

**семантическая сеть** - сеть, в вершинах которой находятся информационные единицы, а дуги характеризуют отношения и связи между ними. **семантическая сеть** является наиболее общей моделью представления знаний.

В зависимости от видов связей между информационными единицами **семантические сети** подразделяются на следующие виды:

1. однородные семантические сети – все **отношения** между информационными единицами одинаковы;
2. неоднородные семантические сети – **отношения** между информационными единицами различны.

Базовыми понятиями для формализации знаний с помощью **SC-кода** являются:

1. **семантические сети** (их удобно хранить и обрабатывать - синтаксический аспект)
2. **теория множеств** (обеспечивает строгость и однозначность формализма и универсальность представления различных видов знаний – семантический аспект).

Согласно статье «Предварительное рассмотрение логической конструкции электронного вычислительного устройства», написанной Дж. фон Нейманом, ЭВМ, построенная по архитектуре фон Неймана, обладает следующими свойствами:

1. память ЭВМ однородна (линейна);
2. доступ к ячейкам памяти ЭВМ осуществляется по адресу;
3. алфавитом, используемым для представления данных и команд, является множество  $\{0, 1\}$ ;
4. каждая программа состоит из набора команд, выполняемых процессором последовательно;
5. возможно присутствие в программах команд условных переходов.

Архитектуре семантического компьютера (SC – Semantic Computer), на который непосредственно и ориентирован **SC-код**, присущи следующие характеристики:

1. память нелинейна;
2. доступ к ячейкам памяти ассоциативный (доступ осуществляется по связям);
3. алфавит состоит из пяти элементов, однако основными являются только два:
  - a. **sc-узел**;
  - b. **sc-дуга основного вида** (sc-дуга константной стационарной принадлежности).

Для адекватного восприятия информации, представленной далее, необходимо владеть терминологией из **Теории множеств** и **Теории отношений**.

Для графического представления текстов **SC-кода** может использоваться **язык SCg** (Semantic Computer graphic Code). Основным принципом, заложенным в основу **SCg-кода** является то, что каждому **sc-элементу** в соответствие ставится sc.g-элемент (графическое отображение), т.е. любому изображаемому на экране sc.g-элементу соответствует некоторый **sc-элемент в базе знаний**. Далее для изображения примеров формализации будем использовать именно **SCg-код**.

Рассмотрим простое математическое выражение:

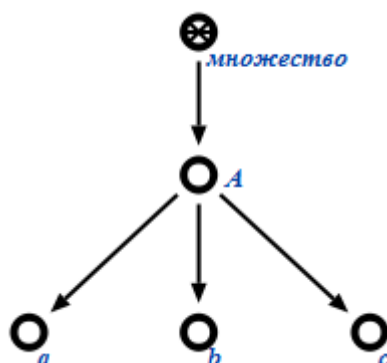
$$A = \{a, b, c\}.$$

На естественном языке это выражение будет выглядеть следующим образом: элементами **множества** A являются **сущности** a, b и c.

Рассмотрим некоторые элементы *алфавита языка SCg*, необходимые для записи выбранного математического выражения:

1. *узел*  $\bigcirc$  обозначает некоторую *сущность*, хранящуюся в *базе знаний*;
2. *узел*  $\bigotimes$  обозначает некоторый *класс сущностей*, хранимых в *базе знаний*;
3. *дуга*  $\rightarrow$  обозначает *принадлежность сущности* некоторому *множеству*, в том числе, - *классу*.

Таким образом, на *языке SCg* рассматриваемое математическое выражение примет следующий вид:



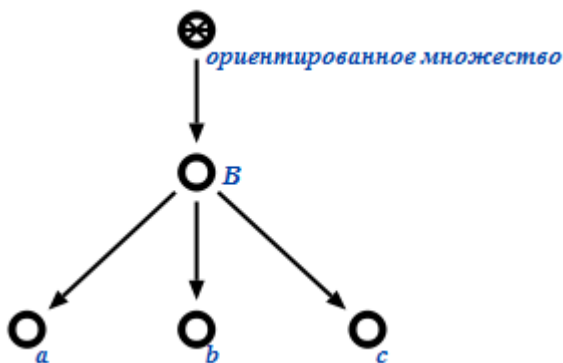
На представленном фрагменте видно, что *A* является *множеством* (является элементом класса «*множество*»), *сущности* *a*, *b* и *c* являются элементами *множества A*.

