

# Документация scr-машины

# **Агент обработки события в sc-памяти, инициирующего агентную scr-программу**

**:=** [sc-агент обработки события в sc-памяти, инициирующего агентную scr-программу]

**⇒** *примечание\**:

[Первичным условием инициирования агента обработки события в sc-памяти, инициирующего агентную scr-программу, является элементарное sc-событие, указанное в первичном условии инициирования абстрактного sc-агента, агентная scr-программа которого инициируется.]

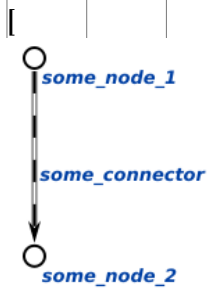
**⇒** *примечание\**:

[Агент обработки события в sc-памяти, инициирующего агентную scr-программу, используется для осуществления логики scr-агентов: реагирование на событие, указанное в первичном условии инициирования и запуск интерпретации scr-программы.]

**⇒** *задачи\**:

- [Среагировать на событие, указанное в первичном условии инициирования абстрактного sc-агента.]
- [Найти в базе знаний scr-программу, указанную во множестве программ соответствующего абстрактного sc-агента.]
- [Начать выполнение действия класса `action_scr_interpretation_request`, указав в качестве аргументов действия найденную scr-программу и ориентированное множество аргументов этой scr-программы, состоящее из двух элементов: scr-программы и sc-дуги, изменение которой инициировало агента обработки события в sc-памяти, инициирующего агентную scr-программу.]

**⇒** *пример входной конструкции\**:



**⇒** *пояснение\**:

[Первичным условием инициирования агента обработки события в sc-памяти, инициирующего агентную scr-программу, может быть любое элементарное sc-событие.]

**⇒** *аргументы агента\**:

- пустое множество

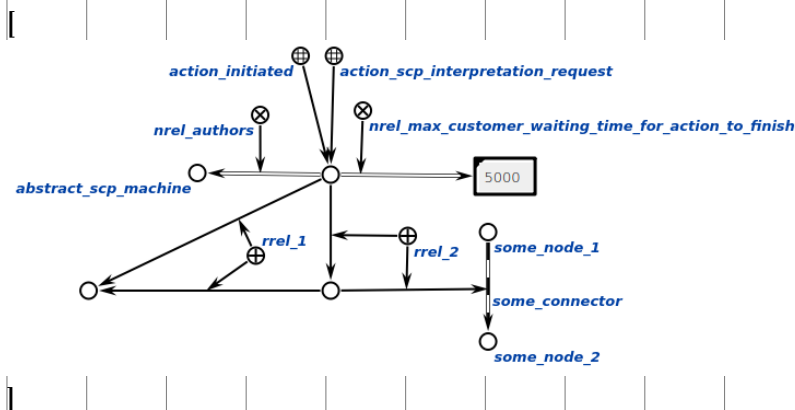
**⇒** *ответ агента\**:

ответ агента обработки события в sc-памяти, инициирующего агентную scr-программу

**⇒** *примечание\**:

[В результате выполнения агентом результат действия никак не меняется, в результате находятся только те элементы, которые были в него добавлены в процессе интерпретации scr-программы.]

**⇒** *пример выходной конструкции\**:



**Агент активации scr-агента**

**:=** [sc-агент активации scr-агента]

**⇒** *примечание\**:

[Первичным условием инициирования агента активации scr-агента является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_sc\_agent.]

**⇒** *задачи\**:

⟨ • [Подписать активного агента на событие, указанное в его первичном условии инициирования.]  
⟩

**⇒** *пример входной конструкции\**:



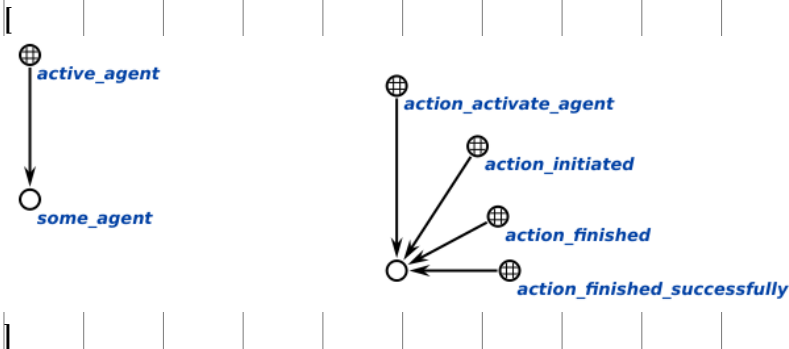
**⇒** *аргументы агента\**:

{ • пустое множество  
}

**⇒** *ответ агента\**:

ответ агента активации scr-агента  
= пустое множество

**⇒** *пример выходной конструкции\**:

**Агент деактивации scr-агента**

**:=** [sc-агент деактивации scr-агента]

**⇒** *примечание\**:

[Первичным условием инициирования агента деактивации scr-агента является удаление выходящей константной постоянной sc-дуги принадлежности из класса active\_sc\_agent.]

**⇒** *задачи\**:

⟨ • [Отписать активного агента от его первичного условия инициирования.]  
⟩

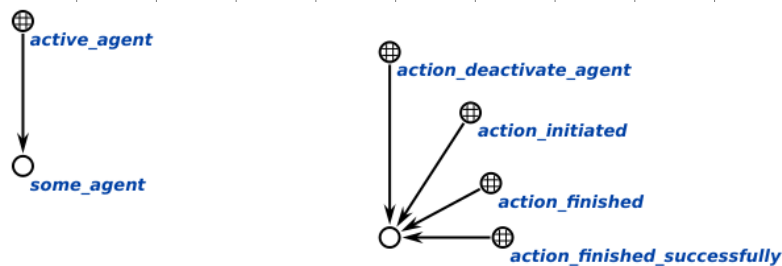
**⇒** *пример входной конструкции\**:



**⇒** *аргументы агента\**:

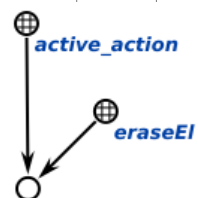
{ • пустое множество  
}

⇒ }  
 ⇒ ответ агента\*:  
 ответ агента деактивации scr-агента  
 = пустое множество  
 ⇒ пример выходной конструкции\*:  
 [

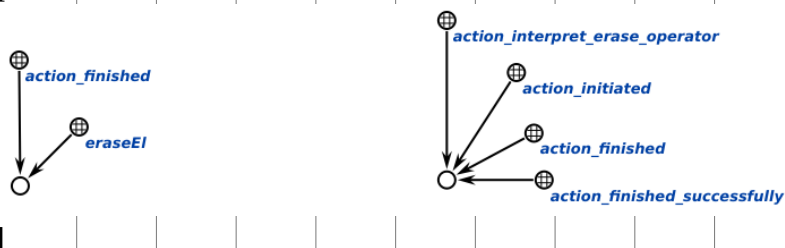


### Агент интерпретации оператора удаления конструкций

:= [sc-агент интерпретации оператора удаления конструкций]  
 ⇒ примечание\*:  
 [Первичным условием инициирования агента интерпретации оператора удаления конструкций является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]  
 ⇒ реализованные операторы\*:  
 {  
 • scr-оператор удаления одноэлементной конструкции  
 • scr-оператор удаления трехэлементной конструкции  
 • scr-оператор удаления пятиэлементной конструкции  
 • scr-оператор удаления множества элементов трехэлементной конструкции  
 • scr-оператор удаления множества элементов пятиэлементной конструкции  
 }  
 ⇒ задачи\*:  
 <• [Интерпретация активного оператора удаления конструкций.]  
 >  
 ⇒ пример входной конструкции\*:  
 [



⇒ ]  
 ⇒ аргументы агента\*:  
 {  
 • пустое множество  
 }  
 ⇒ ответ агента\*:  
 ответ агента интерпретации оператора удаления конструкций  
 = пустое множество  
 ⇒ пример выходной конструкции\*:  
 [



### Агент интерпретации оператора генерации конструкций

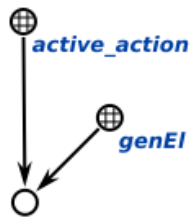
:= [sc-агент интерпретации оператора генерации конструкций]

⇒ *примечание\**:  
[Первичным условием инициирования агента интерпретации оператора генерации конструкций является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]

⇒ *реализованные операторы\**:  
 {
 • scr-оператор генерации конструкции по произвольному образцу  
 • scr-оператор генерации пятиэлементной конструкции  
 • scr-оператор генерации трехэлементной конструкции  
 • scr-оператор генерации одноэлементной конструкции  
 }

