Документация по SC.s

Авторы: Казанцев Павел

Лебединский Роман

Мышкевич Юрий

Протас Виталий

**Содержание**

[Введение 3](#__RefHeading__1151_688780230)

[SC.s-код 1-го уровня 4](#__RefHeading__1153_688780230)

[SC.s-код 2-го уровня 7](#__RefHeading__1155_688780230)

[SC.s-код 3-го уровня 8](#__RefHeading__1157_688780230)

[SC.s-код 4-го уровня 9](#__RefHeading__1157_688780230)

[SC.s-код 5-го уровня 10](#__RefHeading__1157_688780230)

[SC.s-код 6-го уровня 11](#__RefHeading__1157_688780230)

[SC.s-код 7-го уровня 13](#__RefHeading__1157_688780230)

[Базовые sc.s-коннекторы 16](#__RefHeading__1157_688780230)

[Дополнительные sc.s-коннекторы 17](#__RefHeading__1157_688780230)

[Специальные sc.s-коннекторы 18](#__RefHeading__1157_688780230)

**Введение**

***SCs-код*** - строковый (линейный) вариант представления *SC-кода*. Предназначен для представления sc-графов (текстов *SC-кода*) в виде последовательностей символов, которые могут быть отредактированы как при помощи стандартных текстовых редакторов, так и при помощи специализированного *sc.s-редактора*. Следовательно, одним из требований, предъявляемых к ***SCs-коду***, помимо полноты и непротиворечивости, является возможность набора исходных текстов без помощи специализированного редактора, используя только символы стандартной клавиатуры.

В отличие от *SCn-кода*, также являющегося текстовым вариантом отображения sc-графов, в ***SCs-коде*** форматирование не имеет значения, все предложения могут быть записаны в одну строку.

В отличие от *SCg-кода*, в ***SCs-коде*** нет специальных обозначений для указания структурных типов *sc-узлов*, однако есть изображения *sc-коннекторов*, соответствующие ядру и расширению *SCg-кода*, а также изображения *sc-коннекторов* специального типа, для которых нет соответствия в *SCg-коде*.

С формальной точки зрения ***SCs-код*** - множество *sc.s-текстов*. Каждый *sc.s-текст* представляет собой последовательность *sc.s-предложений*, каждое из которых оканчивается разделителем ;; (двойная точка с запятой). Множество *sc.s-предложений* разбивается на множество простых и множество сложных *sc.s-предложений*. Каждое сложное *sc.s-предложение* содержит в своем составе встроенные предложения, ограниченные ограничителем (\*...\*).

В рамках *sc.s-текста* любого уровня допустимо использование комментариев следующего вида:

// однострочный комментарий

/\* многострочный комментарий \*/

В начале файла, содержащего *sc.s-текст* настоятельно рекомендуется указывать уровень и версию используемого ***SCs-кода***. Для этого используются комментарий специального вида:

/\* SCs\_code<" версия ">.Level <" номер уровня ">\*/

Например:

/\* SCs\_code0.1.0.Level 6 \*/

Следует отметить, что комментарий вида /\*…\*/, использованный внутри *sc.s-рамки* (см. соотв. раздел) не опускается при разборе *sc.s-текста*, и становится частью содержимого соответствующей *sc-ссылки*. Например при разборе предложения X => r1: [TEXT1/\*COMMENT\*/TEXT2];;

Будет создана sc-ссылка с содержимым TEXT1/\*COMMENT\*/TEXT2, а не TEXT1TEXT2.

***SCs-код*** условно разделяется на 7 уровней сложности. Все уровни равнозначны по возможностям представления знаний, однако тексты более высоких уровней описывают sc-графы более лаконично и удобно. Введение уровней призвано облегчить работу разработчиков баз знаний при наборе *sc.s-текстов*. При разработке баз знаний рекомендуется использовать уровни до шестого включительно.

**SCs-код 1-го уровня**

***SCs-код уровня 1*** аналогичен *SCg-коду уровня 1* и представляет собой простейшее

универсальное средство кодирования (представления) любого sc-графа в линейном виде,

т.е. в виде цепочки (строки) символов. При этом каждому sc-элементу представляемого

sc-графа взаимно-однозначно ставится в соответствие его строковый идентификатор (имя), что делает необходимым специально придумывать имена для всех (!) используемых

sc-элементов, входящих в состав представляемого sc-графа.

***SC.s-предложение уровня 1*** - *sc.s-предложение*, содержащее только *sc.s-разделители инцидентности* и *имена sc-элементов*.

Каждое такое предложение представляет собой тройку, каждая из которых описывает инцидентность некоторого *sc-коннектора* и двух других *sc-элементов*. Для этого все *sc-элементы* должны быть проименованы. Коннекторы и файлы именуются соответственно правилам идентификации коннекторов и файлов.

Запрещается использование неатомарных идентификаторов, содержащих ограничители <...>, {...}, [...] или [\*...\*]. При этом в каждой описываемой тройке среднее имя соответствует *sc-коннектору*, тип которого может быть уточнен при помощи правил идентификации.

В случае, если указанный *sc-коннектор* ориентированный (т.е. является *sc-дугой*), то первый и третий элемент тройки соответствуют первому и второму элементам связки,

знаком которой является данный *sc-коннектор*. Т.е. имя *sc-элемента*, из которого выходит указанная дуга, указывается первым в тройке, а имя *sc-элемента*, в который входит эта дуга, указывается третьим в тройке.

