7. Навигационно-описковая графодинамическая ассоциативная машина

В данном разделе описывается абстрактная навигационно-поисковая графодинамическая ассоциативная машина, обеспечивающая навигацию и поиск в рамках текущего состояния хранимой в памяти машины базы знаний. Реализация интерфейса этой машины описана в разделе 5 книги [411] (*ПрогрВАМ-2001кн*).

Навигационно-поисковая машина обычно интегрируется в другие графодинамические ассоциативные машины.

Данный раздел может быть использован в качестве учебного пособия по дисциплине «Модели представления знаний, базы данных и СУБД» специальности «Искусственный интеллект».

7.1. Операции навигационно-поисковой графодинамической ассоциативной машины

Ключевые понятия: навигационно-поисковая графодинамическая ассоциативная машина; цель; семантическая окрестность; изоморфный поиск; гипертекстовая семантическая сеть.

При реализации навигационно-поисковой графодинамической ассоциативной машины, так же как и при реализации графодинамической ассоциативной машины вывода (см. раздел 8) в качестве языка микропрограммирования выбирается язык SCP, который описан в разделе 4 [411] (*ПрогрВАМ-2001кн*). Набор операций навигационно-поисковой графодинамической ассоциативной машины строго не фиксирован. Ниже рассмотрим некоторые операции навигационно-поисковой графодинамической ассоциативной машины. Всё множество рассматриваемых операций разбивается на следующие семейства операций:

- семейство операций поиска теоретико-графовой окрестности указываемого sc-элемента;
- семейство операций поиска в рамках указываемой формальной теории всех истинных высказываний, релевантных указываемой высказывательной форме (заданному образцу);
- семейство операций поиска семантических окрестностей указываемого sc-элемента;
- семейство операций поиска семантической связи между (двумя или более) указываемыми sc-элементами;
- семейство навигационно-поисковых операций в гипертекстовой семантической сети.

7.1.1. Информационные конструкции, описывающие состояние навигационно-поисковой графодинамической ассоциативной машины

Ключевые понятия: цель; адресат; результат.

Для представления информационных конструкций, описывающих состояние абстрактной навигационно-поисковой графодинамической ассоциативной машины, вводятся следующие ключевые узлы:

SCs-текст 7.1.1.1 Ключевые узлы навигационно-поисковой машины

```
result ;                    /* Описание ключевого узла result */

пояснение !; □result ,текст пояснения_ :   /" Ключевой узел result является знаком
бинарного отношения, связывающего множество, обозначающие цель, и sc-узел, который
обозначает результат цели "/ □ ;

главный синоним !; result ;


sender ;          /* Описание ключевого узла sender */

пояснение !; □sender ,текст пояснения_ :   /" Ключевой узел sender является знаком
бинарного отношения, связывающего множество, обозначающие цель, и sc-узел, который
```

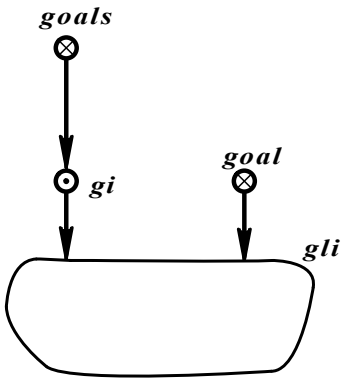
обозначает того, кто инициировал эту цель **"/ □ ;**

главный синоним !; ***sender*** **;**

Опишем типичные конструкции, которые будут необходимы для функционирования операций навигационно-поисковой графодинамической ассоциативной машины.

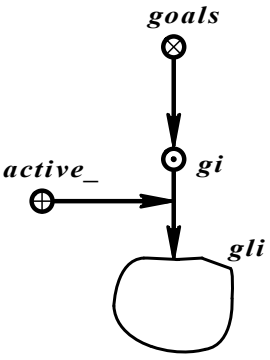
Рассмотрим описание **целей** навигационно-поисковой графодинамической ассоциативной машины.

SCg-текст 7.1.1.1. Общий вид описания цели навигационно-поисковой графодинамической ассоциативной машины



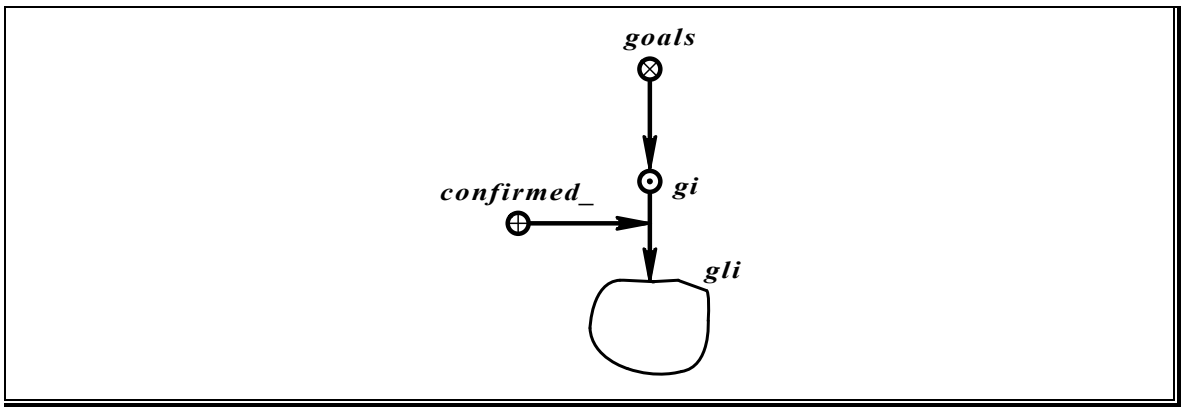
Здесь sc-узел ***gi*** обозначает множество целей, sc-узел ***gli*** обозначает конкретную цель навигационно-поисковой графодинамической ассоциативной машины.

SCg-текст 7.1.1.2. Общий вид описания активной цели навигационно-поисковой графодинамической ассоциативной машины



Здесь sc-узел ***gli*** обозначает конкретную активную цель навигационно-поисковой графодинамической ассоциативной машины.

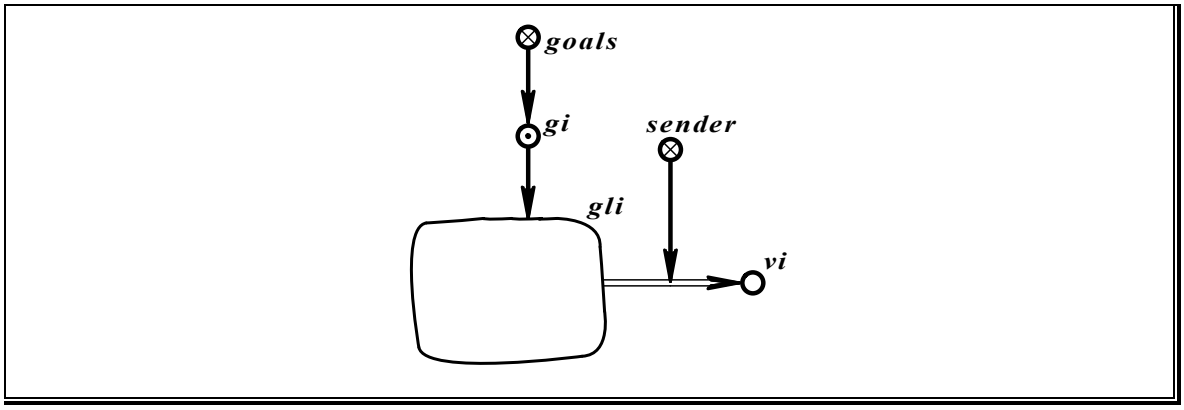
SCg-текст 7.1.1.3. Общий вид описания завершенной цели навигационно-поисковой графодинамической ассоциативной машины



Здесь sc-узел ***gli*** обозначает конкретную завершенную цель навигационно-поисковой графодинамической ассоциативной машины.

Рассмотрим описание **адресата цели**, т.е. кто прислал запрос и кому соответственно необходимо передать результат работы конкретной цели.

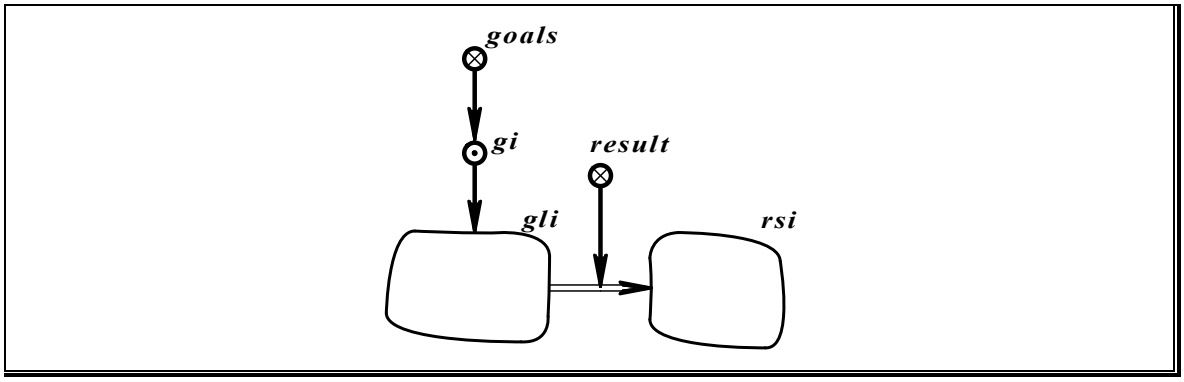
SCg-текст 7.1.1.4. Общий вид описания адресата цели навигационно-поисковой графодинамической ассоциативной машины



Здесь sc-узел ***gi*** обозначает множество целей навигационно-поисковой графодинамической ассоциативной машины, sc-узел ***gli*** обозначает конкретную цель навигационно-поисковой графодинамической ассоциативной машины, sc-узел ***vi*** обозначает адресат цели и связан с sc-узлом ***gli*** бинарным ориентированным отношением ***sender***, которое связывает цель с адресатом.

Рассмотрим описание **результата выполнения цели**.

SCg-текст 7.1.1.5. Общий вид описания результата выполнения цели навигационно-поисковой графодинамической ассоциативной машины



Здесь sc-узел ***gi*** обозначает множество целей навигационно-поисковой графодинамической ассоциативной машины, sc-узел ***gli*** обозначает конкретную цель навигационно-поисковой графодинамической ассоциативной машины, sc-узел ***rsi*** обозначает конкретный результат цели ***gli***. SC-узел цели и sc-узел результата связаны бинарным ориентированным отношением ***result***, которое связывает цель с результатом.

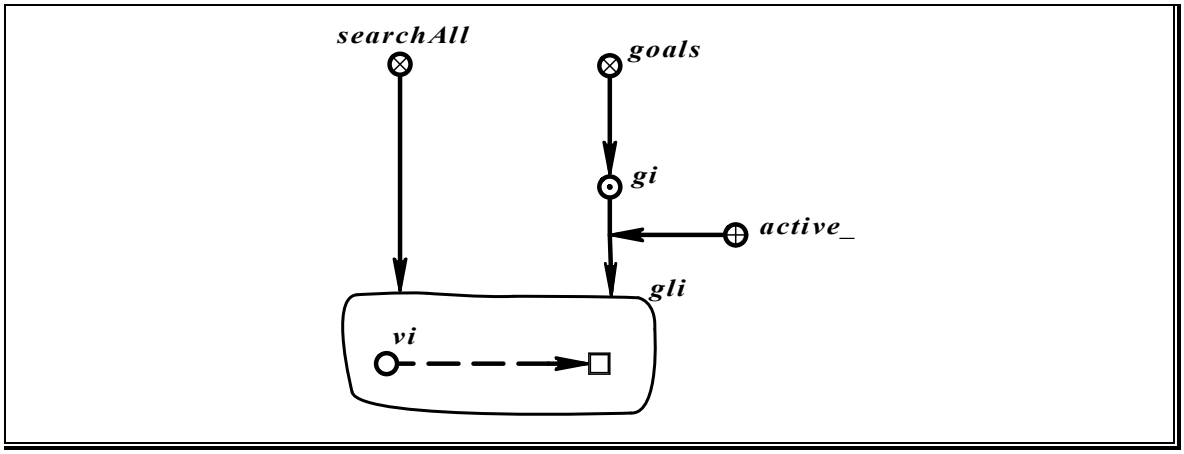
7.1.2. Семейство операций поиска теоретико-графовой окрестности указываемого sc-элемента

В рассматриваемое семейство операций навигационно-поисковой графодинамической ассоциативной машины входят следующие операции:

- операция поиска теоретико-графовой окрестности указываемого sc-элемента по выходящим или входящим парам принадлежности указываемого типа. Указываемый тип может быть составлен из комбинации “константный – переменный – метапеременный” и “позитивная – негативная – нечеткая”;
- операция поиска теоретико-графовой окрестности указываемого sc-элемента по связкам указываемого отношения с указанием атрибута, который должен быть помечен в искомых связках;
- операция поиска теоретико-графовой окрестности указываемого sc-элемента в рамках всей памяти или в рамках указываемой формальной теории и семейства указываемых формальных теорий.

