Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра интеллектуальных информационных технологий

ОТЧЁТ

по лабораторной работе №2

по курсу «Логические основы интеллектуальных систем»

|  |  |
| --- | --- |
| Проверил:  Ивашенко В.П. | Выполнила:  студент гр. 121701  Волкова О. А. |

Минск, 2015

**Выбранное индивидуальное задание:**

* Реализовать процедуру добавления элементов первого домена заданного слотового sc-отношения в заданное sc-множество.
* Реализовать процедуру добавления элементов второго домена заданного слотового sc-отношения в заданное sc-множество.

Для выполнения лабораторной работы, был выполнен набор шагов, отчет о прохождении которых представлен в данном отчете. При демонстрации результатов выполнения программы, из лога была исключена информация, не имеющая отношение к алгоритму выполнения программы (autosegment, process, unlink и проч.).

Тестовая процедура для подготовки тестовых структур и вызова процедуры.

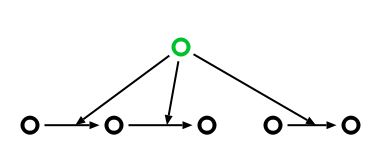


Рисунок 1 – тестовое слотовое отношение

genEl([1\_: assign\_: node\_: const\_: empty])

genEl([1\_: assign\_: node\_: const\_: relA])

genElStr5([

1\_: assign\_: node\_: const\_: nodeA,

2\_: assign\_: arc\_: const\_: pos\_: permanent\_: actual\_: arc2,

3\_: assign\_: node\_: const\_: nodeB,

4\_: assign\_: arc\_: const\_: pos\_: permanent\_: actual\_: arc4,

5\_: fixed\_: relA

])

genElStr5([

1\_: assign\_: node\_: const\_: nodeB,

2\_: assign\_: arc\_: const\_: neg\_: permanent\_: actual\_: arc2,

3\_: assign\_: node\_: const\_: nodeC,

4\_: assign\_: arc\_: const\_: pos\_: permanent\_: actual\_: arc4,

5\_: fixed\_: relA

])

genElStr5([

1\_: assign\_: node\_: const\_: nodeD,

2\_: assign\_: arc\_: const\_: pos\_: permanent\_: actual\_: arc2,

3\_: assign\_: node\_: const\_: nodeE,

4\_: assign\_: arc\_: const\_: pos\_: permanent\_: actual\_: arc4,

5\_: fixed\_: relA

])

genEl([1\_: assign\_: node\_: const\_: state])

genEl([1\_: assign\_: node\_: const\_: result])

genEl([1\_: assign\_: node\_: const\_: logging])

call([1\_: fixed\_: chain\_push\_first, 2\_: fixed\_: prmp1 = [{1\_: relA, 2\_: args\_chain, 3\_: logging}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_push\_first, 2\_: fixed\_: prmp3 = [{1\_: args\_chain, 2\_: history, 3\_: logging}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_push\_first, 2\_: fixed\_: prmp4 = [{1\_: set\_add\_element\_from\_second\_domain, 2\_: parameters, 3\_: logging}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_push, 2\_: fixed\_: prm1 = [{1\_: parameters, 2\_: result, 3\_: parameters, 4\_: logging}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: set\_create\_set\_of\_rel\_first\_domain, 2\_: fixed\_: prmc1 = [{1\_: state, 2\_: empty, 3\_: history, 4\_: parameters, 5\_: logging}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

printEl([1\_: fixed\_: result])

label(finish)

call([1\_: fixed\_: chain\_destroy, 2\_: fixed\_: prmd1 = [{1\_: parameters, 2\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_destroy, 2\_: fixed\_: prmd2 = [{1\_: history, 2\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_destroy, 2\_: fixed\_: prmd3 = [{1\_: args\_chain, 2\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

eraseEl([1\_: fixed\_: f\_: relA])

eraseEl([1\_: fixed\_: f\_: nodeA])

eraseEl([1\_: fixed\_: f\_: nodeB])

eraseEl([1\_: fixed\_: f\_: nodeC])

eraseEl([1\_: fixed\_: f\_: nodeE])

eraseEl([1\_: fixed\_: f\_: result])

eraseEl([1\_: fixed\_: f\_: state])

eraseEl([1\_: fixed\_: f\_: empty])

DBG(M4SCS(print([1\_: fixed\_: leave\_message])))

return()

end()

Процедура добавления элементов первого (второго) домена слотового отношения имеет стандартный интерфейс, в котором в стеке находится цепочка с рассматриваемым отношением, а в качетсве параметров передается множество, в которое будет помещен результат, а также процедура добавления элемента первого (второго) домена в результирующее множетсво, а также процедура создания итератора (можно не указывать). Данное изменение в интерфейсе описано на основании того, что тело процедуры, в зависимости от того, какой домен выбран, изменяется только в месте вызова конкретной процедуры для конкретного домена.