Для указания типов *sc-элементов* могут быть использованы следующие правила

уточнения типа *sc-элемента*:

<" **имя элемента с уточнением типа** "> ::= <" **тип элемента** "> # ["<" **имя элемента** ">"]

<" **тип элемента** "> ::= (" sc\_node || sc\_link || sc\_arc\_common || sc\_edge ||sc\_arc\_main||

sc\_arc\_access ")

<" **имя элемента** "> - произвольная уникальная строка символов, уникальная для

*sc-элементов* заданного типа. В качестве имени рекомендуется использовать системный

идентификатор *sc-элемента*. Если перед глобальным идентификатором ставится знак “.” ,

то такое имя служит только для обозначения элемента в исходных текстах, и не транслируется как системный идентификатор. При этом такое имя также должно быть уникальным в рамках всех исходных тестов базы знаний.

<" **имя элемента** "> может отсутствовать, тогда *sc-элемент*, описанный как <" **тип**

**элемента** "> # будет воспринят как *sc-элемент* указанного типа без идентификатора. *Sc-элемент* без указания типа по умолчанию считается *sc-узлом*.

Обобщенная синтаксическая структура ***sc.s-предложения уровня 1***:

<" **sc.s-предложение уровня 1** "> ::= <" **имя первого элемента тройки** "> | <" **имя**

**коннектора** "> | <" **имя второго элемента тройки** "> ;;

Примеры sc.s-предложений уровня 1:

*Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* | sc\_arc\_common#*pair1* | *Отр(ТчкВ;ТчкС)*;;

*включение*\* | sc-arc-main#*arc2* | sc\_arc\_common#*pair1*;;

***sc.s-текст уровня 1*** - *sc.s-текст*, содержащий только *sc.s-предложения уровня 1*

Пример ***sc.s-текста уровня 1***:

*включение*\* | sc\_arc\_main#*arc1* | sc\_arc\_common#*pair2*;;

*включение*\* | sc\_arc\_main#*arc2* | sc\_arc\_common#*pair1*;;

*включение*\* | sc\_arc\_main#*arc3* | sc\_arc\_common#*pair3*;;

*включение*\* | sc\_arc\_main#*arc4* | sc\_arc\_common#*pair4*;;

*включение* \* | sc\_arc\_main#*arc15* | sc\_arc\_common#*pair5*;;

*сторона*\* | sc\_arc\_main#*arc5* | sc\_arc\_common#*pair1*;;

*сторона*\* | sc\_arc\_main#*arc6* | sc\_arc\_common#*pair2*;;

*сторона*\* | sc\_arc\_main#*arc7* | sc\_arc\_common#*pair4*;;

*сторона*\* | sc\_arc\_main#*arc8* | sc\_arc\_common#*pair3*;;

*сторона*\* | sc\_arc\_main#*arc10* | sc\_arc\_common#*pair5*;;

*Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* | sc\_arc\_common#*pair1* | *Отр(ТчкВ;ТчкС)*;;

*Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* | sc\_arc\_common#*pair2* | *Отр(ТчкC;ТчкD)*;;

*Треугк(ТчкВ;ТчкС;ТчкD)* | sc\_arc\_common#*pair3* | *Отр(ТчкВ;ТчкС)*;;

*Треугк(ТчкВ;ТчкС;ТчкD)* | sc\_arc\_common#*pair4* | *Отр(ТчкC;ТчкD)*;;

*Треугк(ТчкВ;ТчкС;ТчкD)* | sc\_arc\_common#*pair5* | *Отр(ТчкB;ТчкD)*;;

*четырехугольник* | sc\_arc\_main#*arc9* | *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)*;;

*треугольник* | sc\_arc\_main#*arc15* | *Треугк(ТчкВ;ТчкС;ТчкD)*;;

*link1* | sc\_arc\_common#*pair6* | *Треугк(ТчкВ;ТчкС;ТчкD)*;;

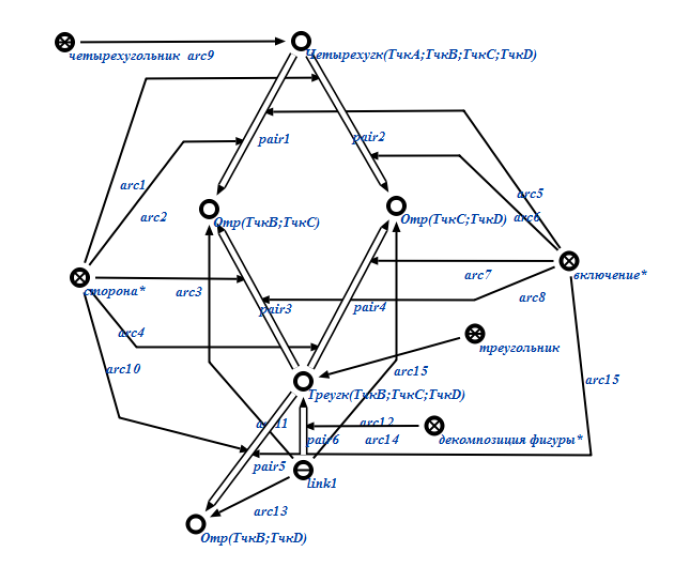
*декомпозиция фигуры*\* | sc\_arc\_main#*arc14* | sc\_arc\_common#*pair6*;;

*link1* | sc\_arc\_main#*arc11* | *Отр(ТчкВ;ТчкС)*;;

