

# ЛЕКЦИЯ 12

Семантические сети.

# Семантическая сеть

Семантическая сеть - структура для представления знаний в виде узлов, соединенных дугами. Самые первые семантические сети были разработаны в качестве языка-посредника для систем машинного перевода, а многие современные версии до сих пор сходны по своим характеристикам с естественным языком. Однако последние версии семантических сетей стали более мощными и гибкими и составляют конкуренцию фреймовым системам, логическому программированию и другим языкам представления знаний.

В системах искусственного интеллекта семантические сети используются для ответа на различные вопросы, изучение процессов обучения, запоминание и рассуждение.

Впервые семантические сети были использованы в системах машинного перевода в конце 50-х - начале 60-х годов. Первая такая система, которую создала Мастерман, включала в себя 100 примитивных концептов таких, как, например, НАРОД, ВЕЩЬ, ДЕЛАТЬ, БЫТЬ. С помощью этих концептов она описала словарь объемом 15000 единиц, в котором также имелся механизм переноса характеристик с типа высшего уровня на подтип. Некоторые системы машинного перевода базировались на корреляционных сетях Цеккато, которые представляли собой набор из 56 различных отношений, некоторые из которых - падежные отношения, отношения подтипа, члена, части и целого. Он использовал сети, состоящие из концептов и отношений для синтаксического анализа и разрешения неоднозначностей.

## Виды семантических сетей

В самом общем случае семантическая сеть представляет собой информационную модель предметной области (совокупность утверждений и фактов из Базы Знаний) и имеет вид графа, вершины которого соответствуют понятиям, а дуги представляют отношения между ними. В семантической сети выражено прежде всего интенциональное знание.

Важной чертой семантических сетей является возможность представлять знания более естественным и структурированным образом, чем в других формализмах (например, в продукционных системах).

Возможные способы организации семантических графов:

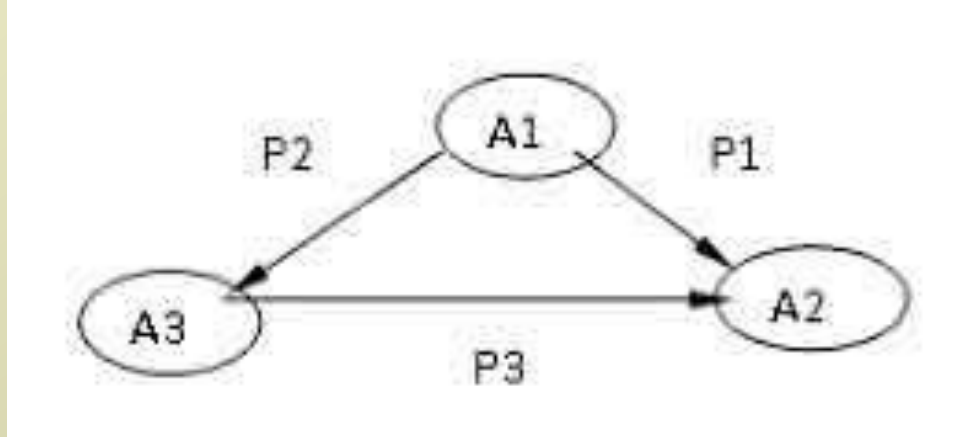
- реляционные графы;
- графы с центром в глаголе;
- пропозициональные графы.

# Реляционные графы

Самые простые сети, которые используются в системах искусственного интеллекта, - реляционные графы. Они состоят из узлов, соединенных дугами. Каждый узел представляет собой понятие, а каждая дуга - отношения между различными понятиями.

*Семантическая сеть* описывает знания в виде сетевых структур.

Например, «программист сел за компьютер и отладил программу». Объектами являются: программист (A1), компьютер (A2), программа (A3). Объекты связаны отношениями: сел за компьютер (p1), отладил (p2), загружена в компьютер программа (p3). На рисунке приведен пример этой простейшей семантической сети.



При работе с реляционными графами возникают проблемы с передачей всего многообразия временных отношений и отношений модальности. Несмотря на то, что многие ученые используют эти графы для решения сложных проблем, они так до сих пор и не разработали общего метода для их разрешения.

# Концептуальные графы Шенка

Проблему многообразия предметов и глаголов решал Шенк

Терминология, которая используется в области семантических сетей, существенно различна.

Шенк ввёл несколько основных примитивов, которые уточнялись дополнительно.

На изображенном графе концептуальных зависимостей Шенка.  $\langle \Rightarrow \rangle$  означает агента. **INGEST** (поглощать) - один из примитивов Шенка. С его помощью представлены глаголы

**ЕСТЬ** - ПОГЛОЩАТЬ твердый объект;

**ПИТЬ** - ПОГЛОЩАТЬ жидкий объект;

**ДЫШАТЬ** - ПОГЛОЩАТЬ газообразный объект.



# Графы с центром в глаголе

Глаголы обычно являются важным связующим звеном для различных предложений. Если нужно изменить модальность, или время, меняется прежде всего глагол.

Глаголы соединяются с группой существительного с использованием падежных отношений. Например, с предложением "Mary gave a book to John", Mary агент который совершает действие, book объект этого процесса, а John реципиент (получатель действия) относительно глагола "давать".