⇒ *задачи\**:  
 {
 • [Интерпретация активного оператора генерации конструкций.]  
 }

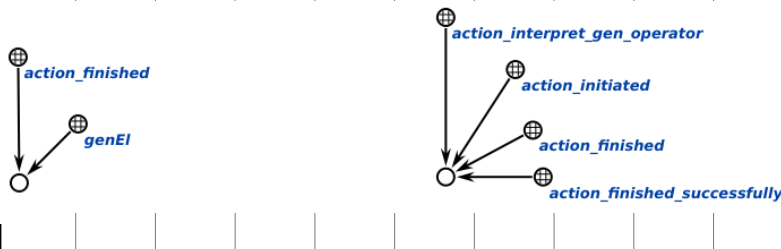
⇒ *пример входной конструкции\**:  
 [



⇒ *аргументы агента\**:  
 {
 • пустое множество  
 }

⇒ *ответ агента\**:  
 ответ агента интерпретации оператора генерации конструкций  
 = пустое множество

⇒ *пример выходной конструкции\**:  
 [



### Агент интерпретации оператора проверки условий

:= [sc-агент интерпретации оператора проверки условий]

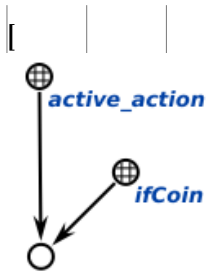
⇒ *примечание\**:  
[Первичным условием инициирования агента интерпретации оператора проверки условий является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]

⇒ *реализованные операторы\**:  
 {
 • scr-оператор сравнения числовых содержимых файлов  
 • scr-оператор проверки равенства числовых содержимых файлов  
 • scr-оператор проверки совпадения значений операндов  
 • scr-оператор проверки наличия значения у переменной  
 • scr-оператор проверки типа sc-элемента  
 }

⇒ *нереализованные операторы\**:  
 {
 • scr-оператор проверки наличия содержимого у файла  
 }

⇒ *задачи\**:  
 {
 • [Интерпретация активного оператора проверки условий.]  
 }

⇒ *пример входной конструкции\**:  
 [



⇒ аргументы агента\*:

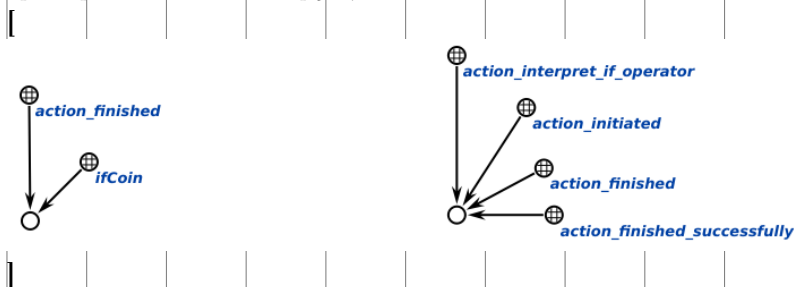
{• пустое множество}

⇒ ответ агента\*:

ответ агента интерпретации оператора проверки условий

= пустое множество

⇒ пример выходной конструкции\*:



### Агент интерпретации оператора обработки числовых содержимых файлов

:= [sc-агент интерпретации оператора обработки числовых содержимых файлов]

⇒ примечание\*:

[Первичным условием инициирования агента интерпретации оператора обработки числовых содержимых файлов является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]

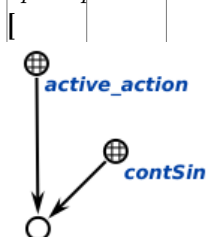
⇒ реализованные операторы\*:

- sc-оператор вычисления синуса числового содержимого файла
- sc-оператор вычисления косинуса числового содержимого файла
- sc-оператор вычисления тангенса числового содержимого файла
- sc-оператор вычисления арксинуса числового содержимого файла
- sc-оператор вычисления арккосинуса числового содержимого файла
- sc-оператор вычисления арктангенса числового содержимого файла
- sc-оператор нахождения остатка от деления числовых содержимых файлов
- sc-оператор нахождения целой части от деления числовых содержимых файлов
- sc-оператор деления числовых содержимых файлов
- sc-оператор умножения числовых содержимых файлов
- sc-оператор вычитания числовых содержимых файлов
- sc-оператор сложения числовых содержимых файлов
- sc-оператор вычисления логарифма числового содержимого файла
- sc-оператор возведения числового содержимого файла в степень