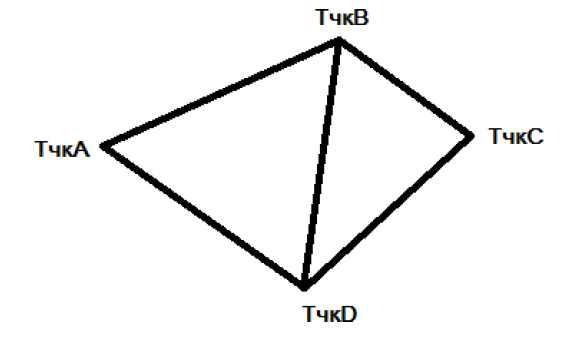
*link1* | sc\_arc\_main#*arc12* | *Отр(ТчкC;ТчкD)*;;

*link1* | sc\_arc\_main#*arc13* | *Отр(ТчкВ;ТчкD)*;;

Данный *sc.s-текст* семантически эквивалентен следующему *sc.g-тексту*:



В свою очередь оба текста соответствуют следующему геометрическому чертежу:



На русском языке текст, семантически эквивалентный указанным выше текстам, выглядит

так:

*sc-элемент Треугк(ТчкВ;ТчкС;ТчкD)* связан выходящими из него дугами, принадлежащими бинарным отношениям *включение\** и *сторона\** с *sc-элементами Отр(ТчкВ;ТчкС), Отр(ТчкВ;ТчкD)* и *Отр(ТчкC;ТчкD)*;

*sc-элемент Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* связан выходящими из него дугами, принадлежащими бинарным отношениям *включение\** и *сторона\** с *sc-элементами Отр(ТчкВ;ТчкС)* и *Отр(ТчкC;ТчкD)*;

*Множество, содержащее элементы Отр(ТчкВ;ТчкС), Отр(ТчкВ;ТчкD)* и *Отр(ТчкC;ТчкD) связано с sc-элементом Треугк(ТчкВ;ТчкС;ТчкD)* отношением *декомпозиция фигуры\**;

*sc-элемент Треугк(ТчкВ;ТчкС;ТчкD)* является элементом множества *треугольник*; *sc-элемент Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* является элементом множества *четырехугольник*.

**SCs-код 2-го уровня**

На данном уровне *SCs-кода* вводятся средства минимизации числа специально придумываемых имен *sc-коннекторов*.

К таким средствам относятся:

1) Алфавит изображений *sc-коннекторов*, отражающих тип изображаемых *sc-коннекторов*.

2) Неатомарные имена *sc-коннекторов*, построенные на основе *sc-коннекторов*. В ***sc.s-предложениях уровня 2*** обеспечивается возможность не вводить имена для всех *sc-коннекторов*.

Для этого вводятся *sc.s-разделители*, являющиеся изображениями *sc-коннекторов*

(например ∍ ), а также неатомарные идентификаторы коннекторов вида (...). Ниже отдельно приведена таблица *sc.s-разделителей*, являющихся изображениями *sc-коннекторов*.

В случае наличия кратных коннекторов между некоторыми *sc-элементами* с целью

избежания омонимии вводятся суффиксы, например (v1 -> v2)1, (v1 -> v2)2.

Использование неатомарных идентификаторов вида <...>, {...}, [\*...\*], [...] не разрешается.

Обобщенная синтаксическая структура ***sc.s-предложения уровня 2***:

<" имя sc-элемента "> ::= (" <" атомарное имя sc-элемента "> || <" неатомарное имя

sc-элемента "> ")

<" неатомарное имя sc-элемента "> ::= ( <" имя sc-элемента "> <" изображение коннектора

"> <" имя sc-элемента "> ) [" суффикс "]

<" sc.s-предложение уровня 2 "> ::= <" имя sc-элемента "> <" изображение коннектора ">

<" имя sc-элемента "> ;;

Примеры ***sc.s-предложений уровня 2***:

*сторона*\* ∍ ( *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкВ;ТчкС)* );;

*включение*\* ∍ ( *Треугк(ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкВ;ТчкС)* );;

***SC.s-текст уровня 2*** - *sc.s-текст*, содержащий хотя бы одно *sc.s-предложение уровня 2* и не содержащий *sc.s-предложений уровня 3* и выше Пример ***sc.s-текста уровня 2***, семантически эквивалентного приведенному выше *sc.s-тексту уровня 1*:

*сторона*\* ∍ ( *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкВ;ТчкС)* );;

*сторона*\* ∍ ( *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкС;ТчкD)* );;

*сторона*\* ∍ ( *Треугк(ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкВ;ТчкС)* );;

*сторона*\* ∍ ( *Треугк(ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкС;ТчкD)* );;

*сторона*\* ∍ ( *Треугк(ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкB;ТчкD)* );;

*включение*\* ∍ ( *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкВ;ТчкС)* );;

*включение*\* ∍ ( *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкС;ТчкD)* );;

*включение*\* ∍ ( *Треугк(ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкВ;ТчкС)* );;

*включение*\* ∍ (Треугк(ТчкВ;ТчкС;ТчкD) => *Отр(ТчкС;ТчкD)* );;

*включение*\* ∍ ( *Треугк(ТчкВ;ТчкС;ТчкD)* => *Отр(ТчкB;ТчкD)* );;

*четырехугольник* ∍ *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)*;;

*треугольник* ∍ *Треугк(ТчкВ;ТчкС;ТчкD)*;;

*декомпозиция фигуры*\* ∍ ( *link1* => *Треугк(ТчкВ;ТчкС;ТчкD)* );;

*link1* ∍ *Отр(ТчкВ;ТчкС)*;;

*link1* ∍ *Отр(ТчкC;ТчкD)*;;

*link1* ∍ *Отр(ТчкВ;ТчкD)*;;

Можно заметить, что в данном примере отсутствуют имена *sc-коннекторов*, которые были искусственно введены для представления данного фрагмента в *SCs-коде уровня 1*, поскольку они заменены соответствующими изображениями *sc-коннекторов*.

