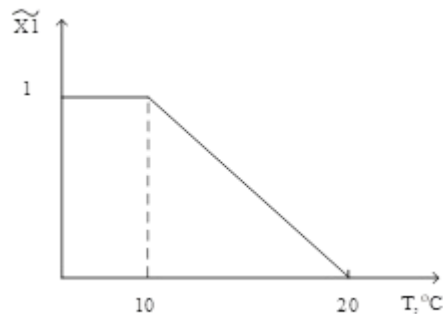
$$X2 = \begin{cases} 1, & \text{если } 10^{\circ}\text{C} < T < 20^{\circ} \\ 0, & \text{иначе} \end{cases}$$

$$R1 = X2 \vee R1 \wedge \overline{R2} \wedge X1.$$

$$R2 = X1 \vee R2 \wedge \overline{R1} \wedge X1$$

$$H1 = R1 \wedge \overline{R2} \vee R2 \wedge \overline{R1}$$

$$H2 = \overline{R1} \wedge R2$$



$$\widetilde{R1} = \widetilde{X2} \widetilde{\vee} \widetilde{R1} \widetilde{\wedge} \widetilde{\overline{R2}} \widetilde{\wedge} \widetilde{X1}$$

$$\widetilde{R2} = \widetilde{X1} \widetilde{\vee} \widetilde{R2} \widetilde{\wedge} \widetilde{\overline{R1}} \widetilde{\wedge} \widetilde{X1}$$

$$\widetilde{H1} = \widetilde{R1} \widetilde{\wedge} \widetilde{\overline{R2}} \widetilde{\vee} \widetilde{R2} \widetilde{\wedge} \widetilde{\overline{R1}}$$

$$\widetilde{H2} = \widetilde{\overline{R1}} \widetilde{\wedge} \widetilde{R2}$$

# Этапы разработки ЭС

Этапы разработки ЭС

Наименование этапа	Решаемые задачи	Сроки
1. Анализ требований	Выбор проблемы; определение режимов работы будущей ЭС; определение требований к адекватности и точности результатов работы будущей ЭС	2–4 недели
2. Анализ предметной области	Выделение сущностей в предметной области, непосредственно относящихся к решаемой задаче; предварительное выделение фактов и правил, непосредственно относящихся к решаемой задаче	До полугода
3. Проектирование: 3.1. разработка прототипа; 3.2. оценка	Предварительный выбор МПЗ; предварительная разработка логической структуры; предварительная разработка программной структуры; предварительное кодирование и испытания на возможно большем материале (в короткие сроки); возврат к проектированию или анализу требований	4–8 недель
4. Программирование и отладка: 4.1. кодирование; 4.2. доработка до промышленной; 4.3. стыковка ЭС	Выбор языка программирования; кодирование, создание интерфейса, службы помощи и документации; интеграция ЭС в технологический процесс предприятия; предварительное обучение путем заполнения баз знаний результатами работы других ЭС	4–8 недель
5. Обучение	Работа ЭС в режиме решения реальных задач под присмотром инженера по знаниям (редактирует БЗ)	До полугода
6. Сверка с эталоном	Проведение показательных экспериментов и определение степени достижения целей проекта по созданию ЭС	1 неделя
7. Поддержка ЭС	Поиск логических и программных ошибок, выпуск новых версий, адаптация к изменившимся условиям эксплуатации	До 5 лет

# Оболочки ЭС

ЭС для решения прикладных задач предметной области

При создании первых ЭС было отмечено, что механизм логического вывода и язык представления знаний могут быть отделены от конкретных знаний и использованы в различных проблемных областях.

"Пустые" проблемно-независимые ЭС с незаполненной базой знаний называются оболочками ЭС.

# Механизм вывода

Механизм вывода. Поиск решений.

Оболочки ЭС — CxPERT, Exsys, GoldWorks, Guru, KDS3, KnowledgePro, Nexpert, Rule Master, VP Expert, MiniExpertSystem

Вывод заключения может быть произведен экспертной системой двумя способами:

- Прямой ход – определяются все входные данные, и на их основании выводится заключение.
- Обратный ход – выбирается гипотеза и проверяются на истинность ее признаки. Если они истинны, то верна и гипотеза, если нет — проверяется следующая гипотеза. Метод резолюций