Рассмотрим более сложное математическое выражение:

$$B = \langle a, b, c \rangle.$$

На естественном языке его можно записать следующим образом: *сущности* *a*, *b* и *c* являются элементами *ориентированного множества B*.

В данном случае недостаточно просто указать *принадлежность* элементов *множеству*:

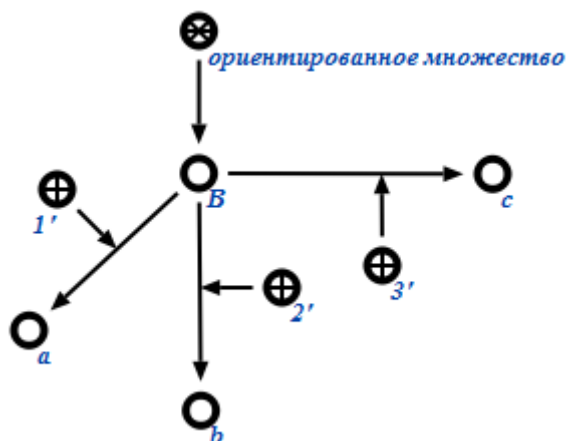


Необходимо каким-то образом зафиксировать порядок элементов *множества*, для этого следует выделить роли каждой *сущности* в рамках представленного *множества*.

На естественном языке преобразованное математическое выражение может выглядеть следующим образом: *сущности* *a*, *b* и *c* являются элементами *ориентированного множества B*, причём *сущность* *a* является первым элементом, *сущность* *b* – вторым, а *сущность* *c* – третьим.

Для представления ролей *сущностей* в рамках некоторого *множества* в языке SCg используется символ  $\oplus$ , обозначающий *ролевое отношение*, которое, в свою очередь, является множеством всех связок принадлежности по указанному атрибуту (подмножеством *отношения принадлежности*).

Таким образом, преобразованное представление рассмотренного математического выражения на языке SCg будет иметь следующий вид:



Рассмотрим следующее *арифметическое выражение*:

$$2^3 = 8.$$

На естественном языке это *выражение* можно записать следующим образом: результатом возведения *числа* 2 в третью степень является *число* 8.

В этом выражении можно выделить следующие значимые единицы:

1. есть числа 2, 3, 8;
2. над этими числами выполняется *арифметическая операция* возведения в степень;
3. основанием степени является *число* 2;
4. показателем степени является *число* 3;
5. результатом выполнения операции является *число* 8.

*Операция* возведения в степень является ничем иным, как *отношением*, заданным на *множестве чисел*. *Связками* этого *отношения* являются всевозможные *тройки* чисел  $\langle a, b, c \rangle$ , удовлетворяющие следующему правилу:  $a^b = c$ , т.е. *связка отношения* возведение в степень для рассмотренного выше *выражения* будет иметь вид:

$$\{ \langle 2, 3, 8 \rangle \},$$

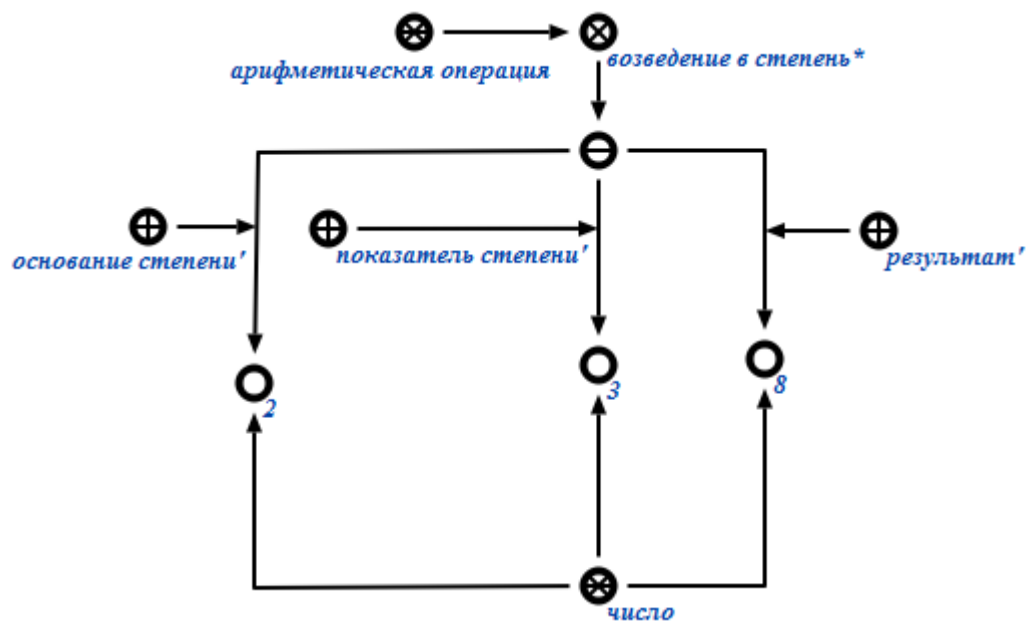
а само *отношение* возведение в степень будет иметь вид:

$$\text{возведение в степень} = \{ \{ \langle 2, 3, 8 \rangle \} \}.$$

*Отношение* возведение в степень является *неролевым*, так как связывает между собой несколько независимых *сущностей*, а не указывает *роль* одной *сущности* в рамках другой, как рассмотренное выше *ролевое отношение*.

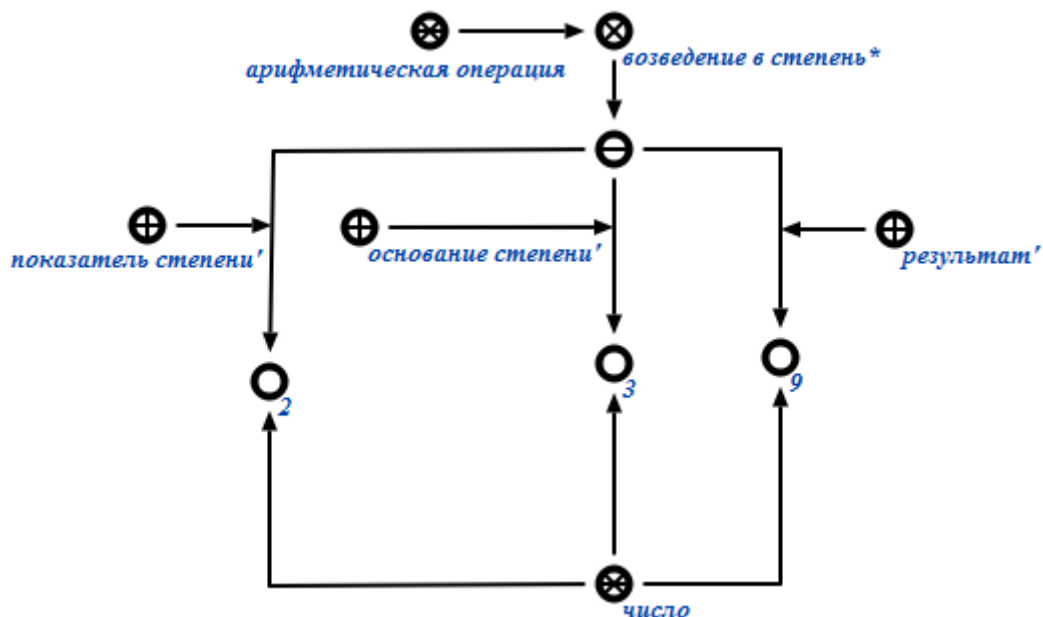
Для обозначения *неролевых отношений* в языке SCg используется символ  $\otimes$ , а для обозначения *связок* таких отношений - символ  $\ominus$ .

Представим рассмотренное выше *выражение* на языке SCg:



Указание *принадлежности операции* возведение в степень\* множеству *арифметических операций* не обязательно в данном контексте.

