**Л 6-1. Представление знаний в виде семантической сети (дополнительная лекция 1)**

Общие понятия и определения. Роль отношений в СС. Свойства отношений. Предикатные семантические сети. Атрибутивные семантические сети. Вывод на семантических сетях.Достоинства и недостатки семантических сетей.

**Общие понятия и определения**

Само название «семантическая сеть» (СС) акцентирует внимание на смысле (семантика - смысловая сторона языка, «семантический» - связанный со смыслом), т.е. на значении тех слов, предложений, ситуаций, состояний, которые входят в модель предметной области и которые необходимо описать адекватно нашему (человеческому) пониманию таким образом, чтобы с этим смыслом можно было работать, т.е. моделировать процессы, рассуждения, выводы, сравнения, сопоставления и т.п. Очевидно, это непросто, хотя и возможно многими способами. Поэтому, не вдаваясь в языковые, психологические и другие подробности, под семантикой будем понимать отношения между некоторыми сущностями, в качестве которых могут выступать отдельные символы, слова, предложения, пакеты, пентаграммы, образы, т.е. все, что необходимо для описания объектов (процессов, состояний и т.п.) предметной области.

В психологии, например, изучались семантические объекты как структурные модели долговременной памяти человека, которые затем использовались для создания программ, понимающих естественный язык (ЕЯ). Эта проблема стала особенно актуальной в связи с необходимостью разработки систем автоматического перевода с одного языка на другой. Формализовать текст - значит построить по тексту на ЕЯ соответствующий текст на некотором формальном языке, описывающий языковую ситуацию. Таким образом, можно говорить о моделях представления семантики знаковых (языковых) систем и процедурах их обработки (понимания, распознавания, классификации и т.п.). С появлением СИИ такой подход стал использоваться для формализации знаний о предметной области. Семантическая сеть (СС) является удобным способом графического описания объектов предметной области. При этом под объектом может пониматься процесс, состояние, какая-либо конкретная сущность или сущность обобщённая (класс) и т.п.

Более строго определим СС как направленный граф с помеченными вершинами и дугами. При этом вершины отождествляются с сущностями (объектами) предметной области, а дуги - с отношениями между ними.

Большое разнообразие СС формируется исходя из следующих соображений:

* является ли вершина СС простой или сложной?
* сколько и какие отношения используются в СС?
* какие процессы необходимо моделировать на СС?

Будем говорить, что вершина СС является простой (ординарной), если она не имеет внутренней структуры. Другими словами, простые вершины отождествляются с терминальными (т.е. более не раскрываемыми) сущностями моделируемой среды. В качестве таких вершин могут, очевидно, фигурировать числа, лексемы, пентаграммы и т.п.

Сложные вершины допускают раскрытие вплоть до семантической сети, в которой участвуют только простые вершины. Таким образом, СС со сложными вершинами может разворачиваться, как матрешка.

В связи с многообразием отношений и сложностью исследования их свойств на практике стараются использовать минимальное число отношений для упрощения СС. И в зависимости от того, какие типы отношений используются, можно выделить несколько видов СС.

* 1. *Однородные СС.* В них используется один тип отношений (неважно какой, но один). В этом смысле любые ориентированные графы являются примерами однородных СС.
  2. *Иерархические СС.* Здесь используются отношения: часть - целое (Part-of-while), класс - подкласс или род - вид (АКО - А Kind Of), экземпляр класса - класс (Is А). Каждое из этих отношений образует свою иерархию. Яркими примерами иерархических СС являются организационные структуры, выстраиваемые по отношению «целое - часть». (Например, схема центральной избирательной комиссии: областная, городская, окружная, районная, участковая.)
  3. *Сценарные СС.* Используются отношения строгого или нестрогого порядка. Сюда же могут быть отнесены СС с отношениями времени, причина - следствие.
  4. *Функциональные СС.* В них используются функциональные отношения типа «вход - выход» (рис. 1).

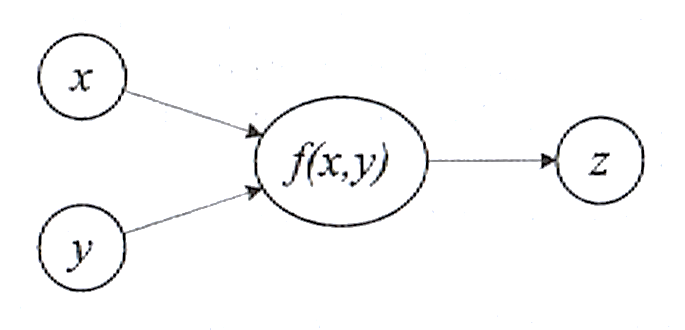
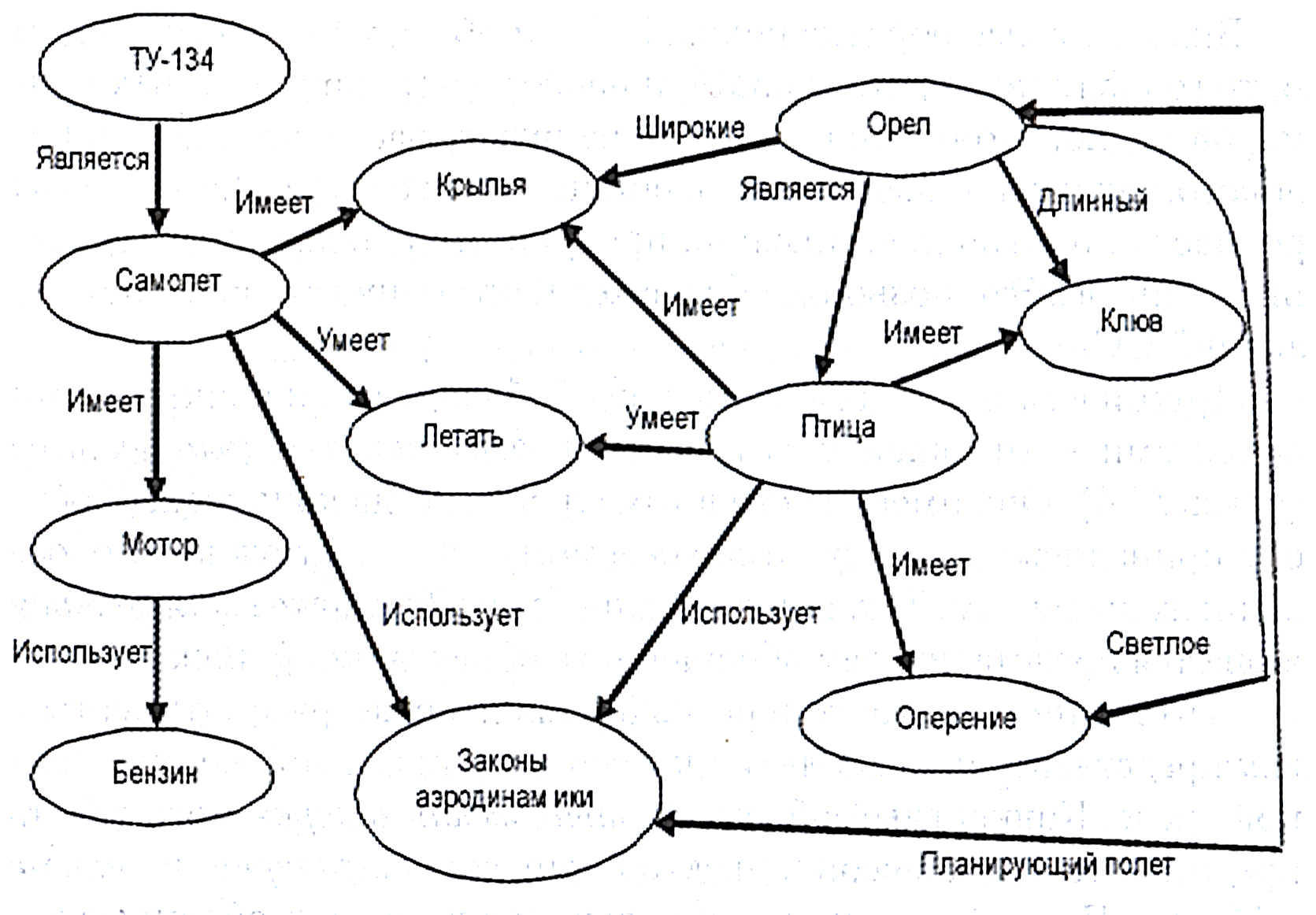