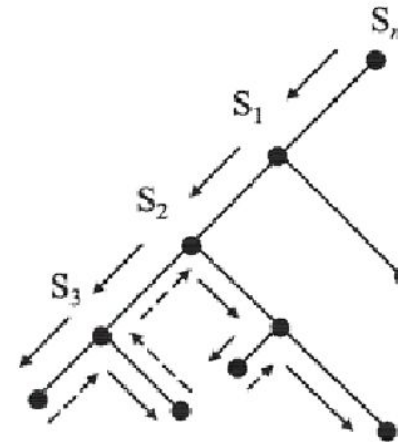
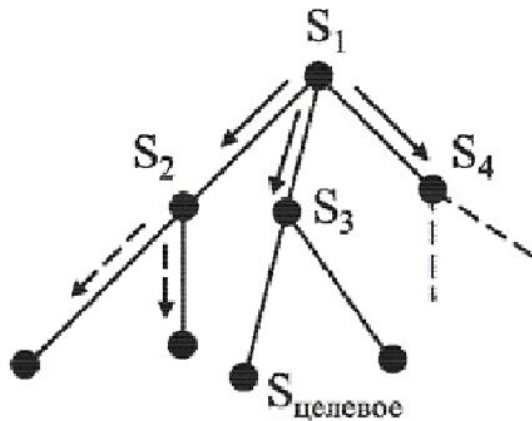
# Механизм вывода

Методы поиска решений:

- полного перебора
- эвристические

Метод полного перебора в ширину.

Метод полного перебора в глубину Поиск с возвратом (Backtracking).



# Механизм вывода. Эвристический поиск

## *Алгоритмы эвристического поиска*

Эвристика (Э) - набор правил для выбора тех ветвей в пространстве состояний, которые с наибольшей вероятностью приведут к приемлемому решению проблемы.

В области ИИ эвристики используются в двух ситуациях.

1. Проблема может не иметь точного решения из-за неопределенности в постановке задачи и/или в исходных данных (пример – медицинская диагностика).

2. Проблема может иметь точное решение, но стоимость его поиска может быть очень высокой. (эвристика позволяет избежать этой проблемы и вести поиск по наиболее «перспективному» пути, исключая из рассмотрения не перспективные состояния и их потомки)

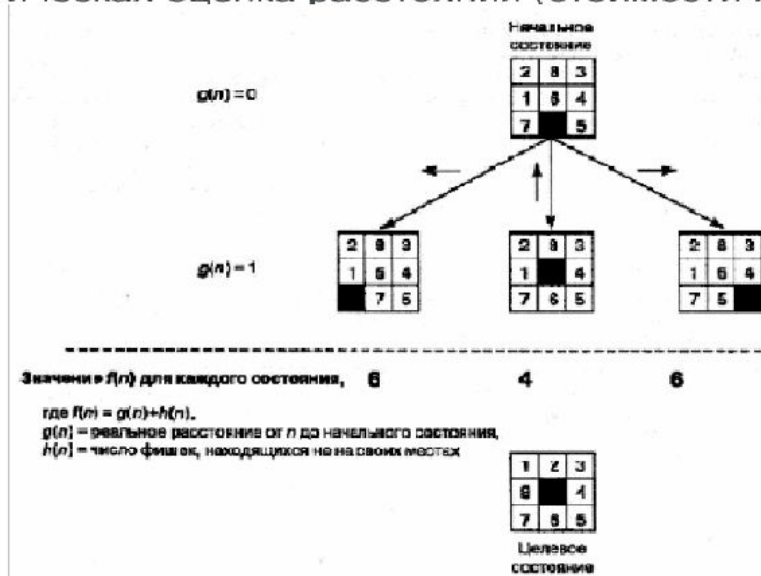
Эвристика - это только предположение следующего шага, который будет сделан на пути решения проблемы.

# Механизм вывода. Эвристический поиск

Алгоритмы оценочных (штрафных) функций

$$f=g(n)+h(n)$$

где  $g(n)$  - фактическая длина пути от произвольного состояния  $n$  к начальному;  
 $h(n)$  - эвристическая оценка расстояния (стоимости пути) от состояния  $n$  к цели.





# Механизм вывода. Эвристический поиск

Поведение эвристики оценивается рядом критериев, и в первую очередь ниже приведенными тремя:

- допустимость;
- монотонность;
- информированность.

## *Мера допустимости*

Алгоритм поиска является допустимым, или приемлемым, если гарантируется нахождение кратчайшего пути к решению.

*Монотонность* - это свойство алгоритма последовательно находить кратчайший путь к каждому состоянию, которое встречается в процессе поиска.

## *Информированные эвристики*

Если из двух эвристик  $h_1$  и  $h_2$  для всех состояний  $n$  в пространстве поиска выполняется неравенство  $h_1(n) \leq h_2(n)$ , то эвристика  $h_2$  является более информативной.

Спасибо за внимание!