Если поменять местами *роли* чисел 2 и 3, получится другая связка *отношения* возведения в степень, результатом которой будет уже *число* 9, а не 8:



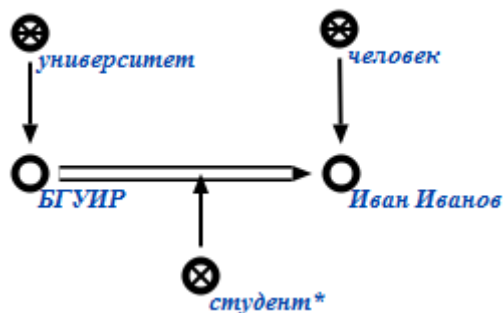
Таким образом, можно сделать вывод, что *атрибутивные отношения* однозначно определяют *роли сущностей* в рамках некоторого *множества* и непосредственно влияют на смысл записанного фрагмента *знаний*.

Для однозначного определения, какому *классу отношений* (*ролевых* или *неролевых*) принадлежит то или иное рассматриваемое *отношение*, в *SC-коде* необходимо добавлять соответствующий знак к *идентификатору отношения*: \* - для *неролевых*, ' - для *ролевых*.

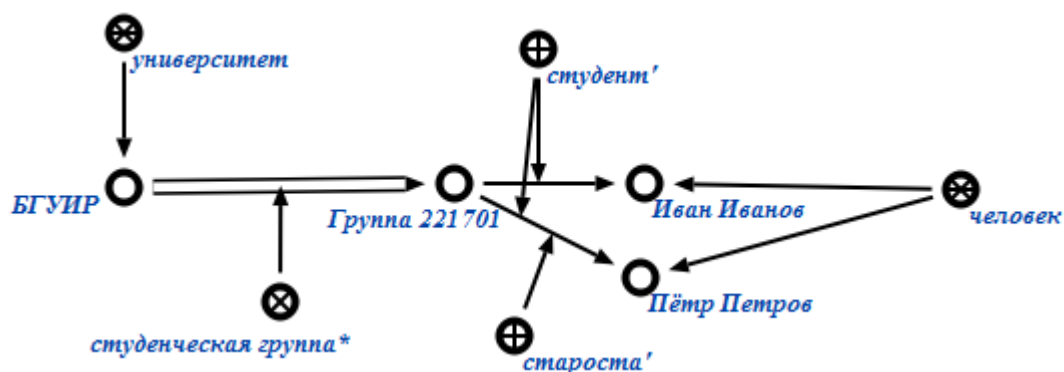
Для *идентификаторов классов сущностей* также существует правило – *идентификаторы классов* пишутся строчными буквами без каких-либо завершающих специальных символов.

Рассмотрим пример, когда одно и то же *отношение* может быть как *ролевым*, так и *неролевым* в зависимости от контекста:

Возьмём *отношение* студент (быть студентом).



В данном случае Иван Иванов является студентом по отношению к БГУИР, но не является его элементом.

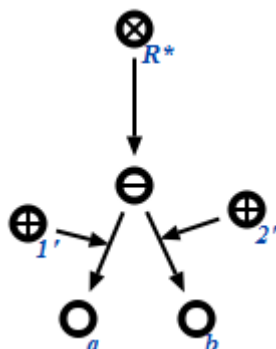


В этом случае Иван Иванов и Пётр Петров являются студентами группы 221701, т.е. они являются элементами *множества* «студенческая группа» и выполняют *роль* «студент» в рамках этого *множества*. Пётр Петров также является старостой группы 221701, что иллюстрирует возможность выполнения *сущностью* нескольких *ролей* в рамках одного и того же *множества*.

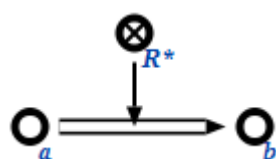
Наиболее часто используемыми *отношениями* при формализации *знаний* являются *бинарные отношения*. Использование преимущественно *бинарных отношений* также позволяет снизить количество *ключевых узлов* в *памяти интеллектуальной системы*. Все остальные *отношения* рекомендуется сводить к *бинарным*, если это возможно. Например, рассмотренное выше *отношение возведение в степень\** также сводимо к *бинарному отношению*. Выполнение такого приведения подробнее рассмотрено во введении к разделу *Раздел. Предметная область чисел и числовых структур*.

Рассмотрим *бинарное отношение*  $R^* = \{ \langle a, b \rangle, \langle b, c \rangle \}$ .