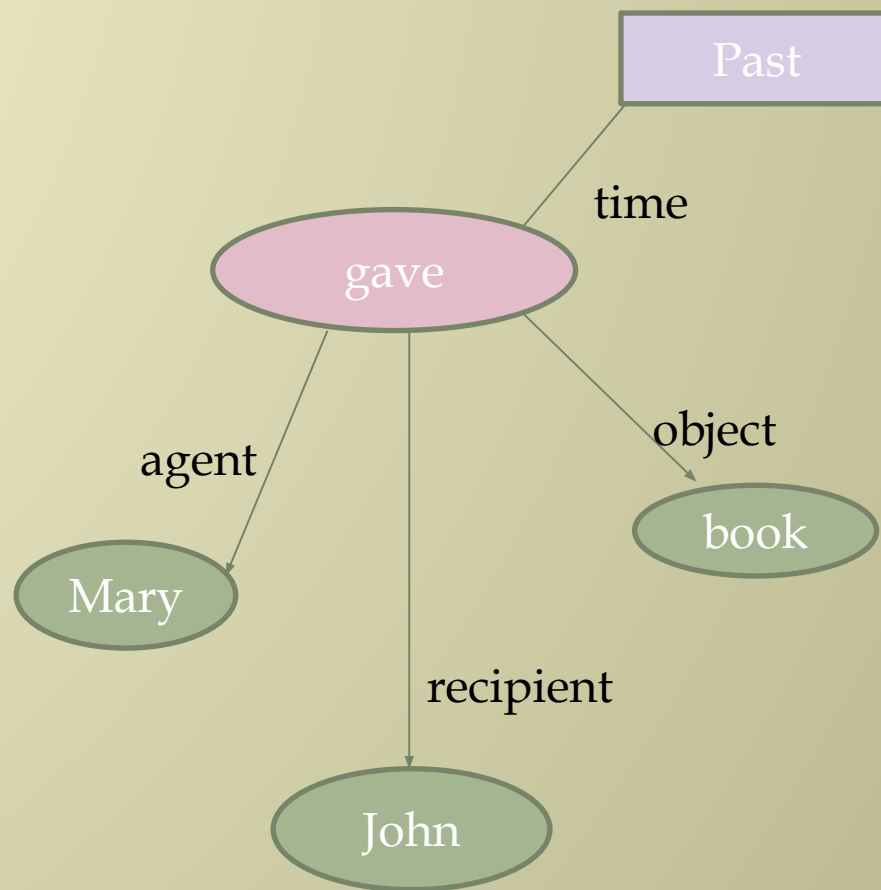
Помимо падежных отношений в предложении в естественном языке также имеются средства для связи отдельных предложений.

Прежде всего, это **союзы**.

Самый простой способ соединить предложения - это поставить между ними союз. Некоторые союзы, как например "и", "или", "если" обозначают логическую связь; некоторые, такие как "после того, как", "когда", "пока", "с тех пор, как" и "потому что", выражают временные отношения и причинно-следственные связи.

## Пример графа с центром в глаголе

В предложении "Mary gave a book to John", Mary - агент который совершает действие, book объект этого процесса, а John реципиент (получатель действия) относительно глагола "давать".





Графы с центром в глаголе - это реляционные графы, где глагол считается центральным звеном любого предложения. Маркеры времени и отношения пишутся прямо рядом с понятиями, которые представляют глаголы. Графы концептуальных зависимостей Роджера Шенка также используют этот подход.

Несмотря на то, что графы с центром в глаголе довольно гибкие по своей структуре, они обладают рядом ограничений. Одно из них заключается в том, что они не проводят разграничение между определителями, которые относятся только к глаголу, и определителями, относящимися к предложению целиком. Эти графы также плохо справляются с предложениями, находящимися внутри других предложений.

