

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.]

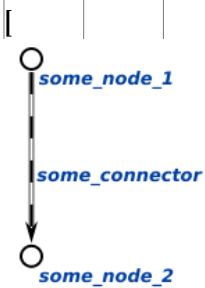
⇒ примечание*:

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

⇒ задача*:

- [Среагировать на первичное условие инициирования абстрактного sc-агента.]
- [Найти в базе знаний scr-программу, указанную во множестве программ соответствующего абстрактного sc-агента.]
- [Вызвать выполнение действия класса `action_scr_interpretation_request`, передавая в качестве аргументов действия найденную scr-программу и ориентированное множество аргументов этой scr-программы, состоящее из двух элементов: scr-программы и sc-дуги, с которой связано первичное условие инициирования.]

⇒ 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

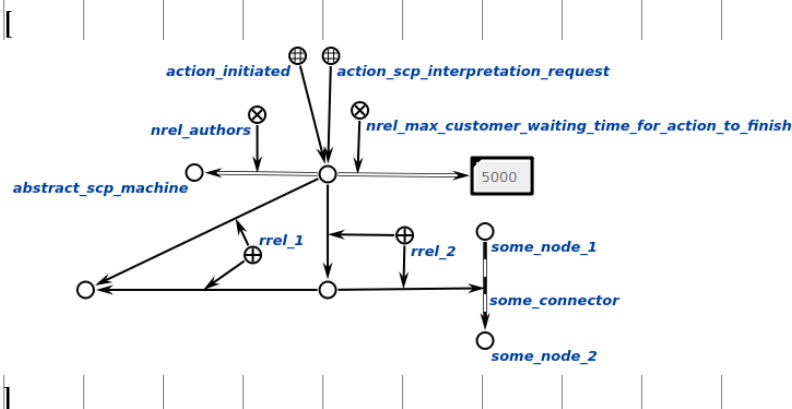
⇒ ответ агента*:

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

⇒ примечание*:

[В результате выполнения агентом ответ действия никак не меняется, в ответе 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 постоянной константной sc-дуги принадлежности из класса active_sc_agent.]

⇒ задача*:

⟨ • [Подписать активного агента на его первичное условие инициирования.]
⟩

⇒ пример входной конструкции*:



⇒ аргументы агента*:

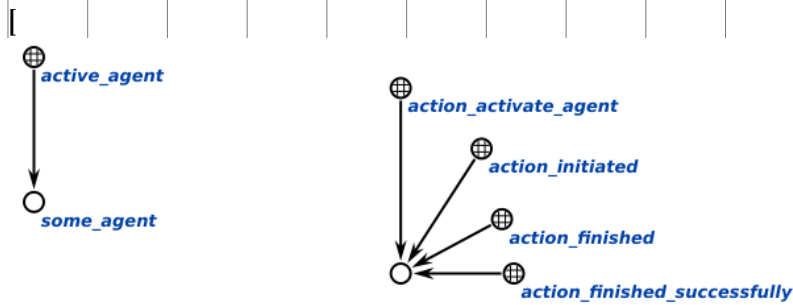
{ • пустое множество
}

⇒ 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 постоянной константной sc-дуги принадлежности из класса active_sc_agent.]

⇒ задача*:

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

⇒ пример входной конструкции*:



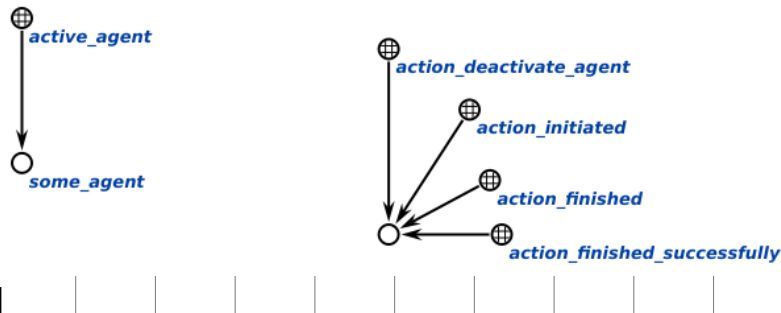
⇒ аргументы агента*:

{ • 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 постоянной константной sc-дуги принадлежности из класса active_action.]

