

## *Лекция 6*

Модели представления знаний:

*Фреймы и семантические сети*

# Фреймы

**Фрейм** (*frame* – каркас, рамка, несущая, конструкция) – модель представления знаний в ИИ с использованием *абстрактного образа* некоторой *сущности* (объекта, явления, события, ситуации, процесса и т.п.), включающего ее *минимально возможное описание*

- термин ввёл М.Минский в 1974 г. как обозначение абстрактного образа для представления некоторого стереотипа восприятия

(М. Минский. Фреймы для представления знаний)

Типы фреймов:

- фреймы-образцы;
- фреймы-экземпляры;
- фреймы-структуры;
- фреймы-роли (клиент, менеджер, кассир... );
- фреймы-сценарии (посещение ресторана, собрание акционеров...);
- фреймы-ситуации (штатный режим устройства, тревога, авария...)

Частным случаем фреймов являются *скрипты* – описания стереотипных сценариев действий с участием определенных объектов

Первые фреймовые языки – KRL, FRL. Дальнейшее развитие привело к созданию ООЯП

## Фреймы как модель представления знаний

*Фрейм* – содержит множество характеристик ситуации, называемых *слотами*, и их значений

### Обобщенная структура фрейма:

(Имя\_фрейма, (Имя\_слота\_1, Значение\_слота\_1),  
(Имя\_слота\_2, Значение\_слота\_2),  
... ..  
(Имя\_слота\_N, Значение\_слота\_N))

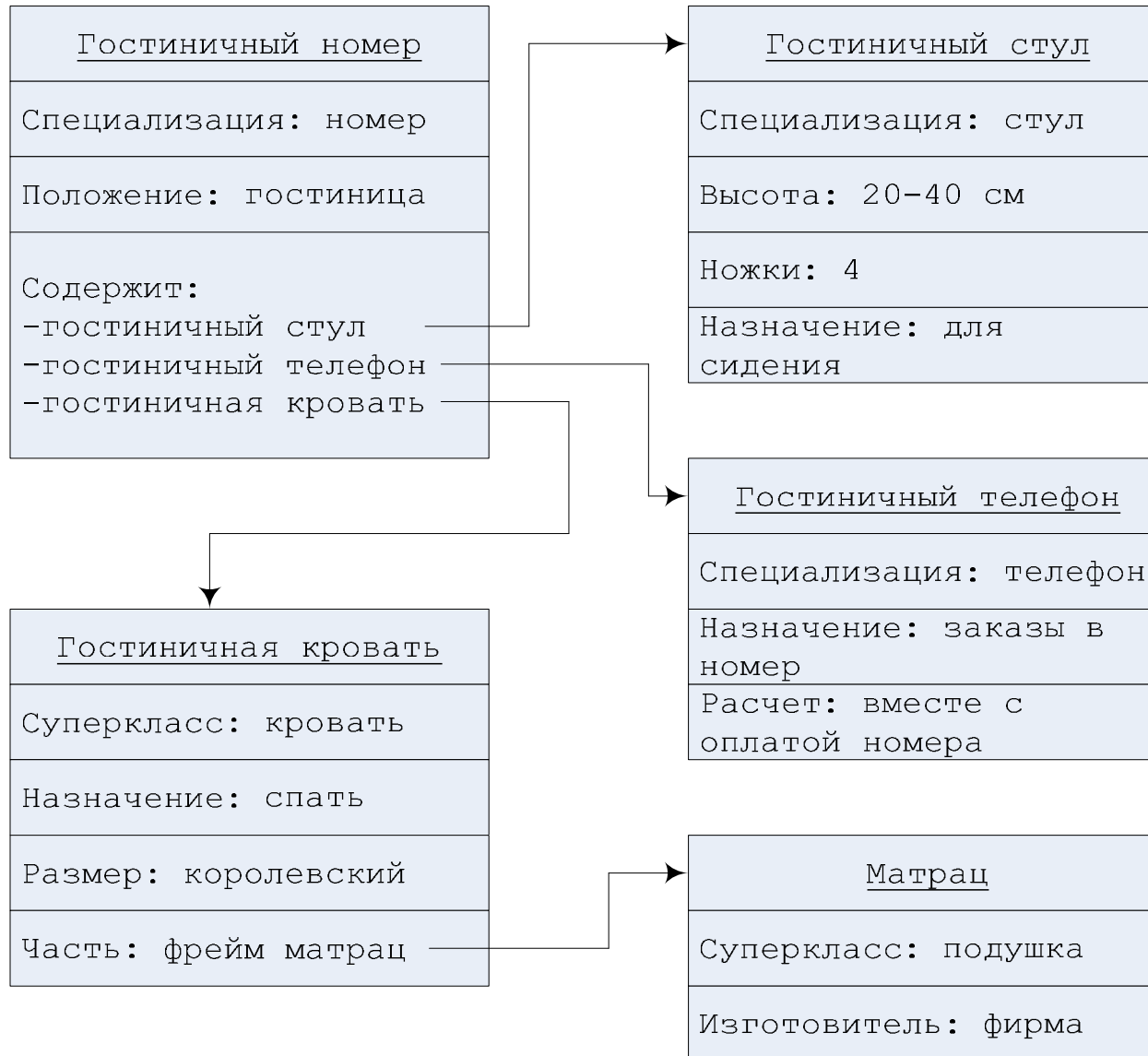
*Имя\_фрейма* – уникальный в рамках данной фреймовой системы идентификатор