**SCs-код 3-го уровня**

На данном уровне *SCs-кода* вводятся дополнительные лаконичные средства описания типологии *sc-коннекторов* - так называемые *модификаторы* и *метамодификаторы sc.s-коннекторов*.

В ***sc.s-предложениях уровня 3*** обеспечивается возможность сокращенно записывать уточнения типов *sc-коннекторов*, для чего вводятся *sc.s-разделители* : и :: (*модификаторы sc-коннекторов*), а также + и ++ (*метамодификаторы sc-коннекторов*).

Если указанные разделители встречаются в предложении несколько раз, то такая запись

указывает, что все уточнения типа относятся к одному и тому же *sc-коннектору*.

Допускается использование уточнения типа *sc-коннектора* в составе неатомарных

идентификаторов, введенных в *sc.s-предложениях уровня 2*, например (v1 => r1 : r2: v2).

Использование неатомарных идентификаторов вида <...>, {...}, [\*...\*] и [...] не разрешается.

Примеры ***sc.s-предложений уровня 3***:

*Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *сторона*\* : *Отр(ТчкВ;ТчкС)*;;

*треугольник* => *идентификатор*\* : *основной англоязычный идентификатор*\*: [triangle];;

***SC.s-текст уровня 3*** - *sc.s-текст*, содержащий хотя бы одно *sc.s-предложение уровня 3* и не содержащий *sc.s-предложений уровня 4* и выше Пример ***sc.s-текста уровня 3***, семантически эквивалентного приведенному выше *sc.s-тексту уровня 2*:

*Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *сторона*\* : *включение*\* : *Отр(ТчкВ;ТчкС)*;;

*Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *сторона*\* : *включение*\* : *Отр(ТчкС;ТчкD)*;;

*Треугк(ТчкВ;ТчкС;ТчкD)* => *сторона*\* : *включение*\* : *Отр(ТчкВ;ТчкС)*;;

*Треугк(ТчкВ;ТчкС;ТчкD)* => *сторона*\* : *включение*\* : *Отр(ТчкС;ТчкD)*;;

*Треугк(ТчкВ;ТчкС;ТчкD)* => *сторона*\*: *включение*\* : *Отр(ТчкB;ТчкD)*;;

*четырехугольник* ∍ *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)*;;

*треугольник* ∍ *Треугк(ТчкВ;ТчкС;ТчкD)*;;

*Треугк(ТчкВ;ТчкС;ТчкD)* <= *декомпозиция фигуры*\*: *link1*;;

*link1* ∍ *Отр(ТчкВ;ТчкС)*;;

*link1* ∍ *Отр(ТчкC;ТчкD)*;;

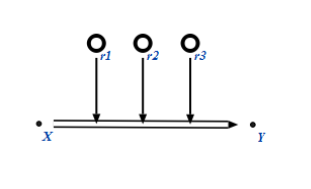
*link1* ∍ *Отр(ТчкВ;ТчкD)*;;

Заметим, что по сравнению с представлением в *SCs-коде 2 уровня*, данный пример содержит меньше *sc.s-предложений*, поскольку несколько *sc.s-предложений уровня 2* могут быть объединены в одно *sc.s-предложение уровня 3*. Также в данном примере отсутствуют ограничители ( ), удаление которых стало возможным благодаря введению разделителя :.

Другой пример ***sc.s-текста 3 уровня***:

*X* => *r1*: *r2*: *r3*: *Y*;;

Данный текст семантически эквивалентен следующему *sc.g-тексту*:



**SCs-код 4-го уровня**

На данном уровне *SCs-кода* вводятся средства лаконичной записи группы *sc.s-предложений уровня 2* или *sc.s-предложений уровня 3*, у которых:

1) В качестве первого (левого) компонента используется одно и то же имя (т.е. имя одного и того же элемента).

2) Полностью аналогичны *sc.s-коннекторы* вместе с группой приписанных им

*модификаторов sc.s-коннекторов*.

Следовательно, ***sc.s-предложение уровня 4*** можно считать описанием семантической окрестности радиуса 1 для заданного *sc-элемента*, т.е описанием известных связей заданного *sc-элемента* с другими смежными ему *sc-элементами*.

В ***sc.s-предложениях уровня 4*** cокращается число изображений *sc-коннекторов* путем введения *sc.s-разделителя* ;.

Более подробно правила использования указанного разделителя рассмотрены ниже в соответствующем разделе.

На данном уровне *SCs-кода* разрешается использование указанного разделителя только для случаев, описанных выше в пояснении к *SCs-коду уровня 4*.

Использование неатомарных идентификаторов вида <...>, {...}, [\*...\*] и [...] не разрешается.

Примеры ***sc.s-предложений уровня 4***:

*Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*;

*Отр(ТчкС;ТчкD)*;;

***SC.s-текст уровня 4* -** *sc.s-текст*, содержащий хотя бы одно *sc.s-предложение уровня 4* и не содержащий *sc.s-предложений уровня 5* и выше.