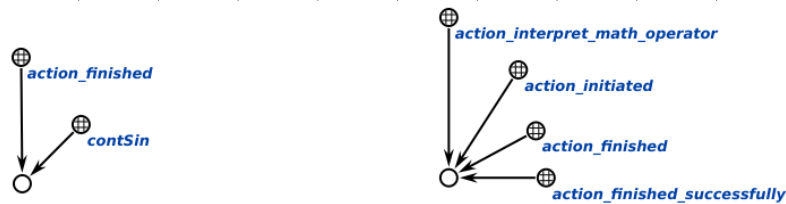
⇒ задачи\*:

- [Интерпретация активного оператора обработки числовых содержимых файлов.]

⇒ пример входной конструкции\*:

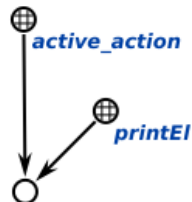


⇒ ]  
 ⇒ аргументы агента\*:  
 {• пустое множество  
 }  
 ⇒ ответ агента\*:  
 ответ агента интерпретации оператора обработки числовых содержимых файлов  
 = пустое множество  
 ⇒ пример выходной конструкции\*:  
 [

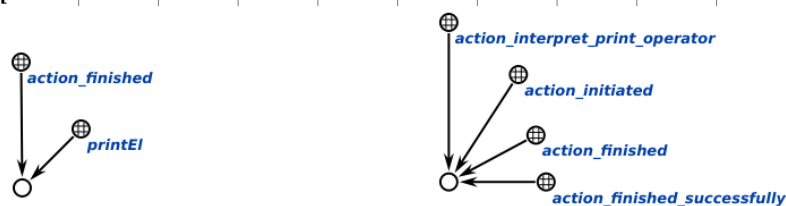


### Агент интерпретации оператора вывода в консоль

⇒ := [sc-агент интерпретации оператора вывода в консоль]  
 ⇒ примечание\*:  
 [Первичным условием инициирования агента интерпретации оператора вывода в консоль является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]  
 ⇒ реализованные операторы\*:  
 {• scr-оператор вывода содержимого файла в консоль.  
 • scr-оператор вывода содержимого файла с переводом на новую строку в консоль.  
 • scr-оператор вывода семантической окрестности элемента в консоль.  
 }  
 ⇒ задачи\*:  
 {• [Интерпретация активного оператора вывода в консоль.]  
 }  
 ⇒ пример входной конструкции\*:  
 [



⇒ ]  
 ⇒ аргументы агента\*:  
 {• пустое множество  
 }  
 ⇒ ответ агента\*:  
 ответ агента интерпретации оператора вывода в консоль  
 = пустое множество  
 ⇒ пример выходной конструкции\*:  
 [

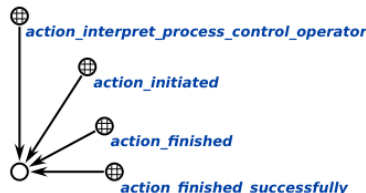
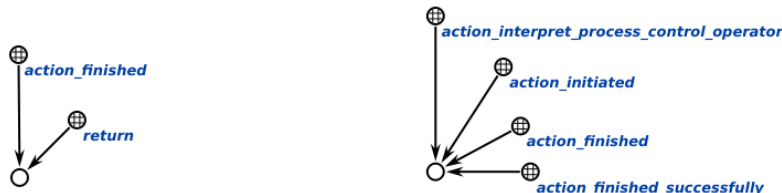


### Агент интерпретации оператора управления scr-процессами

:= [sc-агент интерпретации оператора управления scr-процессами]  
 => *примечание\**:  
 [Первичным условием инициирования агента интерпретации оператора управления scr-процессами является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]  
 => *реализованные операторы\**:  
 {  
 • scr-оператор завершения выполнения программы  
 • scr-оператор ожидания завершения выполнения scr-программы  
 • scr-оператор асинхронного вызова подпрограммы  
 • scr-оператор ожидания события  
 }  
 => *нереализованные операторы\**:  
 {  
 • конъюнкция предшествующих scr-операторов  
 • scr-оператор ожидания завершения выполнения множества scr-программ  
 }  
 => *задачи\**:  
 {  
 • [Интерпретация активного оператора управления scr-процессами.]  
 }  
 => *пример входной конструкции\**:  
 [