Рис. 1. СС типа «вход - выход»

На рис. 2 приведен пример СС с простой структурой.



*Рис. 2.* СС самолета и птицы

Роль отношений в СС

Выделим наиболее важные (фундаментальные) типы отношений между объектами, образующие «скелет (структуру)» и позволяющие более свободно ориентироваться в сложной и большой СС. Так, между двумя классами объектов, представленными в СС своими именами, может существовать родовидовое отношение *АКО (A kind of).* Наличие его между классами объектов *А* и *В,* например, означает, что любой объект, отображаемый понятием *В,* отображается и понятием *А,* и существует такой объект, который отображается понятием *А,* но не отображается понятием *В.* Иными словами, понятие *А* более общее, чем *В.* В таком случае говорят, что *А* является родом *В* или что понятие *А* - родовое для *В.* Так, понятие *животное* (и, соответственно, обобщенный объект *животное* в некоторой проблемной среде) родовое для понятия «человек», а понятие *транспорт -* родовое для понятия *самолет.*

Видовая связь обратна родовой. Если объект *А* является родом *В,* то говорят также, что *В* вид *А,* и наоборот. Из приведенных примеров видно, что родовое понятие не охватывает всех свойств видового, так как видовое понятие богаче содержанием. Все свойства родового понятия, как правило, присущи и видовому (наследование свойств). Это позволяет более компактно представить систему знаний СС.

Другой важный вид связи между обобщенными и конкретными объектами - отношение *является представителем (элементом)* (связь *IS А).* Она имеет место в том случае, когда конкретный объект принадлежит классу, отображаемому соответствующим обобщенным объектом. Так, конкретный объект *Третьяковская галерея* является представителем обобщенного объекта *музей.* Важно отметить, что один и тот же конкретный объект может рассматриваться как представитель нескольких классов в одной и той же проблемной среде. Конкретный объект *Третьяковская галерея* может быть представителем классов объектов *картинная галерея* и *здание в Москве.* Первый из них имеет родовидовую связь с объектом *музей*, второй - нет, так как музей может быть размещен в нескольких зданиях или на открытом воздухе и не всякое здание в Москве - музей. Свойства, присущие обобщенному объекту - классу, характеризуют и любой конкретный объект представитель. Таким образом, множество свойств конкретного объекта содержит в себе подмножество свойств, которыми он наделяется как представитель тех или иных обобщенных объектов (или совпадает с этим подмножеством).

Между сложным (составным) объектом и каким-либо другим объектом проблемной среды может существовать связь «являться частью целого» (Part of while - POW). Ясно, что частью конкретного составного объекта не может являться объект класс.

Типизация объектов и фундаментальные отношения не решают всех проблем представления знаний, но создают хорошую основу для построения прикладной базы знаний. Типизация обеспечивает и реальные осязаемые преимущества, позволяя сжать базу знаний, сделать ее более компактной. Так, для объекта класса, являющегося видом некоторого другого, нет необходимости представлять в базе все свойства, по крайней мере, часть их можно получить благодаря механизму наследования. Для такого объекта целесообразно представлять в базе лишь его собственные (ненаследуемые) свойства, а также свойства, которые хотя и являются наследуемыми, но несут дополнительную информацию.

Отношения многообразны по типу и свойствам. Основные типы отношений, используемых в СИИ:

* лингвистические (падежные или ролевые: кто, что, где, когда, зачем, почему, условие, место, время и т.п.);
* атрибутивные: форма, размер, цвет, вкус и т.п.;
* действия, когда каждый глагол рассматривается как отношение между объектом (субъектом) действия и объектом, выражающим некоторую способность (летает, ходит, стоит и т.п.), реализующуюся во времени как процесс;
* логические: ;
* квантифицированные: ;
* теоретико-множественные (множество, подмножество, объединение, дополнение, пересечение).

Наиболее многообразна группа лингвистических отношений, в которой могут быть выделены подгруппы временных *(раньше, позже, одновременно* и т.п.), пространственных *(над, под, внутри, рядом* и т.п.), причинно-следственных (*каузальных*) и других типов отношений.

Свойства отношений

Классификация отношений важна, поскольку разные типы отношений обладают различными свойствами, которые необходимо учитывать при построении моделей вывода в СС. Речь идет о следующих свойствах отношений: рефлексивность (нерефлексивность, антирефлексивность), симметричность (нессиметричность, антисимметричность), транзитивность (нетранзитивность, антитранзитивность). Соответственно, каждое отношение характеризуется определенным набором этих свойств.

Определим некоторые важные в дальнейшем свойства отношений.

**Рефлексивность.** Если *х* и *у* находятся в отношении *R(x,y),* то это отношение рефлексивно, если оно сохраняется при замене *у* на *x*, т.е. само с собой. Т.е. отношение *R* называется **рефлексивным**, если *xRx*. Так, отношения «быть похожим на», «быть не старше» (я не старше самого себя) рефлексивны. Отношения «быть братом», «быть старше» не рефлексивны. В графе *G(R)* рефлексивного отношения при каждой вершине имеется петля.

Отношение *R* называется **антирефлексивным**, если *R* может выполняться лишь для несовпадающих объектов, т. е. из *xRy* следует, что *х ≠ у.*

Антирефлексивными являются приведенные выше не рефлексивные отношения. В графе *G(R)* антирефлексивного отношения петли отсутствуют.

**Симметричность**. Отношение *R* называется симметричным, если при выполнении *xRy* выполняется и *yRx*. Отношения «быть похожим на», «быть родственником» (но не «быть мужем») симметричны. Другими словами, отношение *xRy* сохраняется, если объекты *х, у* поменять местами: *xRy ~ yRx* (знак ~ означает логическую эквивалентность).