Пример *sc.s-текста уровня 4*, семантически эквивалентного приведенному выше *Sc.s-тексту уровня 3*:

*Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* => *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*;

*Отр(ТчкС;ТчкD)*;; *Треугк(ТчкВ;ТчкС;ТчкD)* => *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*; *Отр(ТчкС;ТчкD)*; *Отр(ТчкB;ТчкD)*;; *четырехугольник* ∍ *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)*;; *треугольник* ∍ *Треугк(ТчкВ;ТчкС;ТчкD)*;; *Треугк(ТчкВ;ТчкС;ТчкD)* <= *декомпозиция фигуры*\*: *link1*;; *link1* ∍ *Отр(ТчкВ;ТчкС)*; *Отр(ТчкC;ТчкD)*; *Отр(ТчкВ;ТчкD)*;;

Обратим внимание, что по сравнению с представлением данного текста в *SCs-коде уровня 3* количество *sc.s-предложений* уменьшилось за счет объединения нескольких схожих *sc.s-предложений уровня 3* в одно *sc.s-предложение уровня 4* с использованием разделителя ;.

**SCs-код 5-го уровня**

На данном уровне *SCs-кода* вводятся средства, позволяющие расширить радиус семантической окрестности описываемого *sc-элемента* благодаря использованию так называемых встроенных предложений уровня 4, описывающих семантические окрестности любых *sc-элементов*, связанных с описываемым *sc-элементом*.

В ***sc.s-предложениях уровня 5*** сокращается число отображаемых имен *sc-элементов* за счет введения *встроенных sc.s-предложений* при помощи разделителя (\*...\*). Более подробно правила использования данного разделителя описаны в соответствующем

разделе. Следует отметить, что использование встроенных предложений допускается только по отношению к последнему (самому правому) элементу *sc.s-предложения* (в том

числе, встроенного), то есть запись вида *x* (\* -> *z*;;\*) -> *y*;; не допустима, правильно будет

в данном случае писать *y* <- *x* (\* -> *z*;;\*) ;;.

Встроенные предложения внутри ограничителя (\*...\*), как и все *sc.s-предложения*,

ограничиваются разделителем ;;.

Использование неатомарных идентификаторов вида <...>, {...}, [\*...\*] и [...] не

разрешается.

Пример ***sc.s-предложения уровня 5***:

*Треугк(ТчкВ;ТчкС;ТчкD)* => *сторона*\*: *Отр(ТчкВ;ТчкС)*; *Отр(ТчкС;ТчкD)*(\* => *граничная точка*\*: *ТчкС*; *ТчкD* ;; \*) ;;

***Sc.s-текст уровня 5*** - *sc.s-текст*, содержащий хотя бы одно *sc.s-предложение уровня 5* и не содержащий *sc.s-предложений уровня 6* и выше Пример ***sc.s-текста уровня 5***, семантически эквивалентного приведенному выше *sc.s-тексту уровня 4*:

*четырехугольник* ∍ *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* (\* => *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*; *Отр(ТчкС;ТчкD)*;; \*);; *треугольник* ∍ *Треугк(ТчкВ;ТчкС;ТчкD)* (\*=> *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*; *Отр(ТчкС;ТчкD)*; *Отр(ТчкB;ТчкD)*;; \*);; *Треугк(ТчкВ;ТчкС;ТчкD)* <= *декомпозиция фигуры*\*: *link1* (\* ∍ *Отр(ТчкВ;ТчкС)*; *Отр(ТчкC;ТчкD)*; *Отр(ТчкВ;ТчкD)*;; \*);;

Обратим внимание, что по сравнению с представлением данного текста в *SCs-коде уровня 4* количество *sc.s-предложений* уменьшилось за счет объединения нескольких схожих *sc.s-предложений уровня 4* в одно *sc.s-предложение уровня 5* с использованием *встроенных предложений* и ограничителя (\*…\*).

**SCs-код 6-го уровня**

На данном уровне *SCs-кода* для именования *sc-структур* и *sc-ссылок* (sc-файлов) разрешается использование таких специальных неатомарных имен *sc-элементов*, как

1) имя-контур *sc-структуры*

2) имя-рамка *sc-файла*

Имя-рамка (*sc.s-рамка*) представляет собой любой линейный текст, заключенный в

квадратные скобки [...].

Имя-контур (*sc.s-контур*) представляет собой *sc.s-текст*, заключенный в квадратные скобки со звездочками [\*...\*]. При этом в sc.s-контуре допускается запись только *sc.s-текстов*.

Использование указанных неатомарных имен превращает *sc.s-предложения* в сложные, в состав которых могут входить другие *sc.s-предложения* и даже целые *sc.s-тексты*. При этом на данном уровне разрешается использовать все виды неатомарных

идентификаторов, например <...> и {...}.

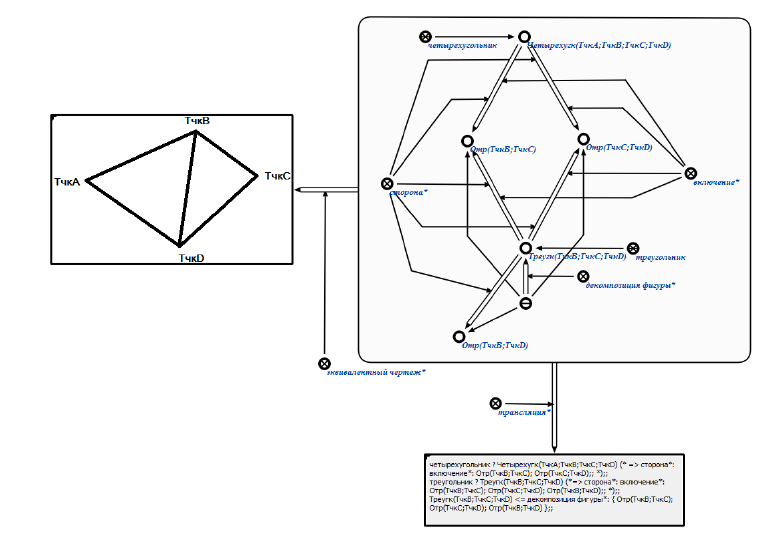
В ***sc.s-предложениях уровня 6*** разрешается использовать любые виды неатомарных идентификаторов.

Пример ***sc.s-предложения уровня 6***, семантически включающегося в приведенный выше *sc.s-текст уровня 5*:

[*Sc-элемент Треугк(ТчкВ;ТчкС;ТчкD)* связан выходящими из него дугами, принадлежащими бинарным отношениям *включение\** и *сторона\** с *sc-элементами Отр(ТчкВ;ТчкС), Отр(ТчкВ;ТчкD)* и *Отр(ТчкC;ТчкD). Sc-элемент Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* связан выходящими из него дугами, принадлежащими бинарным отношениям *включение\** и *сторона\** с *sc-элементами Отр(ТчкВ;ТчкС)* и *Отр(ТчкC;ТчкD). Множество, содержащее элементы Отр(ТчкВ;ТчкС), Отр(ТчкВ;ТчкD)* и *Отр(ТчкC;ТчкD) связано с sc-элементом Треугк(ТчкВ;ТчкС;ТчкD)* отношением *декомпозиция фигуры\*. Sc-элемент Треугк(ТчкВ;ТчкС;ТчкD)* является элементом множества *треугольник*. *Sc-элемент Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* является элементом множества *четырехугольник*.]

<= *трансляция*\*: [\* *четырехугольник* ∍ *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* (\* => *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*; *Отр(ТчкС;ТчкD)*;; \*);; *треугольник* ∍ *Треугк(ТчкВ;ТчкС;ТчкD)* (\*=> *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*; *Отр(ТчкС;ТчкD)*; *Отр(ТчкB;ТчкD)*;; \*);; *Треугк(ТчкВ;ТчкС;ТчкD)* <= *декомпозиция фигуры*\*: { *Отр(ТчкВ;ТчкС)*; *Отр(ТчкC;ТчкD)*; *Отр(ТчкВ;ТчкD)* };; \*] (\* => *эквивалентный чертеж*\* : “file://triangle.png” \*);;

Данный *sc.s-текст* семантически эквивалентен следующему *sc.g-тексту*:



Обратим внимание, что по сравнению с представлением аналогичного текста в *SCs-коде уровня 5*, данный текст содержит сложные идентификаторы вида {…}, […], [\*…\*], что позволяет явно записывать содержимое некоторых *sc-ссылок* в рамках *sc.s-текста*, более лаконично записывать множества из нескольких элементов, а также использовать понятие *sc.s-контура*, аналогичное *sc.g-контуру*.

Другие примеры ***sc.s-предложений уровня 6***:

*треугольник* <= *разбиение*\*:

{

*разностронний треугольник*;

*строго равнобедренный треугольник*;

*равностронний треугольник*

}

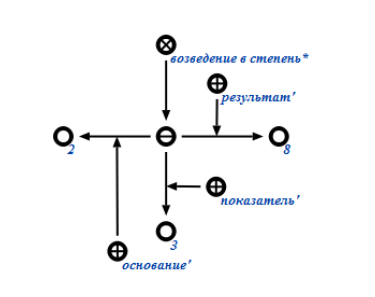
*возведение в степень*\* -> {*основание'* : *2*; *показатель'* : *3*; *степень'* : *8*};;

***sc.s-текст уровня 6*** - *sc.s-текст*, содержащий хотя бы одно *sc.s-предложение уровня 6* и не содержащий *sc.s-предложений уровня 7*

Пример ***sc.s-текста уровня 6***:

*возведение в степень*\* -> {*основание'* : *2*; *показатель'* : *3*; *степень'* : *8*};;

Данный текст семантически эквивалентен следующему *sc.g-тексту*:



На естественном языке текст, семантически эквивалентный указанным выше, звучит так:

«В связку отношения *возведение в степень\** входит sc-элемент *2* под атрибутом *основание'*, sc-элемент *3* под атрибутом *показатель'*, sc-элемент *8* под атрибутом *результат'*.»

**SCs-код 7-го уровня**

На данном уровне *SCs-кода* разрешается перечисление (через разделитель ;) таких

компонентов, которые различным образом связаны с описываемым компонентом *sc.s-предложения* (т.е. разные *sc.s-коннекторы* и/или их модификаторы).

В ***sc.s-предложениях уровня 7*** разрешается использовать *sc.s-разделитель* ; во всех

случаях, для которых предназначен данный разделитель. Более подробно правила использования указанного разделителя рассмотрены ниже в соответствующем разделе.

Правила умолчания (неотображения) разделителей *SCs-кода*:

1) После разделителя ; одинаковые изображения *sc-коннекторов* опускаются. Например, вместо X -> r1: r2: Y; -> r3: Z;; отображается X -> r1: r2: Y; r3: Z;;

2) Если после *sc.s-разделителя* ; присутствуют не только одинаковые изображения

*sc-коннекторов*, но и одинаковые модификаторы коннекторов, то их имена также опускаются. Например, вместо X -> r1: r2: Y; -> r1: r2: Z;; пишется X -> r1: r2: Y; Z;;

3) Ограничитель встроенных предложений (\*...\*) может опускаться в случае, если

встроенное предложение не содержит разделителя ;. Например, вместо X -> r1: r2: Y(\* =>

r3: Z \*);; пишется X -> r1: r2: Y => r3: Z;;

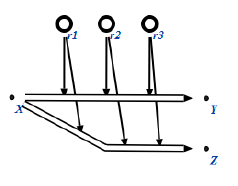
Пример ***sc.s-предложения уровня 7***, семантически эквивалентного приведенному выше *sc.s-предложению уровня 6*:

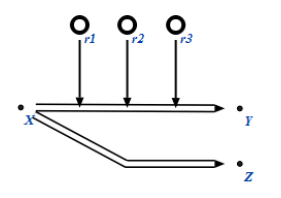
[\* *четырехугольник* ∍ *Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* (\* => *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*; *Отр(ТчкС;ТчкD)*;; \*);; *треугольник* ∍ *Треугк(ТчкВ;ТчкС;ТчкD)* (\*=> *сторона*\*: *включение*\*: *Отр(ТчкВ;ТчкС)*; *Отр(ТчкС;ТчкD)*; *Отр(ТчкB;ТчкD)*;; \*);; *Треугк(ТчкВ;ТчкС;ТчкD)* <= *декомпозиция фигуры*\*: { *Отр(ТчкВ;ТчкС)*; *Отр(ТчкC;ТчкD)*; *Отр(ТчкВ;ТчкD)* };; \*] => *эквивалентный чертеж*\* : “file://triangle.png” ; => *трансляция*\*: [*Sc-элемент Треугк(ТчкВ;ТчкС;ТчкD)* связан выходящими из него дугами, принадлежащими бинарным отношениям *включение\** и *сторона\** с *sc-элементами Отр(ТчкВ;ТчкС), Отр(ТчкВ;ТчкD)* и *Отр(ТчкC;ТчкD). Sc-элемент Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* связан выходящими из него дугами, принадлежащими бинарным отношениям *включение\** и *сторона\** с *sc-элементами Отр(ТчкВ;ТчкС)* и *Отр(ТчкC;ТчкD). Множество, содержащее элементы Отр(ТчкВ;ТчкС), Отр(ТчкВ;ТчкD)* и *Отр(ТчкC;ТчкD) связано с sc-элементом Треугк(ТчкВ;ТчкС;ТчкD)* отношением *декомпозиция фигуры\*. Sc-элемент Треугк(ТчкВ;ТчкС;ТчкD)* является элементом множества *треугольник*. *Sc-элемент Четырехугк(ТчкА;ТчкВ;ТчкС;ТчкD)* является элементом множества *четырехугольник*.] ;;

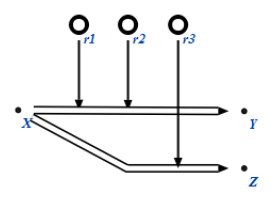
Обратим внимание, что по сравнению с представлением данного *sc.s-предложения* в *SCs-коде уровня 6* запись стала более лаконичной за счет объединения и удаления схожих фрагментов *sc.s-предложений уровня 6*.

***SC.s-текст уровня 7*** - *sc.s-текст*, содержащий хотя бы одно *sc.s-предложение уровня 7* Примеры ***sc.s-текстов уровня 7***:

*X* => *r1*: *r2*: *r3*: *Y*; *Z* ;;

Данный текст семантически эквивалентен следующему *sc.g-тексту*: 

Данный текст семантически эквивалентен следующему *sc.g-тексту*: 

Данный текст семантически эквивалентен следующему *sc.g-тексту*: 

Далее приводится подробное описание всех *sc.s-коннекторов*, используемых в *sс.s-текстах* различных уровней.

**Базовые sc.s-коннекторы**

|  |  |  |
| --- | --- | --- |
| **Синтаксический**  **тип sc.g-элемента** | **Алфавит элементов**  **SCg-кода уровня 1** | **Отображение**  **sc.s-коннектора в**  **стандартных**  **текстовых редакторах** |
| sc.g-ребро | D:\bsuir\kursach\документация\scs10.png | **↔** |
| <>sc.g-дуга  общего вида | D:\bsuir\kursach\документация\scs11.png | **→**  **←** |
| sc.g-дуга  принадлежности  >  < | D:\bsuir\kursach\документация\scs12.png | **..∍**  **D:\bsuir\kursach\документация\scs9.png** |
| Константная позитивная стационарная sc.g-дуга | D:\bsuir\kursach\документация\scs13.png | **∍** |

**Дополнительные sc.s-коннекторы**

|  |  |  |
| --- | --- | --- |
| **Синтаксический**  **тип sc.g-элемента** | **Дополнительный**  **алфавит элементов**  **SCg-кода** | **Отображение**  **sc.s-коннектора в**  **стандартных**  **текстовых редакторах** |
| Константное sc.g-ребро |  | D:\bsuir\kursach\документация\scs21.png |
| Переменное sc.g-ребро  <=> |  | D:\bsuir\kursach\документация\scs22.png |
| Константная sc.g-дуга  \_<=> |  | D:\bsuir\kursach\документация\scs14.png |
| Переменная sc.g-дуга  =>  <= |  | **D:\bsuir\kursach\документация\scs15.png** |
| Переменная позитивная  стационарная  sc.g-дуга принадлежности  \_=>  \_<= |  | D:\bsuir\kursach\документация\scs16.png |
| Константная негативная  Стационарная  sc.g-дуга принадлежности  \_->  \_<- |  | D:\bsuir\kursach\документация\scs17.png |
| Константная нечеткая стационарная  sc.g-дуга принадлежности  ***\_-|>***  ***\_<|-*** |  | D:\bsuir\kursach\документация\scs18.png |
| Переменная нечеткая стационарная |  | D:\bsuir\kursach\документация\scs19.png |
| sc.g-дуга  принадлежности  ***-/>*** |  | D:\bsuir\kursach\документация\scs20.png |