]=  
 => *аргументы агента\**:  
 {  
 • пустое множество  
 }  
 => *ответ агента\**:  
 ответ агента интерпретации оператора управления scr-процессами  
 = пустое множество  
 => *пример выходной конструкции\**:  
 [



### Агент интерпретации оператора ассоциативного поиска конструкций

:= [sc-агент интерпретации оператора ассоциативного поиска конструкций]  
 => *примечание\**:  
 [Первичным условием инициирования агента интерпретации оператора ассоциативного поиска конструкций является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]  
 => *реализованные операторы\**:  
 {  
 • scr-оператор поиска конструкции по произвольному образцу  
 • scr-оператор поиска пятиэлементной конструкции с формированием множеств  
 • scr-оператор поиска трехэлементной конструкции с формированием множеств  
 • scr-оператор поиска пятиэлементной конструкции  
 • scr-оператор поиска трехэлементной конструкции  
 }  
 => *задачи\**:  
 {  
 • [Интерпретация активного оператора ассоциативного поиска конструкций.]  
 }



⇒ пример входной конструкции\*:



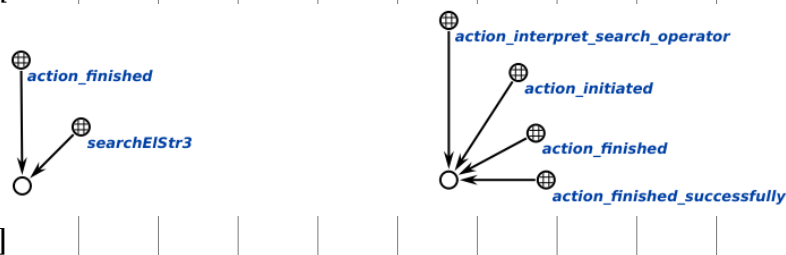
⇒ аргументы агента\*:

{• пустое множество}

⇒ ответ агента\*:

ответ агента интерпретации оператора ассоциативного поиска конструкций  
= пустое множество

⇒ пример выходной конструкции\*:



### Агент интерпретации оператора обработки строковых содержимых файлов

:= [sc-агент интерпретации оператора обработки строковых содержимых файлов]

⇒ примечание\*:

[Первичным условием инициирования агента интерпретации оператора обработки строковых содержимых файлов является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]

⇒ реализованные операторы\*:

- sc-оператор копирования содержимого файла
- sc-оператор получения части строкового содержимого файла по индексам
- sc-оператор поиска строкового содержимого файла в строковом содержимом другого файла
- sc-оператор замены определенной части строкового содержимого файла на содержимое указанной файла
- sc-оператор вычисления длины строкового содержимого файла
- sc-оператор разбиения строки на подстроки
- sc-оператор лексикографического сравнения строковых содержимых файлов
- sc-оператор проверки равенства строковых содержимых файлов
- sc-оператор проверки совпадения конца строкового содержимого файла со строковым содержимым другого файла
- sc-оператор проверки совпадения начальной части строкового содержимого файла со строковым содержимым другого файла
- sc-оператор перевода в нижний регистр строкового содержимого файла
- sc-оператор перевода в верхний регистр строкового содержимого файла

⇒ нереализованные операторы\*:

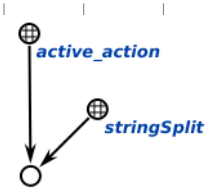
- sc-оператор удаления содержимого файла

⇒ задачи\*:

- [Интерпретация активного оператора обработки строковых содержимых файлов.]