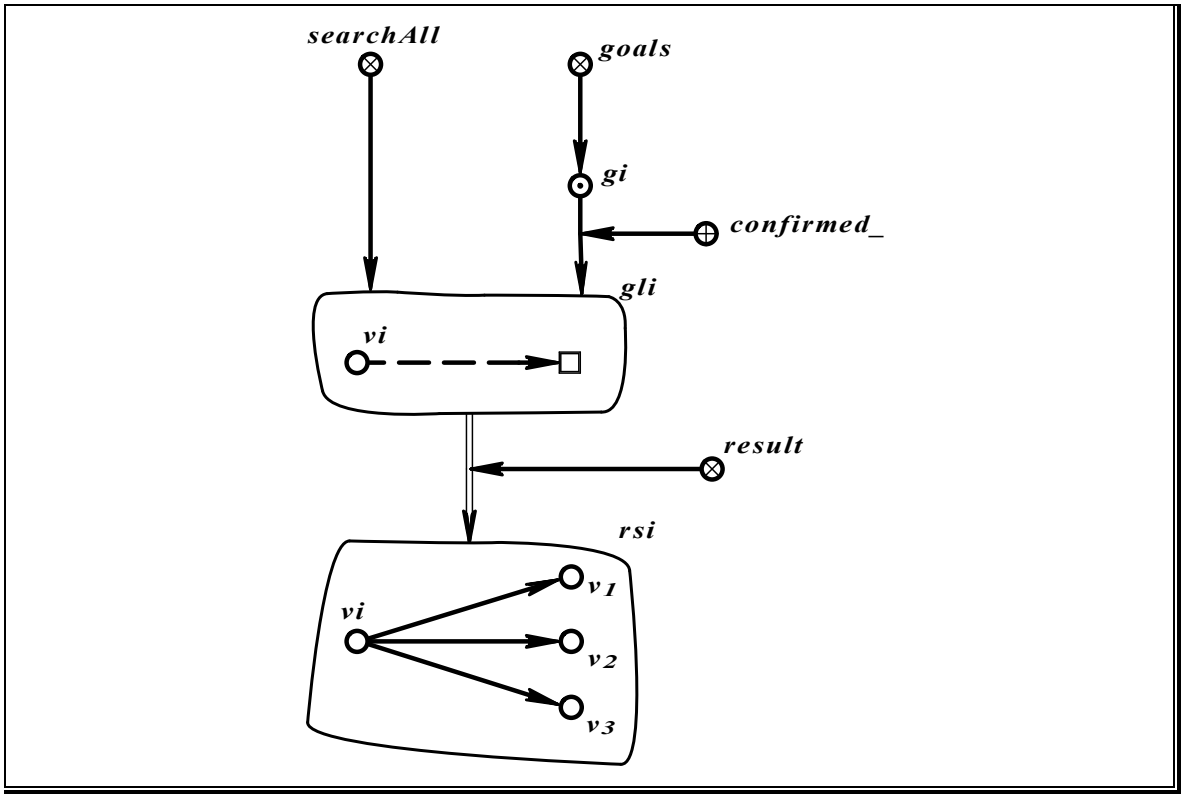
Рассмотрим в качестве примера **операцию поиска теоретико-графовой окрестности указываемого sc-элемента по выходящим парам принадлежности константного позитивного типа**.

Условием выполнения операции поиска теоретико-графовой окрестности указываемого sc-элемента по выходящим парам принадлежности константного позитивного типа является наличие конструкции следующего вида:



Здесь sc-узел ***vi*** является sc-узлом, относительно которого осуществляется поиск теоретико-графовой окрестности указываемого sc-элемента по выходящим парам принадлежности константного позитивного типа. Ключевой sc-узел ***searchAll*** указывает, что необходимо найти все элементы.

Результатом выполнения операции поиска теоретико-графовой окрестности указываемого sc-элемента по выходящим парам принадлежности константного позитивного типа является сформированное следующие множество:



Здесь результат ***rsi*** цели ***gli*** включает sc-узел ***vi***, который является аргументом поиска, и sc-узлы ***v1***, ***v2***, ***v3***, которые связаны с sc-узлом ***vi*** константными позитивными парами принадлежности.

7.1.3. Семейство операций поиска в рамках указываемой формальной теории всех истинных высказываний, релевантных указываемой высказывательной форме (заданному образцу)

К л ю ч е в ы е п о н я т и я : высказывание; изоморфный поиск.

Напомним, что в языке SCL высказывание, релевантное заданной высказывательной форме, и сама эта высказывательная форма являются изоморфными логическими формулами. При этом в рамках указанного изоморфизма:

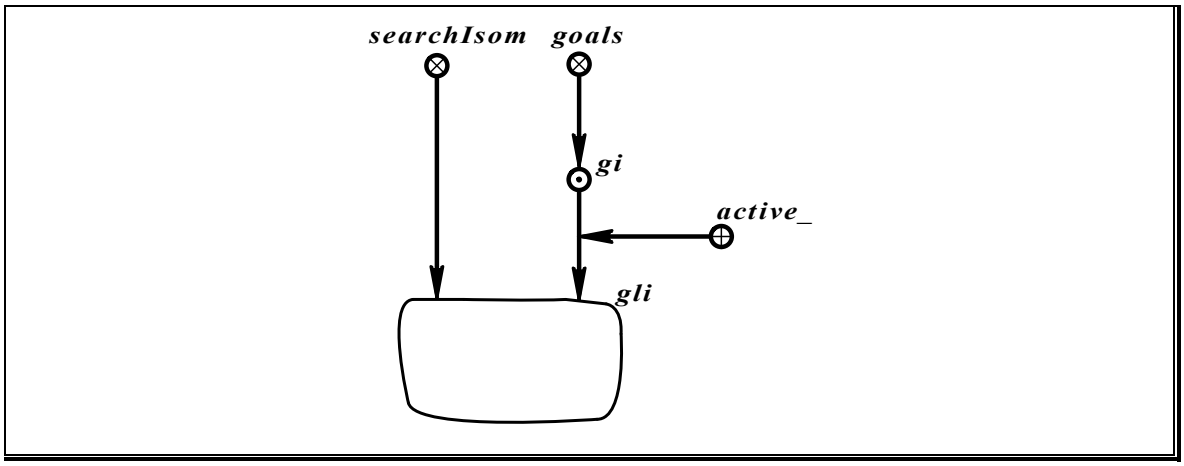
- свободным переменным высказывательной формы ставятся в соответствие константы релевантного высказывания;
- константы высказывательной формы ставятся в соответствие самим себе;
- знаки логических формул, входящих в состав высказывательной формы, ставятся в соответствие знакам логических формул, входящих в состав релевантного высказывания.

Очевидно, что если высказывательная форма является атомарной логической формулой, то релевантное высказывание представляет собой подграф (фрагмент структуры) sc-конструкции, являющейся представлением той предметной области, которая описывается указанной выше формальной теорией.

В рассматриваемое семейство операций входит операция изоморфного поиска по заданному образцу произвольного размера и произвольной конфигурации.

Рассмотрим **операцию изоморфного поиска по заданному образцу**.

Условием выполнения операции изоморфного поиска является наличие конструкции следующего вида:

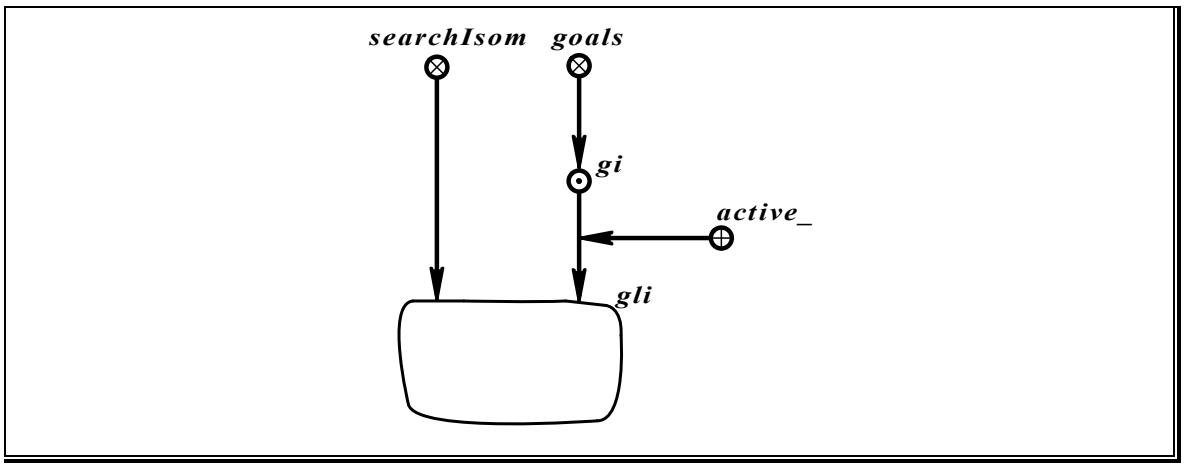


Здесь sc-узел ***gli*** является множеством, которое содержит образец для изоморфного поиска.

Результатом выполнения операции изоморфного поиска является сформированное множество результатов. Связь цели с результатом осуществляется с помощью отношения ***result*** (см scg-текст 7.1.1.5)

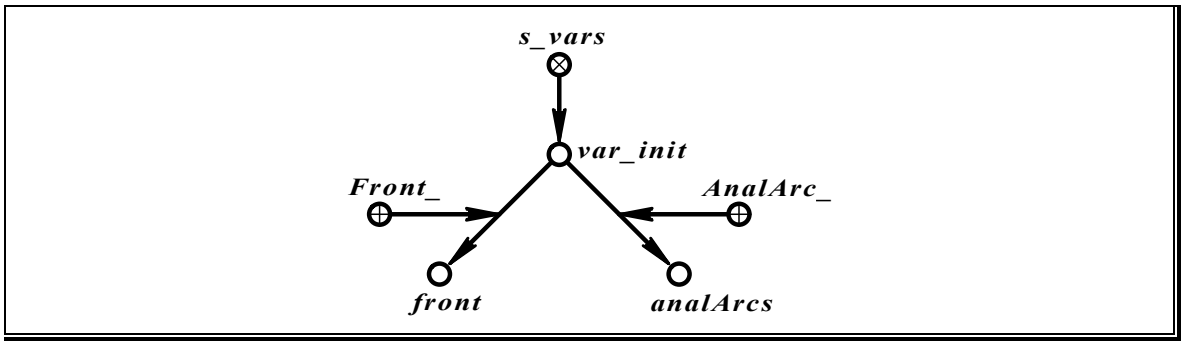
Микрограмма операции изоморфного поиска имеет следующий вид.

Шаг 1. Проверить условие выполнения операции, т.е. найти конструкцию вида:

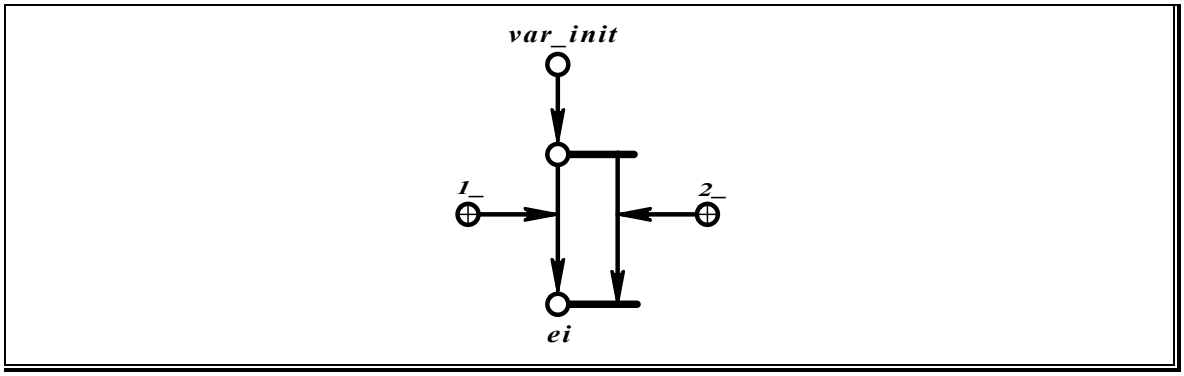


Шаг 2. Если такой конструкции не найдено, то перейти к шагу 1. Это означает, что в текущем состоянии навигационно-поисковой графодинамической ассоциативной машины нет активного запроса на поиск изоморфного подграфа.

Шаг 3. Создать множество выполняемых вариантов (обозначим его ***s_vars***). Под вариантом будем понимать кортеж, в состав которого входят множества: помеченное атрибутом ***Front_***, которое содержит такие элементы, от которых можно осуществлять поиск соседних элементов, помеченное атрибутом ***AnalArc_***, которое содержит лишь те дуги, которые еще не имеют соответствий в рамках текущего варианта, а все остальные, – знаки соответствий. Включить в него начальный выполняемый вариант ***var_init***:



Шаг 4. Просмотреть все элементы шаблона. Все константные элементы поместить в множество ***front***, установить соответствие константного элемента самому себе - сгенерировать конструкцию следующего вида:



Шаг 5. Все переменные дуги поместить в множество ***analArcs***.