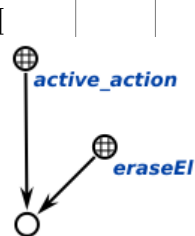
⇒ реализованные операторы*:

- 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

⇒ задача*:

- [Интерпретация активного оператора удаления конструкций.]

⇒ пример входной конструкции*:



⇒ аргументы агента*:

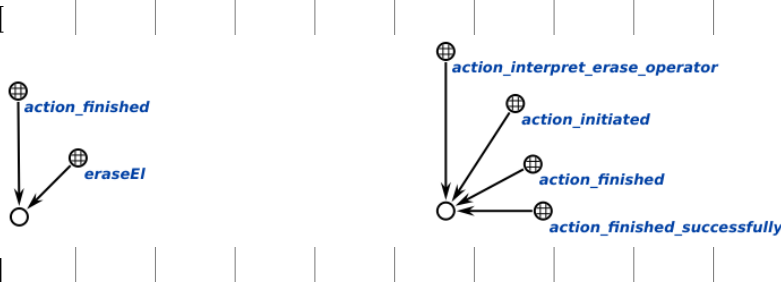
- пустое множество

⇒ 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 постоянной константной sc-дуги принадлежности из класса active_action.]

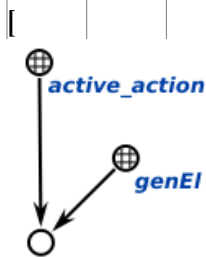
⇒ реализованные операторы*:

- scr-оператор генерации конструкции по произвольному образцу
- scr-оператор генерации пятиэлементной конструкции
- scr-оператор генерации трехэлементной конструкции
- scr-оператор генерации одноэлементной конструкции

⇒ задача*:

- [Интерпретация активного оператора генерации конструкций.]

⇒ пример входной конструкции*:



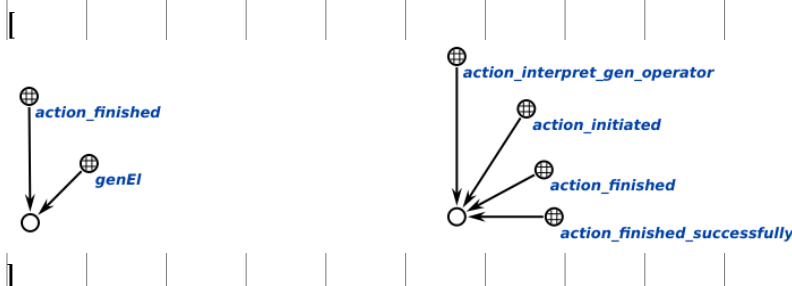
⇒ аргументы агента*:

- 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 постоянной константной 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

⇒ не реализованные операторы*:

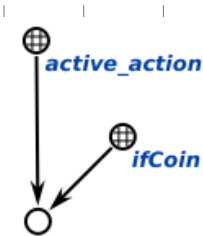
- scr-оператор проверки наличия содержимого у файла

⇒ задача*:

- [Интерпретация активного оператора проверки условий.]

⇒ пример входной конструкции*:

[



⇒ аргументы агента*:

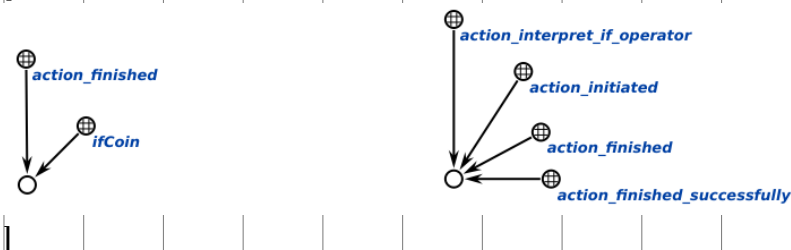
{• пустое множество}

⇒ ответ агента*:

ответ агента интерпретации оператора проверки условий
= пустое множество

⇒ пример выходной конструкции*:

[



Агент интерпретации оператора обработки содержимых файлов, связанной с математическими вычислениями

:= [sc-агент интерпретации оператора обработки содержимых файлов, связанной с математическими вычислениями]

⇒ примечание*:

[Первичным условием инициирования агента интерпретации оператора обработки содержимых файлов, связанной с математическими вычислениями является появление выходящей постоянной константной sc-дуги принадлежности из класса active_action.]