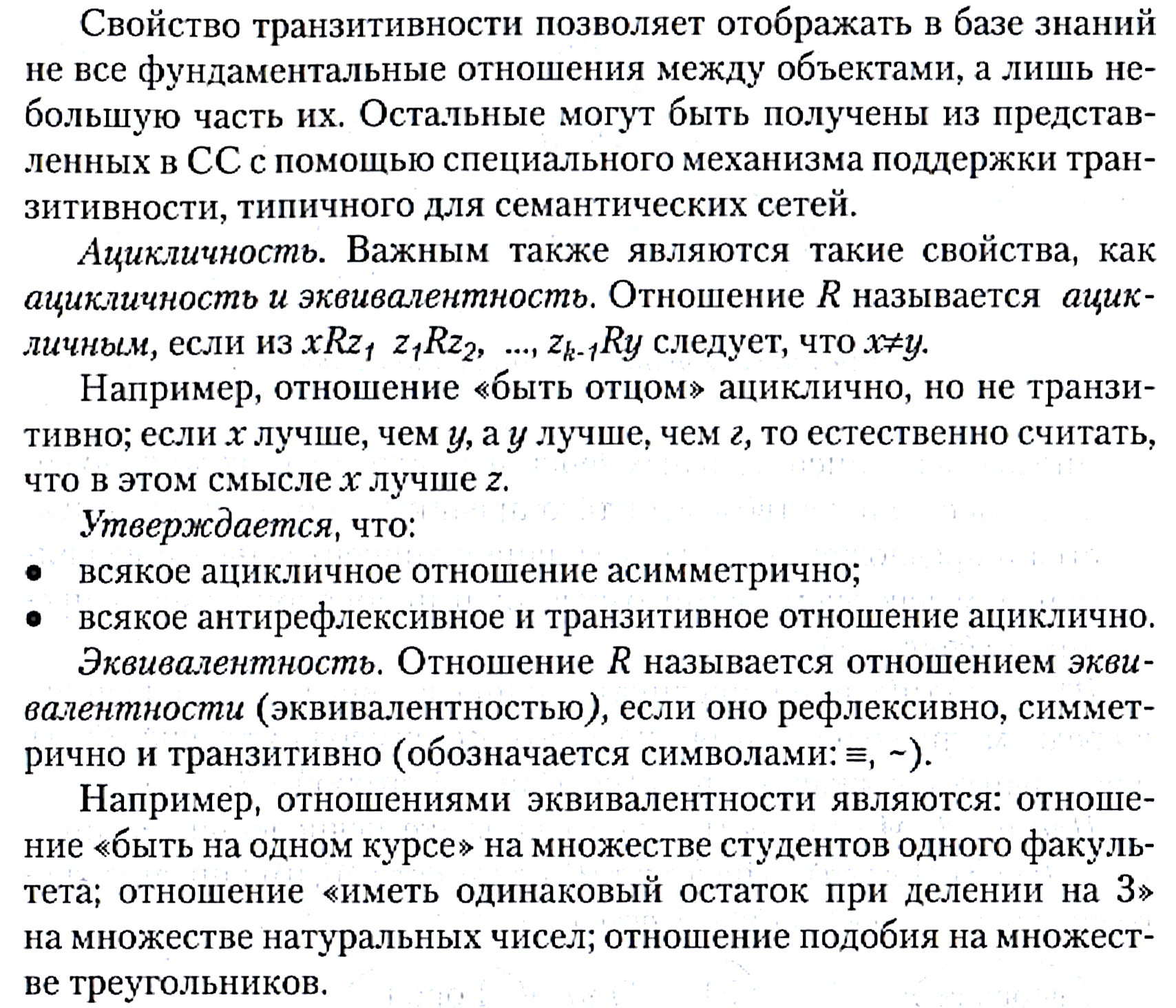
Отношение *R* называется **асимметричным**, если R ∩ R-1 = 0. Иначе говоря, из двух выражений *xRy* и *yRx* но меньшей мере одно несправедливо. Граф *G(R)* не может содержать одновременно дуги вида *(xi, xj)* и *(xj, xi)*. Так, отношение «быть братом» не является ни симметричным, ни асимметричным. Действительно, если Петр - брат Федора, то Федор - брат Петра; но если Игорь - брат Ольги, то неверно, что Ольга - брат Игоря.

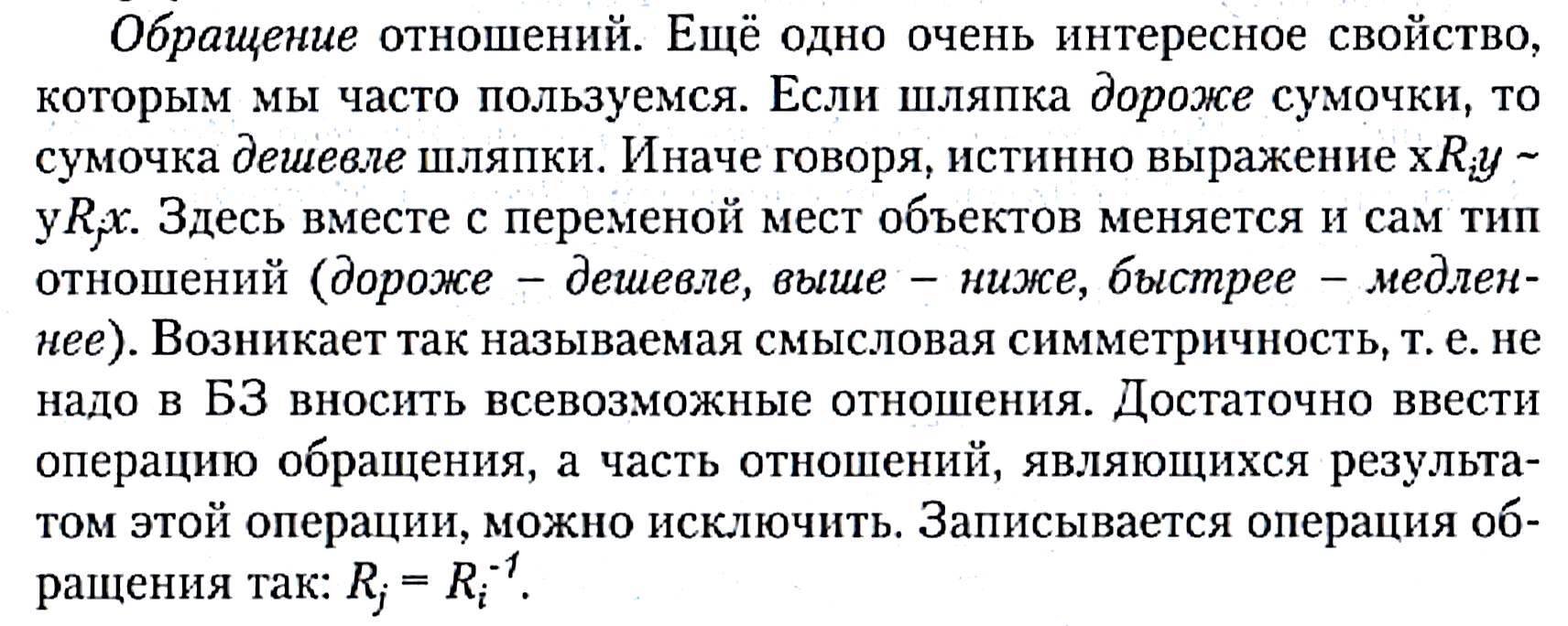
Верно утверждение, что если отношение *R* асимметрично, то оно антирефлексивно.

Отношение *R* называется антисимметричным, если выражения *xRy* и *yRx* справедливы одновременно только тогда, когда *х = у*. Граф *G(R)* не может содержать одновременно дуги вида *(xi, xj)* и *(xj, xi)* при *i ≠ j.*

**Транзитивность**. Отношение R называется транзитивным, если *xRz* и *zRy*, то *xRy*. Отсюда по индукции получаем: если *xRz1, z1Rz2,...,* *zk-1Ry*, то *xRy*. Например, если выполняются отношения х < у и y < z, то выполняется и отношение х < z. Или: *Ваня старше* Пети, *Петя старше Кати*, следовательно, *Ваня старше Кати*; если *Пётр пришёл раньше Ивана*, *Иван пришёл раньше Фёдора*, то *Пётр пришёл раньше Фёдора*.

Свойство транзитивности отношений играет особую роль, поскольку позволяет отображать в базе знаний не все отношения между объектами, а лишь небольшую часть их. Так, перечисленные выше отношения *Is А, АКО (A kind of), POW ((Part-of-while* (часть - целое)*)* являются транзитивными и образуют «скелет» СС, позволяющий более свободно ориентироваться даже в сложной и большой модели. Транзитивность родовидового отношения обеспечивает также возможность наследования свойств не только у представленного в базе знаний родовидового объекта, но и родового для родового. Возможности «сжатия» базы знаний обеспечиваются также транзитивностью родовидового отношения POW(является частью).





Предикатные семантические сети

Рассмотрим тип СС, формируемых на основе анализа текста естественного языка (ЕЯ). Элементарное предложение в предикатных СС состоит из следующих компонентов:

* пропозициональная вершина, обозначающая конкретное предложение (высказывание, суждение);
* множество концептуальных вершин, число которых должно совпадать с количеством мест того предиката, который описывается в предложении. Дуги, ведущие к концептуальным вершинам, помечаются метками, означающими порядок записи в предикате (рис. 3).

Р1 - обозначение предиката, описываемого предложением; в среднем кружочке - имя предикатного символа (стрелка *Pred);* в остальных кружках - термы, первый - *А,* второй - *В.*

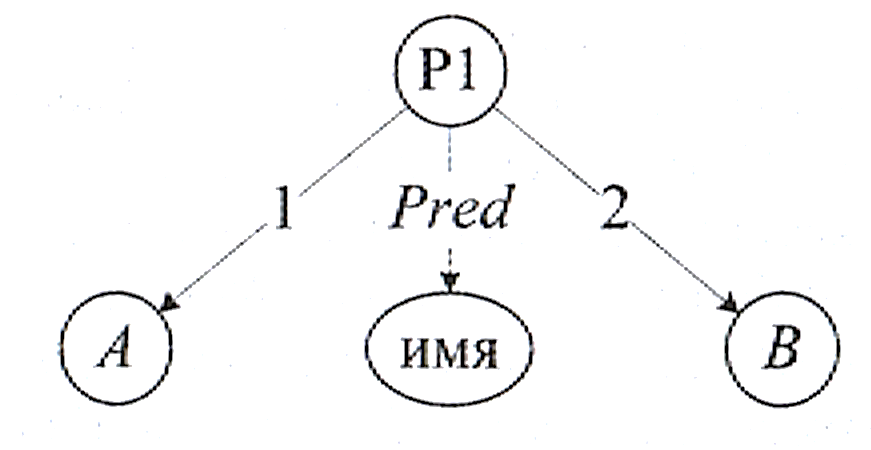
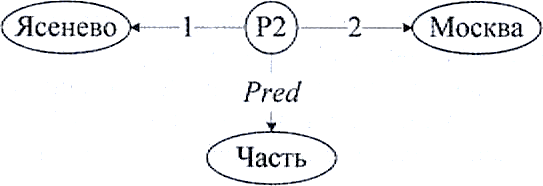


Рис. 3. «Ячейка» предикатной сети

*Пример 1.* Москва есть город (на языке исчисления предикатов - *Город (Москва)).* Предикат Р1 одноместный, имя предикатного символа - *Город,* терм - *Москва.*

G:\МГТУ\МГТУ_2014-15уч.год\Программа_2015_СистемыПоддержкиПринятияРешенийИБМ\Лекция 09-2015 ИБМ6-43\media\image3.jpeg

*Пример 2*. Ясенево - часть Москвы (Часть (Ясенево, Москва)). Здесь предикат Р2 {Часть) двухместный (Ясенево, Москва).



*Пример 3.* Рассмотрим текст, состоящий из следующих предложений. Ваня взял книгу у Маши. Лена взяла книгу у Пети. Петя любит Машу, а Маша любит конфеты.

*Предикатная форма:* (Взял (Ваня, Книга, Маша); Взял (Лена, Книга, Петя); Любит (Петя, Маша); Любит (Маша, Конфеты)).

Семантическая сеть, соответствующая этим предложениям, представлена на рис. 4.

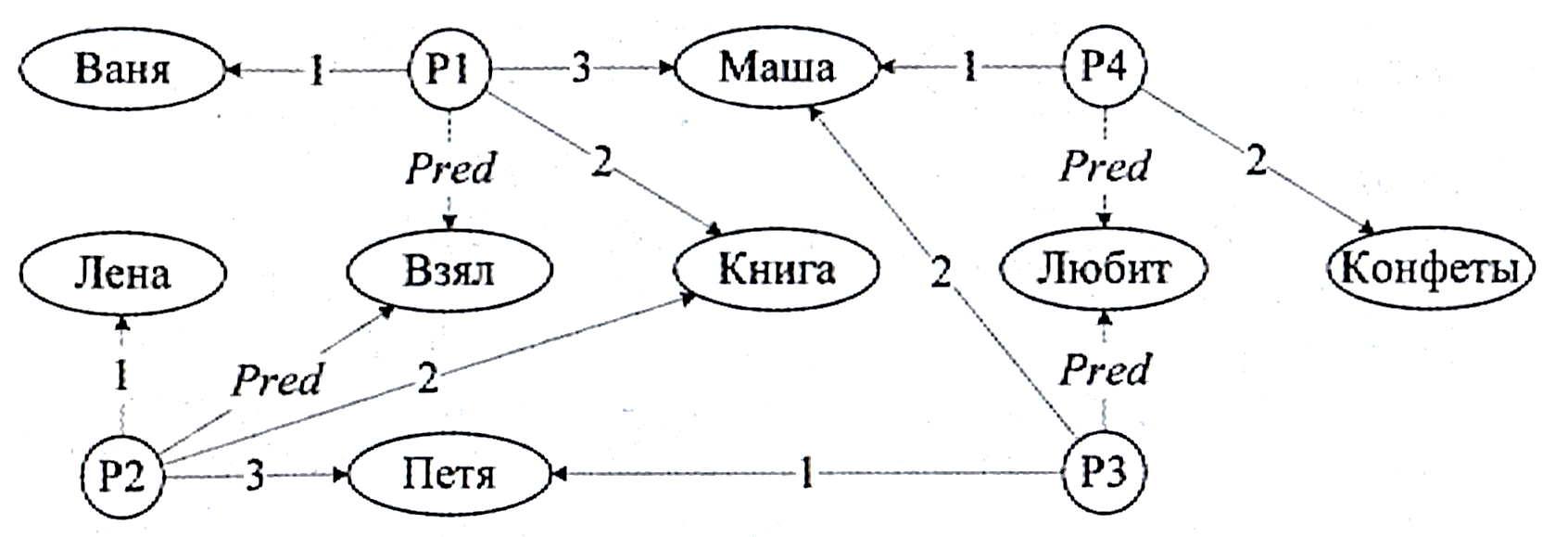


Рис. 4. Предикатная семантическая сеть про книги, любовь и конфеты

Из примеров видно, что:

1. частные факты и общие высказывания представляются в виде вершин графа;
2. предложение состоит из подграфов (подсетей) специального вида со связями между вершинами-понятиями и предикатными вершинами.

