Цель

Разработка экспертной системы родственных связей и изучение возможностей среды разработки.

Описание выбранной предметной области

ЭС устанавливает тип отношений между членами семьи (кто кому в семье приходится). ЭС также подразумевают работу с различными поколениями (например, определение следующих отношений: бабушка, прадедушка, прапрабабушка и т.д.), с различной степенью родства (например, определение следующих отношений: тетя, двоюродный дядя, троюродная тетя и т.д.)

Способ задания фактов: [status] [who] [whom], где [status] – тип отношений, [who] – кто, [whom] – для кого.

[who] является [status] для [whom]. Например, Евгений([who]) является дядей([status]) для Маши([whom]).

ЭС сначала предлагает ввести некоторые известные отношения родства, затем из известных выводит новые.

Рассмотрим простой пример для того, чтобы понять работу ЭС. В ЭС вводятся следующие факты в заданном выше формате команд: Брат Николай Александр, Отец Павел Александр, Мать Екатерина Павел. Следующие правила будут участвовать в выводе новых фактов (исходя из введенных фактов):

```
(Правило1
(Брат ?x ?y)
=>
(Брат ?y ?x)
)
(Правило2
(Брат ?x ?y)
(Отец ?z ?x)
=>
(Отец ?z ?y)
)
(Правило3
```

(Мать ?x ?y)
(Отец ?y ?z)
=>
(Бабушка ?z ?y)
)

Новые добавленные факты: Брат Александр Николай, Отец Павел Николай, Бабушка Екатерина Александр, Бабушка Екатерина Николай.

Для того, чтобы система не была слишком сложной для восприятия, определим отношения снизу вверх (т.е., старшего поколения к младшему), на одном уровне (одно и то же поколение, например, брат/сестра, кузен/кузина), но НЕ сверху вниз (т.е., младшего поколения к старшему). Например, в системе может быть факт «Мать Екатерина Павел», но НЕ «Сын Павел Екатерина».

Также будем различать пол только для следующих отношений: муж и жена, отец и мать, бабушка и дедушка, прабабушка и прадедушка, и т.д. Для отношения «брат» и «сестра» пол не различается, в ЭС это будет один факт записанный следующим образом «Брат/Сестра». Аналогично с отношениями дядя и тетя: «Дядя/Тетя», кузен и кузина: «Кузен/Кузина».

После введенных ограничений система остается полной.

Все правила экспертной системы (в псевдокоде):

1) жена 1 2
=> муж 2 1
2) муж 1 2
=> жена 2 1
3) брат/сестра 1 2
=> брат/сестра 2 1
4) брат/сестра 1 2
   брат/сестра 1 3
   => брат/сестра 2 3

5) мать 1 2
   жена 1 3
=> отец 3 2
6) отец 1 2
   муж 1 3
=> мать 3 2

7) брат/сестра 1 2
   мать 3 1
=> мать 3 2
8) мать 1 2
   мать 1 3
=> брат/сестра 2 3
9) отец 1 2
   отец 1 3
=> брат/сестра 2 3

10) мать 1 2
     брат/сестра 1 3
=> дядя/тетя 3 2
11) отец 1 2
     брат/сестра 1 3
=> дядя/тетя 3 2
12) дядя/тетя 1 2
     мать 1 3
=> кузен/кузина 2 3
13) дядя/тетя 1 2
     отец 1 3
=> кузен/кузина 2 3
14) кузен/кузина 1 2
=> кузен/кузина 2 1

15) мать 1 2
     мать 3 1
=> бабушка 3 2
16) отец 1 2
     мать 3 1
=> бабушка 3 2
17) мать 1 2
     отец 3 1
=> дедушка 3 2
18) отец 1 2
     отец 3 1
=> дедушка 3 2

19) муж 1 2
     мать 3 1
=> свекровь 3 2
20) муж 1 2
     отец 3 1
=> свекор 3 2
21) жена 1 2

мать 3 1
=> теща 3 2
22) жена 1 2
отец 3 1
=> тесть 3 2

23) муж 1 2
брат/сестра 1 3
=> brother/sister-in-law 3 2
24) жена 1 2
брат/сестра 1 3
=> brother/sister-in-law 3 2

25) мать 1 2
бабушка 2 3
=> прабабушка 1 3 1
26) прабабушка 1 3 1
мать 4 1
=> прапрабабушка 4 3 1+1
27) отец 1 2
бабушка 2 3
=> прадедушка 1 3 1
28) прабабушка 1 3 1
отец 4 1
=> прапрадедушка 4 3 1+1
29) отец 1 2
дедушка 2 3
=> прадедушка 1 3 1
30) прадедушка 1 3 1
отец 4 1
=> прапрадедушка 4 3 1+1
31) мать 1 2
дедушка 2 3
=> прабабушка 1 3 1
32) прадедушка 1 3 1
мать 4 1
=> прапрабабушка 4 3 1+1

33) мать 1 2
кузен/кузина 1 3
=> дядя/тетя 3 2 2
34) отец 1 2
кузен/кузина 1 3
=> дядя/тетя 3 2 2
35) дядя/тетя 3 2 2

кузен/кузина 3 4
НЕ (мать 4 2 ИЛИ отец 4 2)
НЕ дядя/тетя 4 2 <2
=> дядя/тетя 4 2 2+1
36) брат/сестра 1 2
кузен/кузина 1 3
=> кузен/кузина 2 3

Текст программы ЭС на языке CLIPS.

```
(defglobal ?*x* = 1)

(defrule data-input
  (initial-fact)
=>
  (printout t crlf "Write in format: [Status] [who] [for whom]" crlf)
  (while (= ?*x* 1)
  (bind ?input (explode$ (readline)))
  (if (eq (nth$ 1 ?input) stop)
   then
      (halt)
           (bind ?*x* 0)
   else
      (assert (data (nth$ 1 ?input) (nth$ 2 ?input)(nth$ 3 ?input)))))
  )

 (defrule r1
 (data wife ?x ?y)
 =>
 (printout t crlf ?y " is husband of " ?x crlf)
 (assert(data husband ?y ?x)))

 (defrule r2
 (data husband ?x ?y)
 =>
 (printout t crlf ?y " is wife of " ?x crlf)
 (assert(data wife ?y ?x)))

 (defrule r3
 (data brother/sister ?x ?y)
 =>
 (printout t crlf ?y " is brother/sister of " ?x crlf)
 (assert(data brother/sister ?y ?x)))
```

```
(defrule r4
(data brother/sister ?x ?y)
(data brother/sister ?x ?z)
(test(neq ?z ?y))
=>
(printout t crlf ?y " is  brother/sister of " ?z crlf)
(assert(data brother/sister ?y ?z)))

(defrule r36
(data father ?x ?y)
(data father ?x ?z)
(test(neq ?y ?z))
=>
(printout t crlf ?z " is brother/sister of " ?y crlf)
(assert(data brother/sister ?z ?y)))

(defrule r37
(data mother ?x ?y)
(data mother ?x ?z)
(test(neq ?y ?z))
=>
(printout t crlf ?z " is brother/sister of " ?y crlf)
(assert(data brother/sister ?z ?y)))

(defrule r0
(data father ?x ?y)
(data mother ?z ?y)
=>
(printout t crlf ?z " is wife of " ?x crlf)
(assert(data wife ?z ?x)))

(defrule r5
(data mother ?x ?y)
(data wife ?x ?z)
=>
(printout t crlf ?z " is  father of " ?y crlf)
(assert(data father ?z ?y)))

(defrule r6
(data father ?x ?y)
(data husband ?x ?z)
=>
(printout t crlf ?z " is  mother of " ?y crlf)
(assert(data mother ?z ?y)))
```

```
(defrule r7
(data mother ?x ?y)
(data brother/sister ?y ?z)
=>
(printout t crlf ?x " is mother of " ?z crlf)
(assert(data mother ?x ?z)))

(defrule r8
(data father ?x ?y)
(data brother/sister ?y ?z)
=>
(printout t crlf ?x " is father of " ?z crlf)
(assert(data father ?x ?z)))

(defrule r9
(data mother ?x ?y)
(data brother/sister ?x ?z)
=>
(printout t crlf ?z " is uncle/aunt of " ?y crlf)
(assert(data uncle/aunt ?z ?y)))

(defrule r10
(data father ?x ?y)
(data brother/sister ?x ?z)
=>
(printout t crlf ?z " is uncle/aunt of " ?y crlf)
(assert(data uncle/aunt ?z ?y)))

(defrule r11
(data uncle/aunt ?x ?y)
(data mother ?x ?z)
=>
(printout t crlf ?y " is cousin of " ?z crlf)
(assert(data cousin ?y ?z)))

(defrule r12
(data uncle/aunt ?x ?y)
(data father ?x ?z)
=>
(printout t crlf ?z " is cousin of " ?y crlf)
(assert(data cousin ?y ?z)))

(defrule r13
(data cousin ?x ?y)
```

```
=>
(printout t crlf ?y " is cousin of " ?x crlf)
(assert(data cousin ?y ?x)))

(defrule r14
(data mother ?x ?y)
(data mother ?z ?x)
=>
(printout t crlf ?z " is grandmother of " ?y crlf)
(assert(data grandmother ?z  ?y)))

(defrule r15
(data father ?x ?y)
(data mother ?z ?x)
=>
(printout t crlf ?z " is grandmother of " ?y crlf)
(assert(data grandmother ?z ?y)))

(defrule r16
(data mother ?x ?y)
(data father ?z ?x)
=>
(printout t crlf ?z " is grandfather  of " ?y crlf)
(assert(data grandfather  ?z  ?y)))

(defrule r17
(data father ?x ?y)
(data father ?z ?x)
=>
(printout t crlf ?z " is grandfather of " ?y crlf)
(assert(data grandfather ?z ?y)))

(defrule r18
(data husband ?x ?y)
(data mother ?z ?x)
=>
(printout t crlf ?z " is svekrov of " ?y crlf)
(assert(data svekrov ?z ?y)))

(defrule r19
(data husband ?x ?y)
(data father ?z ?x)
=>
(printout t crlf ?z " is svekor of " ?y crlf)
(assert(data svekor ?z ?y)))
```

```
(defrule r20
(data wife ?x ?y)
(data mother ?z ?x)
=>
(printout t crlf ?z " is teshcha of " ?y crlf)
(assert(data teshcha ?z ?y)))

(defrule r21
(data wife ?x ?y)
(data father ?z ?x)
=>
(printout t crlf ?z " is test of " ?y crlf)
(assert(data test ?z ?y)))

(defrule r22
(data husband ?x ?y)
(data brother/sister ?x ?z)
=>
(printout t crlf ?z " is brother/sister-in-law of " ?y crlf)
(assert(data brother/sister-in-law ?z ?y)))

(defrule r23
(data wife ?x ?y)
(data brother/sister ?x ?z)
=>
(printout t crlf ?z " is brother/sister-in-law of " ?y crlf)
(assert(data brother/sister-in-law ?z ?y)))

(defrule r24
(data mother ?x ?y)
(data grandmother ?y ?z)
=>
(printout t crlf ?x " is great-grandmother 1 of " ?z crlf)
(assert(data great-grandmother ?x ?z 1)))

(defrule r25
(data mother ?x ?y)
(data great-grandmother ?y ?z ?st)
=>
(printout t crlf ?x " is great-grandmother " ?st " of " ?z crlf)
(assert(data great-grandmother ?x ?z (+ 1 ?st))))

(defrule r26
(data father ?x ?y)
```

```
(data grandmother ?y ?z)
=>
(printout t crlf ?x " is great-grandfather 1 of " ?z crlf)
(assert(data great-grandfather ?x ?z 1)))

(defrule r27
(data father ?x ?y)
(data great-grandmother ?y ?z ?st)
=>
(printout t crlf ?x " is great-grandfather " (+ 1 ?st) " of " ?z crlf)
(assert(data great-grandfather ?x ?z (+ 1 ?st))))

(defrule r28
(data mother ?x ?y)
(data grandfather ?y ?z)
=>
(printout t crlf ?x " is great-grandmother  1 of " ?z crlf)
(assert(data great-grandmother  ?x ?z 1)))

(defrule r29
(data mother ?x ?y)
(data great-grandfather ?y ?z ?st)
=>
(printout t crlf ?x " is great-grandmother " (+ 1 ?st)" of " ?z crlf)
(assert(data great-grandmother ?x ?z (+ 1 ?st))))

(defrule r30
(data father ?x ?y)
(data grandfather ?y ?z)
=>
(printout t crlf ?x " is great-grandfather 1 of " ?z crlf)
(assert(data great-grandfather ?x ?z 1)))

(defrule r31
(data father ?x ?y)
(data great-grandfather ?y ?z ?st)
=>
(printout t crlf ?x " is great-grandfather " (+ 1 ?st) " of " ?z crlf)
(assert(data great-grandfather ?x ?z (+ 1 ?st))))

(defrule r32
(data mother ?x ?y)
(data cousin ?x ?z)
=>
(printout t crlf ?z " is uncle/aunt 2 of " ?y crlf)
```

```
(assert(data uncle/aunt ?z ?y 2)))

(defrule r33
(data uncle/aunt ?x ?y ?st)
(data cousin ?x ?z)
(not(exists(or(data mother ?z ?y)(data father ?z ?y))))
(not(exists(data uncle/aunt ?z ?y $?)))
=>
(printout t crlf ?z " is uncle/aunt " (+ 1 ?st) " of " ?y crlf)
(assert(data uncle/aunt ?z ?y (+ 1 ?st))))

(defrule r34
(data father ?x ?y)
(data cousin ?x ?z)
=>
(printout t crlf ?z " is uncle/aunt 2 of " ?y crlf)
(assert(data uncle/aunt ?z ?y 2)))

(defrule r35
(data brother/sister ?x ?y)
(data cousin ?x ?z)
=>
(printout t crlf ?y " is cousin of " ?z crlf)
(assert(data cousin ?y ?z)))
```

Результаты работы программы ЭС (скриншоты с комментариями).

При запуске программы изначально необходимо ввести факты в заданной форме: [status] [who] [whom], где [status] – тип отношений, [who] – кто, [whom] – для кого.

Небольшое уточнение записи некоторых фактов в среде CLIPS:

«Прабабушка x для y» - будет great-grandmother x y 1. «Прапрабабушка x для y» - будет great-grandmother x y 2

Т.е. последнее число в факте обозначает количество «пра».

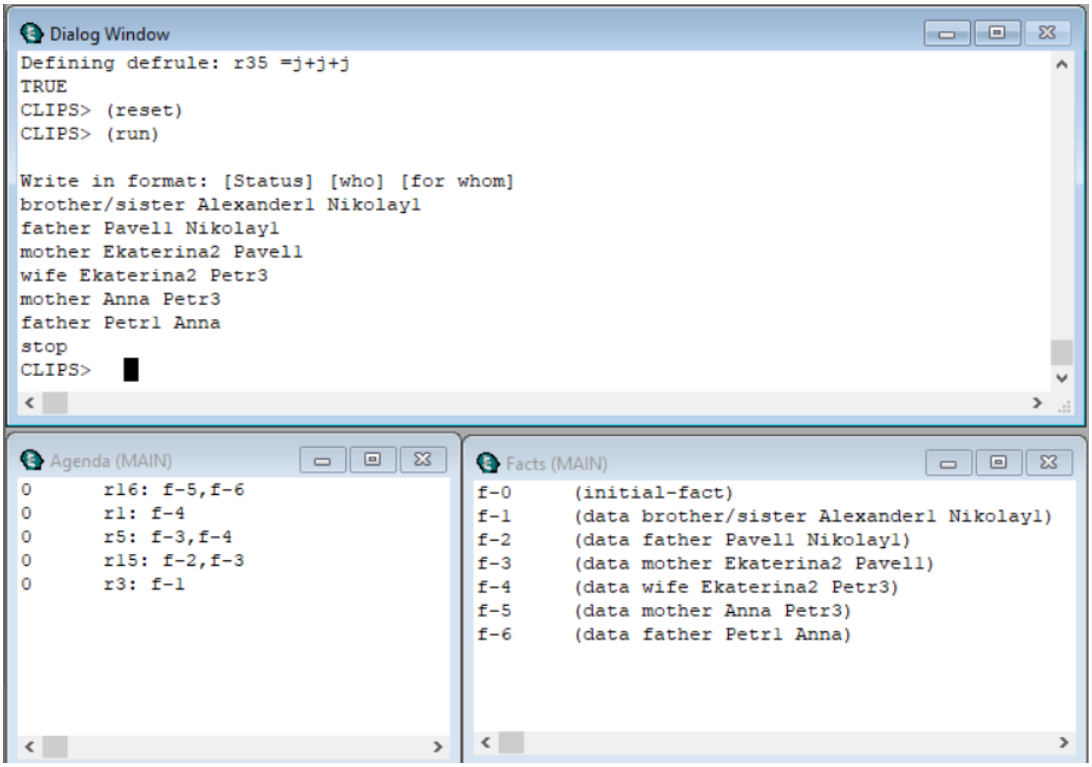
Если необходимо записать двоюродную дядю/тетю, то uncle/aunt x y 2. Троюродная дядя/тетя uncle/aunt x y 3.

Т.е. последнее число в факте обозначает степень родства.

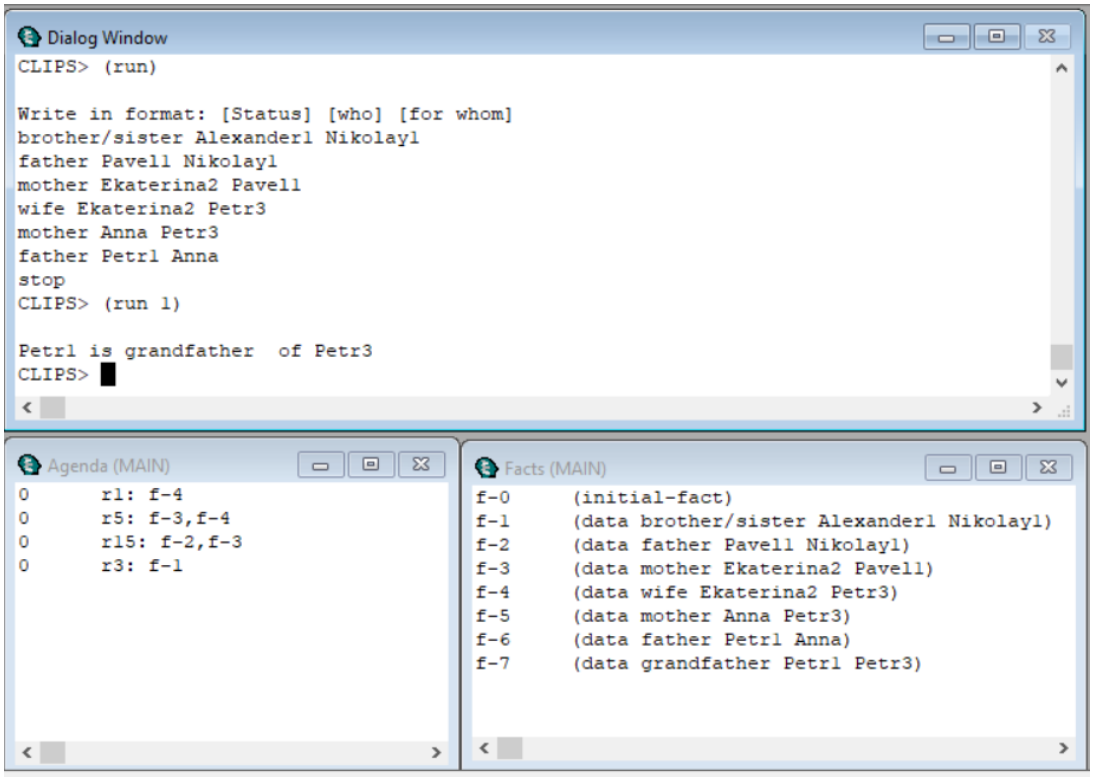
Для окончания ввода фактов необходимо ввести «stop». С помощью (run) запустить МЛВ.

Пошаговый режим

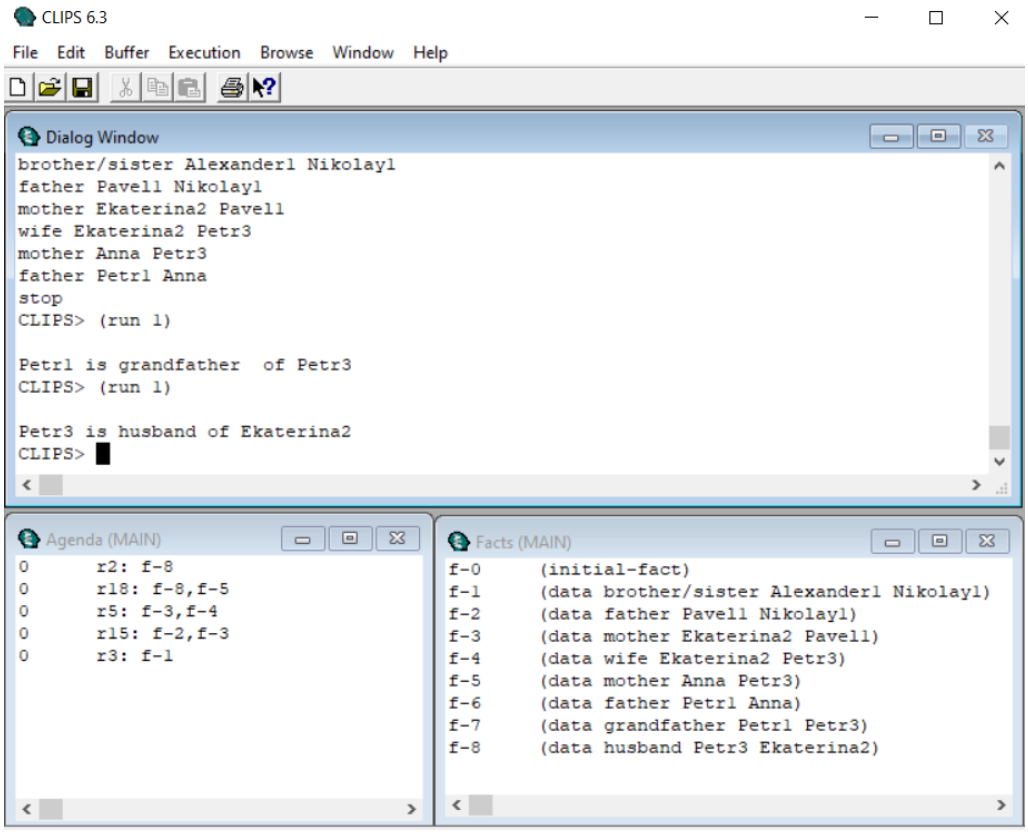
1. Ввод исходных фактов



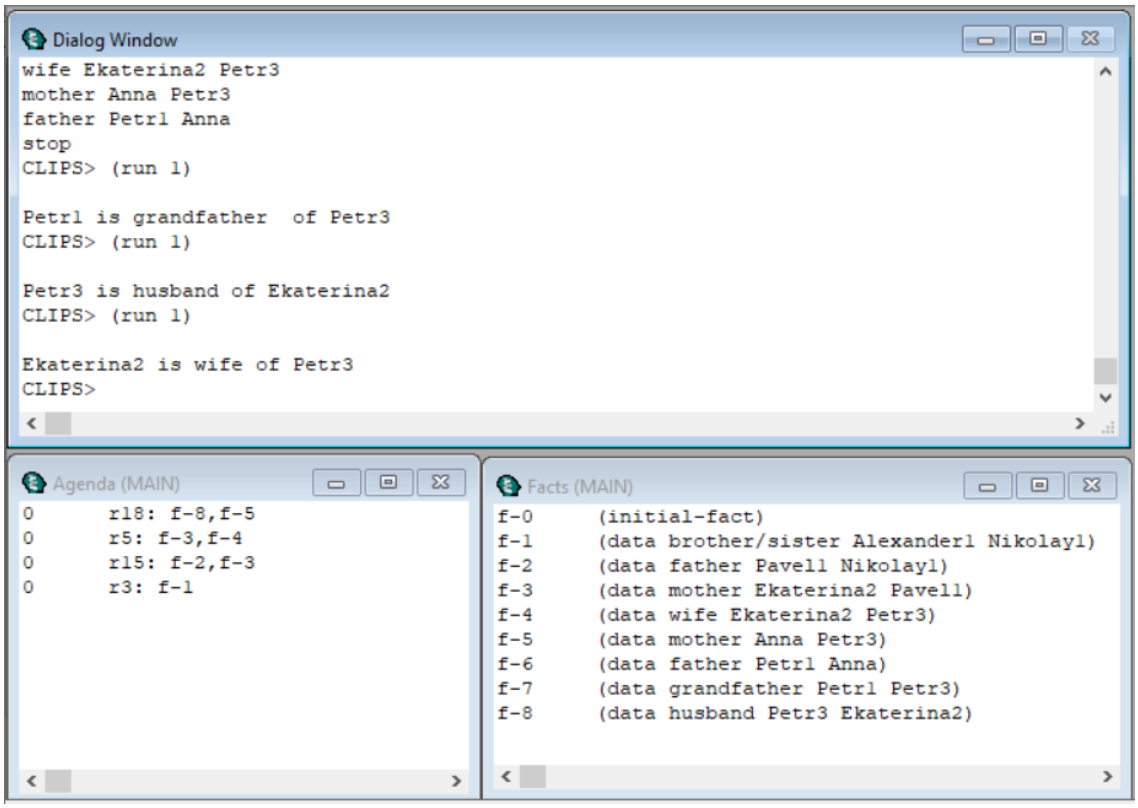
2. Сработало правило 16. Добавился новый факт, «Дед Петр1 Петр3»