⇒ реализованные операторы*:

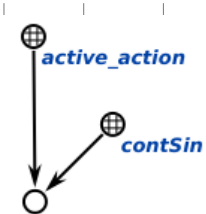
- {• scr-оператор вычисления синуса числового содержимого файла
- scr-оператор вычисления косинуса числового содержимого файла
- scr-оператор вычисления тангенса числового содержимого файла
- scr-оператор вычисления арксинуса числового содержимого файла
- scr-оператор вычисления арккосинуса числового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

⇒ задача*:

- {• [Интерпретация активного оператора обработки содержимых файлов, связанной с математическими вычислениями.]

⇒ пример входной конструкции*:

[

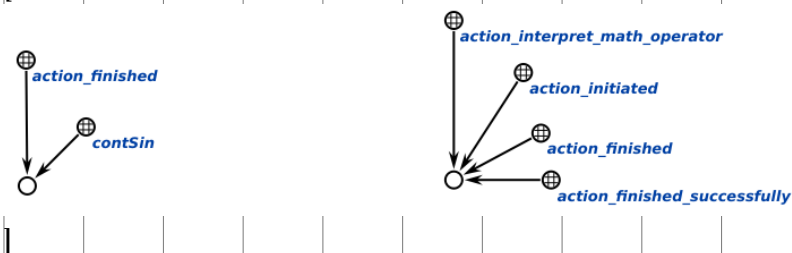


⇒]

⇒ аргументы агента*:
 {• пустое множество
 }

⇒ ответ агента*:
 ответ агента интерпретации оператора обработки содержимых файлов, связанной с математическими вычислениями
 = пустое множество

⇒ пример выходной конструкции*:
 [



Агент интерпретации оператора вывода в консоль

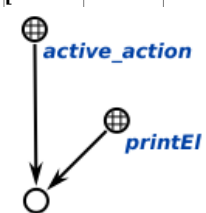
:= [sc-агент интерпретации оператора вывода в консоль]

⇒ примечание*:
 [Первичным условием инициирования агента интерпретации оператора вывода в консоль является появление выходящей постоянной константной sc-дуги принадлежности из класса active_action.]

⇒ реализованные операторы*:
 {• scc-оператор вывода содержимого файла в консоль.
 • scc-оператор вывода содержимого файла с переводом на новую строку в консоль.
 • scc-оператор вывода семантической окрестности элемента в консоль.
 }

⇒ задача*:
 {• [Интерпретация активного оператора вывода в консоль.]
 }

⇒ пример входной конструкции*:
 [

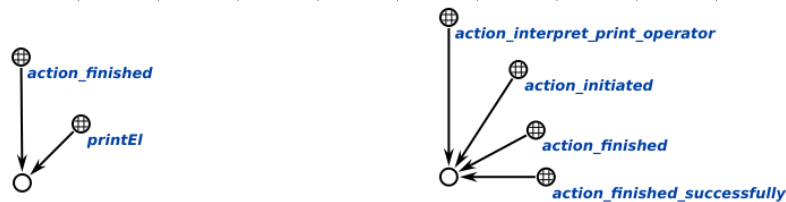


⇒]

⇒ 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 постоянной константной 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

⇒ не реализованные операторы*:

- конъюнкция предшествующих scr-операторов
- scr-оператор ожидания завершения выполнения множества scr-программ

⇒ задача*:

- [Интерпретация активного оператора управления scr-процессами.]

⇒ пример входной конструкции*:



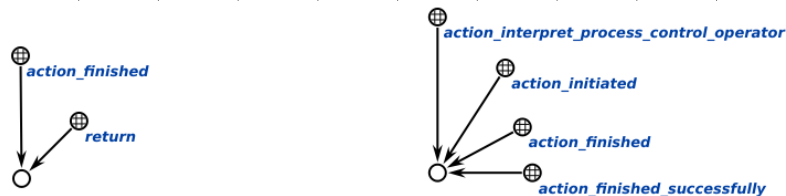
⇒ аргументы агента*:

- 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 постоянной константной 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

⇒ задача*:

⟨ • [Интерпретация активного оператора ассоциативного поиска конструкций.]