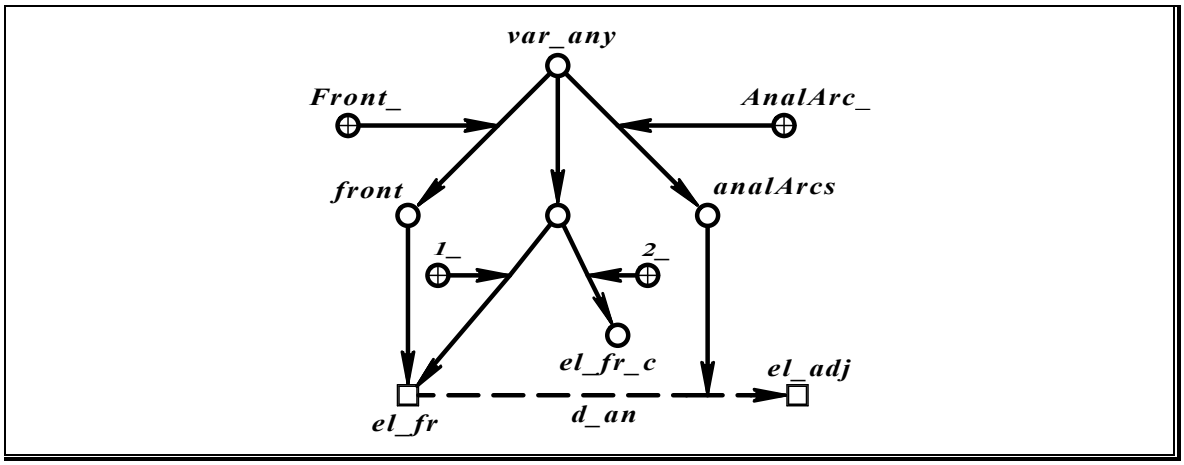
Шаг 6. Просмотреть все элементы множества ***front***. Исключить те элементы, в которые не входят дуги принадлежащие множеству ***analArcs***. Т.е. удалить те элементы, от которых невозможно вести поиск, т.к. их окрестность уже найдена.

Шаг 7. Если множество ***front*** пусто, то перейти к шагу 10.

Шаг 8. Начало цикла по элементам ***var_any*** множества ***s_vars***.

Шаг 9. Если множество ***analArcs*** варианта ***var_any*** пусто, то удалить множество ***front*** и ***analArcs***, исключить ***var_any*** из ***s_vars*** и занести его во множество результатов ***s_result***. Перейти к шагу 8.

Шаг 10. Иначе выбрать элемент множества ***front*** текущего варианта ***var_any*** - sc-элемент ***el_fr***. Среди sc-дуг, входящих в этот элемент или выходящих из него и принадлежащих множеству ***analArcs*** варианта ***var_any*** выбрать одну из них - sc-дугу ***d_an***.



Шаг 11. Если *el_adj* принадлежит множеству *front* варианта *var_any*, т.е. ему уже присвоено соответствие в составе варианта *var_any*, то

Шаг 12. Найти элемент *el_adj_c*, являющийся соответствием элемента *el_adj* в рамках варианта *var_any*.

Шаг 13. Если между элементами *el_adj_c* и *el_fr_c* существует sc-дуга *d_an_c*, "ориентированная" относительно *el_fr_c* точно таким же образом, как *d_an* - относительно *el_fr* (т.е. обе эти дуги - либо входящие, либо выходящие), и не проведенная в варианте *var_any* никакой sc-дуге, то

Шаг 14. Фиксируем, что дуге *d_an* в рамках варианта *var_any* соответствует дуга *d_an_c* и исключаем её из множества *analArcs*.

Шаг 15. Если множество *analArcs* варианта *var_any* становится пустым, то следует перейти к шагу 9 (вариант завершается успешно).

Шаг 16. Если элемент *el_fr* больше не инцидентен ни одной sc-дуге, принадлежащей множеству *analArcs* варианта *var_any*, то *el_fr* исключается из множества *front* варианта *var_any*, как элемент, от которого невозможно осуществить поиск соседей, т.к. всем инцидентным ему дугам уже найдены соответствия.

Шаг 17. Если элемент *el_adj* больше не инцидентен ни одной sc-дуге, принадлежащей множеству *analArcs* варианта *var_any*, то *el_fr* исключается из множества *front* варианта *var_any*.

Шаг 18. Если в sc-дугу *d_an* входят sc-дуги, являющиеся элементами множества *analArcs* варианта *var_any*, то занести *d_an* в множество *front* варианта *var_any*.

Шаг 19. Перейти к шагу 10.

Шаг 20. Иначе sc-дуге *d_an* невозможно присвоить соответствие.

Шаг 21. Удалить множества *front* и *analArcs*, удалить все найденные соответствия, вариант *var_any* исключить из множества *s_vars*, (фиксируется его безуспешное завершение). Перейти к шагу 8.

Шаг 22. Сформировать множество *m_el_adj_c* элементов, которые могут быть поставлены в соответствие элементу *el_adj* и которые не были поставлены другим элементам шаблона в рамках варианта *var_any*. В это множество не могут быть включены элементы, которые уже были поставлены в соответствие некоторому элементу шаблона в рамках текущего варианта.

Шаг 23. Если множество *m_el_adj_c* пусто, то перейти к шагу 22 (вариант *var_any* является неудачным и его следует исключить из множества вариантов).

Шаг 24. Иначе sc-дуга *d_an* исключается из множества *front* варианта *var_any*.

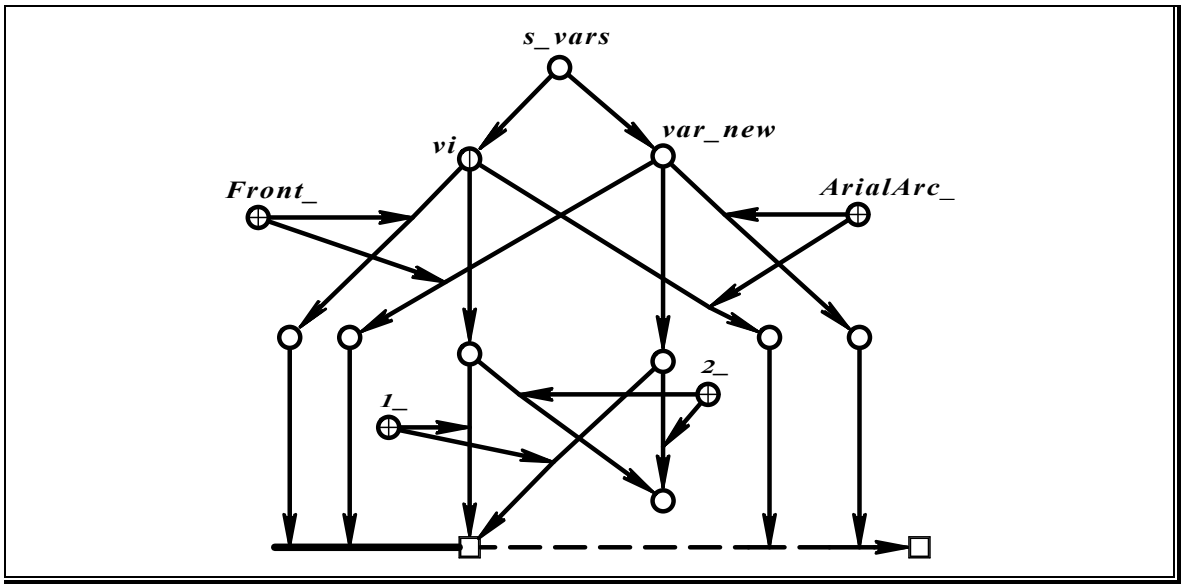
Ш а г 2 5 . Если в sc-дугу ***d_an*** входят sc-дуги, являющиеся элементами множества ***analArcs*** варианта ***var_any***, то занести ***d_an*** во множество ***front*** варианта ***var_any***.

Ш а г 2 6 . Если в sc-элемент ***el_adj*** входят sc-дуги, являющиеся элементами множества ***analArcs*** варианта ***var_any***, то занести ***el_adj*** во множество ***front*** варианта ***var_any***.

Ш а г 2 7 . Если элемент ***el_fr*** больше не инцидентен ни одной sc-дуге, принадлежащей множеству ***analArcs*** варианта ***var_any***, то ***el_fr*** исключить из ***front*** варианта ***var_any***.

Ш а г 2 8 . Начало цикла по элементам ***el_adj_c*** множества ***m_el_adj_c***.

Ш а г 2 9 . Завести новый вариант ***var_new***, множества ***front*** и ***analArcs*** варианта ***var_any*** копировать в соответствующие им множества в варианте ***var_new***, также в новый вариант копировать все найденные соответствия.



Ш а г 3 0 . Еще раньше, при занесении элемента ***el_adj_c*** во множество ***m_el_adj_c*** должна была быть зафиксирована sc-дуга ***d_an_a***, связывающая sc-элементы ***el_fr_c*** и ***el_adj_c*** связью соответствующего направления. Теперь устанавливаем соответствие между элементами ***d_an*** и ***d_an_a***, а также между ***el_adj*** и ***el_adj_c*** в рамках варианта ***var_new***, т.е. присваиваем соответствия элементам ***d_an*** и ***el_adj*** в рамках варианта ***var_new***.

Ш а г 3 1 . Если множество ***analArcs*** варианта ***var_new*** пусто, то удаляется множества ***front*** и ***analArcs*** в рамках варианта ***var_new***, вариант ***var_new*** исключается из множества ***s_vars*** и заносится в результирующее множество ***s_result***. Перейти к шагу 39. (Зафиксировать успешное завершение варианта ***var_new***).

Ш а г 3 2 . Если ***el_adj*** - sc-узел, то перейти к шагу 43.

Ш а г 3 3 . Считаем, что значением ***da*** является ***el_adj***.

Ш а г 3 4 . Зафиксировать ***da_beg*** - начало sc-дуги ***da***; ***da_end*** - конец sc-дуги ***da***.

Ш а г 3 5 . Находим sc-дугу ***da_c*** , соответствующую sc-дуге ***da***. Зафиксировать: ***da_c_beg*** - начало sc-дуги ***da_c***; ***da_c_end*** - конец sc-дуги ***da_c***.

Ш а г 3 6 . Если элемент, являющийся соответствием элемента ***da_beg*** в рамках варианта ***var_new*** определен и не равен ***da_c_beg*** или если элемент, являющийся соответствием элемента ***da_end*** в рамках варианта ***var_new*** определен и не равен ***da_c_end***, то зафиксировать безуспешное завершение

варианта ***var_new*** (уничтожаются множества ***front*** и ***analArcs*** варианта ***var_new***, уничтожаются все найденные соответствия, вариант ***var_new*** исключается из множества ***s_vars***). Перейти к шагу 43.

Шаг 37. Если элемент, являющийся соответствием элемента ***da_beg*** в рамках варианта ***var_new*** не определен, то назначаем соответствующим ему в рамках варианта ***var_new*** элементом ***da_c_beg***.

Шаг 38. Если элемент, являющийся соответствием элемента ***da_end*** в рамках варианта ***var_new*** не определен, то назначаем соответствующим ему в рамках варианта ***var_new*** элементом ***da_c_end.***

Шаг 39. Исключить sc-дугу ***da*** из множества ***analArcs*** варианта ***var_new***.

Шаг 40. Если множество ***analArcs*** варианта ***var_new*** пусто, то перейти к шагу 32. (зафиксировать успешное завершение варианта ***var_new***).