*Имя\_слота\_n* – уникальный в пределах данного фрейма идентификатор слота

### Расширенная структура фрейма:

(Имя\_фрейма, (Имя\_слота\_1, Указатель\_наследования,  
Указатель\_типа\_данных, Значение\_слота, Демон),  
... ..  
(Имя\_слота\_N, Указатель\_наследования,  
Указатель\_типа\_данных, Значение\_слота, Демон))

## Фрейм гостиничного номера



# Определение значений слотов во фреймах

Способы получения слотом значений во фрейме-экземпляре:

- по умолчанию от фрейма-образца;
- через наследование свойств от фрейма, указанного в слоте «род — вид» (наличие связи типа «род — вид» между объектами А и В означает, что понятие А более общее, чем понятие В);
- по формуле, указанной в слоте;
- через присоединенную процедуру;
- явно из диалога с пользователем;
- из базы данных;

Фреймовые системы поддерживают механизм *наследования сущностей*

*Указатель наследования* – используется во фреймовых системах иерархического типа, позволяющих организовать иерархию знаний

- указывает, какую информацию об атрибутах слотов фрейма верхнего уровня наследуют слоты с теми же именами нижнего уровня

## Типовые указатели наследования - U

U (unique) – каждый фрейм может иметь слоты с различными, уникальными значениями

Фрейм верхнего  
уровня

Фрейм	Слот	Значение
Человек	Масса	70



Фрейм среднего  
уровня

Фрейм	Слот	Значение
Ребенок	Масса	40



Фрейм низшего  
уровня

Фрейм	Слот	Значение
Петя	Масса	32

## Типовые указатели наследования – S

S (same) – все слоты должны иметь одинаковые значения

Фрейм верхнего  
уровня

Фрейм	Слот	Значение
Человек	Масса	50



Фрейм среднего  
уровня

Фрейм	Слот	Значение
Ребенок	Масса	50



Фрейм низшего  
уровня

Фрейм	Слот	Значение
Петя	Масса	50

## Типовые указатели наследования – R

R (range) – значение слотов фрейма должны находится в диапазоне указанных значений слотов фрейма верхнего уровня

Фрейм верхнего  
уровня

Фрейм	Слот	Значение
Человек	Масса	3-130



Фрейм среднего  
уровня

Фрейм	Слот	Значение
Ребенок	Масса	3-40



Фрейм низшего  
уровня

Фрейм	Слот	Значение
Петя	Масса	32



## Типовые указатели наследования – О

О (override) – при отсутствии значения оно наследуется от слота верхнего уровня, в случае явного задания значения наследование игнорируется

Фрейм верхнего  
уровня

Фрейм	Слот	Значение
Человек	Масса	70



Фрейм среднего  
уровня

Фрейм	Слот	Значение
Ребенок	Масса	50



Фрейм низшего  
уровня

Фрейм	Слот	Значение
Петя	Масса	50

## Указатель типа данных

*Указатель типа данных* – указывает тип, к которому должно относиться значение слота (аналогично типизации переменных в языках программирования)

Возможные значения указателя:

- FRAME – указатель на другой фрейм;
- INT – целое;
- REAL – вещественное;
- BOOL – булево значение;
- TEXT – текст;
- LIST – список;
- TABLE – таблица;
- EXPR – выражение;

Тип значения должен совпадать с явно указанным.

## Присоединенные процедуры и демоны

*Присоединенные процедура* – процедурный элемент фрейма (слот), программа, запускаемая по сообщениям, переданным из другого фрейма.

*Демон* – процедура, автоматически запускаемая при выполнении некоторого условия и выполняющая определенные действия со значениями слотов

### Типы демонов:

IF-NEEDED – запускается, если в момент обращения к слоту его значение не установлено;

IF-ADDED – запускается при записи в слот некоторого значения;

IF-REMOVED – запускается при удалении значения слота;

*Демоны* можно рассматривать как частный случай *присоединенных процедур*

- например, при каждом изменении значения слота можно запускать процедуру контроля соответствия типов или тест непротиворечивости

## Достоинства фреймового представления знаний

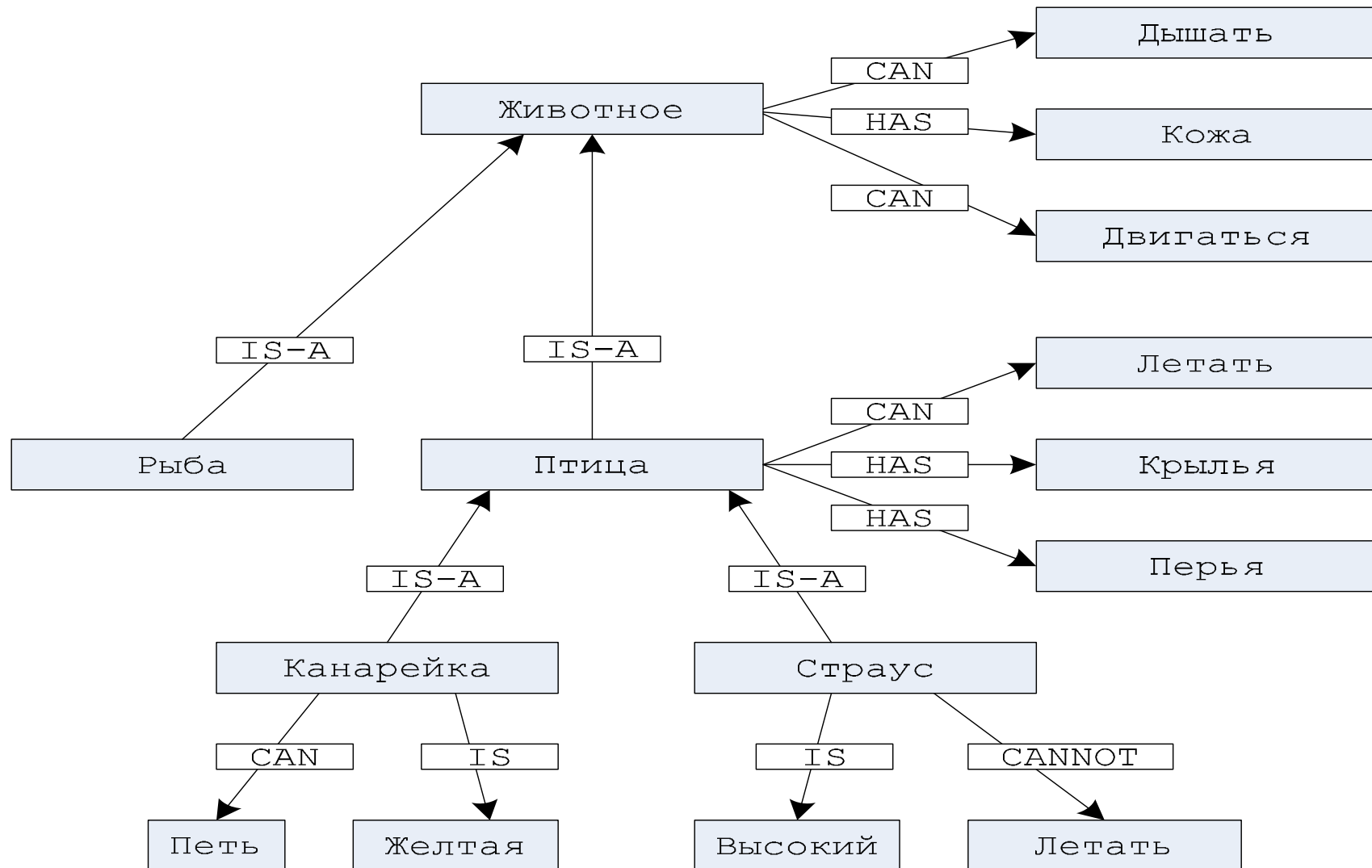
- Иерархичность структуры единиц представления знаний
- Различие степени абстракции единиц знаний
- Возможность комбинации декларативных и процедурных знаний

## Семантические сети

*Семантическая сеть* - помеченный ориентированный граф, вершины которого соответствуют сущностям предметной области (объектам, событиям, свойствам, процессам, явлениям), а дуги – отношениям между ними

С точки зрения математической логики, каждая вершина соответствует элементу предметного множества, а дуга — предикату.

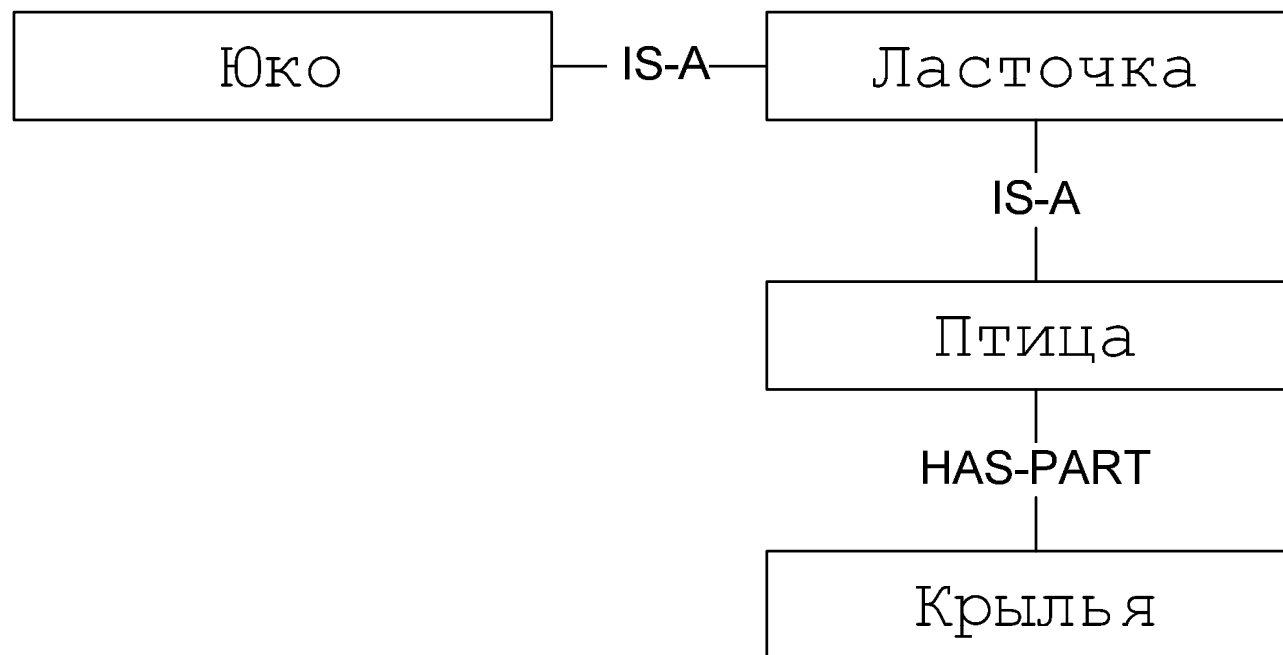
## Пример семантической сети



# Классификация семантических сетей

- *По сложности структуры:*
  - Простые – вершины не имеют внутренней структуры;
  - Иерархические – вершины имеют внутренней структуры;
- *По количеству типов отношений:*
  - Однородные – только один тип отношений ;
  - Неоднородные – разные типы отношений;
- *По арности (кардинальности) отношений:*
  - Бинарные;
  - N-арные;

## Фрагмент простой семантической сети



Представление знаний в ЛППП. Обозначим:

$P(x)$  – “ $x$  - ласточка”

$Q(x)$  – “ $x$  - птица”

$\forall x [P(x) \rightarrow Q(x)]$



## Базовые отношения в семантических сетях

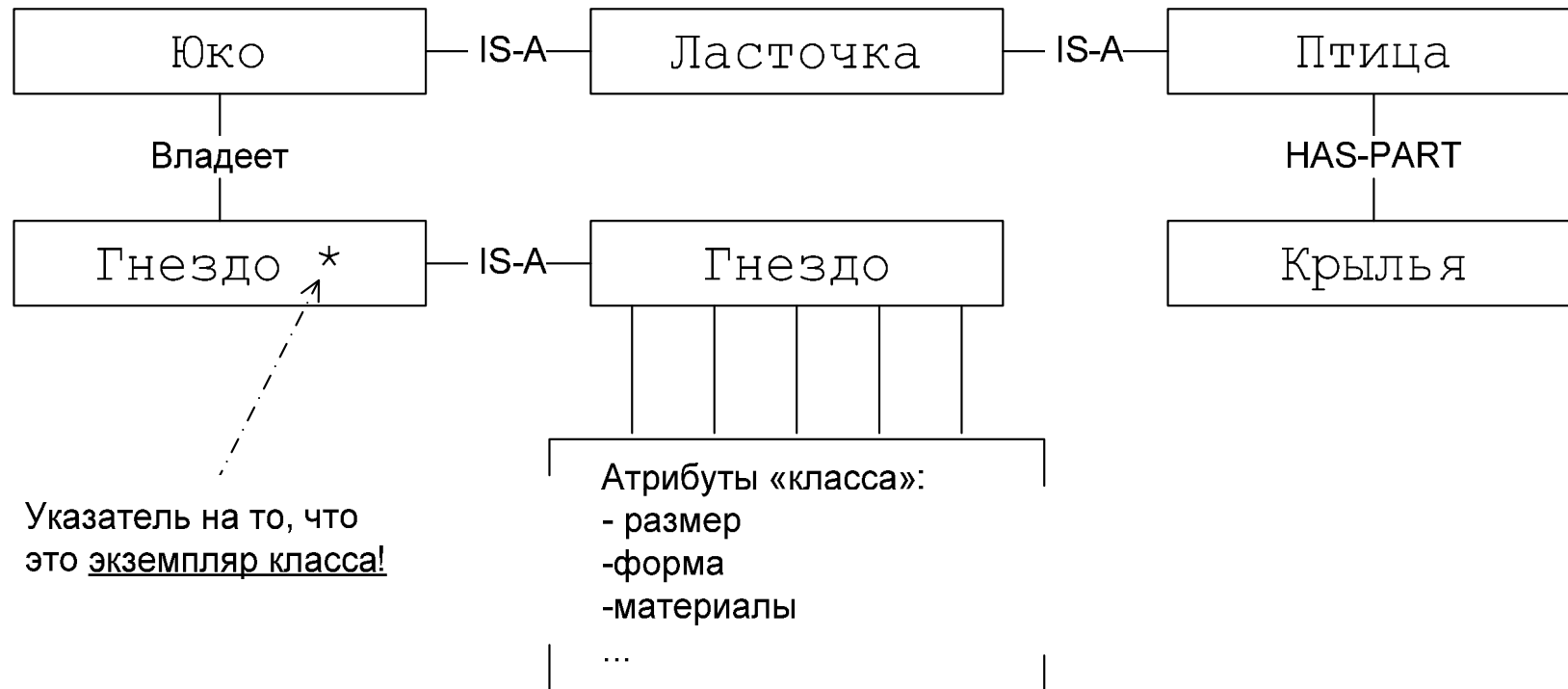
- **IS-A** – отношение между объектом и множеством, означает принадлежность объекта соответствующему множеству (классу)
  - классифицирует объекты;
  - предполагает, что свойства объекта *наследуются* от класса (другое название – MemberOf)

Обратное к IS-A отношение иногда называют «Example» (пример)

- **A-KIND-OF** (AKO, вариант - SubsetOf) – отношение между подмножеством и надмножеством; определяет, что каждый элемент первого множества входит и во второе, свойства первого множества наследуются от второго.
- **HAS-PART** – отношение части и целого (компьютер состоит из системного блока, монитора, клавиатуры, мыши и т. д.)

## Наследование в семантических сетях

Транзитивность свойств, определяющих иерархию понятий в СС, позволяет делать выводы на основе наследования



В данном примере наследование обусловлено транзитивностью отношений IS-A и HAS-PART

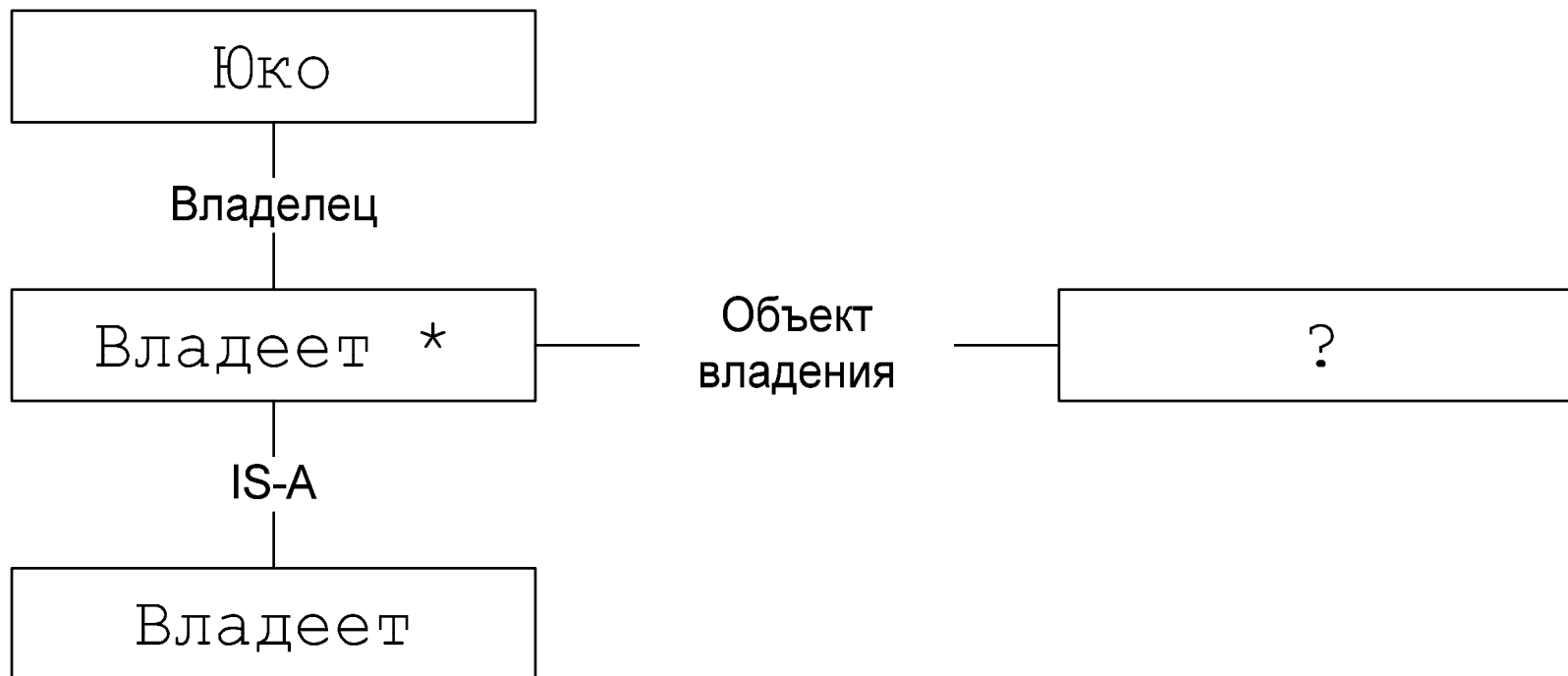
## Отношения как сущности с атрибутами



## Вывод на семантических сетях

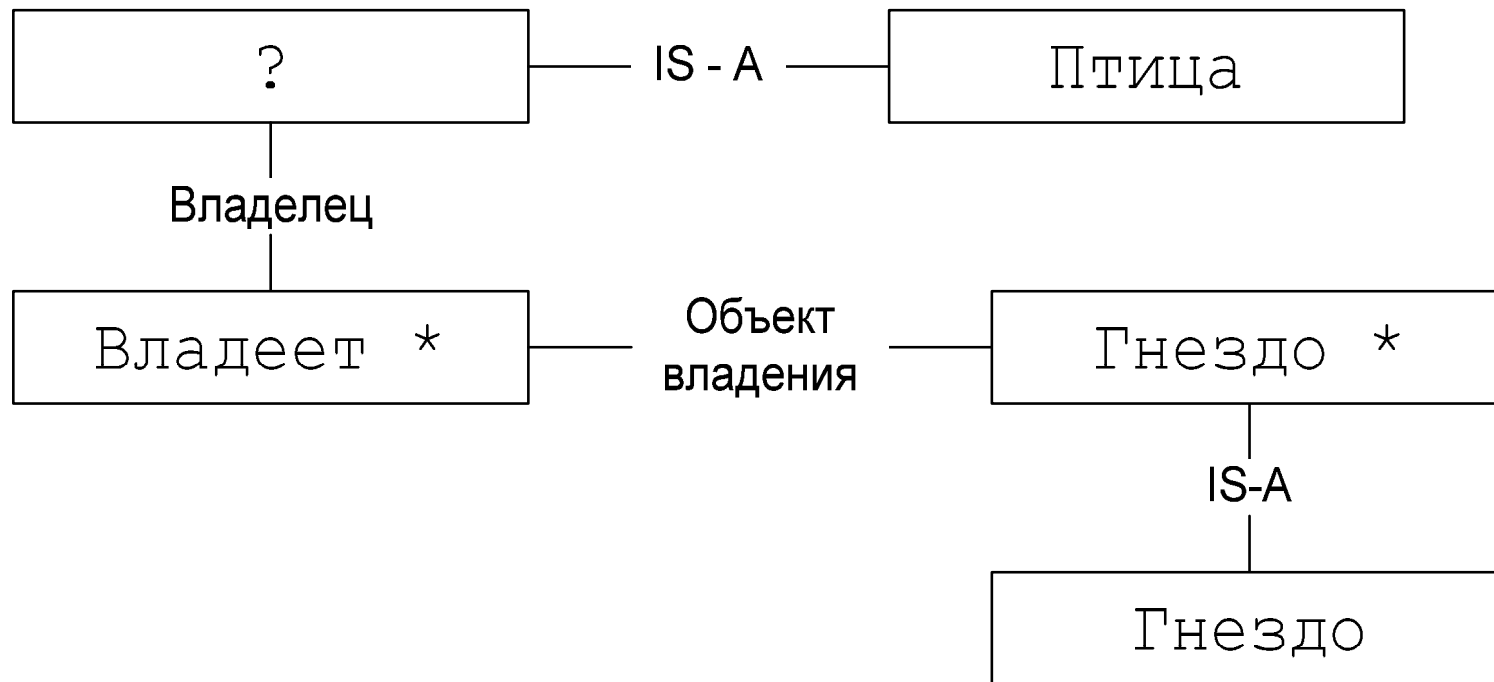
В основе процедур вывода на семантической сети лежит сопоставление частей сетевой структуры (подграфов). Для обработки запроса строится соответствующая подсеть

*Пример.* Запрос: «Чем владеет Юко?». Соответствующая подсеть:



## Вывод на семантических сетях

Пример 2. Запрос: «Существует ли птица, которая владеет гнездом?»  
Соответствующая подсеть:



## Семантические сети и Семантический Web

Теория семантических сетей получает применение при создании *семантического Web* – метода представления данных в сети Интернет, при котором возможна их семантическая обработка.

Основу *семантического Web* составляет семейство стандартов на языки представления знаний – RDF, RDF Schema, OWL, SWRL.

# Достоинства и недостатки семантических сетей

## Достоинства:

- удобство восприятия человеком

## Недостатки:

- Трудности при построении сложных сетей и попытке учета особенностей естественного языка