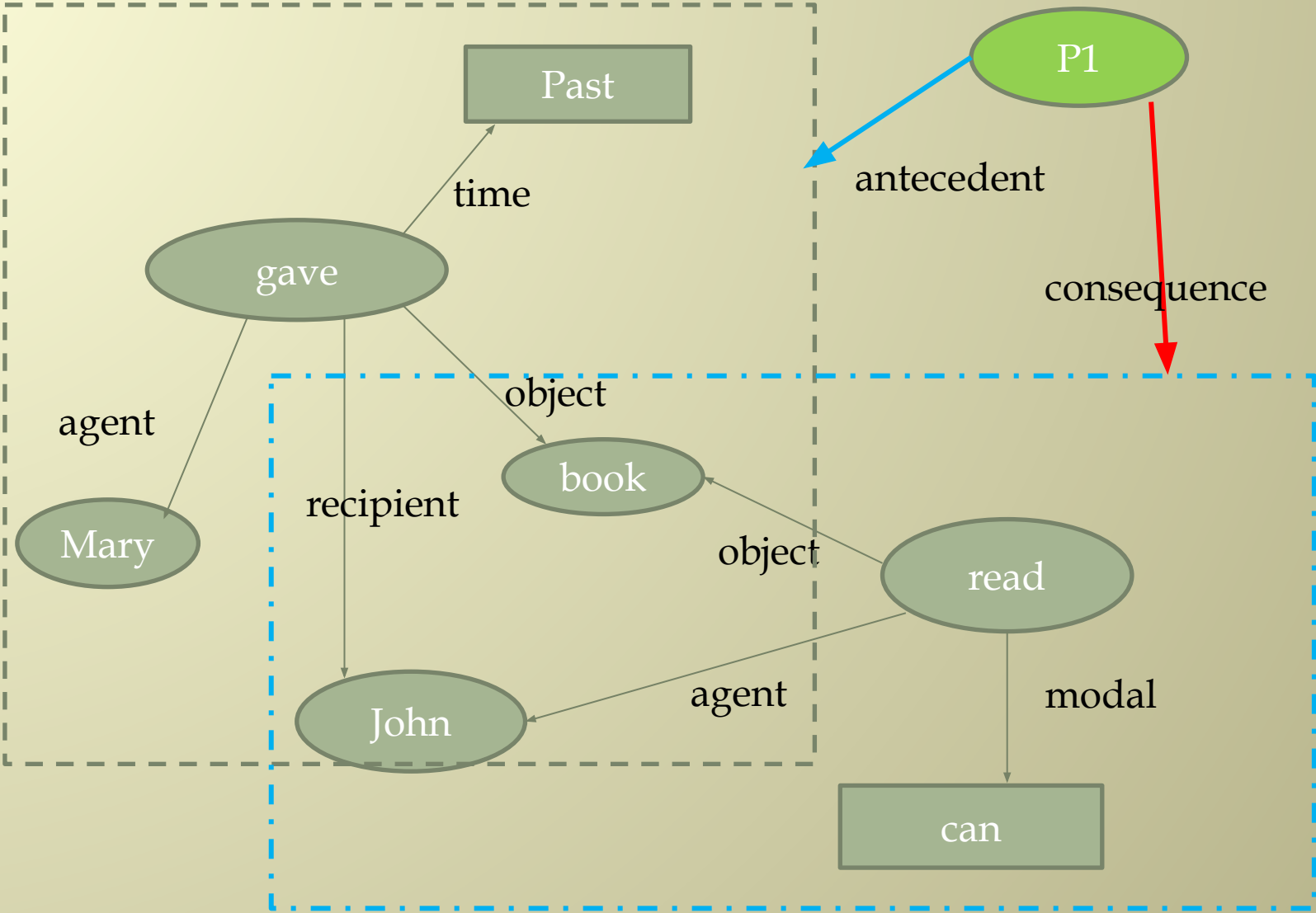
# Пропозиционные сети

- В пропозиционных сетях узлы представляют целые предложения. Эти узлы являются точками соприкосновения для отношений между отдельными предложениями связанного текста. С другой стороны они определяют время и модальность для всего контекста. Представленный ниже пример иллюстрирует отношения, для записи которых необходимы пропозиционные узлы:

- If Mary gave the book to John, then John can read this book.

- В первом предложении для глагола gave используется отдельная вершина. Вторая часть предложения является следствием первой и между ними устанавливается причинно-следственная связь.

Чтобы изобразить такое предложение, необходимо использовать пропозиционные узлы, которые содержат гнездящиеся графы.



Существуют элементы сходства, присущие практически всем семантическим сетям:

- узлы семантических сетей представляют собой понятия - образы предметов, событий, состояний;
- различные узлы одного понятия (концепта) связаны с различными значениями, если не помечено, что они относятся к одному понятию;
- дуги семантических сетей представляют отношения между узлами - понятиями (пометки над дугами указывают на тип отношения);
- отношения между понятиями разнообразны и представляют собой лингвистические падежи (такие как агент, объект, реципиент и инструмент), а также временные, пространственные, логические отношения и отношения между отдельными предложениями;
- понятия организованы по уровням иерархии в соответствии со степенью обобщенности.

# Отношения на сетях

Отношения на семантических сетях делятся на

- лингвистические,
- логические,
- теоретико-множественные,
- квантификационные.

Лингвистические отношения: глагольные (время, вид, род, число, наклонение) и атрибутивные (цвет, размер, форма, материал).

Логические: дизъюнкция, конъюнкция, отрицание, импликация.

Теоретико-множественные: класс – подкласс, класс – элемент класса, элемент – часть элемента.

Квантификационные: логические кванторы (всеобщности, существования); нелогические кванторы (много, несколько), числовые характеристики.

## Иерархические связи

В 70-е годы различными авторами были разработаны самые разные концепции семантических сетей. Такие учёные, как Хендрикс, Шенк, Хайенс, Шапиро, разрабатывали собственные сетевые модели.

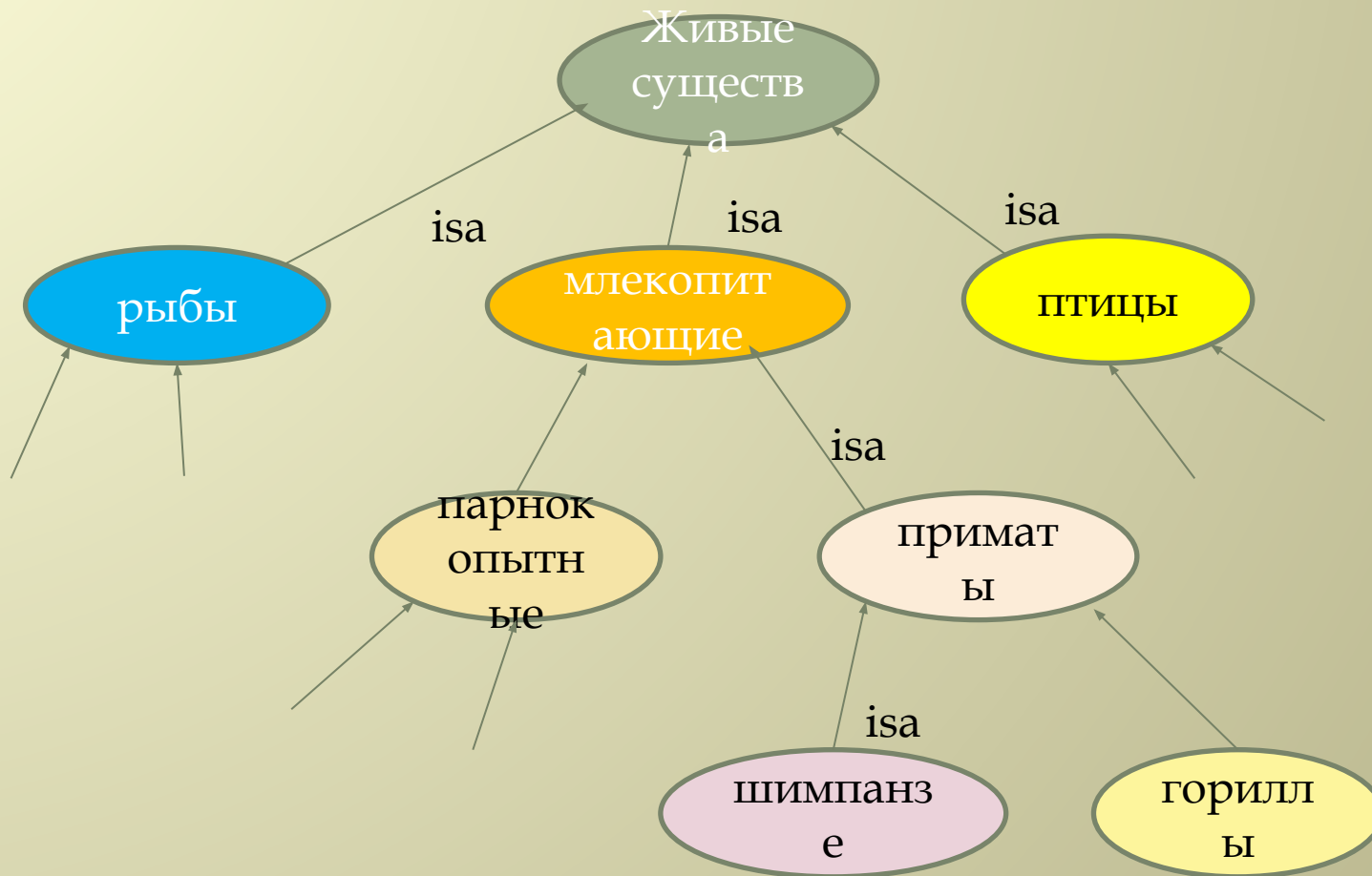
Наиболее важным в организации семантических сетей является введение отношений «множество – подмножество», «множество – элемент множества» и «часть – целое».