⇒ пример входной конструкции\*:

[



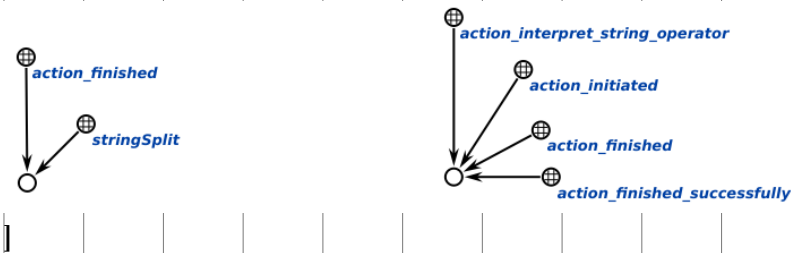
⇒ аргументы агента\*:

{• пустое множество}

⇒ ответ агента\*:

ответ агента интерпретации оператора обработки строковых содержимых файлов  
= пустое множество

⇒ пример выходной конструкции\*:



### Агент интерпретации оператора управления значениями операндов

:= [sc-агент интерпретации оператора управления значениями операндов]

⇒ примечание\*:

[Первичным условием иницирования агента интерпретации оператора управления значениями операндов является появление выходящей константной постоянной sc-дуги принадлежности из класса active\_action.]

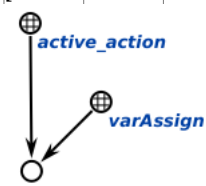
⇒ реализованные операторы\*:

{• scp-оператор удаления значения переменной  
• scp-оператор присваивания значения переменной}

⇒ задачи\*:

<• [Интерпретация активного оператора управления значениями операндов.]>

⇒ пример входной конструкции\*:



⇒ аргументы агента\*:

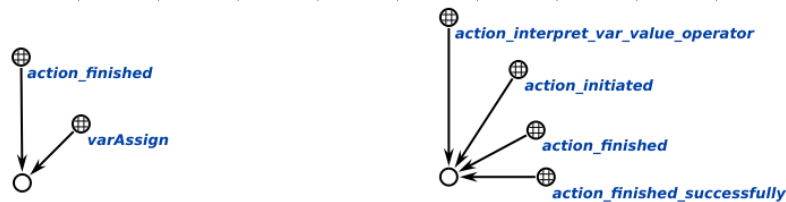
{• пустое множество}

⇒ ответ агента\*:

ответ агента интерпретации оператора управления значениями операндов  
= пустое множество

⇒ пример выходной конструкции\*:

[



### Агент синхронизации процесса интерпретации scp-программ

**:=** [sc-агент синхронизации процесса интерпретации scp-программ]

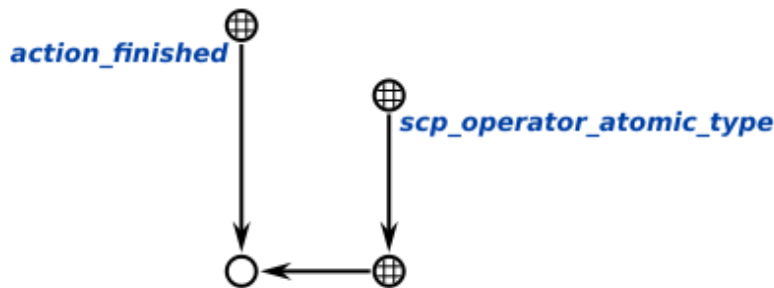
**⇒** *примечание\**:

[Первичным условием инициирования агента синхронизации процесса интерпретации scp-программ является появление выходящей константной постоянной sc-дуги принадлежности из класса action\_finished.]

**⇒** *задачи\**:

- ⟨ • [Добавление операторов, являющихся концом дуг, принадлежащих отношению nrel\_error, начальным элементом которых является завершённый оператор, во множество active\_action, если завершённый оператор был завершён с ошибкой.]
- [Добавление операторов, являющихся концом дуг, принадлежащих отношению nrel\_then, начальным элементом которых является завершённый оператор, во множество active\_action, если завершённый оператор был завершён успешно.]
- [Добавление операторов, являющихся концом дуг, принадлежащих отношению nrel\_else, начальным элементом которых является завершённый оператор, во множество active\_action, если завершённый оператор был завершён безуспешно.]
- [Добавление операторов, являющихся концом дуг, принадлежащих отношению nrel\_goto, начальным элементом которых является завершённый оператор, во множество active\_action, если завершённый оператор не был завершён с ошибкой.]

**⇒** *пример входной конструкции\**:



**⇒** *аргументы агента\**:

- { • пустое множество

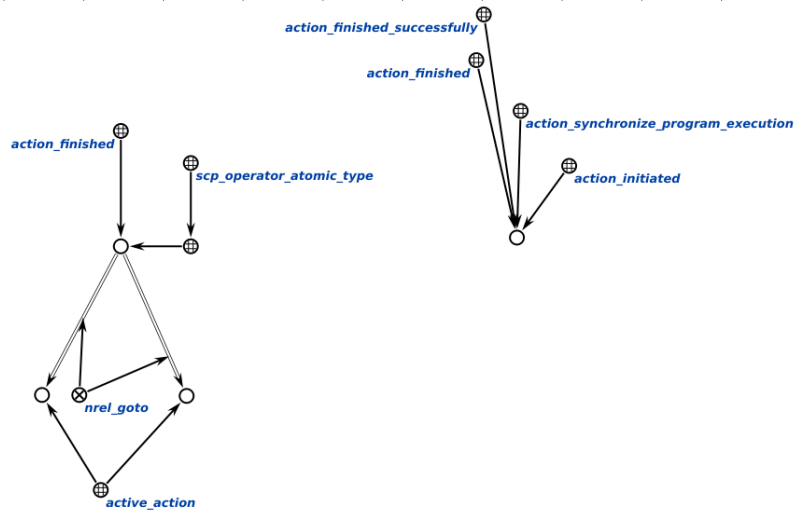
**⇒** *ответ агента\**:

ответ агента синхронизации процесса интерпретации scp-программ

= пустое множество

**⇒** *пример выходной конструкции\**:

[



### Агент интерпретации процесса

:= [sc-агент интерпретации процесса]

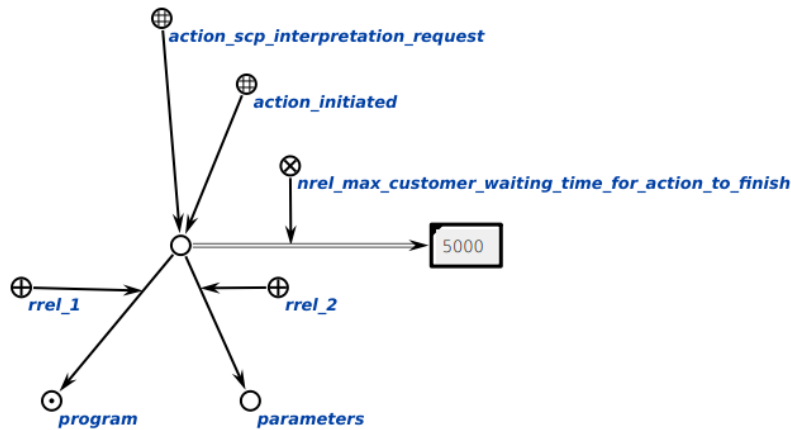
⇒ примечание\*:

[Первичным условием инициирования агента интерпретации процесса является появление выходящей константной постоянной sc-дуги принадлежности из класса action\_initiated.]

⇒ задачи\*:

- [Сгенерировать процесс по шаблону указанной scr-программы, подставив указанные аргументы вместо параметров scr-программы.]
- [Инициировать интерпретацию первого оператора процесса.]
- [Подождать завершения интерпретации процесса.]

⇒ пример входной конструкции\*:



⇒ аргументы агента\*:

- scr-программа
- множество аргументов scr-программы

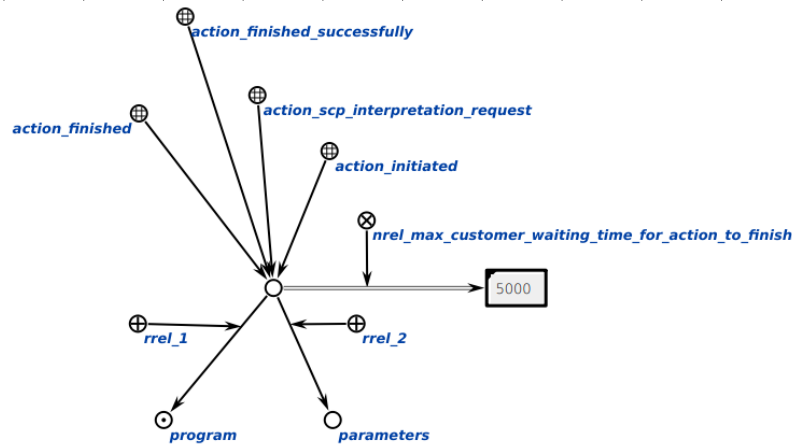
⇒ ответ агента\*:

ответ агента интерпретации процесса

= пустое множество

⇒ пример выходной конструкции\*:

[



### Агент уничтожения scp-процессов

**:=** [sc-агент уничтожения scp-процессов]

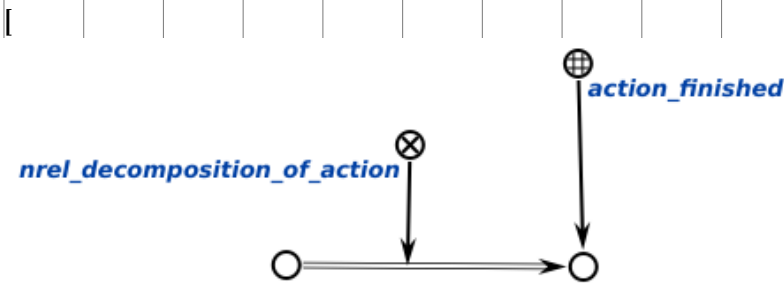
**⇒** *примечание\**:

[Первичным условием инициирования агента уничтожения scp-процессов является появление выходящей константной постоянной sc-дуги принадлежности из класса action\_finished.]

**⇒** *задачи\**:

- ⟨ • [Удалить все операнды и множества пар, являющиеся операндами с ролями rrel\_2 и rrel\_3 у операторов процесса, принадлежащих классам sys\_search и sys\_gen.]
- [Удалить операнд с ролью rrel\_2 у операторов процесса, принадлежащих классу call.]
- [Удалить все операторы процесса.]
- [Удалить множество операторов процесса.]
- [Удалить процесс.]

**⇒** *пример входной конструкции\**:



**⇒** *аргументы агента\**:

- { • пустое множество

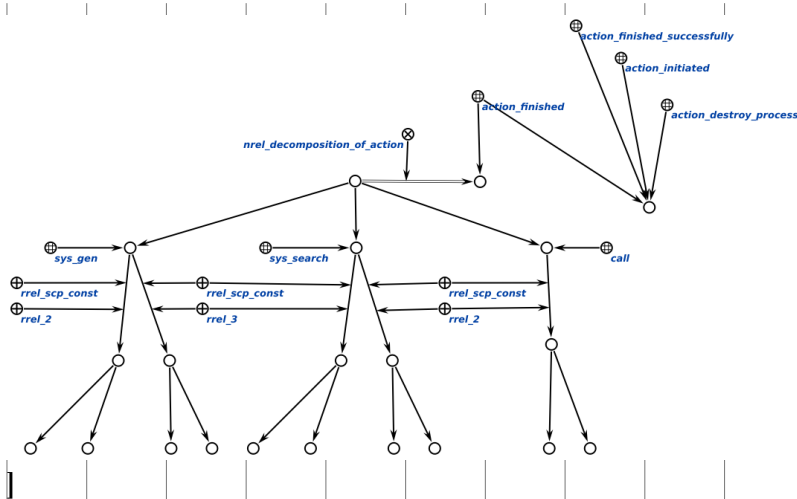
**⇒** *ответ агента\**:

ответ агента уничтожения scp-процессов

= пустое множество

**⇒** *пример выходной конструкции\**:

[



### Агент обработки завершения интерпретации действия

:= [sc-агент обработки завершения интерпретации действия]

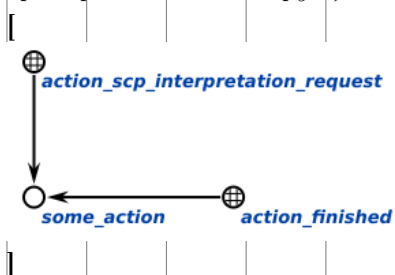
⇒ примечание\*:

[Первичным условием инициирования агента обработки завершения интерпретации действия является появление выходящей константной постоянной sc-дуги принадлежности из класса action\_finished.]

⇒ задачи\*:

⟨ • [Успешное завершение всех активных операторов, принадлежащих классу waitReturn, которые ожидали завершения интерпретации завершённого действия.]

⇒ пример входной конструкции\*:



⇒ аргументы агента\*:

{ • пустое множество }

⇒ ответ агента\*:

ответ агента обработки завершения интерпретации действия  
= пустое множество

⇒ пример выходной конструкции\*:

[