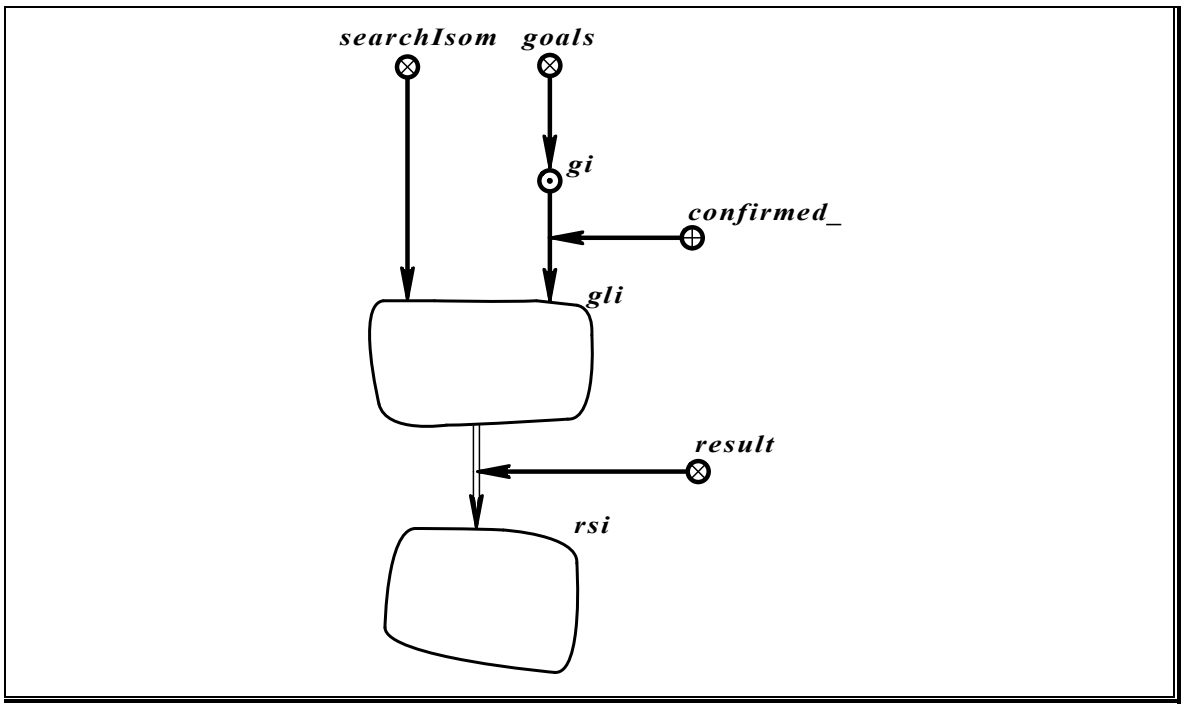
Шаг 41. Если ***da_end*** - sc-дуга, принадлежащая множеству ***analArcs*** варианта ***var_new***, то Считаем, что значением ***da*** является ***da_end***; Перейти к шагу 37.

Шаг 42. Конец цикла по элементам ***el_adj_c*** множества ***m_el_adj_c***.

Шаг 43. Исключить вариант ***var_any*** из множества выполняемых вариантов ***s_vars***.

Шаг 44. Конец цикла по элементам ***var_any*** множества ***s_vars***.

Шаг 45. Пометить исходную цель как достигнутую, что сводится к формированию следующей sc-конструкции:



Конец микропрограммы.

Пример выполнения операции изоморфного поиска приведен в подразделе 7.2.

Реализация операции изоморфного поиска на языке SCP приведена в [411] (***ПрогрВАМ-2001кн***)

7.1.4. Семейство операций поиска семантических окрестностей указываемого sc-элемента

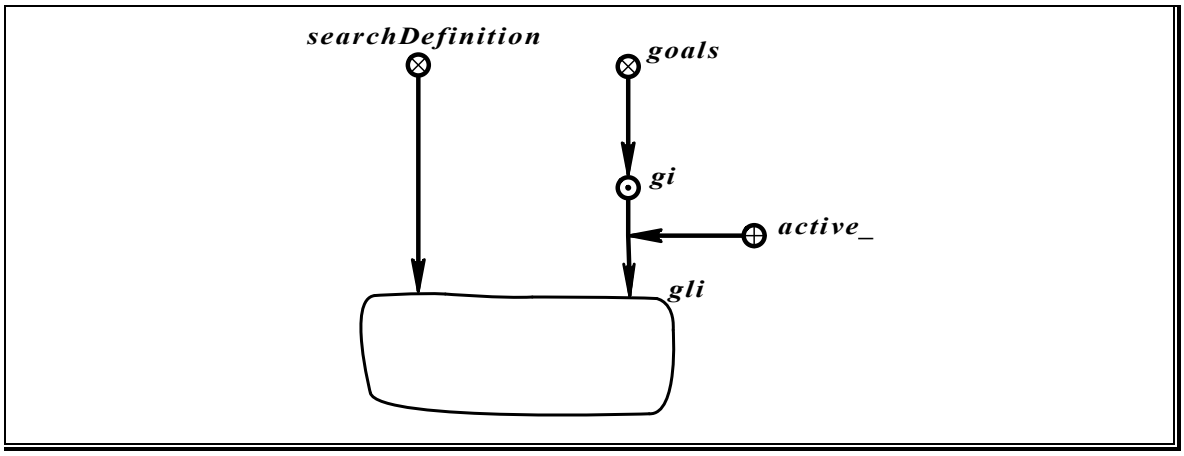
Ключевые понятия: формальная теория; высказывание; определение; семантическая окрестность.

В рассматриваемое семейство операций навигационно-поисковой графодинамической ассоциативной машины входят следующие операции:

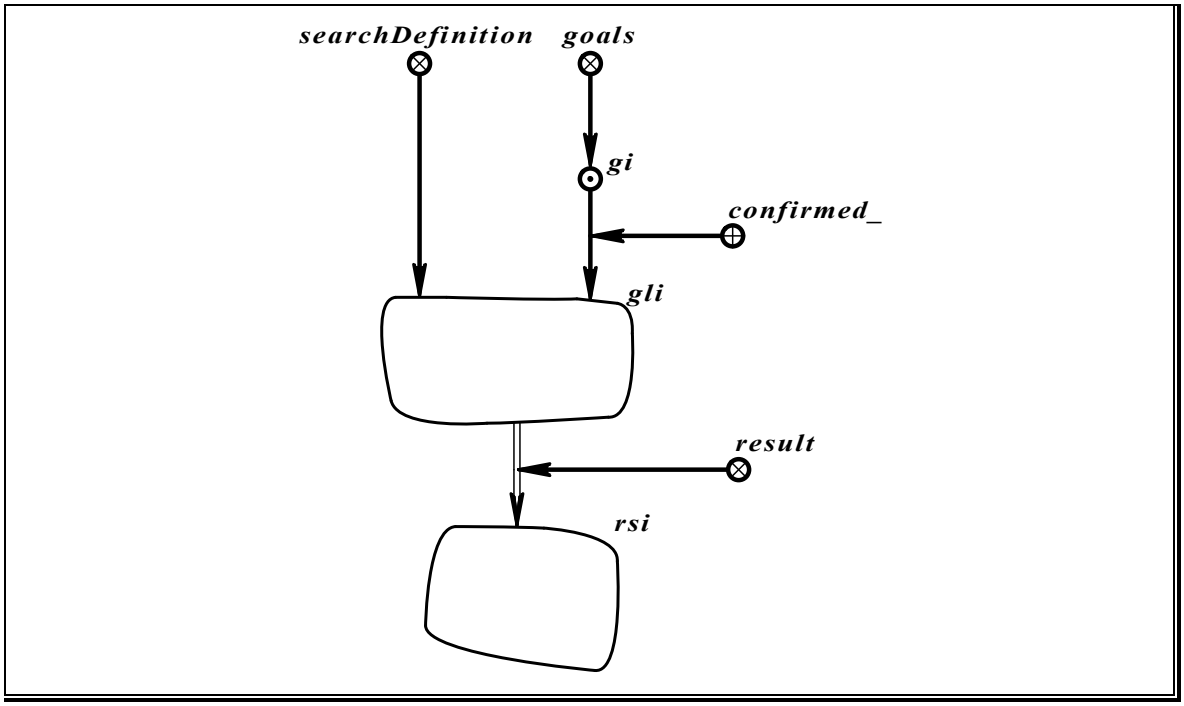
- операция поиска всех высказываний всевозможных формальных теорий, в состав которых указываемый sc-элемент входит в качестве элемента какой-либо атомарной логической формулы;
- операция поиска в рамках указываемой формальной теории всех высказываний, в состав которых указываемый sc-элемент входит в качестве элемента какой-либо атомарной логической формулы;
- операция поиска всех определений указываемого понятия в рамках всевозможных формальных теорий или в рамках указываемой формальной теории;
- операция поиска в рамках указываемой формальной теории всех истинных высказываний, которые являются утверждениями, описывающими свойства (закономерности) понятий, обозначаемого указываемым sc-элементом;
- операция вывода классификационной схемы указываемого понятия.

Рассмотрим **операцию поиска определения понятия** указываемого sc-элемента

Условием выполнения операции поиска определения понятия указываемого sc-элемента является наличие в памяти конструкции вида:



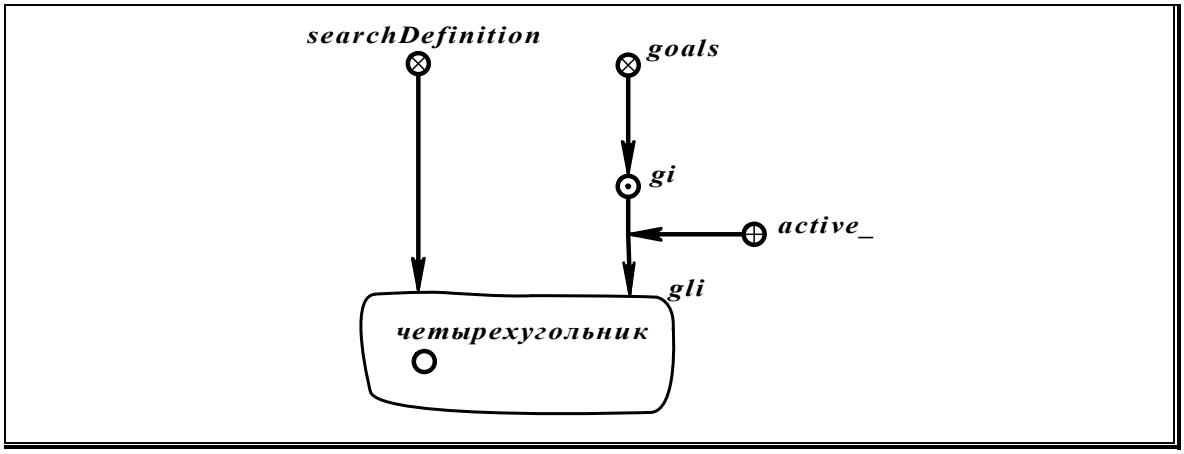
Результатом выполнения операции поиска определения понятия указываемого sc-элемента является генерация конструкции свидетельствующее о том, что запрос успешно обработан. Результаты операции находятся в сформированном множестве ***rsi***.



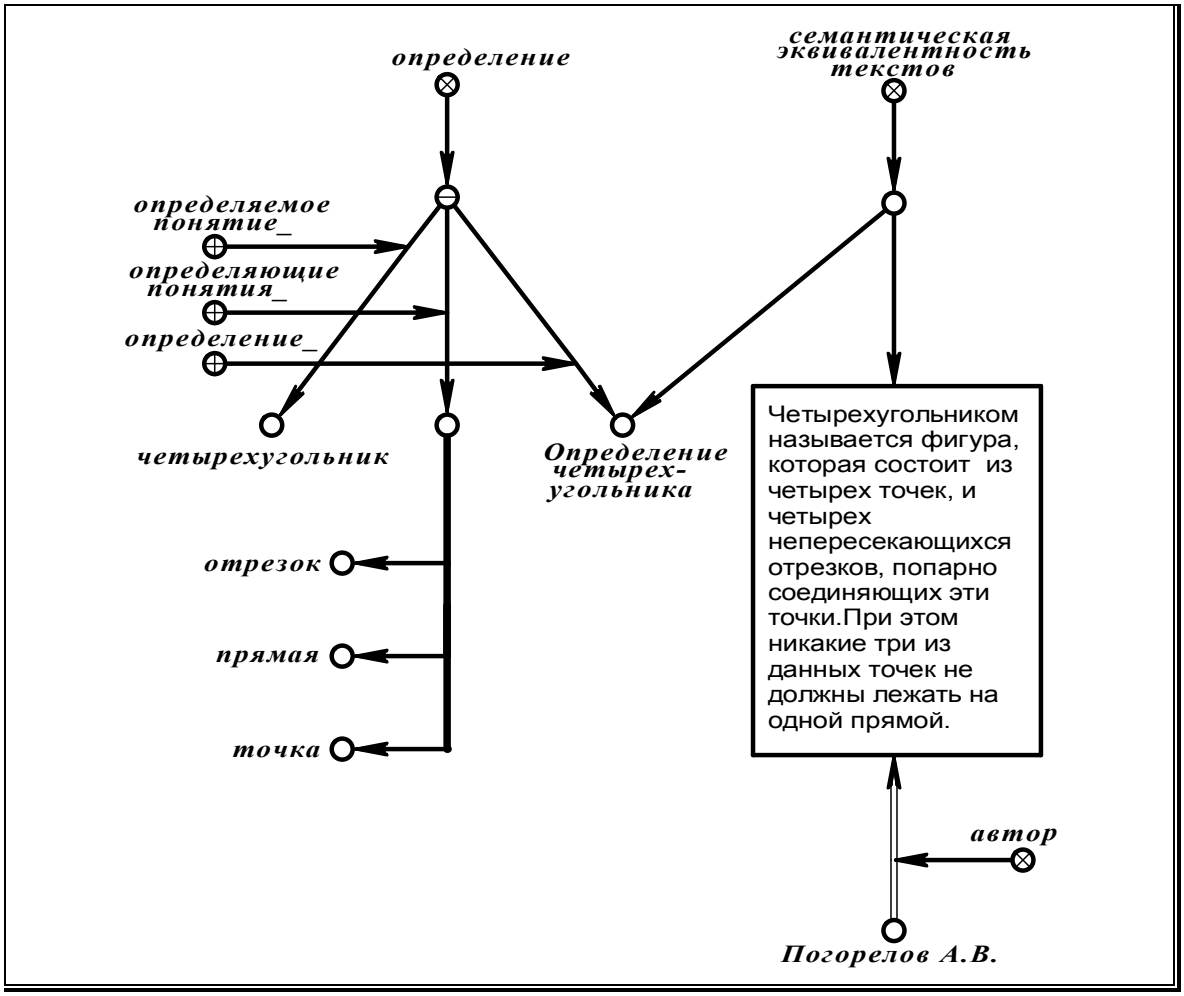
При выполнении операции осуществляется поиск отношения ***определение,*** в которое входит sc-элемент, определение которого мы ищем, с атрибутом ***определяемое понятие.*** Далее ищется sc-конструкция, содержащая запись определения на естественном языке, а также конструкция, описывающая библиографическую ссылку. Все эти найденные конструкции включаются в результирующее множество.