Важным техническим приёмом, используемым в семантических сетях, является иерархия, или введение системы классификации: Все объекты, реальные или абстрактные, относятся к некоторому числу категорий или классов.

Например, множество людей можно разбить на множества мужчин, женщин, детей; дети классифицируются на мальчиков и девочек, и т.п.

Классификации задаются на семантических сетях с помощью отношения *isa*. **A is a B** выполняется для объектов двух типов A и B тогда и только тогда, когда A есть подмножество B.

# Пример иерархии



# Виды иерархий

- Иерархия типов и подтипов является стандартной характеристикой семантических сетей. Иерархия может включать живые объекты:  
ТАКСАisaСОБАКАisaПЛОТояДНОЕisaЖИВОТНОЕisaЖИВОЕ  
СУЩЕСТВОisaФИЗИЧЕСКИЙ ОБЪЕКТisaСУЩНОСТЬ.  
Они также могут включать в себя события:
- ЖЕРТВОВАТЬisaДАВАТЬisaДЕЙСТВИЕisaСОБЫТИЕ  
или состояния:  
ЭКСТАЗisaСЧАСТЬЕisaЭМОЦИОНАЛЬНОЕСОСТОЯНИЕisaСОСТОЯНИЕ.
- Термин "иерархия" обычно обозначает частичное упорядочение, где одни типы являются более общими, чем другие. Упорядочение является частичным, потому, что многие типы просто не подлежат сравнению между собой.

Помимо иерархии типов используется иерархия «часть- целое»:

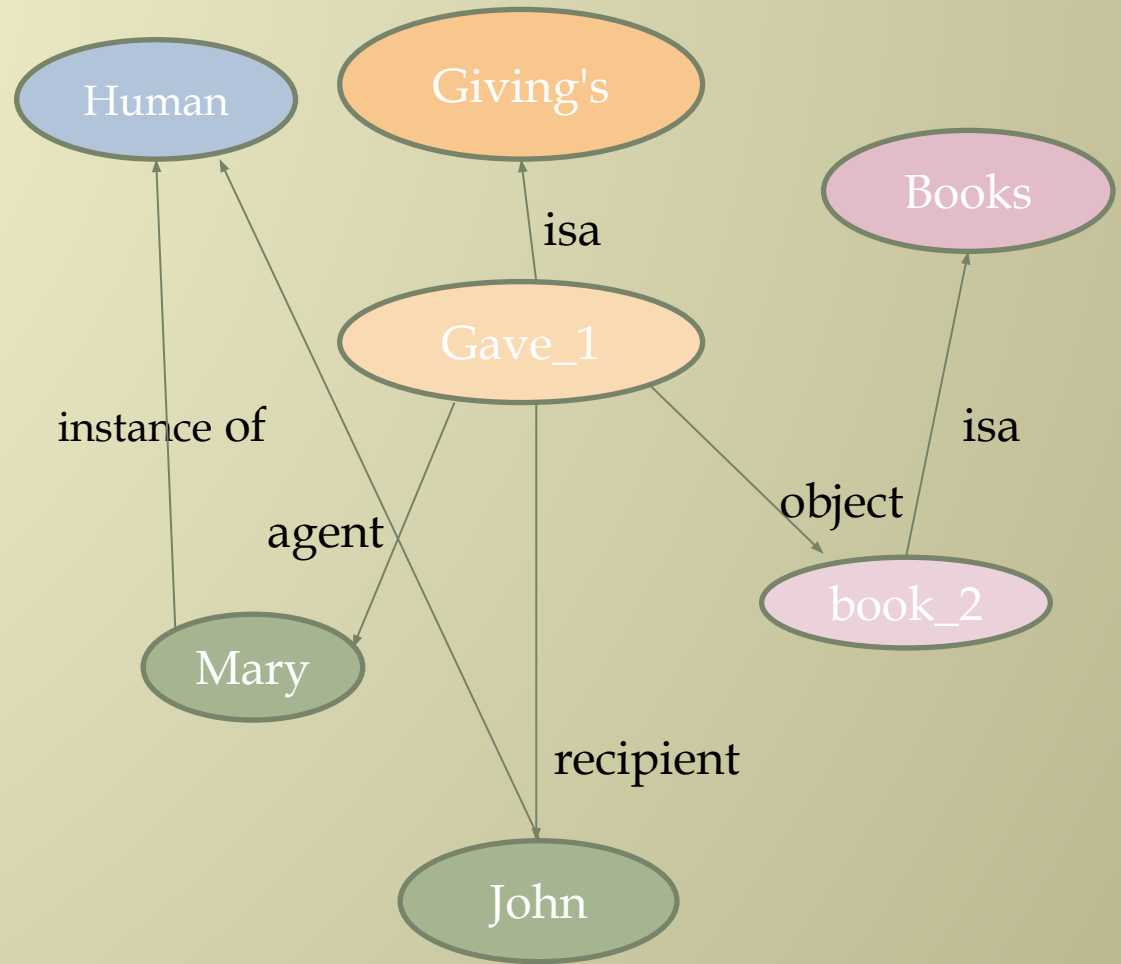
ПАЛЕЦ part of РУКА part of ТУЛОВИЩЕ part of ЧЕЛОВЕК

Отметим, что отношение «часть-целое» транзитивно.