⇒ пример входной конструкции*:

[



]

⇒ аргументы агента*:

- пустое множество

⇒ ответ агента*:

ответ агента интерпретации оператора ассоциативного поиска конструкций

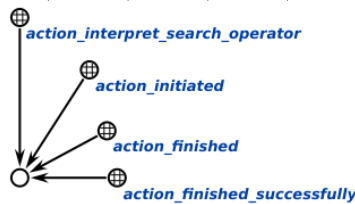
= пустое множество

⇒ пример выходной конструкции*:

[



]



Агент интерпретации оператора обработки содержимых файлов, не связанной с математическими вычислениями

:= [sc-агент интерпретации оператора обработки содержимых файлов, не связанной с математическими вычислениями]

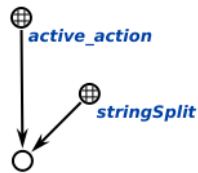
⇒ примечание*:

[Первичным условием инициирования агента интерпретации оператора обработки содержимых файлов, не связанной с математическими вычислениями является появление выходящей постоянной константной sc-дуги принадлежности из класса active_action.]

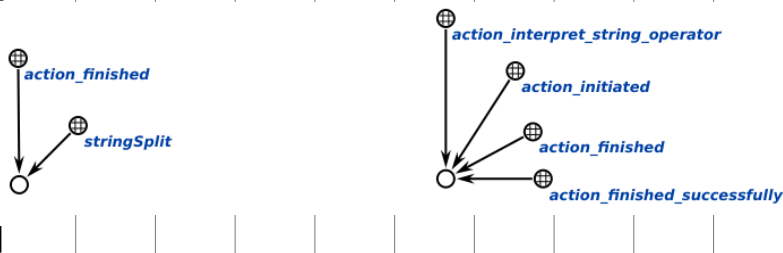
⇒ реализованные операторы*:

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

⇒ не реализованные операторы*:
 {• *scs-оператор удаления содержимого файла*
 }
 ⇒ задача*:
 {• [Интерпретация активного оператора обработки содержимых файлов, не связанной с математическими вычислениями.]
 }
 ⇒ пример входной конструкции*:
 [

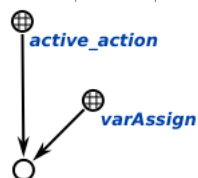


⇒ аргументы агента*:
 {• *пустое множество*
 }
 ⇒ ответ агента*:
 ответ агента интерпретации оператора обработки содержимых файлов, не связанной с математическими вычислениями
 = *пустое множество*
 ⇒ пример выходной конструкции*:
 [



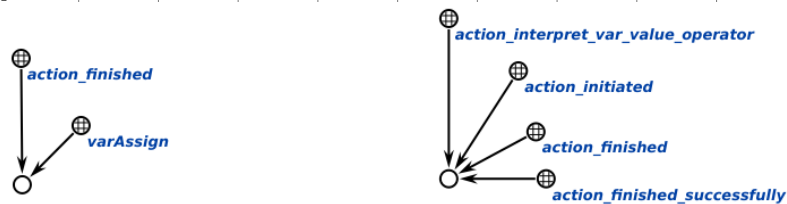
Агент интерпретации оператора управления значениями операндов

:= [scs-агент интерпретации оператора управления значениями операндов]
 ⇒ примечание*:
 [Первичным условием инициирования агента интерпретации оператора управления значениями операндов является появление выходящей постоянной константной scs-дуги принадлежности из класса *active_action*.]
 ⇒ реализованные операторы*:
 {• *scs-оператор удаления значения переменной*
 • *scs-оператор присваивания значения переменной*
 }
 ⇒ задача*:
 {• [Интерпретация активного оператора управления значениями операндов.]
 }
 ⇒ пример входной конструкции*:
 [



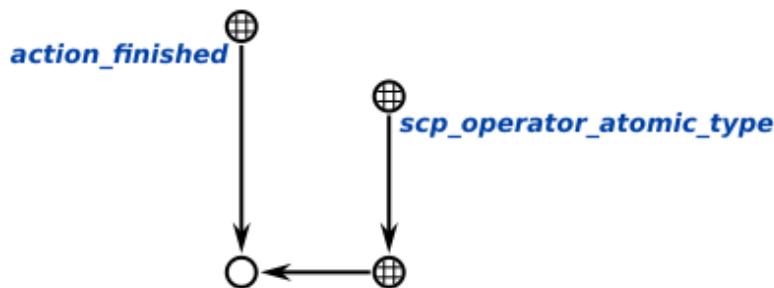
⇒ аргументы агента*:
 {• *пустое множество*
 }