genEl([1\_: assign\_: node\_: const\_: empty])

call([1\_: fixed\_: chain\_get\_next, 2\_: fixed\_: prmg1 = [{1\_: history, 2\_: link, 3\_: args\_chain}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_get\_next, 2\_: fixed\_: prmg4 = [{1\_: args\_chain, 2\_: args\_chain2, 3\_: relationA}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_get\_next, 2\_: fixed\_: prmg2 = [{1\_: parameters, 2\_: link, 3\_: result\_set}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

ifVarAssign([1\_: link],, final)

call([1\_: fixed\_: chain\_get\_next, 2\_: fixed\_: prmg3 = [{1\_: link, 2\_: link, 3\_: n\_domain\_procedure}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

ifVarAssign([1\_: link],, final)

call([1\_: fixed\_: chain\_get\_next, 2\_: fixed\_: prmg5 = [{1\_: link, 2\_: link, 3\_: iterator\_create}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

ifVarAssign([1\_: iterator\_create], do)

varAssign([1\_: assign\_: iterator\_create, 2\_: fixed\_: iterator\_create\_apa\_output])

label(do)

call([1\_: fixed\_: chain\_push\_first, 2\_: fixed\_: prmp1 = [{1\_: iterator\_create, 2\_: outlink1, 3\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_push, 2\_: fixed\_: prmp2 = [{1\_: outlink1, 2\_: empty, 3\_: outlink1, 4\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

genElStr3([

1\_: assign\_: node\_: const\_: singleton,

2\_: assign\_: arc\_: const\_: pos\_: temporary\_: actual\_: arc2,

3\_: fixed\_: result\_set

])

call([1\_: fixed\_: chain\_push\_first, 2\_: fixed\_: prmp3 = [{1\_: singleton, 2\_: outlink2, 3\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_push\_first, 2\_: fixed\_: prmp4 = [{1\_: outlink2, 2\_: result\_set\_link, 3\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_push, 2\_: fixed\_: prmp5 = [{1\_: result\_set\_link, 2\_: n\_domain\_procedure, 3\_: result\_set\_link, 4\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_push, 2\_: fixed\_: prmp6 = [{1\_: outlink1, 2\_: result\_set\_link, 3\_: outlink1, 4\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_push, 2\_: fixed\_: prmp7 = [{1\_: outlink1, 2\_: relationA, 3\_: outlink1, 4\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: iterator\_embedded\_cycle\_filter, 2\_: fixed\_: prmf1 = [{1\_: empty, 2\_: empty, 3\_: history, 4\_: outlink1, 5\_: logging}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

label(final)

call([1\_: fixed\_: chain\_destroy, 2\_: fixed\_: prmd1 = [{1\_: outlink1, 2\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([ 1\_: fixed\_: chain\_destroy, 2\_: fixed\_: prmd2 = [{1\_: result\_set\_link, 2\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([ 1\_: fixed\_: chain\_destroy, 2\_: fixed\_: prmd3 = [{1\_: outlink2, 2\_: empty}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

eraseEl([1\_: fixed\_: f\_: singleton])

eraseEl([1\_: fixed\_: f\_: empty])

Процедура добавления элемента первого домена заданного отношения в результирующее множество:

genEl([1\_: assign\_: node\_: const\_: empty])

call([1\_: fixed\_: chain\_get\_next\_singlet, 2\_: fixed\_: prmg1 = [{1\_: parameters, 2\_: link, 3\_: result\_set}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_get\_next\_singlet, 2\_: fixed\_: prmg2 = [{1\_: history, 2\_: link, 3\_: arcA}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

searchElStr3([

1\_: assign\_: node\_: const\_: relationA,

2\_: fixed\_: arcA,

3\_: assign\_: arc\_: const\_: actual\_: elementArcA

], ,)

searchElStr3([

1\_: assign\_: node\_: const\_: elem1,

2\_: fixed\_: elementArcA,

3\_: assign\_: const\_: elemA

], ,)

label(add\_to\_res\_set)

genElStr3([

1\_: fixed\_: result\_set,

2\_: assign\_: arc\_: const\_: actual\_: pos\_: permanent\_: newElementArc,

3\_: fixed\_: elem1

])

label(final)

eraseEl([1\_: fixed\_: f\_: empty])

Процедура добавления элемента второго домена заданного отношения в результирующее множество:

genEl([1\_: assign\_: node\_: const\_: empty])

call([1\_: fixed\_: chain\_get\_next\_singlet, 2\_: fixed\_: prmg1 = [{1\_: parameters, 2\_: link, 3\_: result\_set}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

call([1\_: fixed\_: chain\_get\_next\_singlet, 2\_: fixed\_: prmg2 = [{1\_: history, 2\_: link, 3\_: arcA}], 3\_: assign\_: descr])

waitReturn([1\_: fixed\_: descr])

searchElStr3([

1\_: assign\_: node\_: const\_: relationA,

2\_: fixed\_: arcA,

3\_: assign\_: arc\_: const\_: actual\_: elementArcA

], ,)

searchElStr3([

1\_: assign\_: node\_: const\_: elem1,

2\_: fixed\_: elementArcA,

3\_: assign\_: const\_: elemA

], ,)

label(add\_to\_res\_set)

genElStr3([

1\_: fixed\_: result\_set,

2\_: assign\_: arc\_: const\_: actual\_: pos\_: permanent\_: newElementArc,

3\_: fixed\_: elemA

])

label(final)

eraseEl([1\_: fixed\_: f\_: empty])