Представим одну из его связок на *языке SCg* способом, рассмотренным выше:



Такое представление является громоздким и обязывает вводить дополнительные *атрибуты* для определения порядка компонентов, участвующих в *отношении*. Язык *SCg* позволяет записать этот же *sc-текст* более компактно, используя *sc-дугу общего вида*  $\Rightarrow$ :



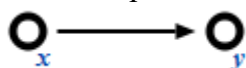
Направление *sc-дуги общего вида* выбирается следующим образом: *sc-дуга общего вида* выходит из *узла*, обозначающего ту *сущность*, которая является первым компонентом в *связке отношения*, т.е. ту *сущность*, которая имеет *атрибут 1'* при развёрнутом описании *отношения*, и входит в *узел*, обозначающий второй компонент *связки отношения*.

Такая краткая запись *связок отношения* уместна только для *бинарных* и *квазибинарных отношений*, т.е. *отношения*, *арность* которых превышает 2 и не может быть сведена к 2, описываются исключительно в развёрнутом виде.

Краткая запись *связок отношения* также не подходит для описания *бинарных отношений*, *роли* компонентов *связок* в которых отличаются от *1'* и *2'*, так как при сворачивании *связки отношения* в *sc-дугу общего вида* *роли* компонентов *связки* отбрасываются, а следовательно, не могут быть восстановлены при последующем описании *связок* в развёрнутом виде.

При использовании краткой записи *бинарного отношения* обязательно должно быть указано, какому *отношению* принадлежит та или иная *связка*, в отличие от *отношения принадлежности* и его *подмножеств*.

Таким образом, запись вида:



является законченной и может быть проинтерпретирована в законченное математическое выражение:  $y \in x$ , так как каждая *sc-дуга основного вида* обозначает *связку отношения принадлежности*, т.е. запись, представленная выше, несёт в себе следующий смысл:

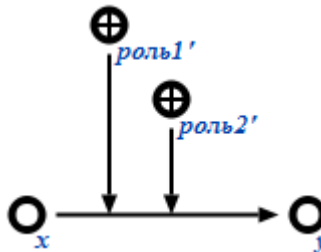


В то время, как запись вида:

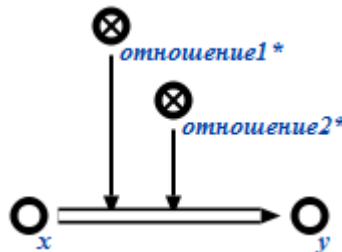


не является законченной и не может быть однозначно проинтерпретирована, так как неизвестно, какому именно *отношению* принадлежит *связка*, обозначаемая *sc-дугой* *общего вида*.

*Связки ролевого отношения* и *связки неролевого отношения* могут одновременно принадлежать не только одному *множеству*, т.е. записи вида:



и



являются вполне допустимыми при формализации каких-либо *знаний* средствами *SC-кода*. Однако вариант со *связками неролевого отношения* редко употребим и не является рекомендуемым.

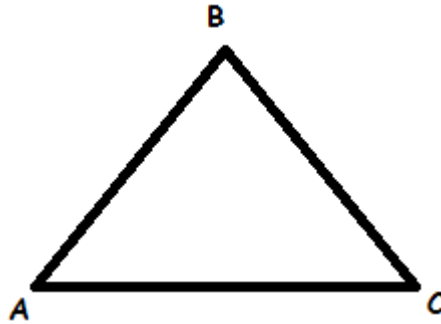
При формализации некоторого фрагмента знаний и выделении *понятий* необходимо определить следующее:

1. является ли описываемое *понятие* абсолютным, т.е. существует ли оно независимо от других понятий;
2. является ли описываемое *понятие* относительным, т.е. существует только в связи с другими понятиями:
  - а. является ли описываемое относительное понятие *ролевым отношением*;
  - б. является ли описываемое относительное понятие *неролевым отношением*:
    - i. какова арность описываемого *неролевого отношения*.

Ответы на эти вопросы помогут выбрать обозначение для вводимого *понятия* и правильно организовать его взаимоотношения с другими.