Приведем пример работы операции поиска определения понятия.

Пусть требуется найти определение для понятия ***"четырехугольник"***. Целевое множество выглядит следующим образом:



В результате выполнения операции в результирующем множестве будет следующая конструкция:



Здесь sc-узел с идентификатором ***"Определение четырехугольника"*** обозначает формальное определение понятия ***четырехугольник***. Так же этот sc-узел связан с определением понятия ***четырехугольник***, которое записано на русском языке.

7.1.5. Семейство операций поиска семантической связи между (двумя или более) указываемыми sc-элементами

В рассматриваемое семейство операций навигационно-поисковой графодинамической ассоциативной машины входят следующие операции:

- операция поиска теоретико-множественных связей между указываемыми sc-элементами;
- операция поиска семантической связи "определяемое-определяющее понятие" между двумя указываемыми sc-элементами;
- операция поиска вхождения указываемых sc-элементов в одни и те же утверждения;
- операция поиска минимальных теоретико-графовых маршрутов указываемых sc-элементов по связкам различных отношений.

7.1.6. Семейство навигационно-поисковых операций в гипертекстовой семантической сети

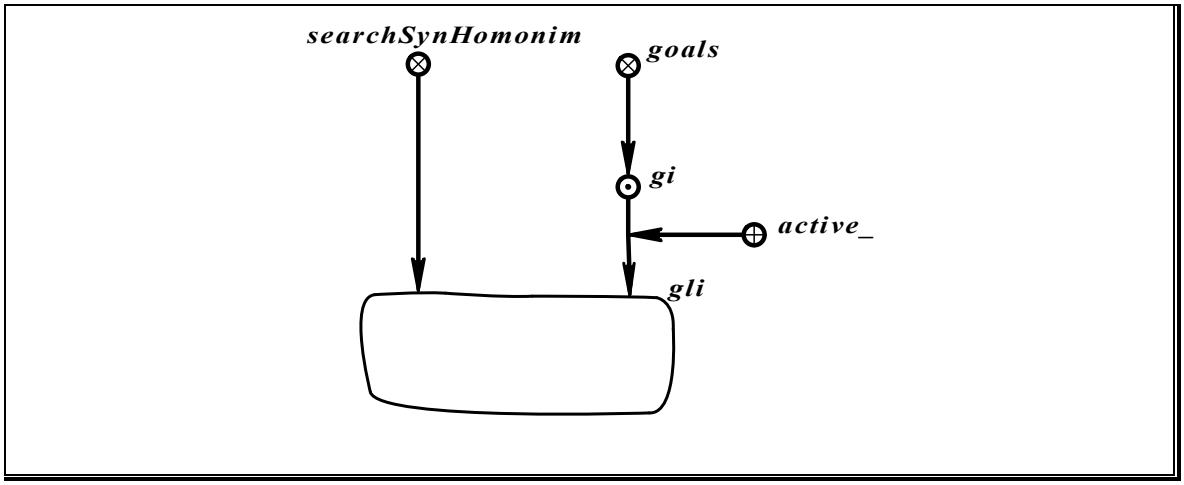
К л ю ч е в ы е п о н я т и я : гипертекстовая семантическая сеть, синонимы, омонимы.

В рассматриваемое семейство операций навигационно-поисковой графодинамической ассоциативной машины входят следующие операции:

- операция поиска всех синонимов и омонимов указываемого sc-элемента (синонимичные sc-элементы – это семантические эквивалентные sc-элементы, имеющие разные идентификаторы; омонимичные sc-элементы – это семантически неэквивалентные sc-элементы, имеющие одинаковые идентификаторы);
- операция поиска всех константных sc-узлов, содержимое каждого из которых представляет собой информационную конструкцию, являющуюся комментарием для указываемого sc-элемента.

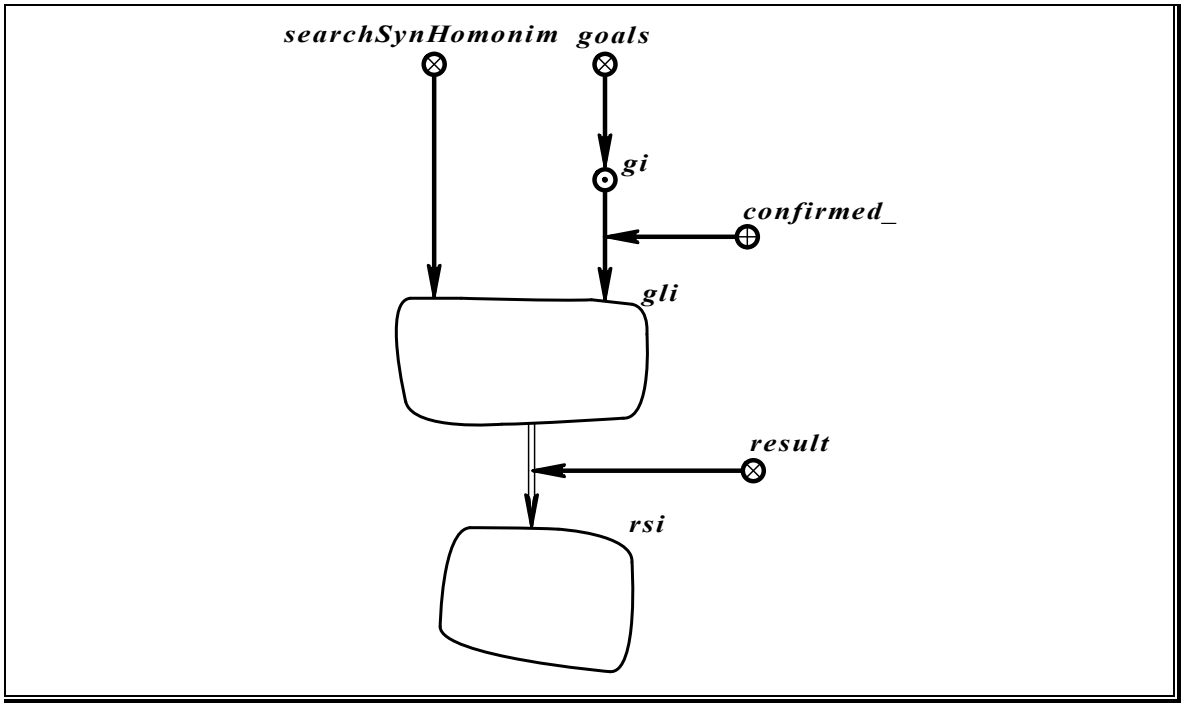
Рассмотрим **операцию поиска синонимов и омонимов указываемого sc-элемента**.

Условием выполнения операции поиска всех синонимов и омонимов указываемого sc-элемента является наличие в памяти конструкции вида:



Здесь в множество gli включаются те sc-элементы, синонимы и омонимы которых необходимо найти.

Результатом выполнения операции поиска всех синонимов и омонимов указываемого sc-элемента является формирование результирующего множества ***rsi***, которое содержит синонимы и омонимы указываемого sc-элемента, а также с указанием отношений ***"синонимичные sc-элементы"***, ***"главный синоним"***, ***"омонимичные sc-элементы"***

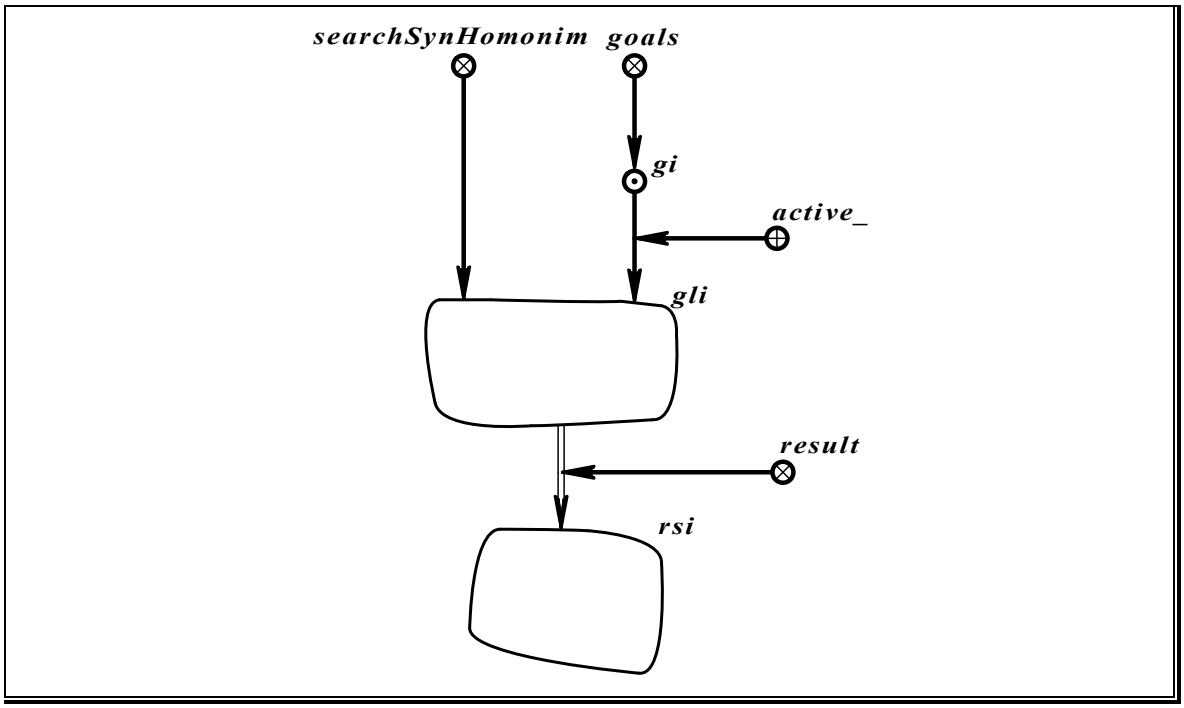


Микропрограмма операции поиска всех синонимов и омонимов указываемого sc-элемента:

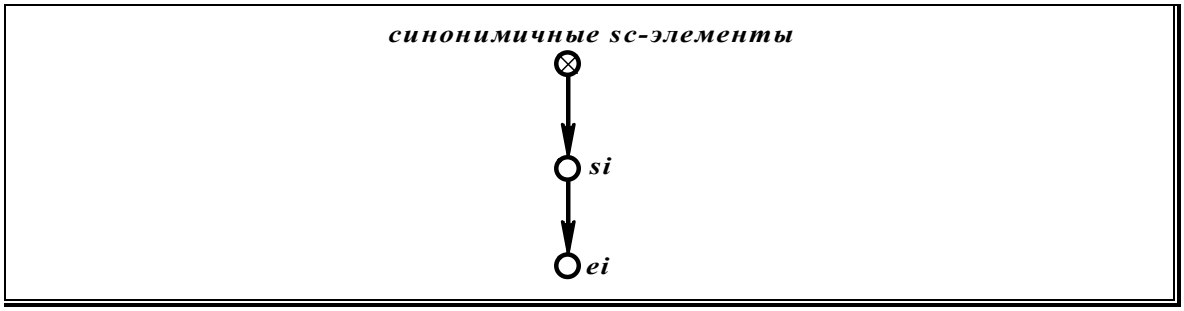
Шаг 1. Проверить условие выполнения операции, если конструкция найдена, то перейти к шагу 2, иначе шаг 1.