## Введение иерархических связей для графа с центром в глаголе

В нашем предложении Mary, как и Джон, является конкретным примером объекта, принадлежащего классу «Человек». Ситуация передачи книги Джону относится к классу ситуаций дарения, которые, в свою очередь, могут относиться к более общему классу различных действий.



# Организация иерархий

- Ациклический граф. Любое частичное упорядочение может быть изображено, как граф без циклов. Такой граф имеет ветви, которые расходятся и сходятся вместе опять, что позволяет некоторым узлам иметь несколько узлов-родителей. Иногда такой тип графа называют путанным.
- Деревья. Самым распространенным видом иерархии является граф с одной вершиной. В такого рода графах налагаются ограничения на ациклические графы: вершина графа представляет собой один общий тип, и каждый другой тип  $X$  имеет лишь одного родителя  $Y$ .
- Решетка. В отличие от деревьев узлы в решетке могут иметь несколько узлов родителей. Однако здесь налагаются другие ограничения: любая пара типов  $X$  и  $Y$  как минимум должна иметь общий надтип  $X$  и  $Y$  и подтип  $X$  или  $Y$ . Вследствие этого ограничения решетка выглядит, как дерево, имеющее по главной вершине с каждого конца. Вместо всего одной вершины решетка имеет одну вершину, которая является надтипом всех категорий, и другую вершину, которая является подтипом всех типов.

# Наследование свойств

- **Наследование свойств.** Основным свойством иерархии является то, что свойства «высших» типов автоматически наследуются низшими типами, или переносятся на них. Так, все характеристики, которые присущи **ЖИВОТНОМУ**, также присущи **МЛЕКОПИТАЮЩЕМУ**, **РЫБАМ** и **ПТИЦАМ**.

Наследование свойств, очевидно, позволяет избежать дублирования информации на сети.

Наследование свойств возможно также для отношений «часть-целое». Так, если **автомобиль** имеет *синий* цвет, **дверца** автомобиля унаследует это свойство. Однако для такой части автомобиля, как **мотор**, это неверно. Необходимо явно указывать наследуемые признаки при построении иерархии.

В целом введение отношений «часть целое» и «множество – подмножество» оказалось весьма эффективным для решения многих проблем из области принятия решений.

# Языки разработки СС

- ▣ Семантические сети могут быть записаны практически на любом языке программирования на любой машине. Самые популярные в этом отношении языки LISP и PROLOG. Однако многие версии были созданы и на FORTRANe, PASCALe, C и других языках программирования. Для хранения всех узлов и дуг необходима большая память, хотя первые системы были выполнены в 60-х годах на машинах, которые были гораздо меньше и медленнее современных компьютеров.
- ▣ Один из самых распространенных языков, разработанных для записи естественного языка в виде сетей, - это PLNLP (Programming Language for Natural Language Processing) Язык Программирования для Обработки Естественного Языка, созданный Хайдерном. Этот язык используется для работы с большими грамматиками с обширным покрытием.

# Наиболее важные принципы, положенные в основу работы семантических сетей

1. Принцип группирования информации (блоки, соответствующие отдельным утверждениям).
2. Специализация – описание индивидуальных понятий через общие.
3. Наследование свойств.
4. Иерархия понятий.
5. Использование механизмов дедукции и обобщения.
6. Использование семантических падежей и развитой системы отношений.

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети, отражающей поставленный запрос к базе. Преимущества заключаются в простоте и наглядности описания предметной области. Однако последнее свойство с усложнением семантической сети теряется и, кроме того, существенно увеличивается время вывода. Также к недостаткам семантических сетей относят сложность обработки различного рода исключений.

# Дедуктивный вывод на семантических сетях

В семантических сетях связи между понятиями делятся на структурные, логические и процедурные.

Наиболее распространёнными являются структурные связи, меньше применяются логические и процедурные.

Поскольку дедуктивные возможности системы определяются прежде всего логическими связями, рассмотрим организацию логических связей на примере семантической сети Deliyanny A., Kowalsky R.

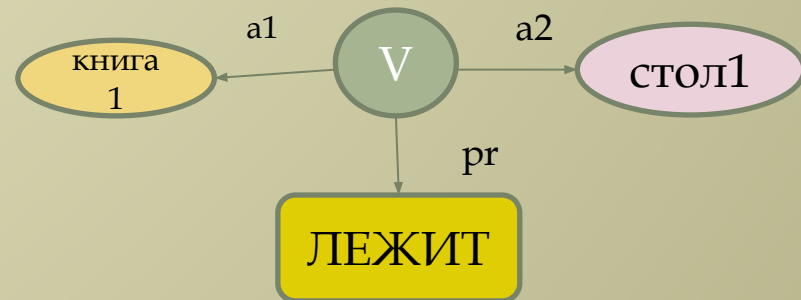
Рассмотрим высказывание  $V = \text{«Книга1 лежит на столе 1»}$

В исчислении предикатов 1 порядка утверждение примет вид:

$V = \text{ЛЕЖИТ(книга1, стол1)}$

Представим его на сети:

$V$  – вершина, соответствующая всему высказыванию,  $a_1$ ,  $a_2$  – аргументы предиката ЛЕЖИТ.



## Причинно-следственные связи

Если выражение отражает причинно-следственные связи, на семантической сети потребуется введение нового типа отношений: *условие* и *заключение*.