**Результат выполнения программы (второй домен)**

test\_create\_set\_of\_rel\_first\_domain: entering

result set:

printEl: node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@14

Output arcs:

Input arcs:

nodes:

printEl: node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@7

Output arcs:

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@9 >- node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@8

Input arcs:

printEl: node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@b

Output arcs:

arc|const|neg|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@d >- node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@c

Input arcs:

printEl: node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@c

Output arcs:

Input arcs:

arc|const|neg|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@d -< node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@b

printEl: node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@f

Output arcs:

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@11 >- node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@10

Input arcs:

printEl: node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@10

Output arcs:

Input arcs:

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@11 -< node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@f

relation:

printEl: node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@6

Output arcs:

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@a >- arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@9

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@e >- arc|const|neg|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@d

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@12 >- arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@11

Input arcs:

chain\_push\_first: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_first: leaving

chain\_push\_first: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_first: leaving

chain\_push\_first: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_first: leaving

chain\_push: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push: leaving

Process 024F7FF8 is dying now

set\_create\_set\_of\_rel\_first\_domain: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_first: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_first: leaving

chain\_push: entering

chain\_push: leaving

chain\_push\_first: entering

chain\_push\_first: leaving

chain\_push\_first: entering

chain\_push\_first: leaving

chain\_push: entering

chain\_push: leaving

chain\_push: entering

chain\_push: leaving

chain\_push: entering

chain\_push: leaving

iterator\_embedded\_cycle\_filter: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

set\_add\_element\_from\_second\_domain: entering

chain\_get\_next\_singlet: entering

chain\_get\_next\_singlet: leaving

chain\_get\_next\_singlet: entering

chain\_get\_next\_singlet: leaving

set\_add\_element\_from\_second\_domain: leaving

chain\_pop\_destroy\_singlet: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_pop\_destroy\_singlet: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_singlet: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_singlet: leaving

set\_add\_element\_from\_second\_domain: entering

chain\_get\_next\_singlet: entering

chain\_get\_next\_singlet: leaving

chain\_get\_next\_singlet: entering

chain\_get\_next\_singlet: leaving

set\_add\_element\_from\_second\_domain: leaving

chain\_pop\_destroy\_singlet: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_pop\_destroy\_singlet: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_singlet: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_push\_singlet: leaving

set\_add\_element\_from\_second\_domain: entering

chain\_get\_next\_singlet: entering

chain\_get\_next\_singlet: leaving

chain\_get\_next\_singlet: entering

chain\_get\_next\_singlet: leaving

set\_add\_element\_from\_second\_domain: leaving

chain\_pop\_destroy\_singlet: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_pop\_destroy\_singlet: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

iterator\_embedded\_cycle\_filter: leaving

chain\_destroy: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_destroy: leaving

chain\_destroy: entering

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_get\_next: entering

chain\_get\_next: leaving

chain\_destroy: leaving

chain\_destroy: entering

chain\_destroy: leaving

set\_create\_set\_of\_rel\_first\_domain: leaving

printEl: node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@14

Output arcs:

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@e0 >- node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@8

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@109 >- node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@c

arc|const|pos|permanent|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@132 >- node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@10

Input arcs:

arc|const|pos|temporary|actual:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@3d -< node|const:/tmp/autoseg/@@3785e451-0c57-4692-b4b9-d2425095c3dd@31

chain\_destroy: entering

chain\_destroy: leaving

chain\_destroy: entering

chain\_destroy: leaving

chain\_destroy: entering

chain\_destroy: leaving

**Результат выполнения программы (первый домен)**

printEl: node|const:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@14

Output arcs:

arc|const|pos|permanent|actual:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@d3 >- node|const:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@7

arc|const|pos|permanent|actual:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@fc >- node|const:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@b

arc|const|pos|permanent|actual:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@125 >- node|const:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@f

Input arcs:

arc|const|pos|temporary|actual:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@32 -< node|const:/tmp/autoseg/@@17689efa-1570-4692-8582-51d124e22b2a@30

**Вывод:**

В ходе лабораторной работы было изучено использование итераторов множеств, их виды. При реализации алгоритма определения первого (второго) домена слотового отношения были использованы итераторы для перебора выходящих из заданного множества дуг. Помимо самой структуры итератора, в ходе работы были изучены процедуры для работы с ними, которые представляют собой циклический перебор характеризованной в теле итератора дуги заданного множества до тех пор, пока условие истинно, либо множество непросмотренных дуг непустое. В ходе создания кода процедур были использованы тексты процедур создания композиции слотовых отношений (автор – Яльчик Илья).