Шаг 2. Найти множество sc-элементов (обозначим его ***gli***), которое является описанием задачи. Элементами этого множества являются sc-элементы, для которых надо найти синонимичные sc-элементы и омонимичные sc-элементы

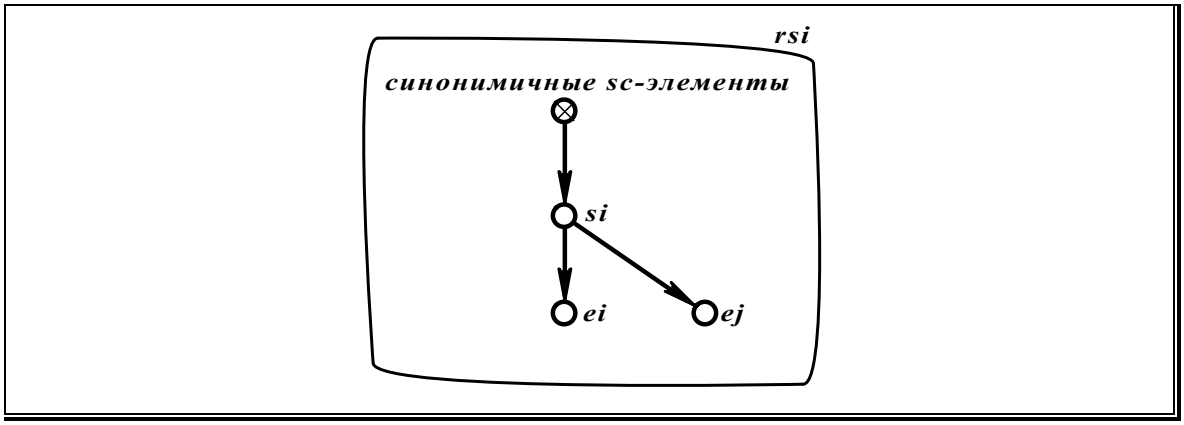
Шаг 3. Сформировать множество (обозначим его ***rsi***), описывающее результат обработки запроса, т.е. надо сформировать следующую sc-конструкцию:



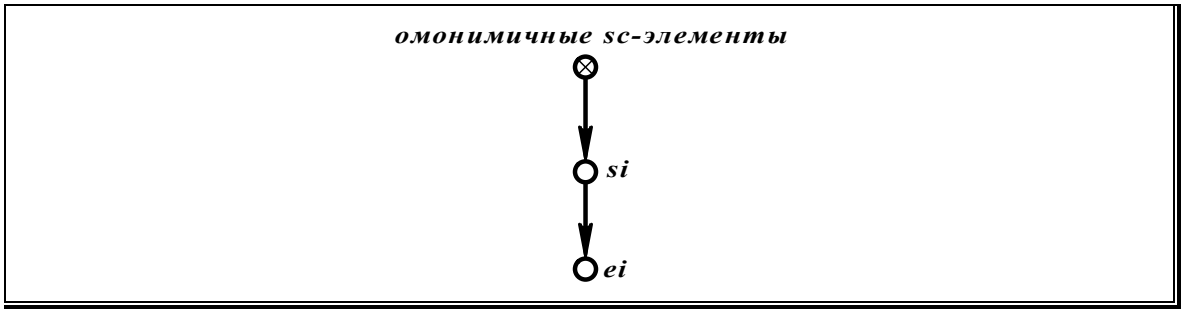
Шаг 4. Для каждого sc-элемента (обозначим его ***ei***) из множества ***gli*** найти знак множества синонимичных sc-элементов (обозначим его ***si***), т.е. надо найти sc-конструкцию вида:



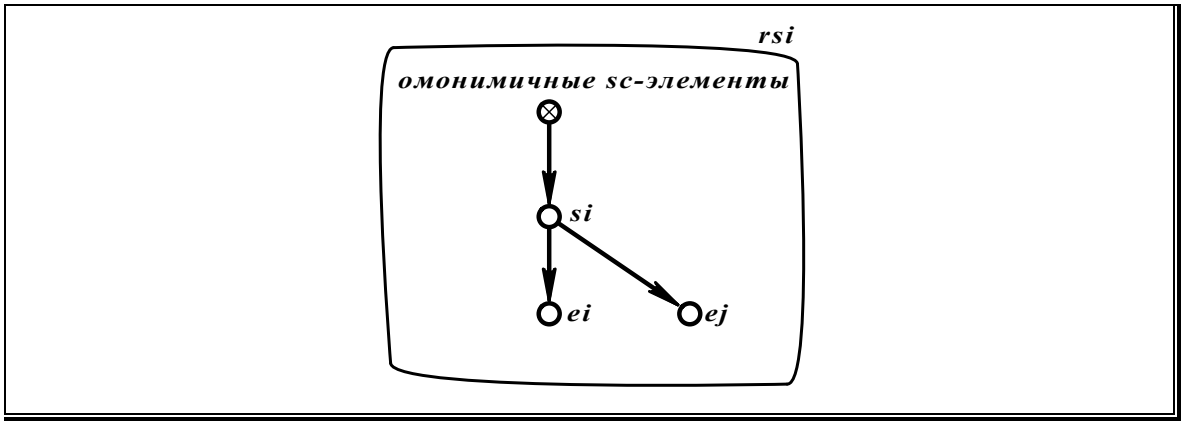
Шаг 5. Включить все элементы множества ***si*** в результирующее множество (***rsi***), т.е. надо сформировать sc-конструкцию вида:



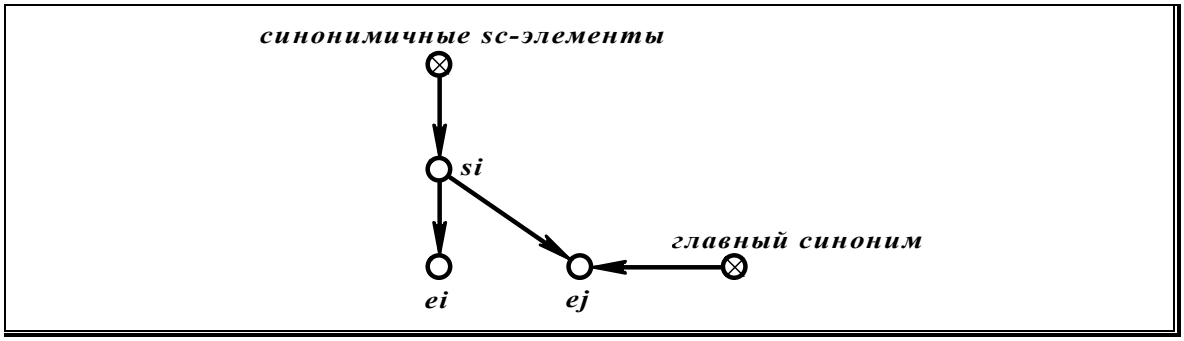
Шаг 6. Для каждого sc-элемента (обозначим его ei) из множества gli найти знак множества омонимичных sc-элементов (обозначим его si), т.е. надо найти sc-конструкцию вида:



Шаг 7. Включить все элементы множества si в результирующее множество (rsi), т.е. надо сформировать sc-конструкцию вида:

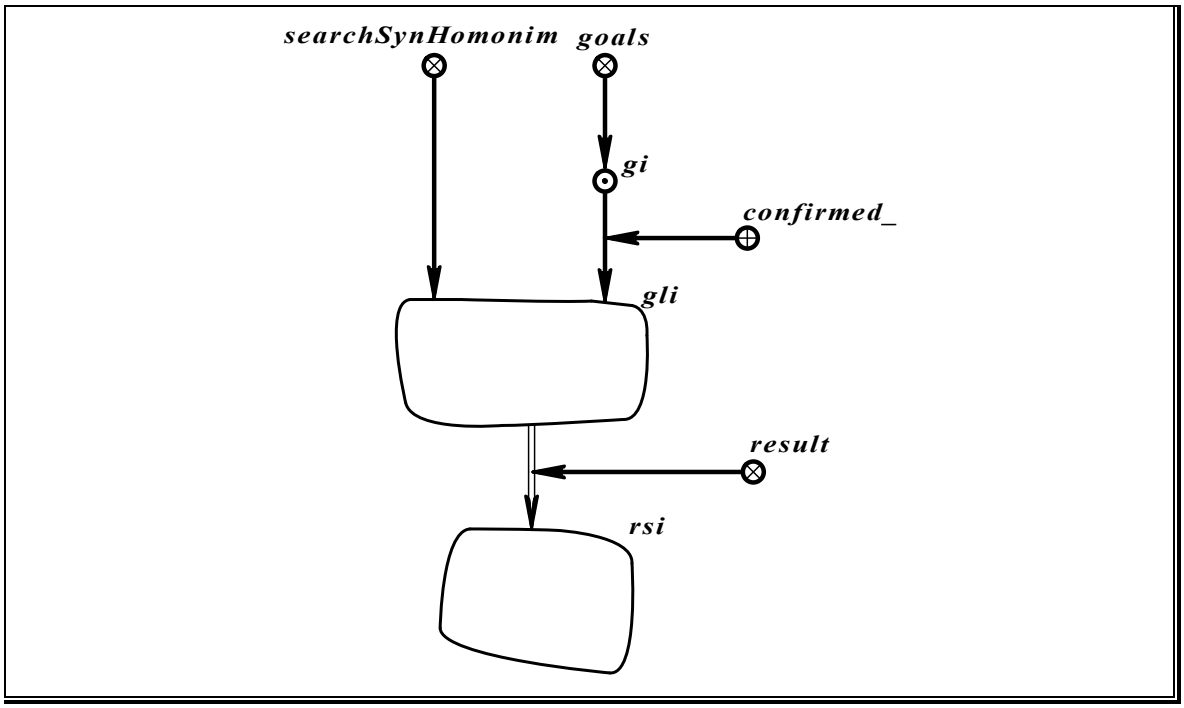


Шаг 8. Для каждого sc-элемента, омонима, (обозначим его vi) из множества si найти главный синоним, т.е. надо найти sc-конструкцию вида:



Шаг 9. Включить найденную sc-конструкцию в результирующее множество (rsi).

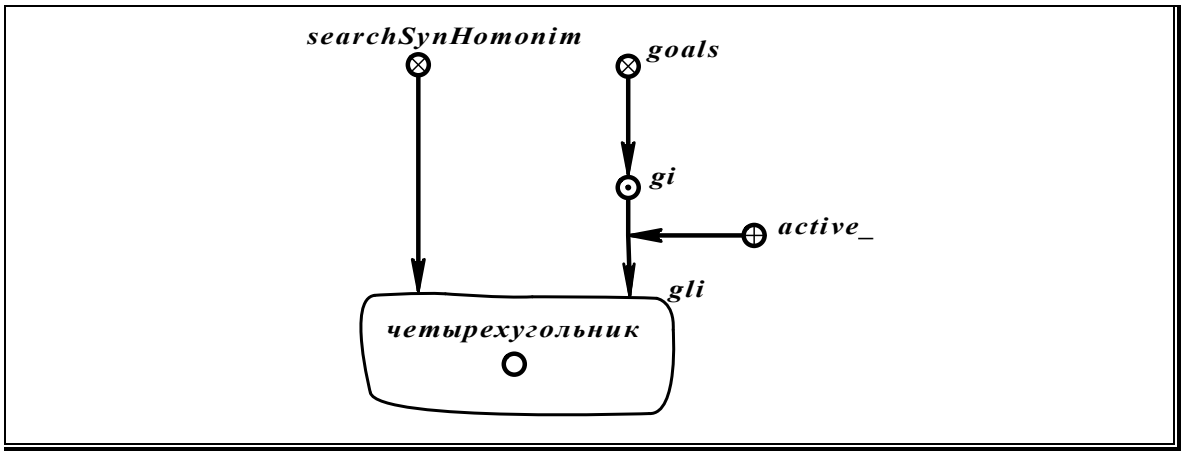
Шаг 10. Сгенерировать sc-конструкцию, свидетельствующую о том, что цель обработана, т.е удалить sc-дугу между ключевым узлом ***active*** и sc-узлом цели (***gli***) и сгенерировать sc-дугу между ключевым узлом ***confirmed*** и sc-узлом цели (***gli***):



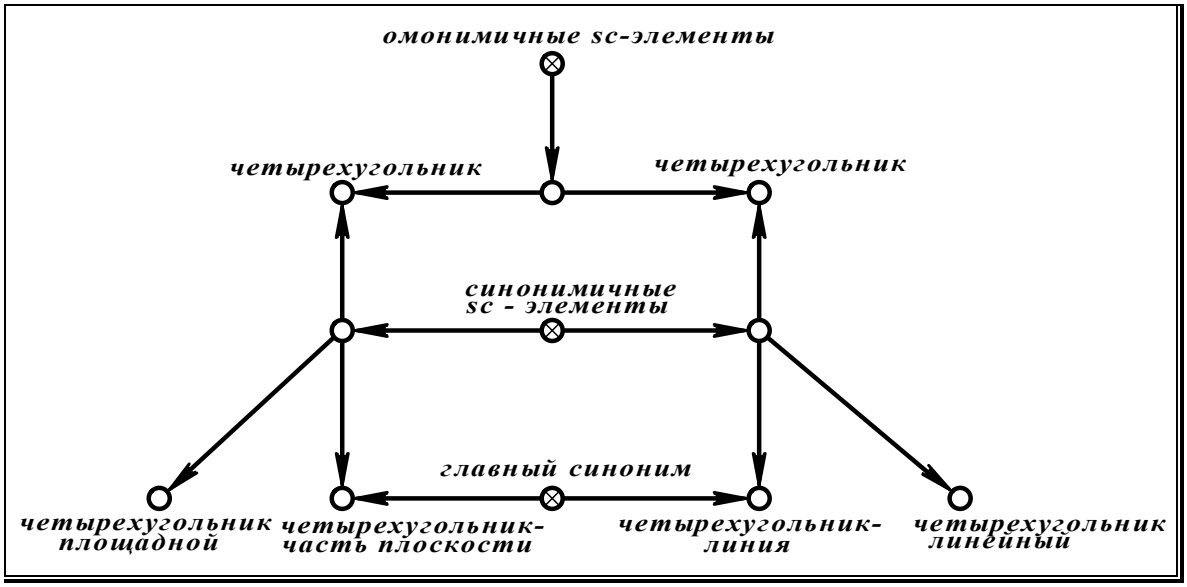
Конец микропрограммы.

Приведем пример работы операции поиска всех синонимов и омонимов указываемого sc-элемента.

Пусть требуется найти все синонимы и омонимы для понятия ***"четырехугольник"***. Целевое множество выглядит следующим образом:



В результате выполнения операции в результирующем множестве будет следующая конструкция:



Здесь sc-узлы с идентификаторами ***"четырехугольник площадной"*** ***"четырехугольник – часть плоскости"*** являются синонимами понятия ***четырехугольник***. А sc-узел с идентификатором ***"четырехугольник"*** является нетривиальным омонимом (см. подраздел 6.4) понятию ***четырехугольник***. Эти два понятия различаются ***главными синонимами***.

Резюме к подразделу 7.1

Приведенная классификация операций навигационно-поисковой графодинамической ассоциативной машины является открытой, т.е. набор операций может быть легко расширен.

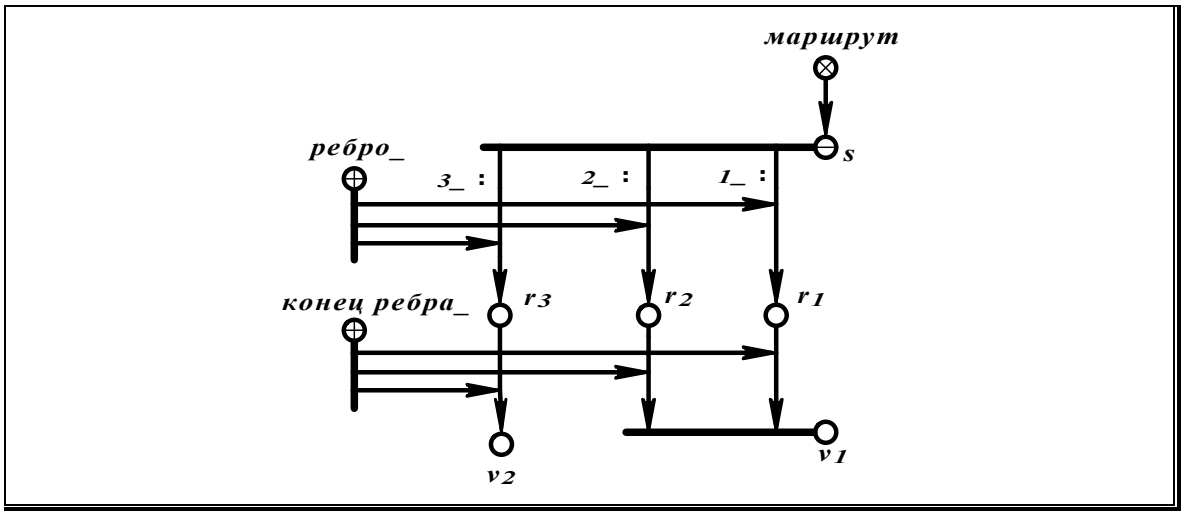
Если в структуре цели указан отправитель, т.е. цель инициирована пользователем, то результат выполнения цели выводится пользователю на виртуальный экран графодинамической ассоциативной машины (см. раздел 5 в [411] (***ПрогрВАМ-2001кн***))

7.2. Пример работы навигационно-поисковой графодинамической ассоциативной машины

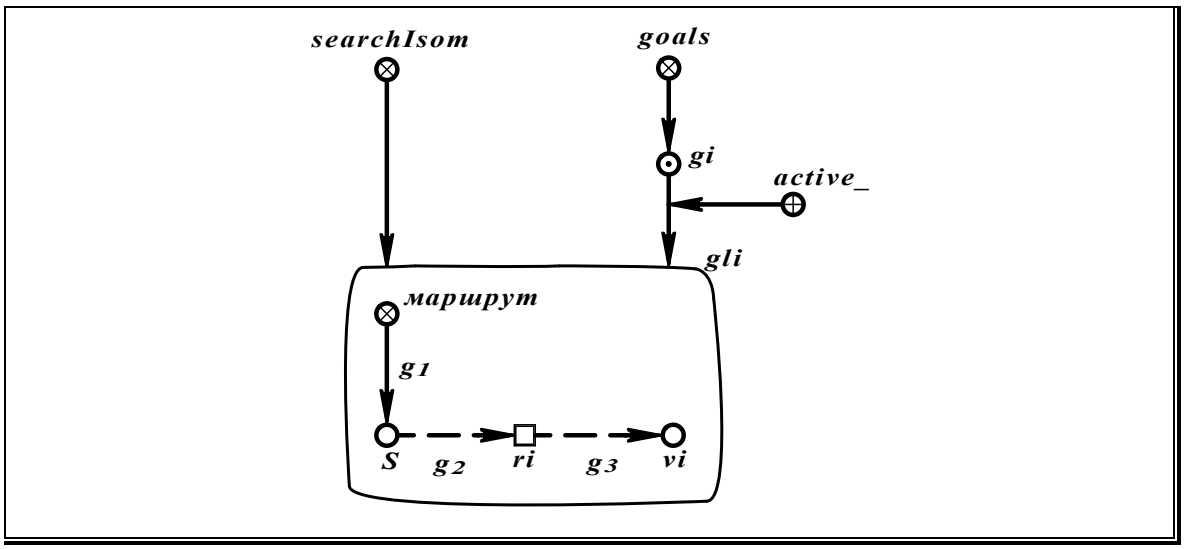
Ключевые понятия: цель; изоморфный поиск; результат.

Рассмотрим работу навигационно-поисковой графодинамической ассоциативной машины на примере операции изоморфного поиска.

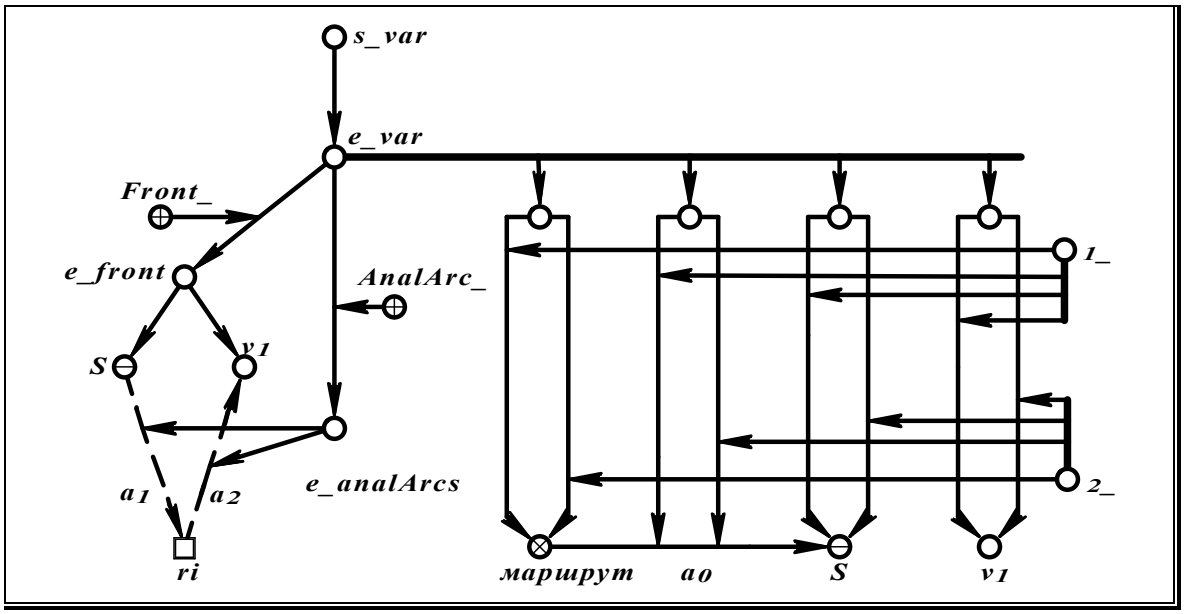
Пусть в памяти имеется конструкция, описывающая маршрут S в некотором графе, и нам необходимо найти участки этого маршрута, которые проходят через вершину $v1$.



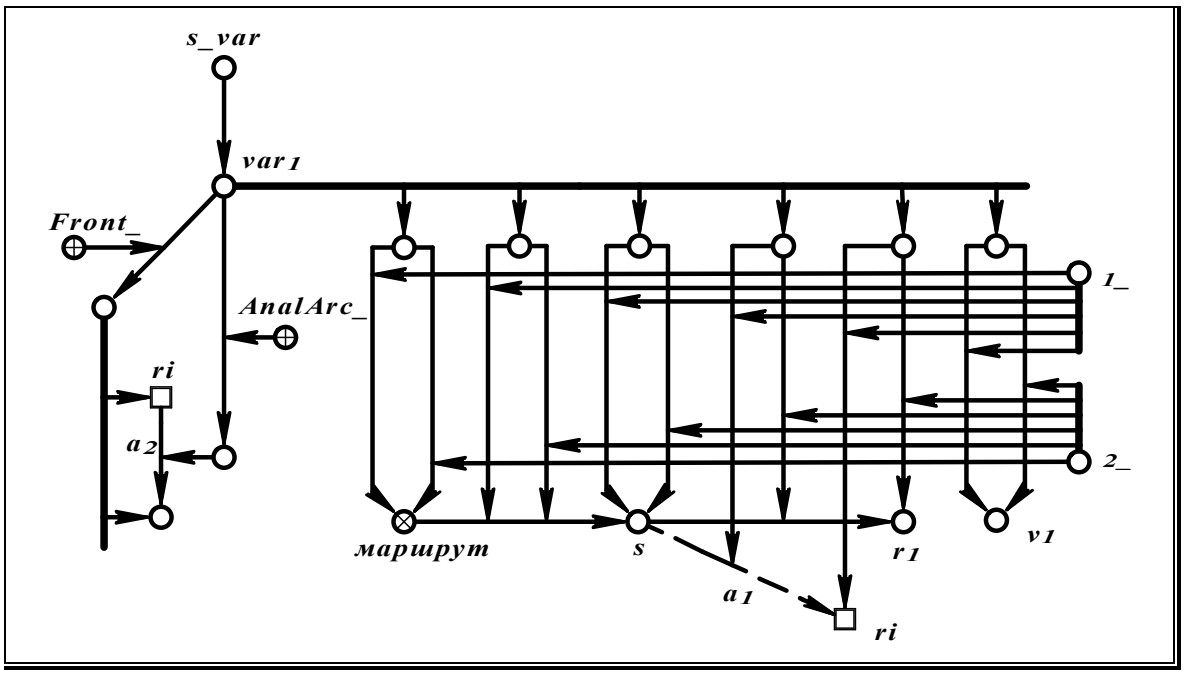
Конструкция цели выглядит следующим образом.