⇒ }
 ⇒ ответ агента*:
 ответ агента интерпретации оператора управления значениями операндов
 = пустое множество
 ⇒ пример выходной конструкции*:
 [

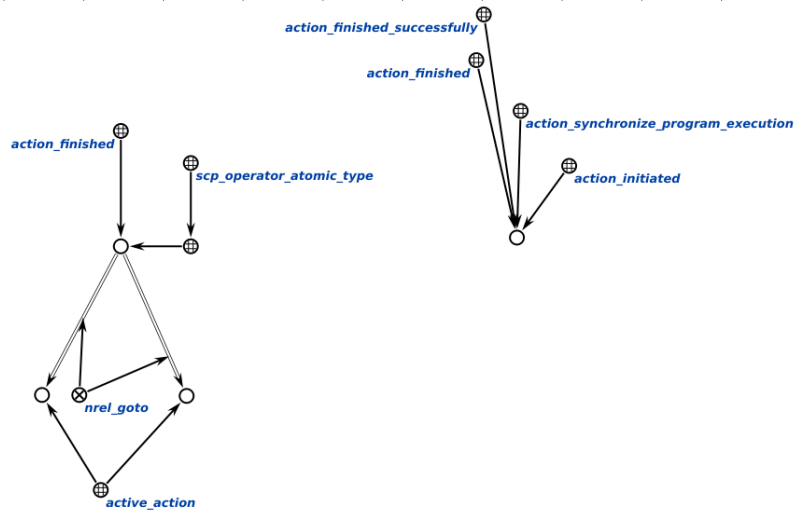


Агент синхронизации процесса интерпретации scp-программ

:= [sc-агент синхронизации процесса интерпретации scp-программ]
 ⇒ примечание*:
 [Первичным условием инициирования агента синхронизации процесса интерпретации scp-программ является появление выходящей постоянной константной sc-дуги принадлежности из класса action_finished.]
 ⇒ задача*:
 <• [Добавление операторов, связанных с завершённым оператором отношением nrel_error, во множество active_action, если завершённый оператор был завершён с ошибкой.]
 • [Добавление операторов, связанных с завершённым оператором отношением nrel_then, во множество active_action, если завершённый оператор был завершён успешно.]
 • [Добавление операторов, связанных с завершённым оператором отношением nrel_else, во множество active_action, если завершённый оператор был завершён безуспешно.]
 • [Добавление операторов, связанных с завершённым оператором отношением nrel_goto, 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]

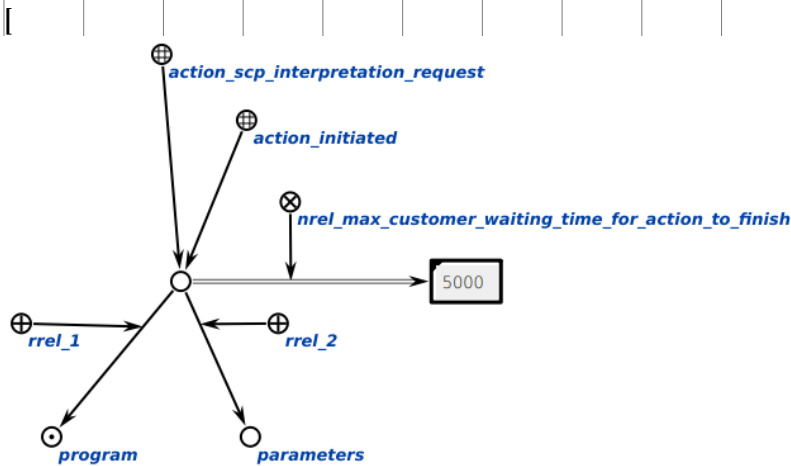
⇒ примечание*:

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

⇒ задача*:

- [Сгенерировать процесс по шаблону переданной scp-программы, используя переданные аргументы вместо аргументов scp-программы.]
- [Инициировать интерпретацию первого оператора процесса.]
- [Подождать завершения интерпретации процесса.]