Подграф называется фрагментом сети или предложением. Для объединения составных предложений вводятся специальные вершины «*И*», *«ИЛИ»,* «*НЕ*» (). Их предикатная форма: рис. 4.



Рис. 4. Примеры «ячеек» предикатной сети, моделирующих пропозициональные связки

*Пример 5.* Говорят, что: *ИЛИ* Петя любит Машу (Р1), *ИЛИ* Паша любит Машу (Р2), *ИЛИ* Ваня любит Машу (РЗ), *И* Ваня выше Пети (Р4). Рисунок 5 фиксирует эту ситуацию. Здесь введены дополнительные вершины: P5((Pl, Р2, РЗ)) и P6(P5P4) для предложений типа «*ИЛИ*» и «Я».

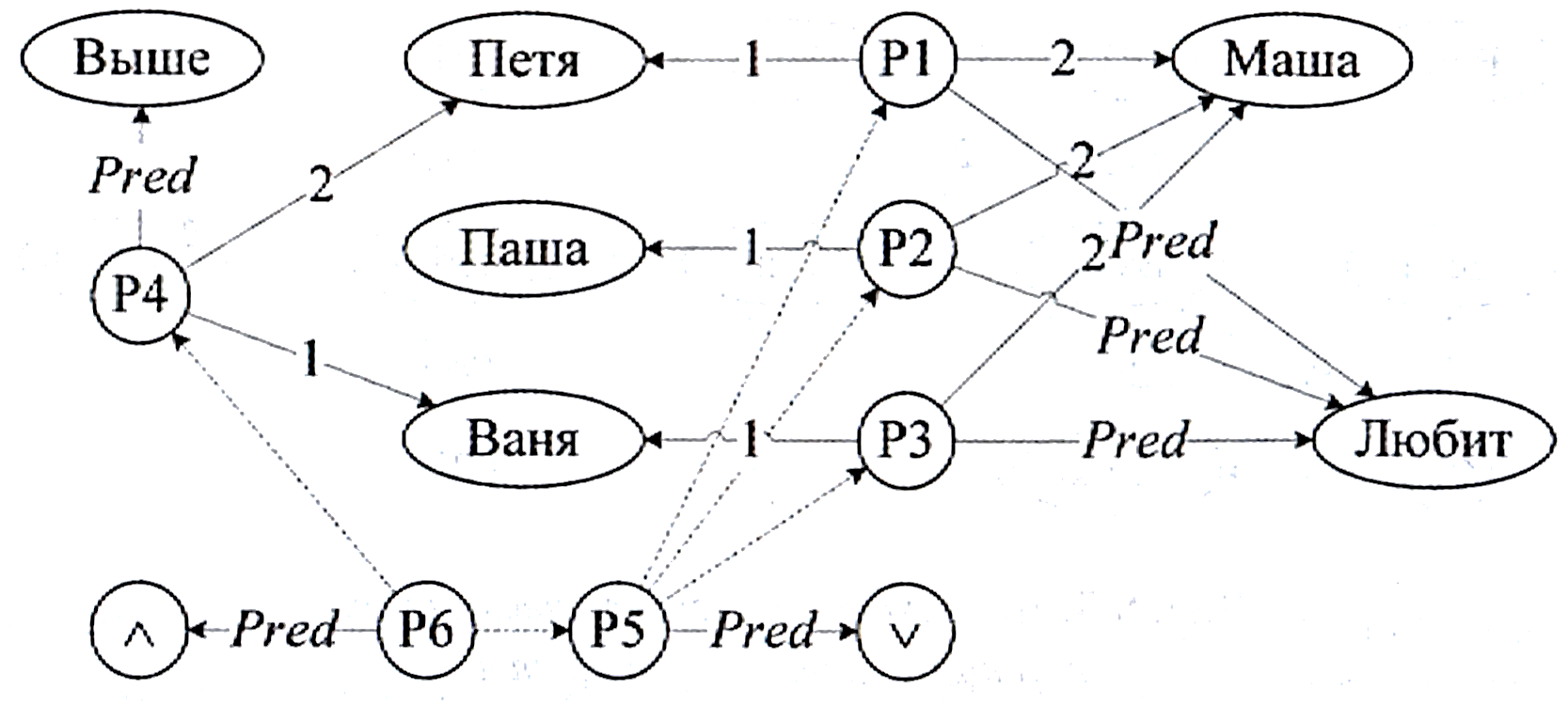


Рис. 5. Предикатная сеть с вершинами типа «И» и «ИЛИ»

По такой сети легко восстановить первоначальный смысл каждого предложения, составить текст на каком-либо другом языке. Действительно, СС подобного типа чаще всего используют в системах машинного перевода, а также для обработки научно-технической информации, например, для аннотирования, реферирования. При этом процесс обработки текста можно представить в виде схемы.

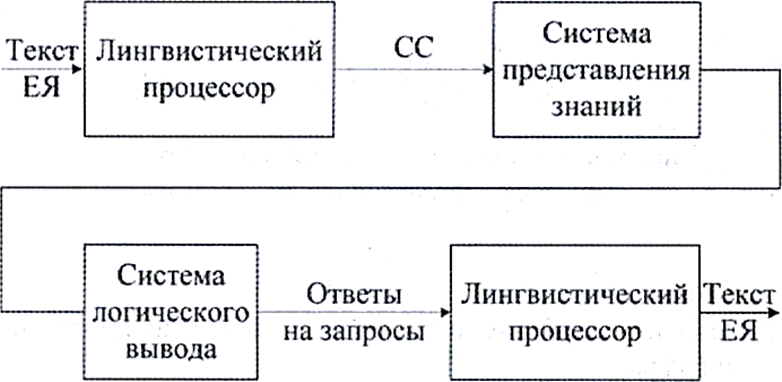


Рис. 6. Схема системы машинного перевода

Лингвистический процессор приводит текст на естественном языке (ЕЯ) к виду СС, которая затем обрабатывается системами представления и обработки знаний для получения ответов на запросы на ЕЯ. Перевод с ЕЯ на язык СС и обратно также осуществляется лингвистическим процессором (рис. 6).

Атрибутивные семантические сети

Этот популярный тип семантических сетей определяется через систему унарных отношений. Как известно, унарные отношения определяются одноместным предикатом или свойством, описываемого понятия. Например, мы говорим: яблоко красное и вкусное. Это отношение соответствует записям: *ЯБЛОКО (ЦВЕТ -* красный), *ЯБЛОКО (ВКУС* - сладкий). Такого рода отношения называют атрибутами, а соответствующие СС - атрибутивными. Здесь предполагается известным: имя объекта (класса, подкласса, единичного экземпляра), перечень его атрибутов и область задания каждого атрибута. Например, атрибут яблока ЦВЕТ можно задать перечнем (красный, желтый, белый, зеленый,...). Совокупность атрибутов вместе с их множествами задания определяют объект как нечто целое в виде отношения:

При этом очевидно, что атрибуты составляют постоянную (статическую) часть определения объекта, а область задания - переменную (динамическую) часть. Атрибутивное задание в конкретных реализациях чаще всего осуществляется путем позиционирования атрибутов, т.е. фиксации места каждого атрибута в общем описании:

поз. 1 поз. 2 поз. 3 поз. 4

*R (ЯБЛОКО) = (ЦВЕТ СОРТ ВКУС ВЕС)*

*R* (имя объекта) = (),),...).