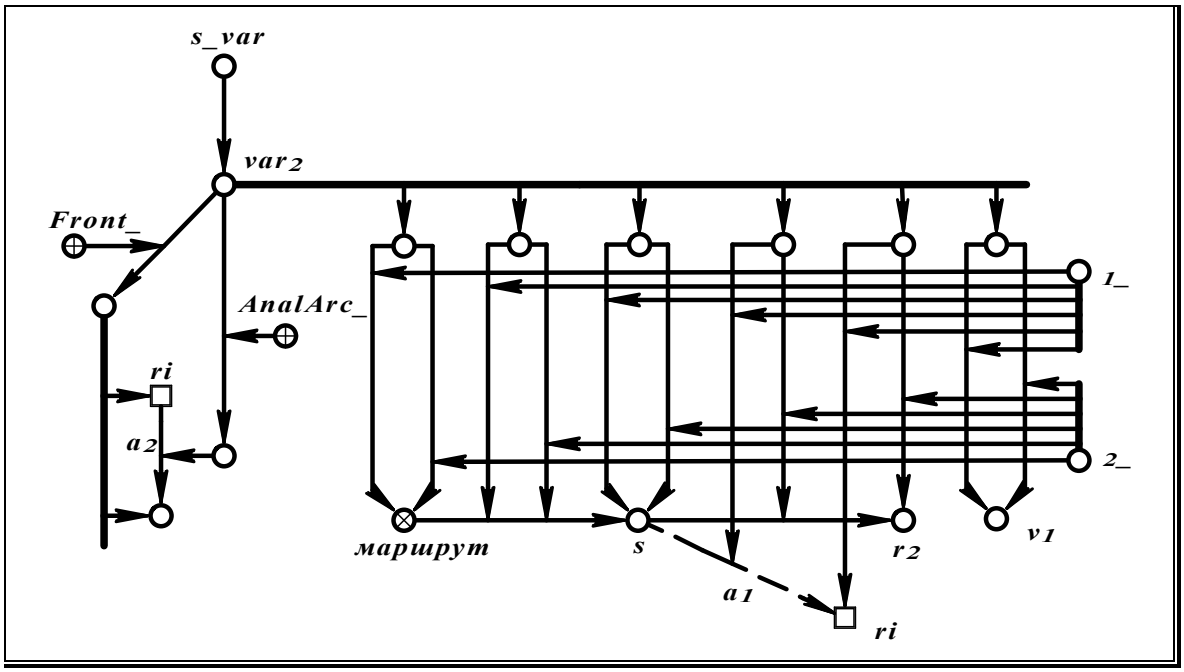


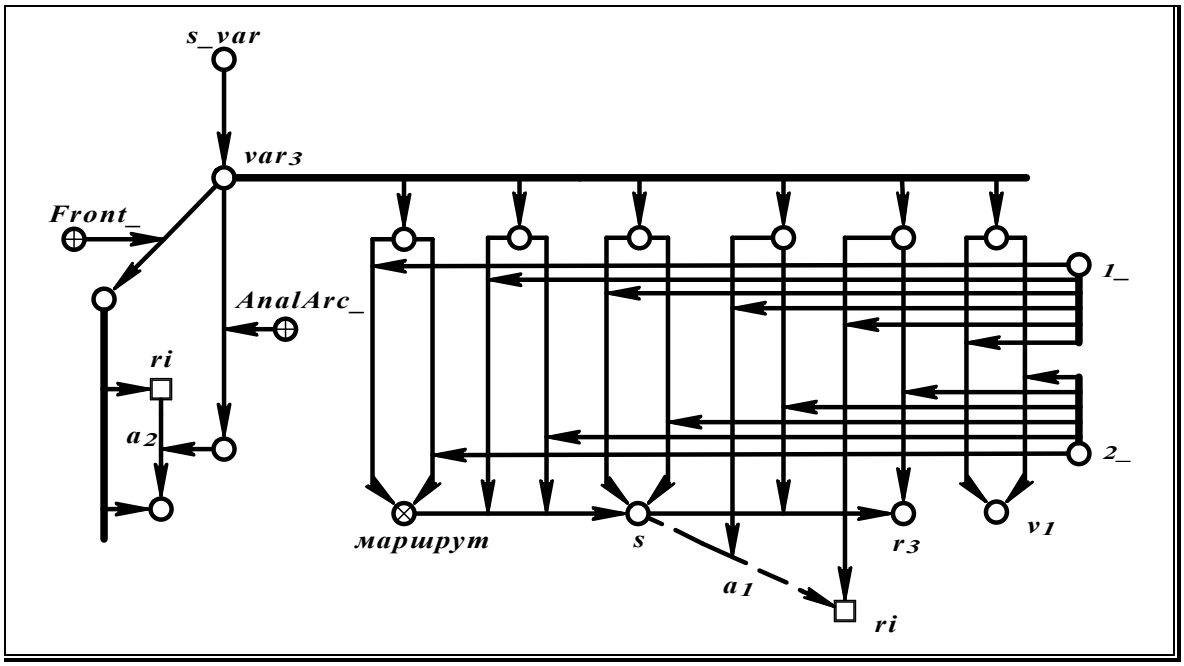
На первом этапе (шаги 1-7) происходит поиск запроса, разбор шаблона. Так константные элементы шаблона заносятся во множество $front$, ставятся в соответствие сами себе в рамках начального варианта, переменные дуги заносятся во множество $analArcs$:



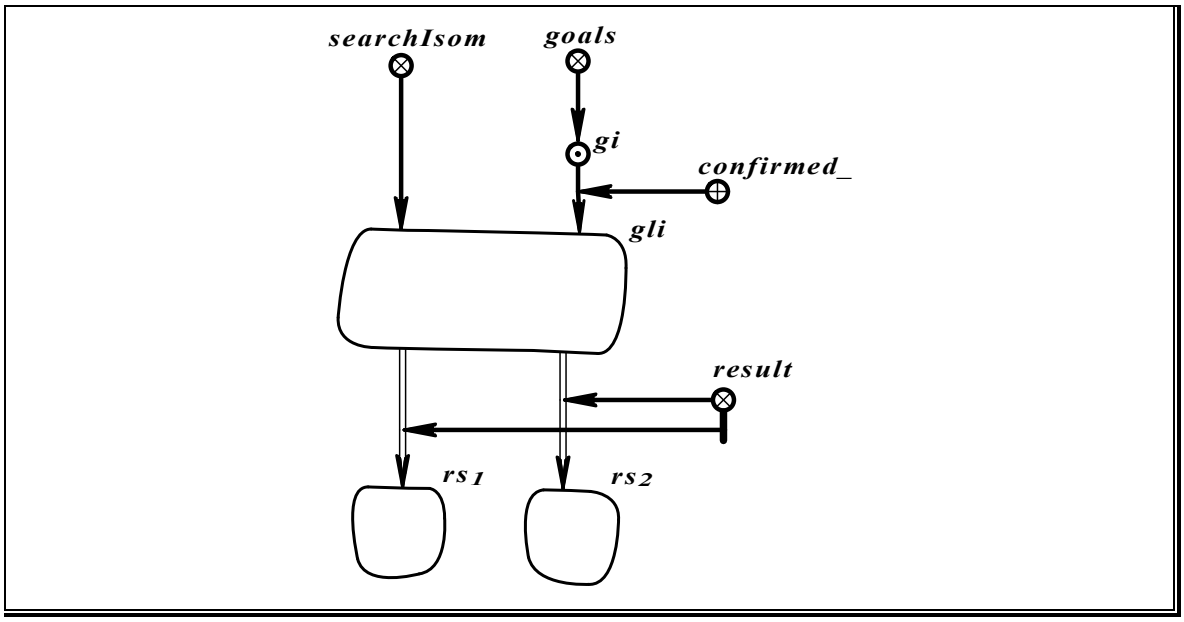
Далее начинается разбор вариантов, находящихся во множестве s_var. Берем любой элемент множества e_front, например, возьмем узел S. Берем любую дугу инцидентную узлу S и принадлежащую множеству $e\_analArcs$. В нашем случае это будет дуга $a1$, как единственно возможная. Затем формируется множество дуг, которые могут быть поставлены в соответствие этой дуге. То есть ищем все константные дуги, выходящие из узла S (т.к. этот узел константный, то он соответствует сам себе) и которые входят в константный узел (дуга $a1$ входит в переменный узел), при этом найденные дуги и узлы, в которые они входят, должны не иметь соответствий в рамках данного варианта. В нашем случае это будут три дуги: $(S\ !;r1)$, $(S\ !;r2)$, $(S\ !;r3)$. Формируем новые варианты и удаляем старый. Так, в результате обработки начального варианта получится три новых варианта:



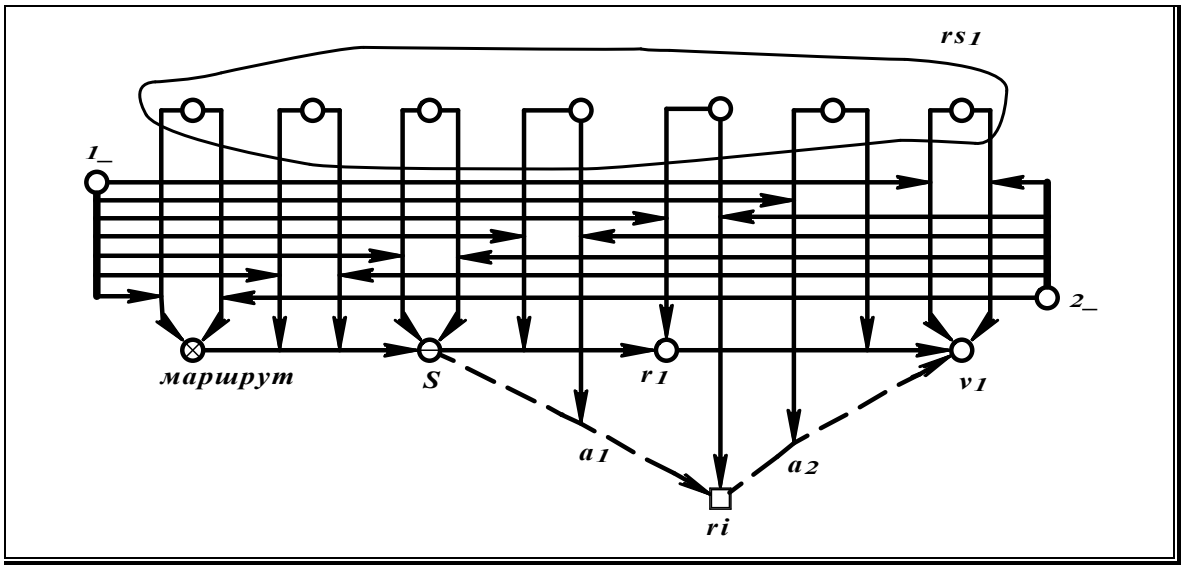




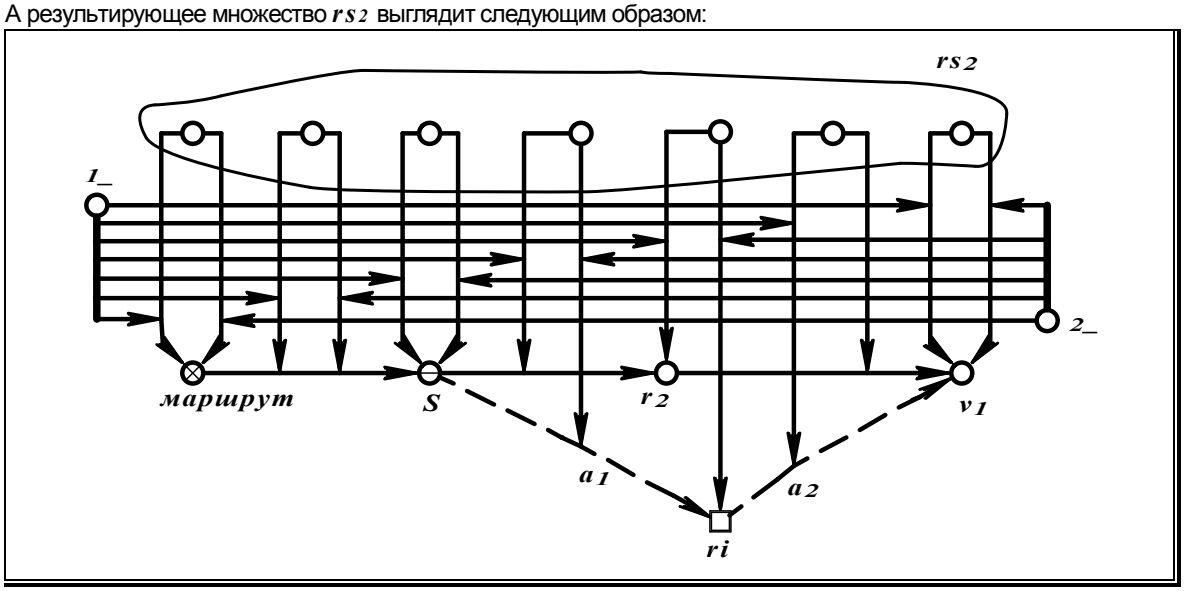
Далее процесс повторяется: берем вариант из множества вариантов, проверяем, закончен ли он (если закончен, то множество ***analArcs*** будет пустым). Если вариант не закончен, то пытаемся развить его дальше. То есть берем элемент фронта, инцидентную ему дугу из множества ***analArcs*** и ищем те элементы, которые могут быть сопоставлены им. Если таких элементов не оказывается, то данный вариант уничтожается и процесс начинается сначала. В противном случае для каждого возможного соответствия генерируется новый вариант. Если множество вариантов оказывается пустым, то уничтожаем знак этого множества и указываем, что операция завершилась. Так выглядит результат выполнения операции в нашем примере:



Здесь результирующее множество *rs1* выглядит следующем образом:



А результирующее множество rs_2 выглядит следующим образом:



Выводы к разделу 7

Рассмотренная в данном разделе навигационно-поисковая графодинамическая ассоциативная машина является частью любой другой графодинамической ассоциативной машины.

Одним из актуальных направлений использования навигационно-поисковой графодинамической ассоциативной машины являются ассоциативные электронные учебники, в которых учебный материал представлен в виде гипертекстовой семантической сети (см. подраздел 6.4). Подробно ассоциативные электронные учебники рассмотрены в книге [236] (***ИнтелОСиВУО-2001кн***).

Реализация навигационно-поисковой графодинамической ассоциативной машины и пользовательский интерфейс рассмотрены в книге [411] (***ПрогрВАМ-2001кн***).