Рассмотрим ещё один пример и его формализацию:

Пусть дан рисунок:



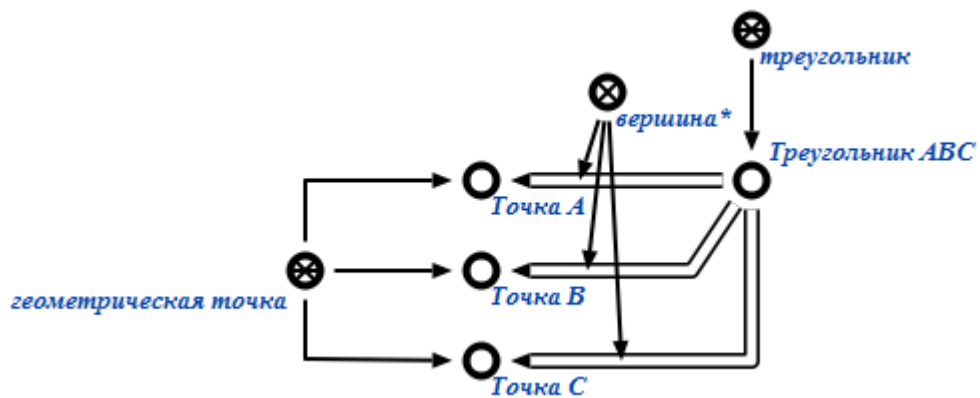
Представленную на этом рисунке информацию на естественном языке можно записать следующим образом: дан треугольник ABC, вершинами которого являются точки A, B и C, а сторонами – отрезки AB, BC и AC.

Запишем это на **языке SCg** по шагам:

- 1) Треугольник ABC является элементом множества всех треугольников:



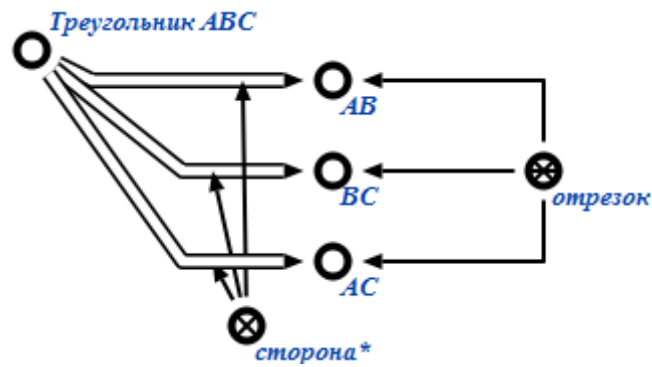
- 2) Геометрические точки A, B и C являются вершинами треугольника ABC:



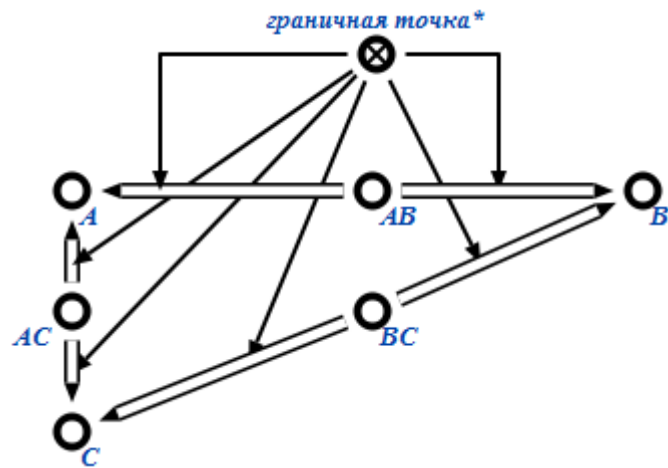
Направление *sc-дуги общего вида* в данном случае выбрано по принципу: «от объекта к его свойству».

- 3) Отрезки AB, BC и AC являются сторонами треугольника ABC:





- 4) Также необходимо указать, что точки А, В и С являются граничными точками для отрезков АВ, ВС и АС:



Нет необходимости повторно указывать *принадлежность сущностей* соответствующим *классам*, так как в *языке SC* и его подязыках *сущности*, имеющие одинаковые *имена*, считаются одной и той же *сущностью*.

Таким образом, окончательная запись формализации рассмотренного примера будет иметь вид:

