

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ЭЛЕКТРОННОГО  
СРЕДСТВА УЧЕБНОГО НАЗНАЧЕНИЯ  
«ИНТЕЛЛЕКТУАЛЬНАЯ СПРАВОЧНАЯ СИСТЕМА ПО  
ПЛАНИМЕТРИИ»**

**МИНСК 2014**

# 1 Назначение системы

Интеллектуальная справочная система по геометрии предназначена для информационного обслуживания пользователя, путём ответа на вопросы предметной области планиметрии и осуществления решения задач из этой же области.

Данная система предназначена как для школьников, учащихся СУЗов, ВУЗов, преподавателей геометрии, так и для людей, интересующихся геометрией.

При изучении геометрии учащиеся могут испытывать трудности. Причём зачастую с этими трудностями учащиеся сталкиваются дома при выполнении домашних заданий, подготовке к контрольным работам. Им необходима консультация, помощь в решении задач, объяснение способа решения задачи проверка своих знаний по пройденному материалу.

Преподаватели могут использовать данную систему в качестве вспомогательного материала на уроках (занятиях и т.д).

Основой для реализации системы по геометрии является открытая семантическая технология проектирования интеллектуальных систем OSTIS. Больше информации о технологии можно найти на сайте проекта OSTIS <http://ostis.net> и сайте интеллектуальной Метасистемы OSTIS <http://ims.ostis.net>.

## **2 Основные возможности системы**

К основным возможностям системы относятся следующие:

- Осуществление навигации по базе знаний системы путем:
  - осуществления перехода от одной семантической окрестности элемента семантической сети к другой по ссылке
  - использования поисковых операций в пункте меню графического интерфейса системы
- Осуществление решения задач из предметной области геометрии на основе имеющихся у системы знаний, включая объяснение способа решения
- Предоставление возможности создания геометрических чертежей, понимаемых системой при помощи специализированного редактора.

### 3 Пользовательская настройка системы

#### 1. Системные требования

Система поставляется как web-приложение, которое может быть развернуто как на локальном сервере, так и на удаленном ресурсе.

Требования к серверной части:

	Минимальные	Рекомендуемые
Программные требования	ОС: ОС на базе Linux	ОС: Ubuntu 14.04 LTS
Аппаратные требования	Процессор: 1 ядро 385 МГц ОЗУ: Свободное место на HDD: 10 Гб Доступ к Internet	Процессор: 2 ядра по 1 ГГц ОЗУ: Свободное место на HDD: 20 Гб Доступ к Internet

Требования к клиентской части:

Web-браузер с поддержкой последних стандартов HTML и JavaScript, например Google Chrome

#### 2. Установка системы

Для установки системы на сервере должна быть предустановлена операционная система, указанная в системных требованиях.

Далее для работы необходимо установить систему контроля версий Git. Это можно сделать при помощи команды

```
sudo apt-get install git
```

После этого необходимо загрузить установочные скрипты в какую-либо папку на сервере при помощи команды

```
git clone https://github.com/geometryostis/geometry.ostis.git
```

Для установки и сборки системы достаточно перейти в нужный каталог и выполнить скрипт сборки:

```
cd geometry.ostis/scripts
```

```
./prepare.sh
```

Далее при установке нужно руководствоваться сообщениями, которые скрипт будет выводить в терминал.

После успешной установки системы для ее запуска необходимо выполнить следующие команды в том же каталоге

```
./build_kb.sh
```

```
./run_sctp.sh &
```

```
./run_sweb.sh &
```

Для остановки системы нужно выполнить команды

```
pkill python
```

```
pkill sctp-server
```

## 4 Внутреннее представление знаний в системе (SC-код)

Достоинствами SC-кода являются следующие его свойства:

- Все основные семантические связи между текстами (семантическая эквивалентность, семантическое включение, семантическое пересечение) в SC-коде становятся теоретико-множественными (равенство, включение, пересечение множеств).
- Неограниченная возможность перехода от sc-текстов к sc-метатекстам, содержащим знаки описываемых sc-текстов.
- Тексты SC-кода (в том числе и те тексты, sc-знаки которых явно вводятся в рамках соответствующих им метатекстов) могут быть иерархическими структурами, имеющими любое число уровней иерархии, поскольку sc-элемент может обозначать множество, состоящее из любых sc-элементов (в т.ч. и из sc-элементов, обозначающих любые множества других sc-элементов). В отличие от этого, например, такие традиционные структуры, как алгебраические системы, являются трехуровневыми:
  - На первом уровне – элементы носителя (основного множества) алгебраической системы;
  - На втором уровне – кортежи, элементами которых являются элементы носителя;
  - На третьем уровне – отношения, элементами которых являются указанные кортежи.

Формализация знаний на основе SC-кода предполагает теоретико-множественную интерпретацию всех sc-элементов, не являющихся знаками внешних сущностей. Все такие sc-элементы являются знаками множеств sc-элементов и необходимо, прежде всего, уточнять то, какие sc-элементы являются знаками этих множеств. При этом совсем не обязательно, чтобы все эти sc-элементы были представлены в текущем состоянии sc-памяти.

Тексты SC-кода будем называть sc-текстами. Знаки, входящие в состав sc-текстов, будем называть sc-элементами. Переход от общего понятия семантической сети к унифицированным семантическим сетям (sc-текстам) рассмотрим как задание целого ряда ограничений на семантические сети общего вида, но таких ограничений, которые не снижают семантической мощности языка семантических сетей, претендующего на универсальность.

**Ограничение 1.** Семантическая нормализация описываемых множеств. Если элемент семантической сети (sc-элемент) является знаком некоторого множества, то каждый элемент этого множества, представляющий собой сущность, не являющуюся знаком (sc-элементом), заменяется на знак указанной сущности (на sc-элемент, обозначающий эту сущность). Таким образом, все описываемые sc-текстом множества становятся множеством знаков, а точнее, множеством sc-элементов (sc-множествами). Такие множества условно будем называть семантически нормализованными. Очевидно, любое множество может быть представлено в семантически нормализованном виде. Для описания связей между sc-множествами и элементами этих множеств вводятся связи принадлежности. Каждая из этих связей связывает знак некоторого sc-множества с одним из элементов этого множества, который всегда является знаком (sc-элементом) благодаря семантической нормализации sc-множества.

**Ограничение 2.** Каждая связь описываемых сущностей трактуется как множество, элементами которого являются описываемые сущности, связываемые этой связью. Семантическая нормализация каждого такого множества означает то, что все

связи, входящие в состав семантической сети, будут связывать не сами описываемые сущности, а знаки этих сущностей за исключением случаев, когда указанные сущности уже являются знаками. Напомним при этом, что знаки описываемых сущностей, связи между знаками описываемых сущностей и знаки этих связей включаются в число сущностей, описываемых семантической сетью. Подчеркнем также, что связь- $i$  между  $sc$ -элементом, который является знаком некоторой связи- $j$ , и  $sc$ -элементом, который является компонентом этой связи- $j$  (т.е. является знаком, связываемым связью- $j$ ), в  $sc$ -тексте может быть представлена двумя способами:

- с помощью явно вводимого  $sc$ -элемента, обозначающего связь принадлежности;
- с помощью синтаксически задаваемой пары инцидентности, для которой соответствующий ей  $sc$ -элемент (знак) не вводится.

**Ограничение 3.** Если в состав семантической сети входит знак небинарной связи, то связи между этим знаком и компонентами обозначаемой им связи задаются не с помощью пар инцидентности (на синтаксическом уровне), а явно с помощью явно вводимых связей принадлежности. Это означает, что в текстах  $SC$ -кода все небинарные связи представляются с помощью бинарных (сводятся к бинарным).

**Ограничение 4.** Каждая семантическая сеть и, в частности, каждая унифицированная семантическая сеть ( $sc$ -текст), которая описывается и, соответственно, обозначается в другой семантической сети (метасети), трактуется как семантически нормализованное множество, элементами которого являются все те и только те знаки, которые входят в состав описываемой семантической сети.

**Ограничение 5.** В рамках  $SC$ -кода четко задается и минимизируется алфавит элементов семантических сетей (Алфавит знаков, входящих в состав унифицированных семантических сетей, Алфавит  $sc$ -элементов). Указанный алфавит представляет собой семейство тех классов элементов унифицированных семантических сетей ( $sc$ -элементов), принадлежность  $sc$ -элементов которым задается не с помощью явно вводимых связей принадлежности, а с помощью синтаксически задаваемых меток  $sc$ -элементов. При этом каждой такой метке взаимно однозначно соответствует свой синтаксически задаваемый класс  $sc$ -элементов. Семейство таких классов, т.е. Алфавит  $sc$ -элементов, включает в себя:

- Класс  $sc$ -узлов
- Класс  $sc$ -ссылок на внешние информационные ресурсы
- Класс  $sc$ -ребер
- Класс  $sc$ -дуг общего вида
- Класс  $sc$ -дуг принадлежности или не принадлежности
- Класс стационарных константных  $sc$ -дуг принадлежности

**Ограничение 6.** В рамках  $SC$ -кода четко задаются правила перехода  $sc$ -элементов из одного синтаксически задаваемого класса  $sc$ -элементов в другой при полном сохранении его семантики (т.е. при сохранении обозначаемого им денотата). Речь идет об изменении синтаксического типа (метки)  $sc$ -элемента при появлении определенного вида новой (дополнительной) информации об этом  $sc$ -элементе. Так, например,  $sc$ -элемент может перейти:

- из Класса  $sc$ -узлов в Класс  $sc$ -ссылок;
- из Класса  $sc$ -узлов в Класс  $sc$ -ребер;
- из Класса  $sc$ -ребер в Класс  $sc$ -дуг общего вида;
- из Класса  $sc$ -дуг общего вида в Класс  $sc$ -дуг принадлежности или не принадлежности;
- из Класса  $sc$ -дуг принадлежности или не принадлежности в Класс стационарных константных  $sc$ -дуг принадлежности.

**Ограничение 7.** Важнейшая особенность любого языка, текстами которого являются семантические сети, (в том числе и  $SC$ -кода) заключается в том, что все элементы (примитивы, атомарные фрагменты) семантических сетей ( в т.ч. и  $sc$ -

элементы) являются знаками. При этом в SC-коде сущности, обозначаемые sc-элементами, делятся на два вида:

- семантически нормализованные множества, элементами которых являются sc-элементы;
- внешние описываемые сущности, не являющиеся множествами.

Таким образом, каждый sc-элемент есть либо знак множества (но не просто множества, а множества, элементами которого являются sc-элементы), либо "внешней" описываемой сущности (сущности, которая множеством не является). Особо подчеркнем то, что sc-элементов, являющихся знаками множеств sc-элементов большинство:

- каждая связь трактуется как множество sc-элементов, обозначающих связываемые сущности;
- каждая структура трактуется как множество всех sc-элементов, входящих в эту структуру;
- каждое понятие трактуется как множество sc-элементов, обозначающих сущности, являющиеся экземплярами этого понятия.

Следовательно, имеет место четкая теоретико-множественная трактовка таких сущностей, как связи, структуры и понятия. Это означает то, что SC-код имеет четкую базовую теоретико-множественную семантическую интерпретацию всех его sc-элементов – одни sc-элементы могут быть только элементами каких-либо множеств, а другие могут быть как элементами одних множеств, так и знаками других множеств. Все sc-элементы, кроме знаков связей принадлежности или не принадлежности, можно разбить на следующие уровни иерархии:

- первичные sc-элементы – знаки "внешних" сущностей;
- sc-элементы второго уровня – знаки множеств, элементами которых являются только первичные sc-элементы;
- sc-элементы третьего уровня – знаки множеств, среди элементов которых есть по крайней мере, один sc-элемент второго уровня, но нет ни одного sc-элемента более высокого уровня;
- и так – до бесконечности.

**Ограничение 8.** Все sc-элементы имеют базовую семантическую типологию, описанную в SC-тексте базовой типологии sc-элементов.

## 5 Языки общения с системой

### 5.1 SCg-код

#### 5.1.1 SCg-код и его sc-модель

SCg-код - один из возможных способов визуального представления sc-текстов (далее sc-текст отображенный с помощью SCg-кода будем называть sc.g-текстом). Основным принципом заложенным в основу SCg-кода является то, что каждому sc-элементу в соответствие ставится sc.g-элемент (графическое отображение). Другими словами любому изображаемому на экране sc.g-элементу соответствует некоторый sc-элемент в базе знаний. В рамках SCg-кода выделено несколько его уровней.




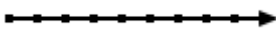

#### 5.1.2 SCg-код 1-го уровня (SCg-ядро)

SCg-ядро - это наиболее близкий способ представления sc-текстов в графическом виде. Каждому элементу алфавита SCg-кода соответствует элемент алфавита SC-кода. Основными свойствами SCg-ядра являются:

- каждый sc.g-текст записанный с помощью SCg-ядра (SCg-кода 1-го уровня), изоморфен тому sc-тексту, который он графически изображает. В этом смысле средства SCg-ядра максимально близки к SC-коду.
- в sc.g-текстах, использующих SCg-ядро, как и в sc-текстах, запрещена синонимия элементов этих текстов.

Тексты записанные с помощью SCg-ядра удобны для иллюстрации синтаксических и семантических свойств SC-кода, но не удобны для широкого практического использования в качестве языка визуализации sc-текстов, т.к. не являются наглядными. Разработка SCg-кода как раз и направлена на разрешение этого противоречия. В таблице №1 представлен алфавит SCg-ядра.

**Таблица №1. Алфавит SCg-ядра.**

Алфавит элементов SC-кода	Алфавит элементов SCg-ядра	Синтаксический тип sc.g-элемента
<i>sc-узел</i>	•	<i>sc.g-узел общего вида</i>
<i>sc-ссылка</i>		<i>sc.g-ссылка общего вида</i>
<i>sc-ребро общего вида</i>		<i>sc.g-ребро общего вида</i>
<i>sc-дуга общего вида</i>		<i>sc.g-дуга общего вида</i>
<i>sc-дуга принадлежности</i>		<i>sc.g-дуга принадлежности</i>
<i>константная позитивная стационарная sc-дуга принадлежности</i>		<i>константная позитивная стационарная sc.g-дуга принадлежности</i>

Некоторым из sc.g-элементов могут быть приписаны строковые идентификаторы, которые являются именами изображаемых sc-элементов (то как эти идентификаторы представлены в базе знаний, будет описано далее). Это нужно для того, чтобы пользователь мог читать sc.g-тексты. Указанные идентификаторы могут быть:

- глобальными (системными), которые являются абсолютно уникальными в рамках системы. В основном эти идентификаторы используются самой системой и её разработчиками;



- основными, которые являются уникальными в рамках режима диалога, ориентированного на носителя соответствующего естественного языка. Другими словами не может быть два объекта с одинаковым основным русскоязычным идентификатором\* или основным англоязычным идентификатором\* и т. д.;
- дополнительными, которые являются различными синонимами.

Идентификатор изображается рядом с идентифицируемым sc.g-элементом (на небольшом расстоянии). К примеру, узел с идентификатором *бинарное отношение* изображается следующим образом:



Рис. 5.1 - Пример изображения идентификатора

При большом количестве идентифицируемых sc.g-узлов на экране становится сложно определить к какому из них относится тот или иной идентификатор. Чтобы немного упростить эту задачу, идентификаторы sc.g-узлов могут располагаться лишь в 4 позициях относительно идентифицируемого объекта. Чтобы описать возможные позиции идентификатора, введем два прямоугольника: прямоугольник *A* - это прямоугольник, который описан вокруг идентифицируемого узла, а прямоугольник *B* - это прямоугольник, который описан вокруг идентификатора узла. Тогда расположение идентификатора относительно идентифицируемого узла может быть следующим:

- справа внизу (рекомендуемая позиция для расположения идентификаторов). При таком расположении, левый верхний угол прямоугольника Б, должен совпадать с нижним правым углом прямоугольника А;
- справа вверху (левый нижний угол прямоугольника Б совпадает с правым верхним углом прямоугольника А);
- слева вверху (правый нижний угол прямоугольника Б совпадает с левым верхним углом прямоугольника А);
- слева внизу (правый верхний угол прямоугольника Б совпадает с левым нижним углом прямоугольника А).

Более подробно возможные варианты расположения идентификаторов относительно sc.g-узлов можно посмотреть на рисунке 5.2.

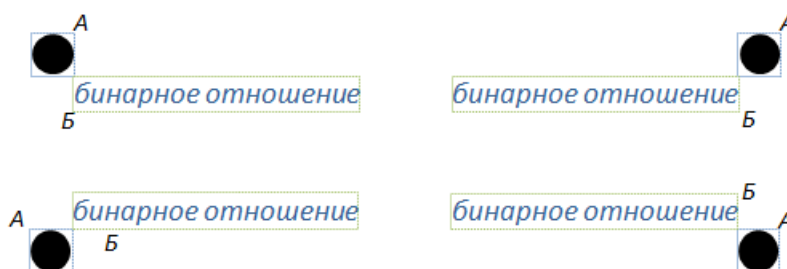


Рис 5.2 - Возможные варианты расположения идентификаторов относительно sc.g-узлов

Разрешено использовать дополнительные средства для обозначения принадлежности идентификатора некоторому sc.g-узлу. К примеру, при наведении указателя мыши на идентификатор или sc.g-узел, можно менять их цвет. Важно, чтобы эти дополнительные средства не загромождали и не усложняли, отображаемый на экране, sc.g-текст.

Рассмотрим примеры некоторых sc.g-текстов, с помощью SCg-ядра (SCg-кода 1-го уровня):

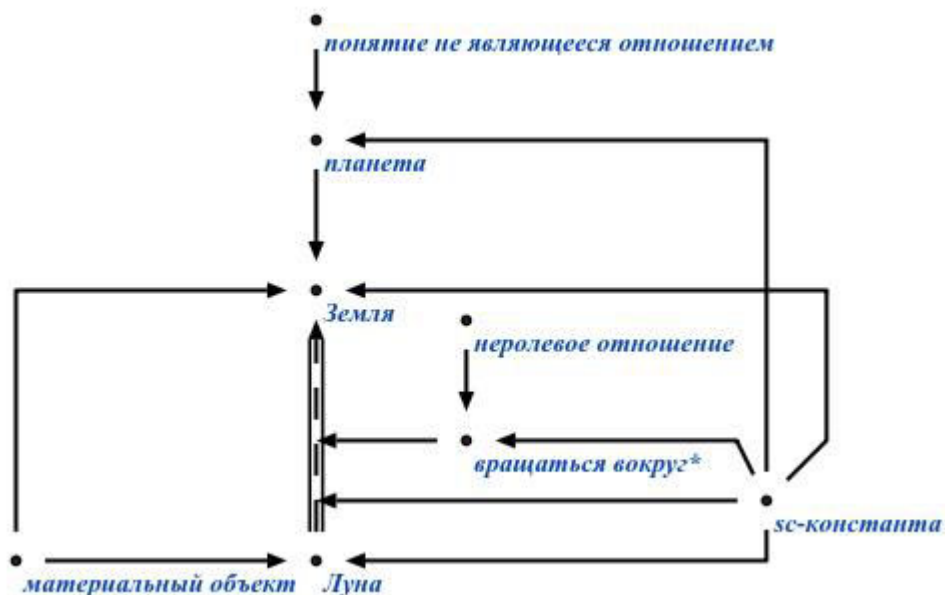


Рис 5.3 - Запись факта: “Луна вращается вокруг планеты Земля” с помощью SCg-кода 1-го уровня

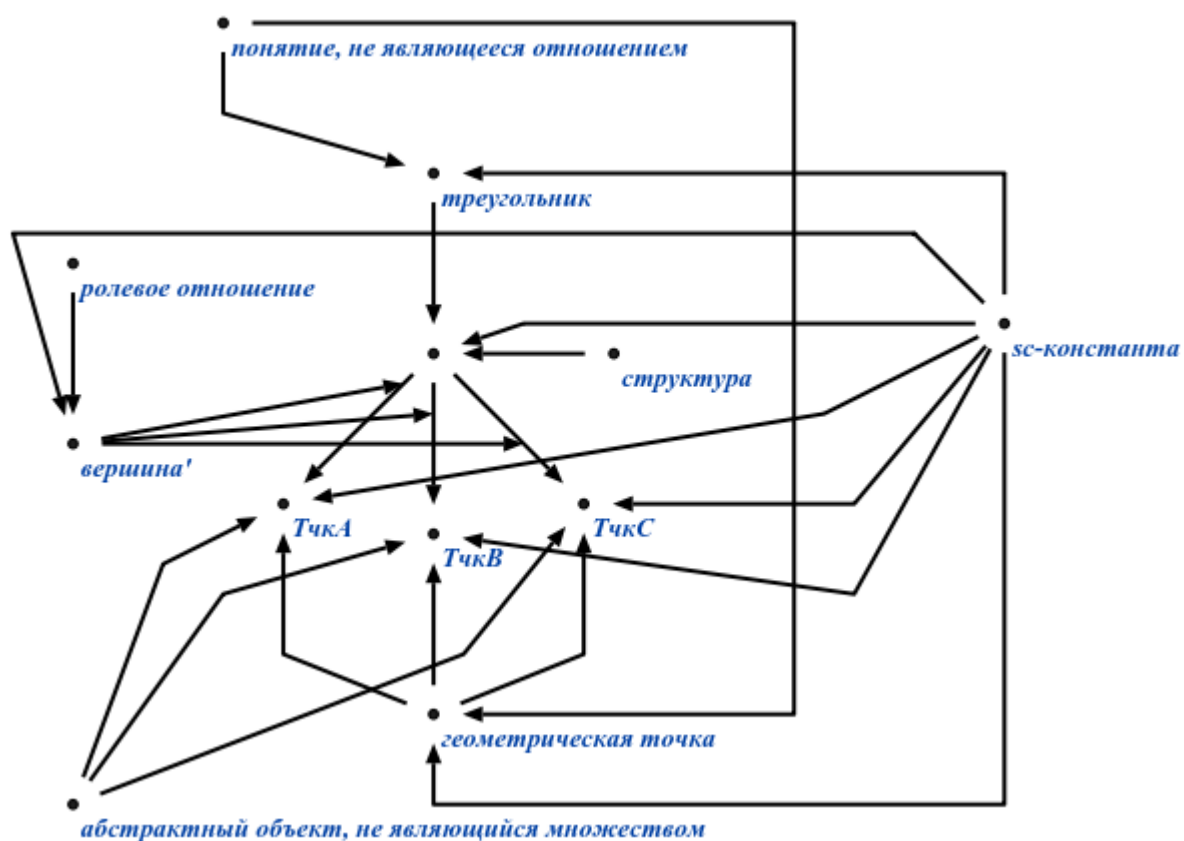


Рис 5.4 - Запись факта: “Имеется треугольник с вершинами в точках A, B и C” с помощью SCg-кода 1-го уровня

Из примеров видно, что читабельность sc.g-текстов записанных с помощью SCg-ядра не высока, отчасти это обусловлено большим количеством изображаемых sc.g-элементов. Поэтому одним из способов повышения читабельности таких текстов является уменьшение количества изображаемых объектов. Важно, чтобы при уменьшении количества объектов тексты не теряли записанную в них информацию.

### 5.1.3 SCg-код 2-го уровня

Второй уровень SCg-кода направлен на уменьшение количества отображаемых на экране sc.g-элементов для повышения наглядности (читабельности) sc.g-текстов. Основная идея, которая лежит в основе SCg-кода 2-го уровня заключается в том, что изображение наиболее часто используемых дуг принадлежности sc-элементов к типовым (базовым) sc-множествам, можно заменить на некоторый, характерный для них, графический признак. Стоит отметить что графические признаки отличаются для sc.g-узлов и sc.g-дуг, поэтому их будем рассматривать по отдельности.

Множество всех sc-узлов разбивается по двум признакам. Первым признаком является константность (узел может быть элементом множества *sc-констант*, либо множества *первичных sc-переменных*). На рисунке 5.5 представлен пример записи sc.g-текста, с помощью SCg-ядра, в котором присутствуют sc-узлы являющиеся элементами множества *sc-констант* и sc-узлы, которые являются элементами множества *первичных sc-переменных*.

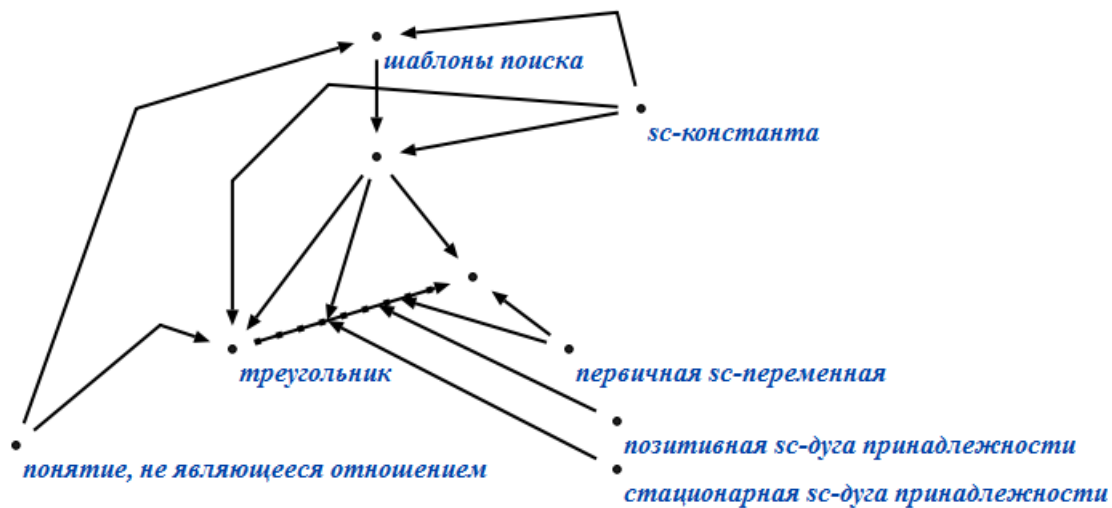


Рис 5.5 - Пример записи поискового шаблона, для поиска всех элементов множества треугольников, с помощью SCg-ядра

Вместо явного изображения, с помощью SCg-кода, sc-дуг принадлежности и sc-узлов обозначающих эти два множества, используются следующие признаки:

- sc-узлы, которые являются элементами множества *sc-констант*, изображаются в виде окружности (рекомендуемая толщина линий и габариты всех sc.g-элементов можно посмотреть в приложении №);
- sc-узлы, которые являются элементами множества *первичных sc-переменных*, изображаются в виде квадратов.



Рис 5.6. Пример изображения константного sc-узла (слева) и переменного sc-узла (справа), с помощью SCg-кода 2-го уровня

Очевидно, что даже это небольшое упрощение значительно повысило наглядность изображаемого sc.g-текста, за счет уменьшения количества изображаемых элементов.

Множество sc-узлов, кроме разбиения по признаку константности, разбивается на следующие подмножества, по структурному признаку:

- множество sc-узлов, обозначающих небинарные связки;
- множество sc-узлов, обозначающих структуры;
- множество sc-узлов, обозначающих ролевые отношения;
- множество sc-узлов, обозначающих неролевые отношения;
- множество sc-узлов, обозначающих понятия, не являющиеся отношениями;
- множество sc-узлов, обозначающих абстрактные объекты, не являющиеся множествами;
- множество sc-узлов, обозначающих материальный объект.

Принадлежность sc-узла к одному из выше перечисленных множеств изображается с помощью специальных графических примитивов, которые расположены внутри фигур изображающих константные или переменные sc-узлы. Пример изображения структурного признака представлен на рисунке 5.7.



Рис 5.7. Пример изображения константного и переменного sc-узлов, обозначающих понятия, не являющиеся отношениями, с помощью SCg-кода 2-го уровня

Используя приведенный выше подход, можно упростить изображение, представленное на рисунке 5.7, до конструкции на рисунке 5.8.

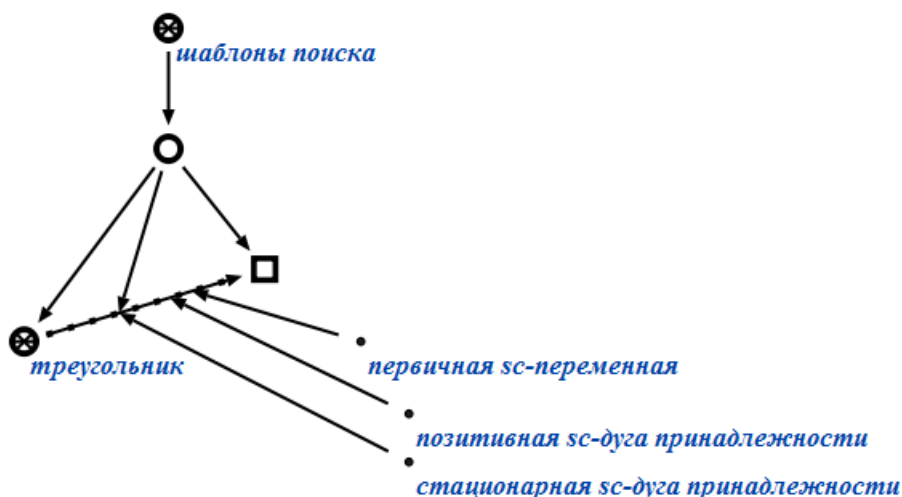


Рисунок 5.8. Пример записи поискового шаблона с упрощенным отображением sc.g-узлов

Аналогичный подход используется для отображения sc-коннекторов. SC-дуги (ребра) общего вида изображаются в виде двух параллельных линий, с достаточно большим промежутком между ними. Признак константности или переменности таких дуг указывается с помощью заполнения этого пустого пространства между линиями. Для переменных дуг оно заполняется штриховой линией. В случае константных дуг, это пространство остается пустым (рисунок 5.9).

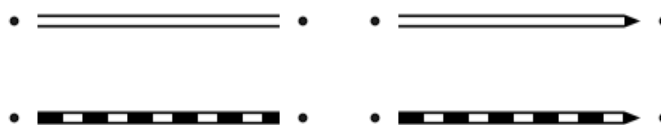
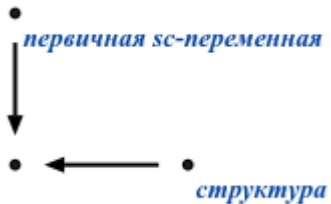



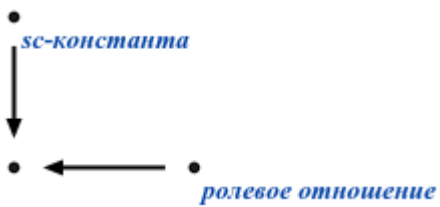

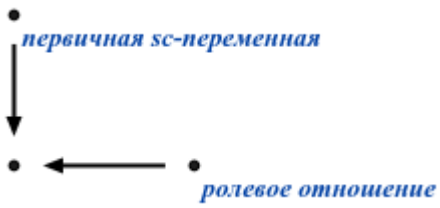







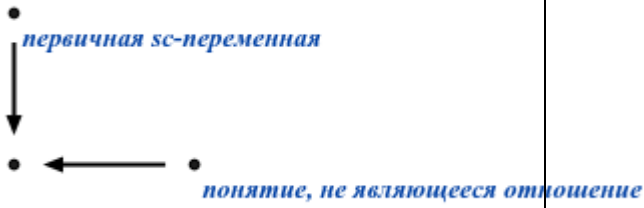

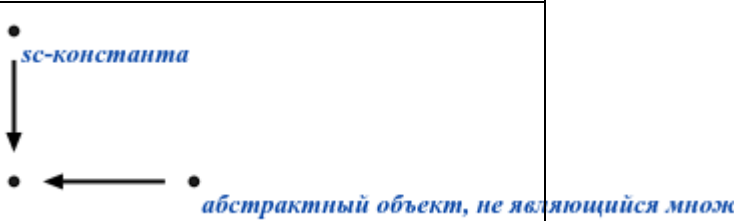


Рисунок 5.9. Пример изображения константных и переменных sc-ребер (дуг)

В случае с дугами принадлежности, используется тот же признак - переменные дуги принадлежности изображаются штрихованной линией. Помимо признака константности множество sc-дуг принадлежности разбивается на подмножества по следующим признакам:

- признак позитивности. По этому признаку они разбиваются на множество позитивных, негативных и нечетких sc-дуг принадлежности;
- признак стационарности. По этому признаку они разбиваются на множество стационарных и нестационарных sc-дуг принадлежности.

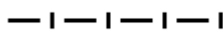
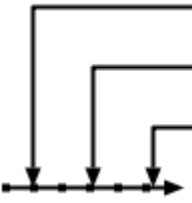
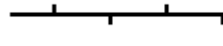
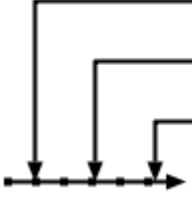
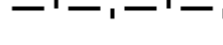
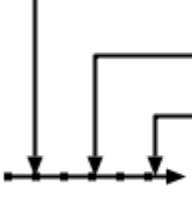

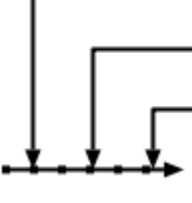

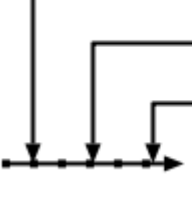

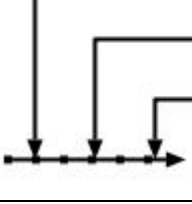

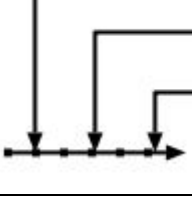
Полную таблицу правил SCg-кода второго уровня можно посмотреть ниже:

Расширение алфавита sc.g-элементов	Синтаксический тип sc.g-элемента	Представление изображаемых sc-элементов в SC-коде (точнее, в SCg-коде 1-го уровня)
○	константный sc.g-узел	
□	sc.g-узел, являющийся первичной переменной	
⊖	константный sc.g-узел обозначающий небинарную связку	
⊞	первичная sc.g-переменная, обозначающая небинарную связку	
⊙	константный sc.g-узел, обозначающий структуру	
⊠	первичная sc.g-переменная, обозначающая структуру	

	константный sc.g-узел, обозначающий ролевое отношение	
	первичная sc.g- переменная, обозначающая ролевое отношение	
	константный sc.g-узел, обозначающий неролевое отношение	
	первичная sc.g- переменная, обозначающая неролевое отношение	
	константный sc.g-узел, обозначающий понятие, не являющееся отношением	
	первичная sc.g- переменная, обозначающая понятие, не являющееся отношением	
	константный sc.g-узел, обозначающий абстрактный объект, не являющийся множеством	
	первичная sc.g- переменная, обозначающая абстрактный объект, не являющийся множеством	

	константный sc.g-узел, обозначающий материальный объект	
	первичная sc.g-переменная, обозначающая материальный объект	
	константное sc.g-ребро	
	первичная sc.g-переменная, обозначающая пару sc-элементов	
	константная sc.g-дуга	
	первичная sc.g-переменная, обозначающая ориентированную пару sc-элементов	
	первичная переменная позитивная стационарная sc.g-дуга принадлежности	<ul style="list-style-type: none"> <li>первичная sc-переменная</li> <li>позитивная sc-дуга принадлежности</li> <li>стационарная sc-дуга принадлежности</li> </ul>
	константная негативная стационарная sc.g-дуга принадлежности	<ul style="list-style-type: none"> <li>sc-константа</li> <li>негативная sc-дуга принадлежности</li> <li>стационарная sc-дуга принадлежности</li> </ul>



	первичная переменная негативная стационарная s.c.g-дуга принадлежности	 <ul style="list-style-type: none"> <li>первичная s.c-переменная</li> <li>негативная s.c-дуга принадлежности</li> <li>стационарная s.c-дуга принадлежности</li> </ul>
	константная нечеткая стационарная s.c.g-дуга принадлежности	 <ul style="list-style-type: none"> <li>s.c-константа</li> <li>нечеткая s.c-дуга принадлежности</li> <li>стационарная s.c-дуга принадлежности</li> </ul>
	первичная переменная нечеткая стационарная s.c.g-дуга принадлежности	 <ul style="list-style-type: none"> <li>первичная s.c-переменная</li> <li>нечеткая s.c-дуга принадлежности</li> <li>стационарная s.c-дуга принадлежности</li> </ul>
	константная позитивная нестационарная s.c.g- дуга принадлежности	 <ul style="list-style-type: none"> <li>s.c-константа</li> <li>позитивная s.c-дуга принадлежности</li> <li>нестационарная s.c-дуга принадлежности</li> </ul>
	первичная переменная позитивная нестационарная s.c.g- дуга принадлежности	 <ul style="list-style-type: none"> <li>первичная s.c-переменная</li> <li>позитивная s.c-дуга принадлежности</li> <li>нестационарная s.c-дуга принадлежности</li> </ul>
	константная негативная нестационарная s.c.g- дуга принадлежности	 <ul style="list-style-type: none"> <li>s.c-константа</li> <li>негативная s.c-дуга принадлежности</li> <li>нестационарная s.c-дуга принадлежности</li> </ul>
	первичная переменная негативная нестационарная s.c.g- дуга принадлежности	 <ul style="list-style-type: none"> <li>первичная s.c-переменная</li> <li>негативная s.c-дуга принадлежности</li> <li>нестационарная s.c-дуга принадлежности</li> </ul>





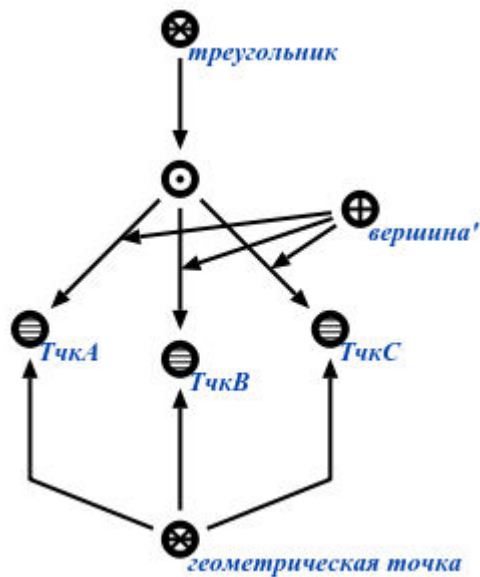


Рис. 5.11 Имеется треугольник с вершинами в точках А, В и С (запись с помощью SCg-кода 2-го уровня).

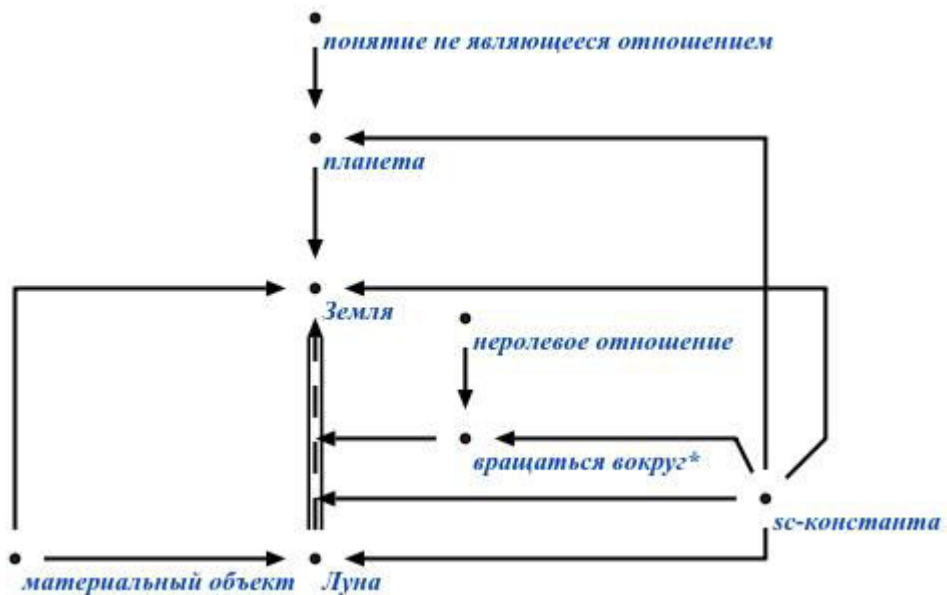


Рис. 5.12 Луна вращается вокруг планеты "Земля" (запись с помощью SCg-кода 1-го уровня).



Рис. 5.13 Луна вращается вокруг планеты "Земля" (запись с помощью SCg-кода 2-го уровня).

Из примеров видно, что SCg-код 2-го уровня значительно сокращает количество элементов присутствующих на экране. Это в свою очередь позволяет повысить читабельность sc.g-текстов.

#### 5.1.4 SCg-код 3-го уровня

При больших объемах sc.g-текстов и второго уровня может оказаться недостаточно. Как и при переходе с 1-го уровня на 2-й уровень, повышение читабельности sc.g-текстов можно достичь еще более существенным сокращением, изображаемых на экране, элементов. Для этого введен 3-й уровень SCg-кода.

Часто возникает ситуация, когда из sc.g-узла выходит много sc.g-дуг(пар) (или же большое количество sc.g-дуг (пар) в него входит). В таком случае sc.g-текст становится сложно читать, так как вокруг этого sc.g-узла рисуется много sc.g-дуг (пар), которые могут накладываться на текстовый идентификатор. Кроме того в таком тексте средняя длина sc.g-дуг (пар) увеличивается, что приводит к трудностям в поиске начальных и конечных элементов этих sc.g-дуг (см. рисунок ниже)

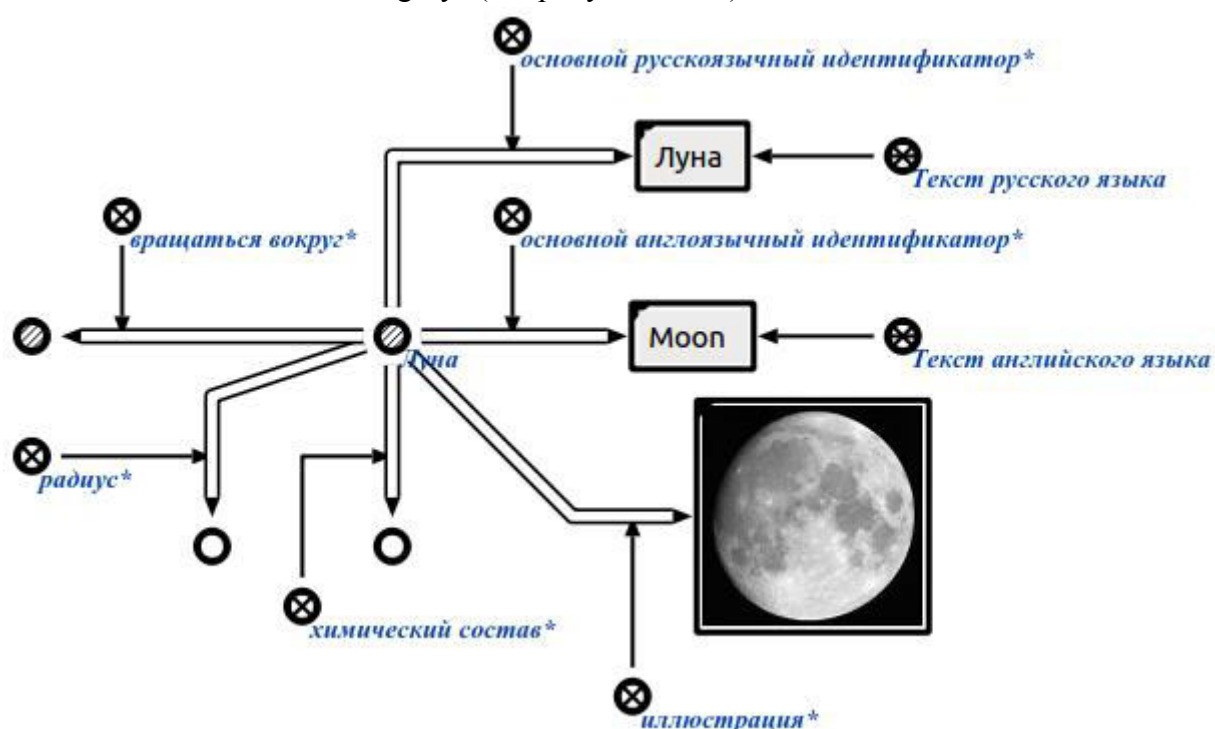


Рис.5.14 Запись некоторой информации о луне с помощью SCg-кода 2-го уровня

Очевидно, что если потребуется добавить еще информации об объекте, то единственным выходом будет его дублирование. Другими словами придется создать еще один узел с идентификатором *Луна* и связывать новые свойства sc.g-парами с ним. Решить данную проблему можно увеличив контактную площадь узла. Для этого вводится новый элемент - sc.g-шина. Пример того же sc.g-текста записанного с помощью SCg-кода 3-го уровня можно посмотреть на следующем изображении:

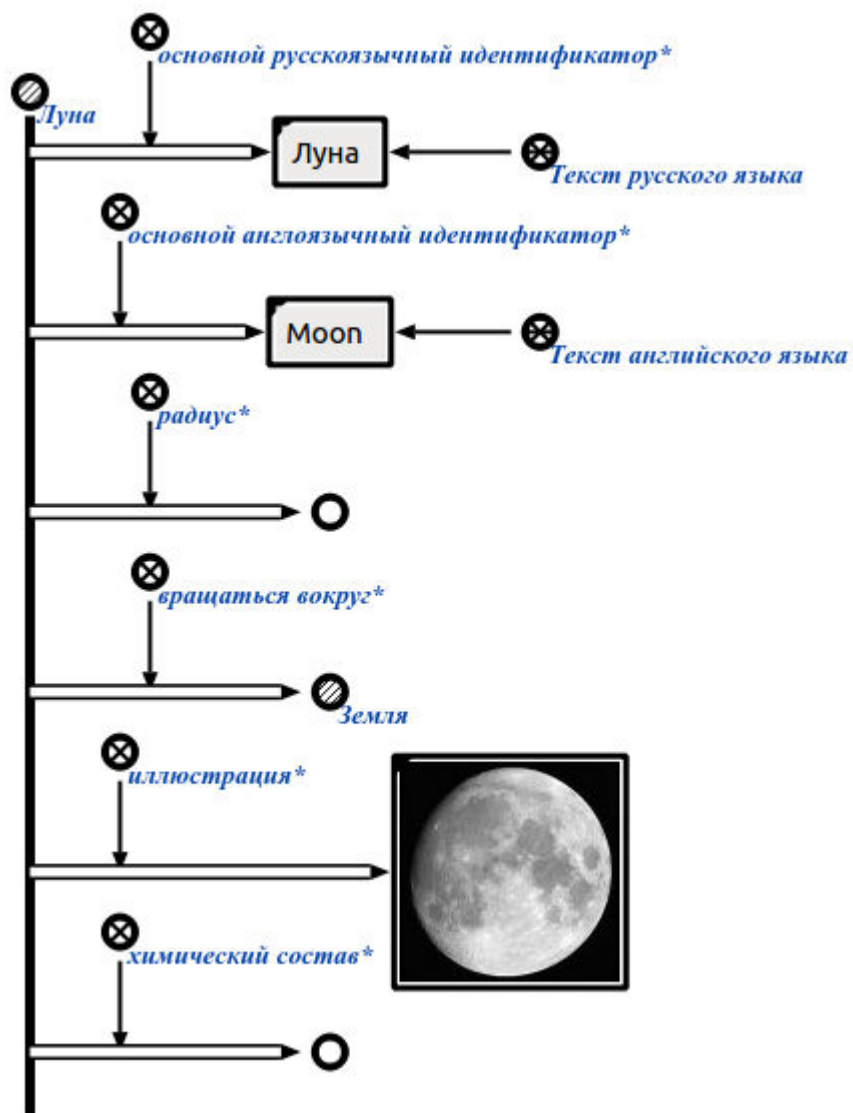
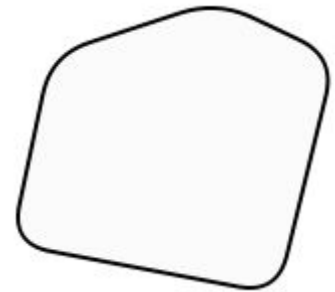

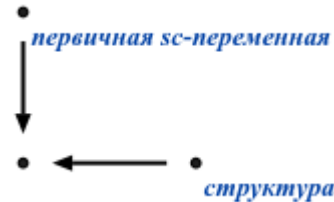






Рис. 5.15 Запись некоторой информации о луне с помощью SCg-кода 3-го уровня

Переход от элементов SCg-кода 1-го уровня к SCg-коду 3-го уровня осуществляется по правилам описанным в таблице:

Расширение алфавита sc.g-элементов	Синтаксический тип sc.g-элемента	Представление изображаемых sc- элементов в SC-коде
	sc.g-шина	
	константный sc.g-контур	

	<i>первопеременный sc.g-контур</i>	
	<i>sc.g-рамка</i>	
	<i>sc.g-узел</i>	

### 5.1.5 SCg-код 4-го уровня

SCg-код является не только способом визуального представления sc-текстов, а также используется в графическом пользовательском интерфейсе для организации диалога пользователя с системой. Конечно, пользователь может производить общение с помощью трех первых уровней (пользователь рисует сообщения вручную). Но, чтобы повысить эффективность диалога, на 4-ом уровне SCg-кода вводятся sc.g-элементы управления. Нажатие на такой объект инициирует некоторое действие, которое он обозначает.

### 5.2 SCn-код

**SCn-код** - строковый нелинейный вариант представления SC-кода. С формальной точки зрения **SCn-код** - множество *sc.n-текстов*.

Предназначен для представления *sc-графов* в виде отформатированных по заданным правилам последовательностей символов, в которых также могут быть использованы базовые средства гипермедиа, такие как графические изображения, а также средства навигации между частями *sc.n-текстов*. **SCn-код** имеет много общего с *SCs-кодом* и, за исключением некоторых особенностей, является его двумерным форматированным вариантом.

Каждый *sc.n-текст* может быть представлен в нескольких вариантах идентификации *sc-элементов* представляемого *sc-графа*:

- с использованием *системных sc-идентификаторов*, которые носят интернациональный характер;
- с использованием *основных sc-идентификаторов* для русскоязычных пользователей;
- с использованием *основных sc-идентификаторов* для англоязычных пользователей;

Каждый *sc.n-текст* представляет собой последовательность *sc.n-статей* (это аналог форматированного *естественно-языкового текста*). Каждая *sc.n-статья*, в свою очередь, представляет собой последовательность *sc.n-предложений*, в начале которой помещается, *заголовок sc.n-статьи\**, который представляет собой идентификатор ключевого *sc-элемента* того *sc-графа*, который представляется данной

*sc.n-статьей*. С семантической точки зрения указанный *sc-граф* является *семантической окрестностью*, центром которой является указанный ключевой *sc-элемент*. При этом ключевой *sc-элемент* некоторой статьи обязательно входит в состав каждого из *sc.n-предложений*, но необязательно является компонентом ключевого (самого первого, считая от начала предложения) *sc-коннектора* данного *sc.n-предложения*.

Входящие в *sc.n-статью* *sc.n-предложения* являются *sc.s-предложениями* либо некоторыми их модификациями. Разделителями *sc.n-предложений* (как и *sc.s-предложений*) являются *двойные точки с запятой*. Этот же разделитель отделяет *заголовок sc.n-статьи* от первого предложения этой *sc.n-статьи*. При этом *заголовок sc.n-статьи* можно трактовать как вырожденное *sc.n-предложение*, состоящее только из одного *sc-идентификатора*.

Одним из вариантов хранения *sc.n-текстов* являются *sc.s-тексты*, которые проще в обработке и хранении при помощи традиционных средств. При этом разрешается использовать все возможности *SCs-кода*. Для того, чтобы в *sc.s-тексте* указать ключевой элемент представляемой *sc.n-статьи* в начале соответствующего фрагмента *sc.s-текста* используется комментарий следующего вида:

<" заголовок *sc.s-текста*, являющегося *sc.n-статьей* уровня 1 "> ::= **/\* keyword:**  
<" системный идентификатор ключевого *sc-элемента* данной *sc.n-статьи* "> **\*/**

При хранения *sc.n-текстов* на основе *SCs-кода* используются только *системные идентификаторы sc-элементов* (*системные sc-идентификаторы*).

Редактирование *sc.n-текстов* может осуществляться как непосредственно с помощью специализированного визуального редактора, так и при помощи стандартного тестового редактора на уровне хранимых *sc.s-текстов*.

Так же, как и другие варианты представления *SC-кода*, ***SCn-код*** условно разделяется на уровни. Можно выделить четыре уровня *SCn-кода*, для представления *sc-графов* в настоящий момент рекомендуется использовать уровни до третьего включительно.

Так же, как и в других вариантах представления *SC-кода*, на любом уровне ***SCn-кода***, начиная с уровня 2, разрешается также использовать и *sc.n-предложения* более низких уровней.

Следует также упомянуть о вариантах стилистики в ***SCn-коде***. Под стилистикой понимается какие-либо дополнительные правила оформления *sc.n-текстов*, не относящиеся к какому-либо конкретному уровню ***SCn-кода***.

Можно выделить следующие варианты классификации различных стилистик оформления ***SCn-текстов***:

- 1) Правила упорядочивания *sc.n-предложений* в *sc.n-статье*;
- 2) Правила упорядочивания *sc.n-статей* в *sc.n-тексте* (например, по алфавиту, логически и т.п.);
- 3) Правила введения фрагментов одной *sc.n-статьи* в другую (использование встроенных *sc.n-статей*);
- 4) Глубина детализации описания ключевого *sc-элемента* *sc.n-статьи* (например, семантическая окрестность радиуса 1 или радиуса 2 и т.д.);

### 5.2.1 SCn-код уровня 1

**SCn-код уровня 1** представляет собой определенный способ форматирования произвольных *sc.s-текстов*.

Правила форматирования:

- 1) Для заданного *sc.s-текста* выделить ключевые описываемые *sc-элементы* и их имена.
- 2) Множество предложений заданного *sc.s-текста* разбить на группы в соответствии с выделенными ключевыми *sc-элементами* (очевидно, что в каждое такое предложение должно входить имя соответствующего ключевого *sc-элемента*).
- 3) Каждую выделенную группу *sc.s-предложений* оформить как *sc.n-статью*, заголовком которой становится выделенное жирным шрифтом имя соответствующего *sc-элемента*, оформленное как специальное *sc.s-предложение*, состоящее только из одного имени и ограниченное *разделителем двойная точка с запятой* (как и любое *sc.s-предложение*).
- 4) Каждое *sc.s-предложение*, входящее в состав некоторой *sc.n-статьи* записать с новой строки (начиная с первого символа строки).
- 5) Увеличить интервал между разными *sc.n-статьями*.
- 6) Все вхождения идентификатора, который указан в заголовке соответствующей *sc.n-статьи*, в *sc.s-предложение*, выделить шрифтом.

Примеры *sc.n-текстов уровня 1*:

**треугольник;;** /\* заголовок *sc.n-статьи* уровня 1 \*/

**треугольник**  $\Rightarrow$  разбиение\*: {разносторонний треугольник; равносторонний треугольник; строго равнобедренный треугольник};;

**Треугк(Тчка;ТчкВ;ТчкС);;** /\* заголовок *sc.n-статьи* уровня 1 \*/

**Треугк(Тчка;ТчкВ;ТчкС)**  $\in$  пример': *треугольник*;;

**Треугк(Тчка;ТчкВ;ТчкС)**  $\Rightarrow$  сторона\*: *Отр(Тчка;ТчкВ)* (\*  $\Rightarrow$  *границная точка*\*: *Тчка; ТчкВ* \*); *Отр(ТчкВ;ТчкС)* (\*  $\Rightarrow$  *границная точка*\*: *ТчкВ; ТчкС* \*);;

### 5.2.2 SCn-код уровня 2

*SCn-код уровня 2* отличается от предыдущего уровня тем, что форматирование *sc.s-предложений* производится не только в вертикальном, но и горизонтальном направлении.

Правила форматирования:

- 1) Если входящее в *sc.n-статью sc.s-предложение* уровня 2 и выше начинается с имени ключевого *sc-элемента* данной *sc.n-статьи* (т.е. совпадает с заголовком *sc.n-статьи*), то это имя можно удалить. При этом первый *sc.s-коннектор* указанного *sc.s-предложения* переместится в начало строки. Все остальные *sc.s-коннекторы*, входящие в рассматриваемое *sc.s-предложение*, но не в состав встроенного в него предложения, также переносятся в начало следующей строки.
- 2) Первый *sc.s-коннектор* встроенного *sc.s-предложения* размещается точно под тем *sc-именем*, которое предшествует этому встроенному *sc.s-предложению*. При этом скобки, ограничивающие данное встроенное предложение, удаляются. Заметим, что встроенное *sc.s-предложение* может входить в состав другого встроенного *sc.s-предложения*. Аналогичным образом под *sc-идентификатором*, предшествующим встроенному *sc.s-предложению* размещаются и остальные *sc.s-коннекторы*

размещаемого встроенного *sc.s-предложения*, не входящие в состав встроенного в него *sc.s-предложения*.

Если в *sc.s-предложении* (в том числе встроенном) встречается *точка с запятой* (как разделитель), то соответствующий *sc.s-коннектор* и все его модификаторы повторяются столько раз, сколько раз встречается указанный разделитель. При этом дублируемый *sc.s-коннектор* и модификаторы отображаются в полном соответствии с правилами данного пункта.

3) Удаляются *разделители двойная точка с запятой sc.s-предложений*. При этом для обеспечения наглядности необходимо увеличить расстояние по вертикали между изображенными *sc.s-предложениями* (отступы), но не превышая интервала между *sc.n-статьями*. Следует отметить, что если *sc.n-предложение* не начинается с ключевого элемента *sc.n-статьи*, то *разделитель двойная точка с запятой* не удаляется в конце указанного *sc.n-предложения*, а также в конце предыдущего *sc.n-предложения*, если оно не состоит только из заголовка *sc.n-статьи*.

4) При использовании модификаторов *sc.s-коннектора* вся часть *sc.s-предложения*, стоящая справа от последнего *разделителя двоеточие* изображается с новой строки, начиная с символа, стоящего под первым символом первого модификатора.

5) Вводятся закадровые (фоновые) вертикальные линии для более четкого определения уровня вложенности *sc.s-предложений* в рамках *sc.n-статьи*.

6) Удаляется *sc.s-разделитель точка с запятой*.

Примеры *sc.n-текстов уровня 2*:

***треугольник*** /\* заголовок *sc.n-статьи* уровня 2 \*/  
⇒ *разбиение\**: /\* ключевой элемент *sc.n-статьи* опускается \*/  
{*разносторонний треугольник*; *равносторонний треугольник*; *строго равнобедренный треугольник*}  
/\* второй компонент связки указывается с новой строки \*/

***Треугк(ТчкаА;ТчкаВ;ТчкаС)*** /\* заголовок *sc.n-статьи* уровня 2 \*/  
∈ *пример'*: /\* ключевой элемент *sc.n-статьи* опускается \*/  
*треугольник*  
⇒ *сторона\**:  
*Отр(ТчкаА;ТчкаВ)*  
⇒ *граничная точка\**: /\* встроенное *sc.s-предложение* смещается вправо \*/  
*ТчкаА* /\* ключевой элемент встроенного *sc.s-предложения* опускается \*/  
⇒ *граничная точка\**:  
*ТчкаВ*  
⇒ *сторона\**:  
*Отр(ТчкаВ;ТчкаС)*  
⇒ *граничная точка\**:  
*ТчкаВ*  
⇒ *граничная точка\**:  
*ТчкаС*

### 5.2.3 SCn-код уровня 3



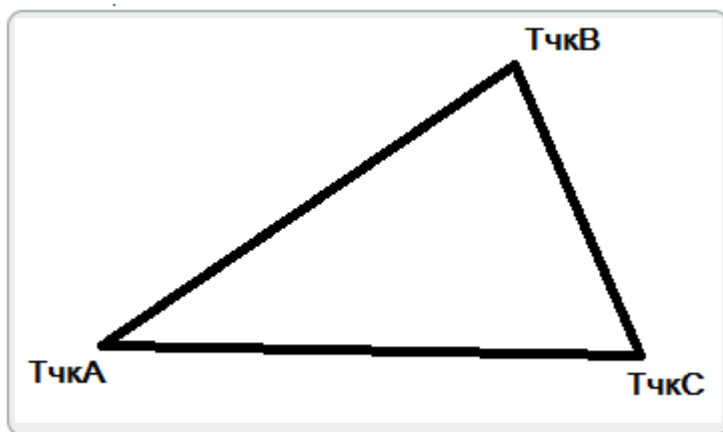
**SCn-код уровня 3** отличается от предыдущих уровней тем, что на данном уровне появляется возможность использовать средства гипермедиа.

Вводимые элементы:

- 1) *Контурные sc-идентификаторы*, каждый из которых идентифицирует *sc-элемент*, обозначающий множество всех *sc-элементов* того *sc-графа*, который является результатом перевода в *SC-код* той информации, которая ограничена контуром.
- 2) *Рамочные sc-идентификаторы*, каждый из которых идентифицирует *sc-элемент* (точнее, *sc-ссылку*), который обозначается файл, отображаемый в рамке.
- 3) Внутри *контурных sc-идентификаторов* и *рамочных sc-идентификаторов* могут находиться не только линейные тексты (как в *SCs-коде*), но и информационные конструкции любого вида (*sc.g-тексты*, рисунки, изображения). Таким образом осуществляется переход к гипермедийному тексту.

Пример *sc.n-текста* уровня 3:

**Треугк(ТчкА;ТчкВ;ТчкС)** /\* заголовок sc.n-статьи уровня 3 \*/  
⇒ иллюстрация\*:



/\* контурный sc-идентификатор \*/

⇒ пояснение\*:

Треугольник Треугк(ТчкА; ТчкВ; ТчкС) задается точками ТчкА, ТчкВ и ТчкС, которые являются его вершинами

/\* Здесь использован рамочный sc-идентификатор, соответствующий sc-узлу, обозначающему текстовый файл, отображенный в рамке \*/

#### 5.2.4 SCn-код уровня 4

В *SCn-коде уровня 4* улучшается визуальное восприятие текстов за счет введения разделителя перечисления и удаления некоторых частей *sc.s-предложений*.

На данном уровне вводится разделитель перечисления •. Данный разделитель используется в следующих случаях:

- 1) При отображении составных идентификаторов {...} или <...>
- 2) При отображении уточнений типа *sc.s-коннектора* с использованием *разделителя двоеточия*.

При этом, если модификаторы *sc.s-коннектора* полностью совпадают для всех *sc-элементов*, стоящих в правой части *sc.s-коннектора*, то перечисляются только сами *sc-элементы*.

В противном случае перечисление может происходить в несколько уровней, при этом на каждом уровне перечисляются *sc-элементы* и модификаторы, имеющие общую часть на вышестоящем уровне.

3) При использовании разделителя небинарная связка  $\Theta$  (данный разделитель не имеет аналога в *SCs-коде*).

4) При описании идентификаторов *sc-элементов* используется *разделитель равенства* = (синтаксически аналогичный *sc.s-коннектору равенства*).

Примеры *sc.n-текстов уровня 4*:

**треугольник** /\* заголовок *sc.n-статьи уровня 4* \*/

⇒ *разбиение\**:

- {
- *разносторонний треугольник*
- *равносторонний треугольник*
- *строго равнобедренный треугольник*
- }

/\* изображение составного идентификатора с использованием разделителя перечисления \*/

**v1** /\* заголовок *sc.n-статьи уровня 4* \*/

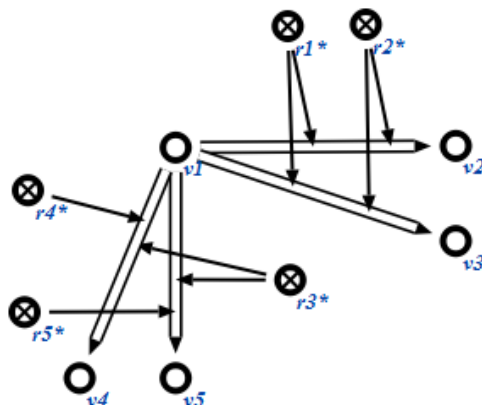
⇒ *r1\**: *r2\**:

- *v2*
- *v3*

⇒ *r3\**:

- *r4\**:
- *v4*
- *r5\**:
- *v5*

Приведем *sc.g-текст*, семантически эквивалентный данному *sc.n-тексту*:



**8** /\* заголовок *sc.n-статьи уровня 4* \*/

$\Theta$  *возведение в степень\**:

- *основание'*: 2 /\* *sc.n-предложение, описывающее тернарную связку* \*/
- *показатель'*: 3 /\* *отношения возведения в степень* \*/

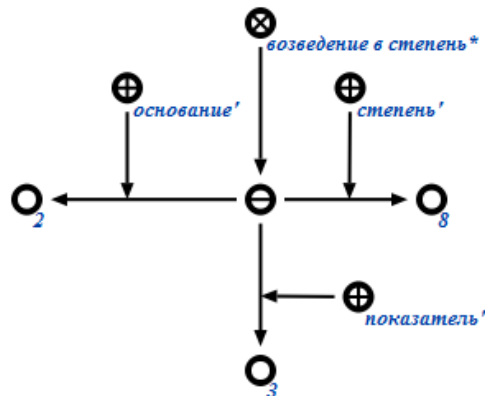
- *степень'*: 8

$$= 2^3$$

$$= 2^3$$

*/\* связка отношения, описывающего арифметическую операцию возведения в степень \*/*

Приведем *sc.g-текст*, семантически эквивалентный данному *sc.n-тексту*:



**канторовское множество** */\* заголовок sc.n-статьи уровня 4 \*/*

= классическое множество

= множество без кратных вхождений элементов

= множество без кратных принадлежностей элементов

*/\* несколько вариантов идентификации одного и того же понятия\*/*

Указанный *sc.n-текст* уровня 4 семантически эквивалентен следующему *sc.n-тексту* уровня 3:

**канторовское множество**

⇒ идентификатор\*:

классическое множество

⇒ идентификатор\*:

множество без кратных вхождений элементов

⇒ идентификатор\*:

множество без кратных принадлежностей элементов

### 5.3 Переключение способов отображения информации (мультимодальность)

Для переключения способов ввода используется соответствующий компонент управления

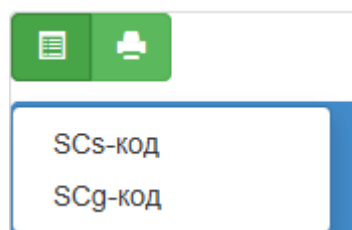


Рис.5.16 – Переключение способов отображения

## 6 Пользовательский интерфейс системы

### 6.1 Изменение языка ведения диалога (русский, английский)

Для изменения языка диалога используется специальный компонент

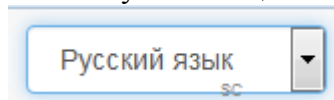


Рис. 6.1 Переключения языка диалога

### 6.2 Использование пунктов меню, всплывающих подсказок, ведение диалога с системой

Пользовательский интерфейс системы состоит из главного окна в рамках которого размещены все элементы управления. Диалог с системой осуществляется путем обмена сообщениями. Для задания вопросов используются пункты меню. При задании вопроса необходимо указать аргументы вопроса.

В качестве аргументов запросов могут выступать любые элементы в рамках главного окна (при наведении на них справа от курсора появляется небольшая надпись sc). Используя эту возможность вы можете получить справку по всему пользовательскому интерфейсу.

Система содержит обучающий ролик, описывающий пример ведения простейшего диалога. Ролик доступен при нажатии кнопки в верхней правой части экрана:



Рис. 6.2 Кнопка помощи

Система позволяет задавать любые вопросы к любым компонентам интерфейса, обучаясь таким образом использованию этих компонентов.

### 6.3 Основные команды меню

#### Для произвольных объектов предметной области

##### *Команда поиска иллюстраций*

Команда поиска иллюстраций для заданного понятия предназначена для нахождения иллюстраций для заданного понятия. Единственным аргументом команды является понятие, для которого необходимо найти иллюстрации. Результатом выполнения запроса является множество изображений, найденных для понятия, само понятие и множество связывающих их дуг с отношением трансляция sc-текста\* и атрибутом пример'.

##### *Команда поиска определения/пояснения*

Команда поиска определения/пояснения для заданного понятия предназначена для нахождения определения либо пояснения для заданного понятия. Единственным аргументом команды является понятие, для которого необходимо найти определение или пояснение. Результатом выполнения запроса является множество из определения и пояснений, найденных для понятия с их основными идентификаторами\*, само понятие и множество связывающих их дуг с отношением трансляция sc-текста\* и атрибутом пример'.

##### *Для произвольных понятий*

##### *Команда поиска понятий, на основе которых определяется данное*

Команда поиска понятий, на основе которых определяется заданное понятие предназначена для нахождения понятий, через которые определяется заданное понятие. Единственным аргументом команды является понятие, для которого запрашиваются определяющие понятия. Результатом выполнения команды является узел, включающий множество понятий, через которые определяется заданное понятие, связанное отношением используемые константы\* с заданным понятием.

#### *Команда поиска отношений, заданных на понятии*

Команда поиска отношений для заданного понятия предназначена для нахождения отношений, заданных на указанном понятии. Единственным аргументом команды является понятие, для которого необходимо найти заданные на нем отношения. Результатом выполнения запроса являются отношения, найденные для понятия, само понятие и множество связывающих их дуг.

#### *Команда поиска примеров*

Команда поиска примеров для заданного понятия предназначена для нахождения примеров заданного понятия. Единственным аргументом команды является понятие, для которого необходимо найти примеры. Результатом выполнения команды является множество примеров, найденных для понятия, само понятие и множество связывающих их дуг с атрибутом пример'.

#### *Команда поиска утверждений*

Команда поиска утверждений для заданного понятия предназначена для нахождения утверждений для заданного понятия. Единственным аргументом команды является понятие, для которого необходимо найти утверждения. Результатом выполнения запроса является множество утверждений, найденных для понятия, само понятие и множество связывающих их дуг с отношением трансляция sc-текста\* и атрибутом пример'.

#### **Для относительных понятий**

##### *Команда поиска области определения отношения*

Команда поиска области определения для заданного отношения предназначена для нахождения области определения для заданного отношения. Единственным аргументом команды является отношение, для которого необходимо найти область определения. Результатом выполнения команды является область определения\* отношения, найденная для отношения, само понятие и все домены отношения.

#### **Для утверждений**

##### *Команда поиска доказательства*

Команда поиска доказательства для заданного утверждения предназначена для нахождения доказательства для заданного утверждения. Единственным аргументом команды является утверждение, для которого необходимо найти доказательство. Результатом выполнения команды является полная декомпозиция доказательства с указанием последовательности действий и промежуточными результатами, а также сам знак утверждения.