Такое представление задает как бы свертку *n* унарных отношений в одном. Это чрезвычайно экономит память для сложных описаний и существенно повышает быстродействие. Имя атрибута при этом понимается по умолчанию, на местах значений атрибутов указываются их конкретные значения. Другими словами, мы определяем описание каждого объекта или вершины СС как отношение с перечнем атрибутов и областями их определения. Если обозначить область определения каждого атрибута как DOM (DOMain - область определения), то можно написать:

где - имя объекта; - имена атрибутов.

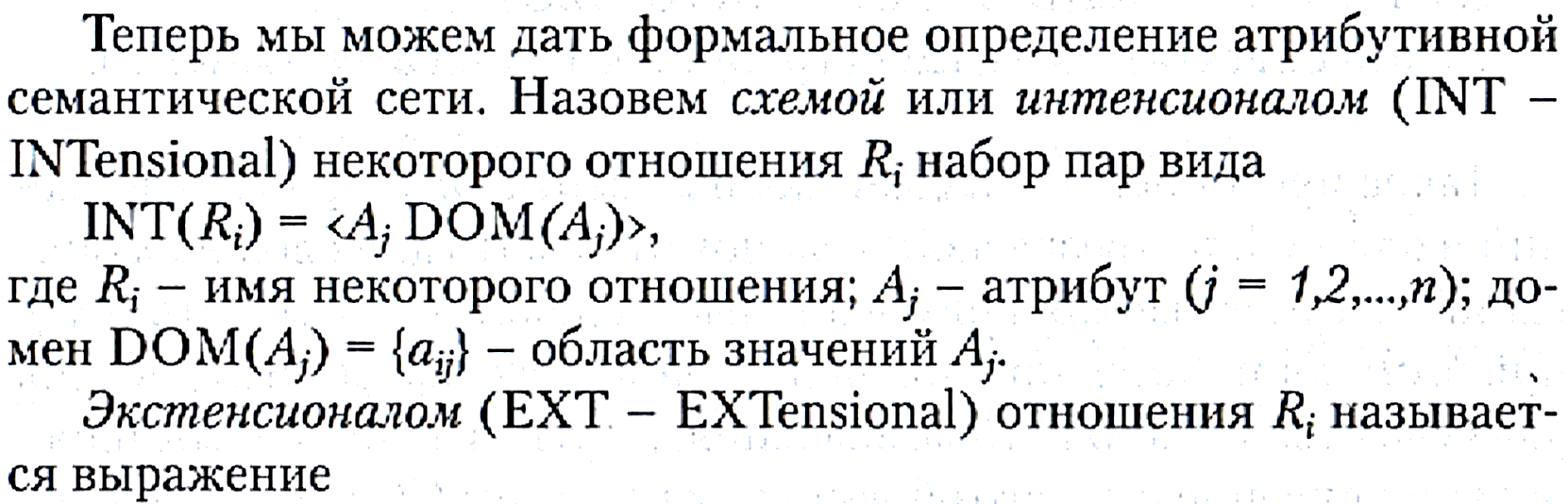
На основе такого представления можно построить множество конкретных описаний. Будем называть каждое такое описание фактом (или примером). В нашем примере мы можем определить, например, факты:

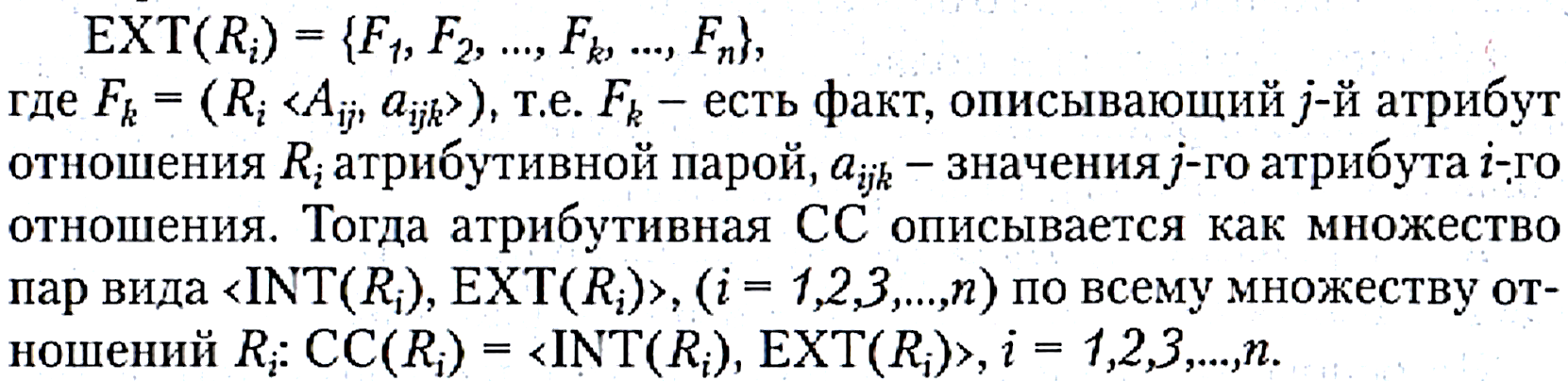
;

.

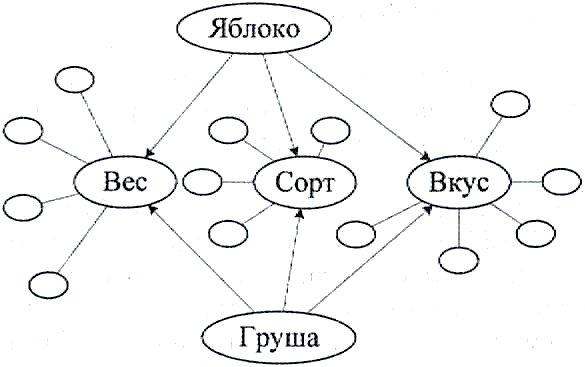
Максимальное число фактов при трех значениях атрибута *ЦВЕТ* и двух - атрибута *ВКУС -* очевидно равно 6.

задает статическую, или *интенсиональную (внутреннюю)* составляющую описания объекта; конкретные факты *-* динамическую, т.е. *экстенсиональную (внешнюю)* составляющую описания.





Графически атрибутивную СС можно представить в виде звездчатых графов (рис. 7).



*Рис. 7.* Пример графа атрибутивной СС (стрелки означают отношение ИМЕЕТ)

Вывод на семантических сетях

Многообразие СС определяет сложность процессов вывода решений, поскольку каждому типу СС (однородная, иерархическая, сценарная, функциональная и т. п.) соответствует своя модель вывода в зависимости от свойств используемых отношений (рефлексивность, симметричность, транзитивность). В практических реализациях наиболее часто используется механизм вывода, основанный на наследовании свойств от класса объектов к его конкретным представителям (класс - подкласс, класс - экземпляр класса). Более строго, наследование - это способ, которым происходит передача значений свойств объектов друг другу.

Механизм наследования реализуется на иерархии понятий, выстроенной по отношениям IsA и АКО, или Part-of-while (часть - целое). Иерархия показывает отношение включения понятий более низкого уровня в более высокий. При этом совокупность понятий, вносимых в конкретную СС, зависит от целевого назначения системы и целей ее конкретного использования. Выделяются объекты: классы, подклассы, экземпляры, свойства экземпляров, свойства значений экземпляров.

Класс - это множество объектов, имеющих несколько или большинство общих свойств, т.е. класс является обобщением или концептом (понятием). Класс хранит информацию, присущую всем его объектам. Объекты, когда нужно, могут наследовать от класса эту информацию.

Подкласс - это класс объектов, составляющих подмножество класса более высокого уровня. Например, класс «пассажирский самолет» является подклассом класса «самолет», а «самолет» - подклассом класса «летательные аппараты».

Классы и подклассы также состоят из объектов - экземпляров, наследующих свойства объектов уровнем выше.

Рисунок 8 описывает иерархию отношений между классом *ЯБЛОКО* и его подклассами *БЕЛЬ, АНИС, АПОРТ* и отношения подкласс - экземпляр класса.

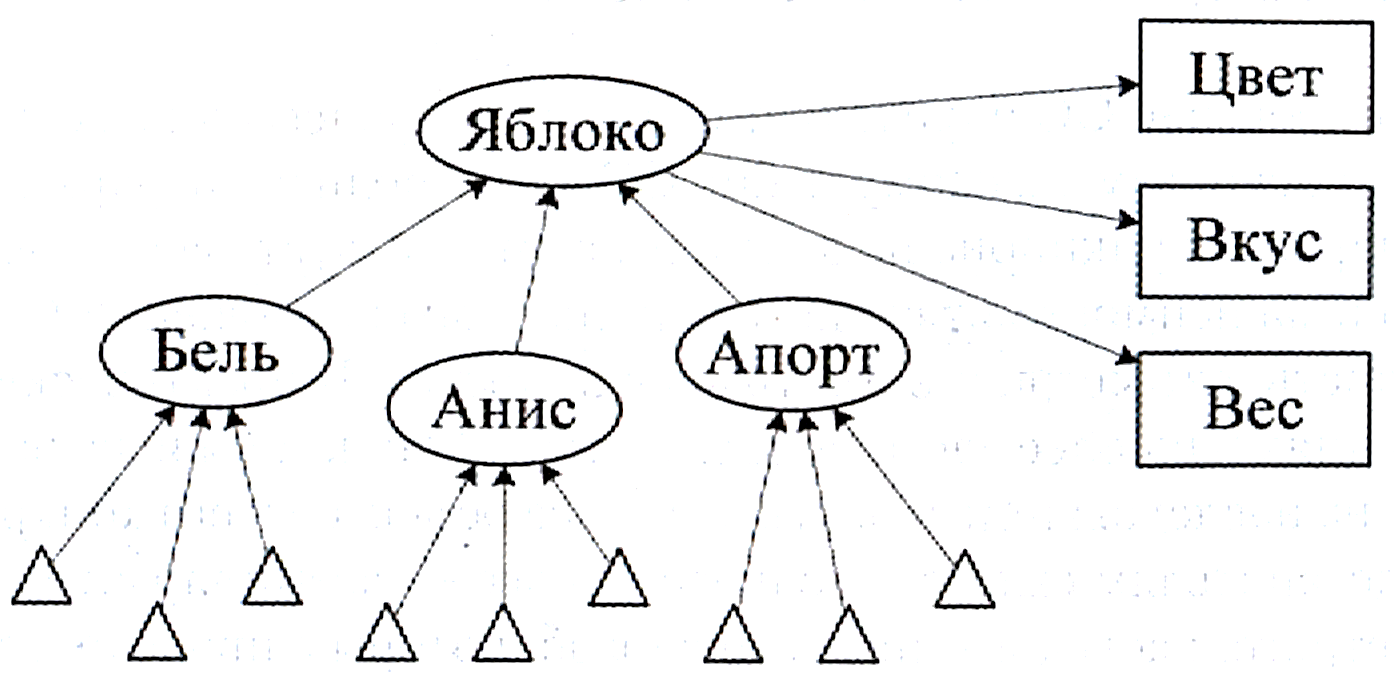


Рис. 8. Пример семантической сети

Наследование свойств означает, что свойства, приписываемые объекту (классу) высокого уровня, автоматически присваиваются объектами более низкого уровня по умолчанию. Но тут есть особенность. Рассмотрим процесс подробнее.

1. Отношение *класс - подкласс.* Это отношение специализации.

Здесь свойства и значения свойств наследуются по умолчанию по связям специализации (АКО). Пример: *САМОЛЕТ - Ту-104 (Ту-104* наследует многие свойства класса *САМОЛЕТ).* Обратное отношение (Part-of) - отношение обобщения, т.е. подкласс специализирует, а класс - обобщает.

1. Отношение *класс - объект.* Это - отношение IsA. Свойства наследуются вниз по умолчанию. Это отношение классификации. Обратное отношение - отношение интеграции.
2. Отношение *объект (экземпляр) - подобъект (свойства).* Это отношение декомпозиции. Объект разлагается на составляющие. Обратите внимание - свойства по этой связи не наследуются! В самом деле, если на линии IsA *МЛЕКОПИТАЮЩИЕ - КОШКИ, КОШКИ -* экземпляр, то дальнейшее движение по этой линии уже приводит к разделению объекта *КОШКИ* на составляющие: *ФОРМА УШЕЙ, ХВОСТ, ЛАПЫ и* т.д. Свойства класса *МЛЕКОПИТАЮЩИЕ* здесь уже не наследуются. Обратное отношение - агрегирование. Что еще интересно: если по линии IsA число экземпляров может варьироваться и не влияет на целостность класса, то в обратном отношении Part-of должны присутствовать все части или какой-то обязательный минимум этих частей. Как уже говорилось, одним из наиболее распространенных методов вывода на сетях является метод сопоставления частей сетевой структуры.

Рассмотрим пример сопоставления.

Имеется участок сети базы данных относительно класса *САМОЛЕТЫ* (рис. 9).



Рис. 9. Участок СС САМОЛЕТЫ

Требуется для начала ответить на вопрос: «Что имеет Як-40?». Участок сети, соответствующий этому вопросу, показан на рис. 10. Для ответа на этот вопрос проводится сопоставление. Сначала ищется вершина *ИМЕТЬ* такая, которая имеет соединение с *Як-40.* Теперь три левые вершины совпали, и по линии «Что имеет» мы делаем вывод: «Як-40 имеет двигатель Д1».

Рассмотрим процесс вывода подробнее. Пусть требуется ответить на вопрос: «Существует ли самолет, имеющий двигатель типа Д1?». Сеть, соответствующая этому вопросу, показана на рис.11. Однако в таком виде она не годится для сопоставления с базой данных. Если же ввести вершину *Як-40* и линию, показывающую, что *Як-40 - ПАССАЖИРСКИЙ САМОЛЕТ* то уже можно сопоставить вершину *ИМЕТЬ* с вершиной *ИМЕЕТ* а вершину *ДВИГАТЕЛЬ* с *ДВИГАТЕЛЬ типа Д1?.* Отсюда следует вывод: «Да, это Як-40».

В процессе вывода модификация базы знаний на СС сводится к удалению и добавлению новых вершин и ребер. Базовые операции поиска информации в сети обеспечивают поиск вершины или ребра по имени, переходы от одной вершины к другой по связям и от одной связи к другой через смежные вершины. Цель поиска - получение знаний, представленных в сети и требуемых для решения задачи.

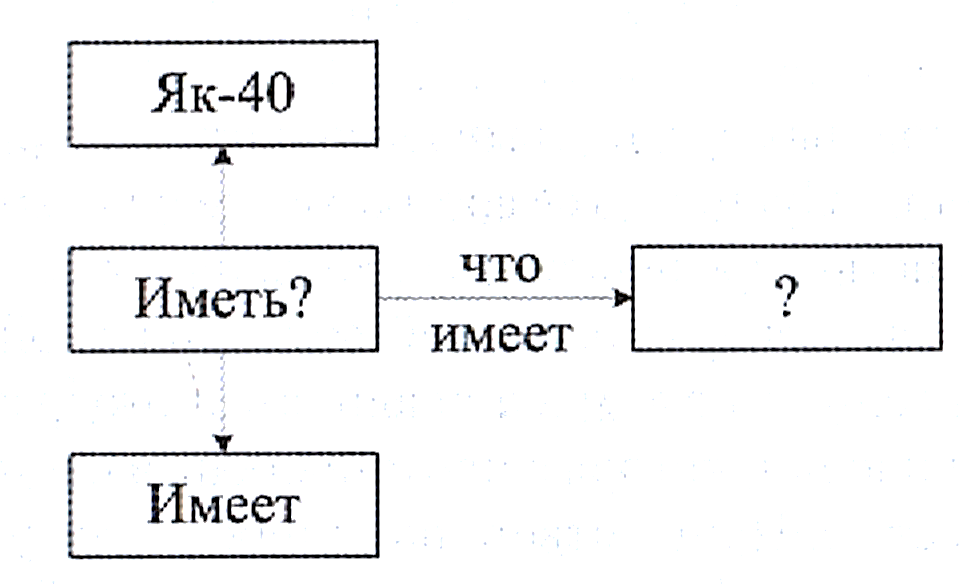


Рис. 10. Участок сети «Что имеет Як-40?»

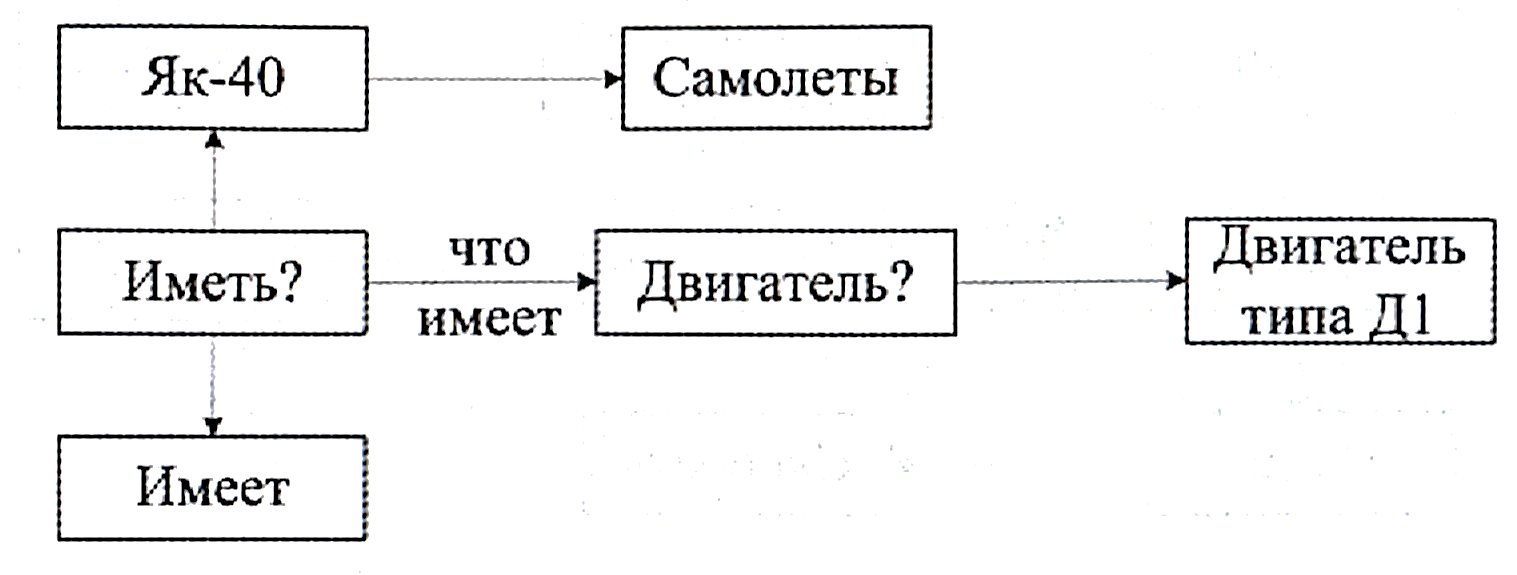


Рис. 11. Подсеть для вопроса «Существует ли самолет,

имеющий двигатель типа Д1?»

Одним из мощных средств вывода является операция сопоставления с образцом. Сущность ее состоит в следующем. Информационная потребность, определяющая содержание и цель запроса к базе знаний, описывается автономной семантической сетью -сетью запроса, построенной по тем же правилам и отображающей те же объекты и отношения, которые представлены в системе знаний семантической сетью. Поиск ответа на запрос реализуется сопоставлением сети запроса с фрагментами семантической сети, т.е. путём установления отношения эквивалентности между СС запроса и СС фрагмента общей СС, представляющей систему знаний. Положительный результат отождествления позволяет получить один из ответов на запрос. Все ответы можно получить путем обнаружения всех сопоставимых с сетью запроса фрагментов.

В общем виде операцию сопоставления для СС можно описать следующим образом. Для сети, представляющей систему знаний, задается набор допустимых преобразований, переводящий исходную сеть (или ее фрагменты) в логически ей эквивалентную. Операция сопоставления выявляет все фрагменты исходной или эквивалентных ей сетей, изоморфные сети запроса. Набор допустимых преобразований для семантических сетей дополняет сети новыми связями, полученными из транзитивности фундаментальных отношений и наследования свойств, но не ограничивается этим. В зависимости от специфики решаемых задач и особенностей того или иного конкретного средства набор эквивалентных преобразований может существенно расширяться. Поскольку теория СС не дает универсальных средств, позволяющих описывать допустимые преобразования, операция сопоставления может рассматриваться как базовая лишь методически (в том смысле, что поиск в СС всегда есть какое-то сопоставление), но не технически (в том смысле, что любой требуемый поиск может быть выражен операцией сопоставления в некотором универсальном языке представления знаний). Именно поэтому базовыми для СС здесь названы не операции сопоставления с образцом, а гораздо более примитивные операции перехода по сети. Фактически совокупность последних в каждом конкретном случае поиска реализует то или иное требуемое сопоставление.

**Достоинства и недостатки семантических сетей**

1. Простота и прозрачность описания. Однако с увеличением размеров сети существенно увеличивается время поиска, теряется наглядность.

2. Открытость, позволяющая дополнять и модифицировать СС при необходимости.

3. По сравнению с логикой предикатов СС имеют важное преимущество: вся точно известная информация расположена в базе знаний вокруг соответствующей вершины, т.е. фокусируется в одном месте.

Это все были достоинства. В качестве недостатков отметим.

1. Неоднозначность описания.

2. Отсутствие формального аппарата установления противоречивости описания.

3. Сложность внесения изменений.

Основное применение семантические сети находят в системах обработки естественных языков, а также в системах распознавания образов, в которых СС используются для хранения знаний о структуре, форме и свойствах физических объектов. Могут создаваться СС и более сложной структуры, как мы увидим далее.