**Специальные sc.s-коннекторы**

|  |  |  |
| --- | --- | --- |
| **Название**  **разделителя** | **Семантика разделителя** | **Отображение**  **sc.s-коннектора** |
| константный  sc.s-коннектор  включение | Изображение связки отношения включение\*  Пример:  треугольник ⊂ многоугольник ;;  действительное число ⊃ натуральное число;; | **⊂**  **⊃** |
| переменный  sc.s-коннектор  включение | Изображение переменной связки отношения включение\*  Пример:  v1 \_⊂ v2;;  =  v1 \_=> включение\*:: v2;;  <=> семантическая эквивалентность\*: | **\_⊂**  **\_⊃** |
| константный  sc.s-коннектор  невключение |  | Изображение связки, не  Принадлежащей отношению включение\*  Пример:  треугольник ⊄ правильный  многоугольник;;  натуральное число ⊅ целое  число;; |
| переменный  sc.s-коннектор  невключение⊄  ⊅ |  | Изображение связки, не  принадлежащей отношению включение\*  Пример:  v1 \_⊄ v2;;  =  включение\* \_∌  (v1 \_=> v2);;  <=>  семантическая  эквивалентность\*: |
| константный  sc.s-коннектор  пересекающихся  множеств \_⊄  \_⊅ | Изображение связки отношения пара  пересекающихся множеств\*  Пример:  треугольник ⊃⊂ правильный многоугольник;;  натуральное число ⊃⊂ целое число;; | **⊃⊂** |
| переменный  sc.s-коннектор  пересекающихся  множеств | Изображение связки отношения пара  пересекающихся множеств\*  Пример:  v1 \_⊃⊂ v2;;  =  v1 \_<=> пара пересекающихся множеств\*:: v2;;  <=> семантическая эквивалентность\*: | **\_⊃⊂** |
| константный  sc.s-коннектор  непересекающихся  множеств | Изображение связки, не принадлежащей  отношению пара пересекающихся множеств\*  Пример:  треугольник ⊃|⊂ квадрат;;  натуральное число ⊃|⊂ иррациональное число;; | **⊃|⊂** |
| переменный  sc.s-коннектор  непересекающихся  множеств | Изображение связки, не принадлежащей  отношению пара пересекающихся множеств\*  Пример:  v1 \_⊃|⊂ v2;;  =  пара пересекающихся множеств\* \_∌ (v1 \_<=> v2);;  <=> семантическая эквивалентность\*: | **\_⊃|⊂** |
| sc.s-коннектор  идентификации | Изображение связки отношения  идентификатор\*.  Разделяет имя sc-объекта и имя файла, в котором хранится один из возможных идентификаторов указанного sc-элемента. Каждому sc-элементу может быть поставлено в соответствие несколько его идентификаторов.  Пример:  треугольник =∙ ["triangle"];; | **=∙**  **∙=** |
| sc.s-коннектор  семантически  эквивалентных  файлов | Изображение связки отношения семантическая эквивалентность\*. Разделяет имена двух файлов, которые являются семантически эквивалентными.  Пример:  ["треугольник"] ≗ ["triangle"];; | **≗** |
| sc.s-коннектор  семантически  неэквивалентных файлов | Изображение связки, не принадлежащей  отношению семантически эквивалентные файлы\*.  Разделяет имена двух файлов, которые не являются семантически эквивалентными, если есть необходимость подчеркнуть наличие  семантической разницы между содержимым некоторых файлов.  Пример:  ["линейный треугольник"] \*\*\* ["треугольник"];; |  |
| sc.s-коннектор  трансляции | Изображение связки, принадлежащей  отношению трансляция\*.  Разделяет обозначение sc-графа и имя файла, семантически эквивалентного данному sc-графу.  Пример:  [Треугк(ТчкА;ТчкВ;ТчкС) <- прямоугольный  треугольник] ≗∙ ["Треугольник  Треугк(ТчкА;ТчкВ;ТчкС) является прямоугольным"]  ;; | **≗∙**  **∙≗** |
| sc.s-коннектор  семантического  включения | Изображение связки, принадлежащей  отношению семантическое включение\*.  Разделяет имя файла и обозначение sc-графа,  которые связаны указанным отношением, либо  имена двух файлов, связанных указанным отношением.  Пример:  ["Треугольник Треугк(ТчкА;ТчкВ;ТчкС) является  прямоугольным"] ≗> [Треугк(ТчкА;ТчкВ;ТчкС) <-  прямоугольный треугольник; равнобедренный  треугольник;;];; | **≗>**  **<≗** |
| sc.s-коннектор  равенства | Разделяет имена синонимичных sc-элементов  Пример:  Поток 72170x = {721701;721702;721703};; | **=** |
| sc.s-коннектор  неравенства | Разделяет имена sc-элементов, не являющихся синонимичными.  Это иногда требуется, чтобы подчеркнуть наличие семантической разницы между двумя знаками.  Пример:  треугольник ≠ линейный треугольник;; | **≠** |