Рассмотрим утверждения:

«Куб2 находится в том же месте, что и Куб1»

«Куб1 находится в месте А»

«Куб1 не находится в месте С»

Введём предикат  $V(x, y)$ : « $x$  находится в месте  $y$ »

Тогда утверждение1 запишем  $V(\text{Куб1}, y) \rightarrow V(\text{Куб2}, y)$

Утверждение 2:  $I \rightarrow V(\text{Куб1}, A)$

или  $\rightarrow V(\text{Куб1}, A)$  в сокращенной записи.

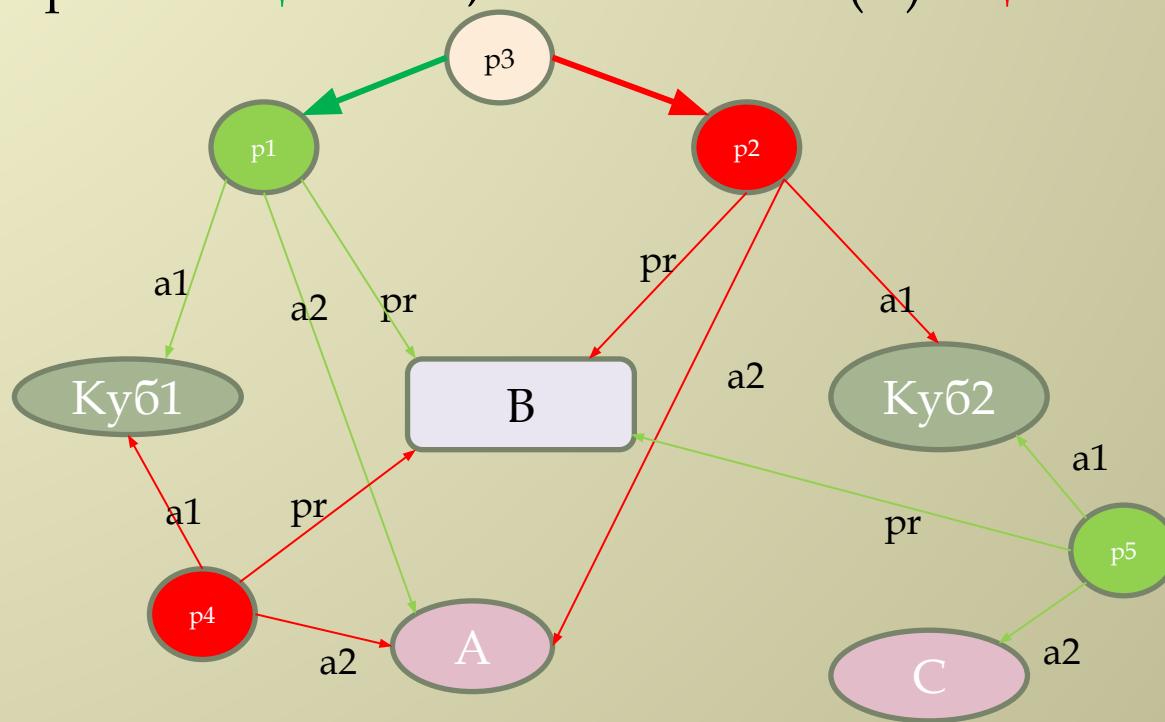
Утверждение 3:  $V(\text{Куб2}, C) \rightarrow L$

или  $V(\text{Куб2}, C) \rightarrow$  в сокращенной записи.



# Пример раскраски сети

Введём семантическую сеть для множества дизъюнктов. При этом используем операцию раскраски семантической сети: представим сеть как раскрашенный граф, где для дизъюнкта  $B \rightarrow A$  условие будет раскрашено **цветом 1**, а заключение (A) – **цветом 2**.



# Логическая сеть

Для удобства работы введём далее упрощение семантической сети: будем изображать на сети только вершины, соответствующие дизъюнктам, предикатные вершины и дуги, раскрашенные двумя цветами.

На таких сетях рассмотрим процедуру логического вывода (L – сеть).

Пусть логическая модель содержит дизъюнкты

$$g1: \rightarrow A$$

$$g2: A \rightarrow B$$

$$g3: B \rightarrow C$$

Необходимо доказать истинность утверждения C (то есть  $\rightarrow C$ ).

Для доказательства применим правило modus tollens:

$$B \rightarrow A$$

$$\underline{A \rightarrow}$$

$$B \rightarrow$$

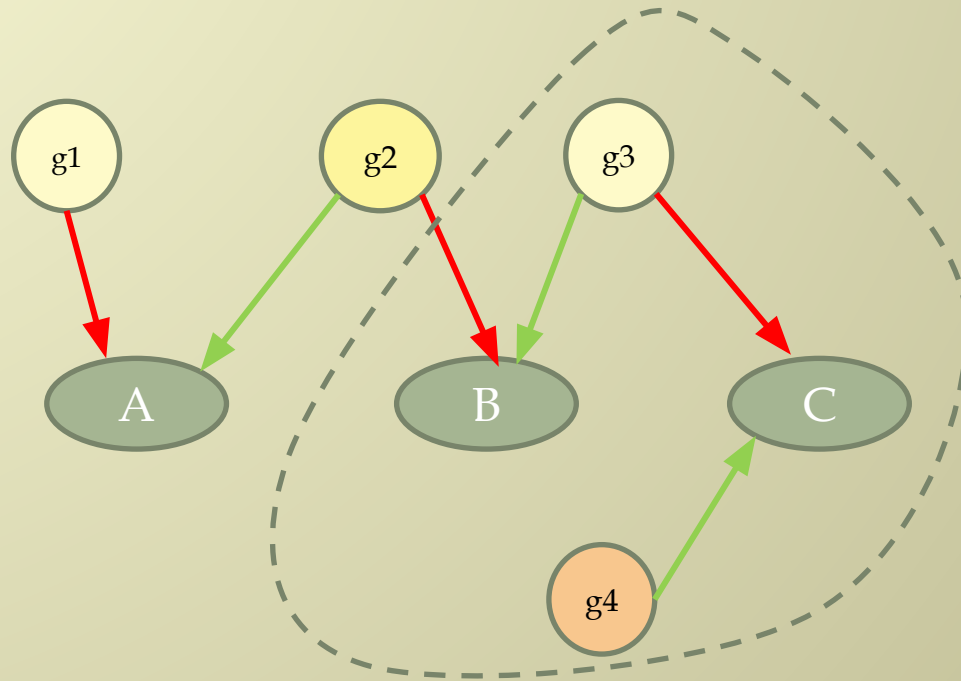
отрицаний:

Это правило предназначено для получения

Если B то A; A ложно; следовательно, B ложно.

# Пример логического вывода шаг 1

Построим L-граф и добавим к нему отрицание доказываемого  $g4$ :  
 $C \rightarrow$



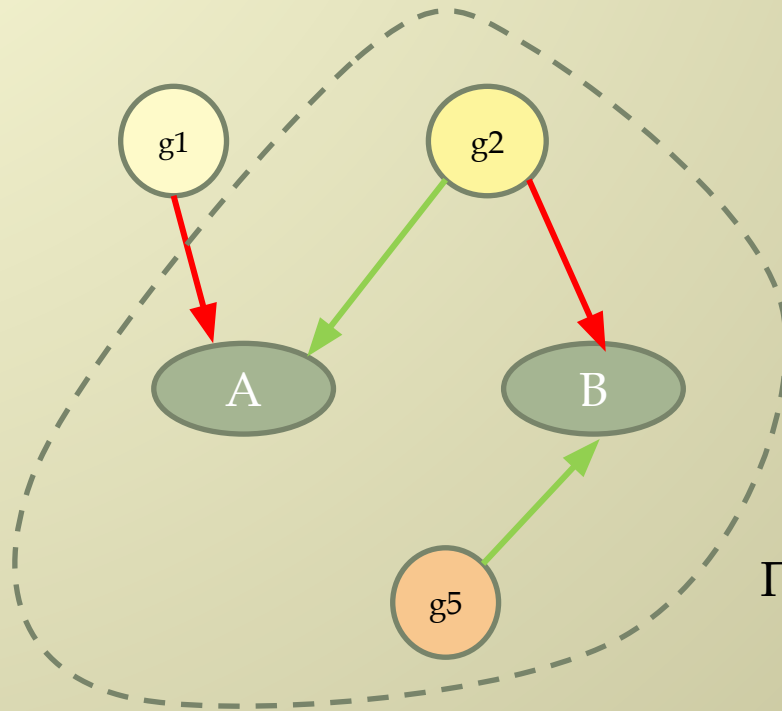
Для выделенной части графа применяем правило  $B \rightarrow C$

$$\frac{C \rightarrow}{B \rightarrow}$$

(доказали ложность  $B$ )

## Пример логического вывода шаг 2

Добавляем к L-графу новое утверждение о ложности В  $g5: B \rightarrow$



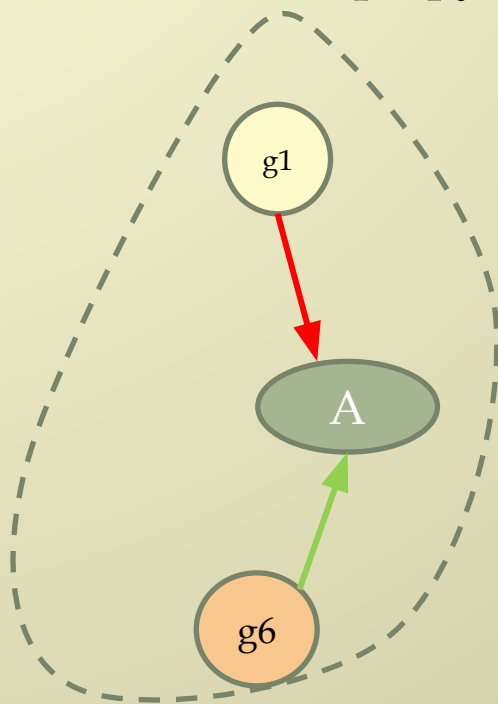
Применяем  $A \rightarrow B$

$$\frac{B \rightarrow}{A \rightarrow}$$

Доказали ложность А  $g6: A \rightarrow$

## Пример логического вывода шаг 3

Добавляем к L-графу новое утверждение  $g_6: A \rightarrow$



Получили противоречие  $g_1: \rightarrow A$

$g_6: A \rightarrow$

Таким образом доказано С «от

противного».

# Лекция 13

Фреймы и сети фреймов

# Понятие фрейма

- *Фрейм* (англ. frame — каркас или рамка) предложен М. Минским в 70-е гг. как структура знаний для восприятия пространственных сцен. Фрейм (дословно — «рамка») — это единица представления знаний, детали которой могут изменяться в соответствии с текущей ситуацией. Фрейм - это минимально возможное описание сущности какого-либо явления, события, ситуации, процесса или объекта.
- Одним из возможных новых путей организации машинной модели реального мира является подход, развиваемый М. Минским. В соответствии с этим подходом знания о мире - машинная модель реального мира - должны быть представлены в памяти ЭВМ в виде достаточно большой совокупности определенным образом структурированных данных, представляющих собой стереотипные ситуации. Эти структуры запомненных данных получили название "фреймы".

В общем случае фрейм – это структура данных, предназначенная для представления некоторой стандартной ситуации. С каждым фреймом связана разнообразная информация, как представленная явно, так и процедурная. Таким образом фрейм содержит информационные и процедурные элементы, обеспечивающие преобразование информации и связь с другими фреймами.

В случае возникновения конкретной ситуации, например, необходимости совершить роботом, управляемым ЭВМ, определённое действие, воспринять с помощью сенсоров, связанных с ЭВМ, какой-то зрительный образ и т.д., из памяти ЭВМ должен быть выбран фрейм, соответствующий данному классу ситуаций и согласован с рассматриваемой конкретной ситуацией из этого класса путем конкретизации данных, необходимых для заполнения фрейма.



# Структура фрейма

Фрейм можно представить себе в виде сети, состоящей из узлов и связей между ними, где каждый узел играет свою роль при описании тех или иных характерных черт ситуации, которой он соответствует. В общем случае во фрейме можно выделить несколько уровней, иерархически связанных друг с другом. Узлы фрейма, принадлежащие к верхним уровням, представляют собой более общие свойства, которые всегда справедливы в отношении предполагаемой ситуации. Эти узлы уже заполнены своими данными.

Например, узел самого верхнего уровня фрейма обычно заполнен названием ситуации, т. е. названием зрительного образа (это может быть, например, "куб"), названием действия (например, "уборка комнаты"). Узлы нижних уровней по большей части не заполнены данными. Такие незаполненные узлы называют терминалами.

# Определение фрейма по М.Минскому

**Фрейм** – это абстрактный образ для представления некоего стереотипа восприятия. В психологии и философии известно понятие абстрактного образа. В теории фреймов такой образ называется фреймом. Фреймом называется также и формализованная модель для отображения образа.

В настоящее время вместо названия *фрейм* чаще используют понятия объектно-ориентированный, структурированный объект, и подобные структуры легко реализуются на объектно-ориентированных языках программирования.

Фреймы могут объединяться в сети фреймов. Результаты изменения данных в одном фрейме могут, таким образом, отражаться в других, связанных с исходным, фреймах системы. Это удобно, если происходят типичные изменения одной и той же ситуации.

# Понятие слота

Важным понятием при разговоре об организации фрейма является **слот**.

**Слот** – это незаполненный участок во фрейме. Слот означается при обращении ко фрейму. С каждым слотом связаны условия, при которых становится возможным означивание.

Значения слотов могут быть разных типов (логическое, числовое, текст, изображение и т.п.) и могут быть получены из различных источников.

Мы рассматривали семантическую сеть как граф с помеченными вершинами и дугами, с этой точки зрения фрейм – это выделенный (помеченный) подграф такой сети. Фреймы следовательно также можно объединять в сети.

# Общая структура фрейма

Структуру фрейма можно представить так:

Имя фрейма:

{имя 1-го слота: значение 1-го слота}  
{имя 2-го слота: значение 2-го слота}

.....

{имя n-го слота: значение n-го слота}

Значение слота может быть взято по умолчанию (default),  
может быть вычислено процедурно,  
может быть получено из Базы Данных путём обращения к  
БД (запрос), может быть получено из другого фрейма  
(ссылка на имя другого фрейма).

## Пример организации фрейма

В качестве примера рассмотрим фрейм для понятия «взятие»:



**«Взятие»:**

(*Субъект, X1*);

(*Объект, X2*);

(*Место, X3*);

(*Время, X4*);

(*Условие, X5*).

В этом фрейме указаны имена слотов (субъект, объект и т.д.), но вместо их значений стоят переменные (X1, X2 и т.д.). Такой фрейм называется фреймом-прототипом, или протофреймом. Мы видим, что понятие «взять» связано с наличием слотов с указанными именами. Взятие осуществляет X1 в месте X3 во время X4, если выполнено условие X5. Берет X1 нечто, обозначенное как X2.

Подставляя вместо всех переменных конкретные значения, получим конкретный факт-описание:

**«Взятие»:**

(*Субъект, Робот*);

(*Объект, Деталь*);

(*Место, Приемный бункер*);

(*Время, X4*);

(*Условие, В бункере есть деталь, а у робота ее нет*).

# Типы фреймов

- Различают *фреймы-образцы*, или *прототипы*, хранящиеся в базе знаний, и *фреймы-экземпляры*, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:
  - · *фреймы-структуры*, использующиеся для обозначения объектов и понятий (заем, залог, вексель);
  - · *фреймы-роли* (менеджер, кассир, клиент);
  - · *фреймы-сценарии* (банкротство, собрание акционеров, празднование именин);
  - · *фреймы-ситуации* (тревога, авария, рабочий режим устройства) и др.

# Наследование свойств

- Важнейшим свойством теории фреймов является заимствованное из теории семантических сетей наследование свойств. И во фреймах, и в семантических сетях наследование происходит по *ISA-связям*. Слот ISA указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т.е. переносятся, значения аналогичных слотов. Основным преимуществом фреймов как модели представления знаний является способность отражать концептуальную основу организации памяти человека, а также ее гибкость и наглядность.

# Языки для создания сетей фреймов

Существуют специальные языки представления знаний в сетях фреймов FRL (Frame Representation Language) и другие позволяют эффективно строить промышленные Экспертные Системы. Широко известны такие фрейм-ориентированные экспертные системы, как ANALYST, МОДИС.

Построенная на одном из таких языков информационно-поисковая сеть может делать выводы на основе условий, описывающих некую начальную ситуацию, отвечать на запросы, в том числе и делать прогнозы относительно развития текущей ситуации.



# Применение семантических сетей и фреймов.

Перечислим основные области применения семантических сетей и фреймов.

1. Анализ естественно – языковых текстов. Требуется по исходным данным – связному тексту – получить структуру, описывающую ситуацию.
2. Порождение текста по структуре (фрейму). Пример создания связного текста – генерация сказок, стихотворных произведений.
3. Использование в информационно-поисковых системах для ответов на запросы.
4. Машинное обучение на структурах. В случае, когда программа машинного обучения работает со структурными данными, преимущество – в лёгкости сравнения, добавления и удаления как вершин, так и дуг графа.
5. Представление знаний о Предметной области для экспертных систем и систем поддержки принятия решений.