⇒ пример входной конструкции*:



⇒ аргументы агента*:

- scp-программа
- множество аргументов scp-программы

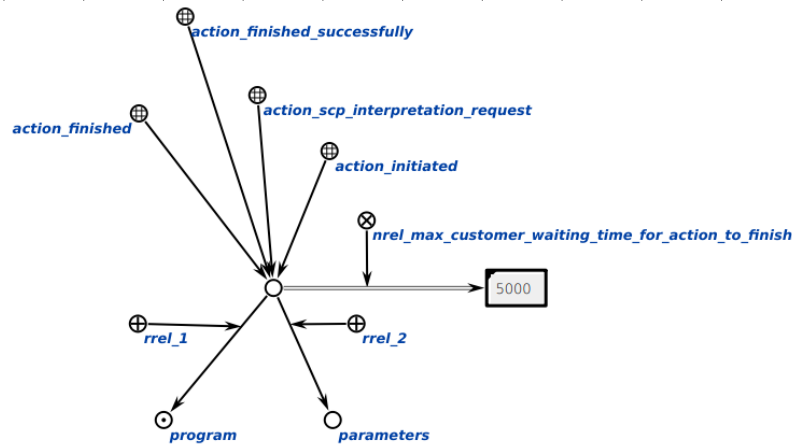
⇒ ответ агента*:

ответ агента интерпретации процесса

= пустое множество

⇒ пример выходной конструкции*:

[



]

Агент уничтожения scp-процессов

:= [sc-агент уничтожения scp-процессов]

⇒ примечание*:

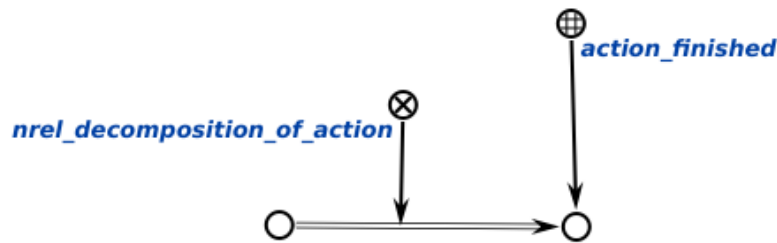
[Первичным условием инициирования агента уничтожения scp-процессов является появление выходящей постоянной константной sc-дуги принадлежности из класса action_finished.]

⇒ задача*:

- ⟨ • [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

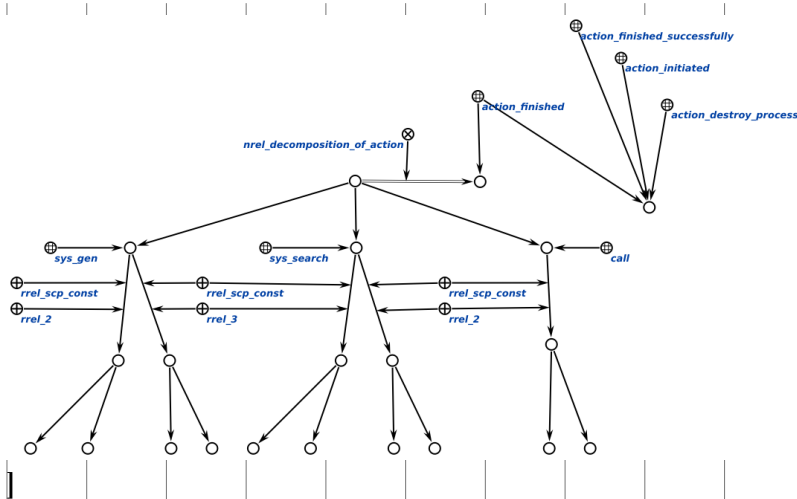
⇒ ответ агента*:

ответ агента уничтожения scp-процессов

= пустое множество

⇒ пример выходной конструкции*:

[



Агент обработки завершения интерпретации действия

:= [sc-агент обработки завершения интерпретации действия]

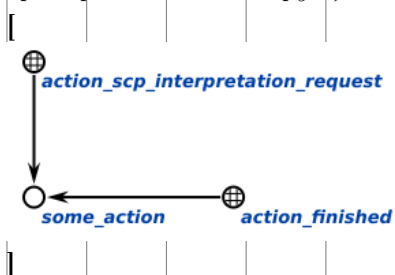
⇒ примечание*:

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

⇒ задача*:

⟨ • [Завершение с успехом 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*:

[