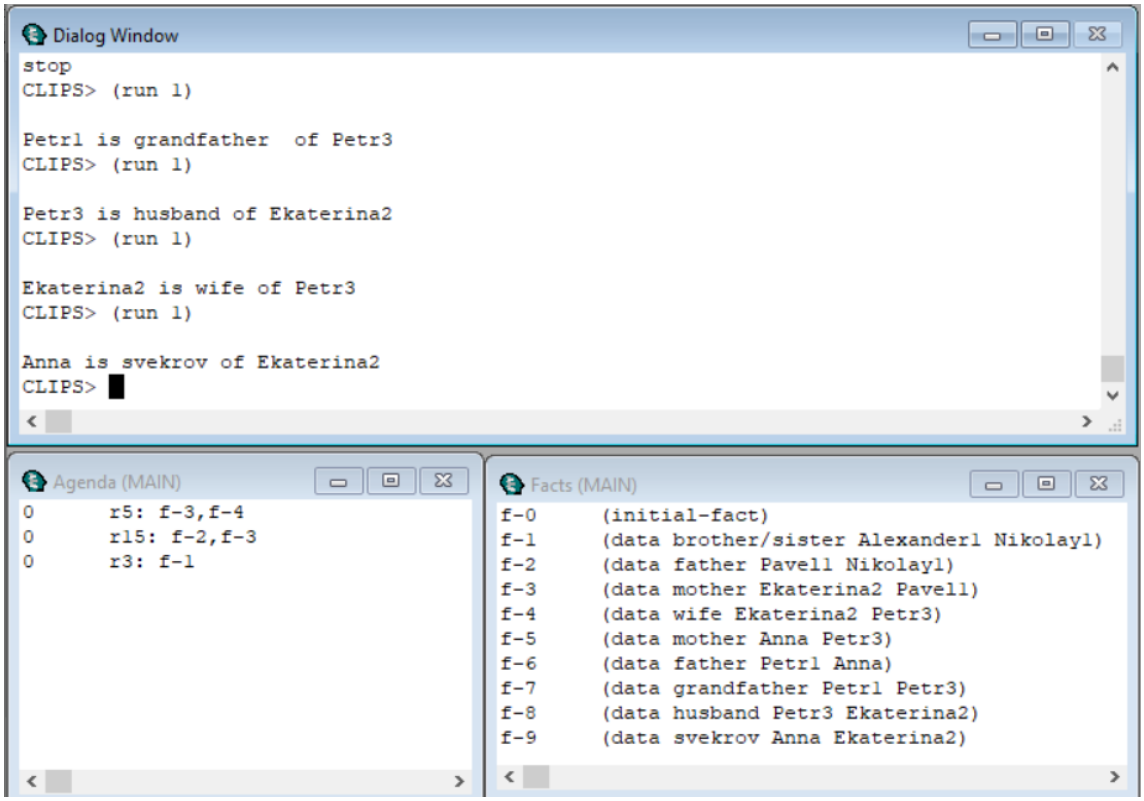
3. Сработало правило 1. Добавился новый факт, «Муж Петр3 Екатерина2»



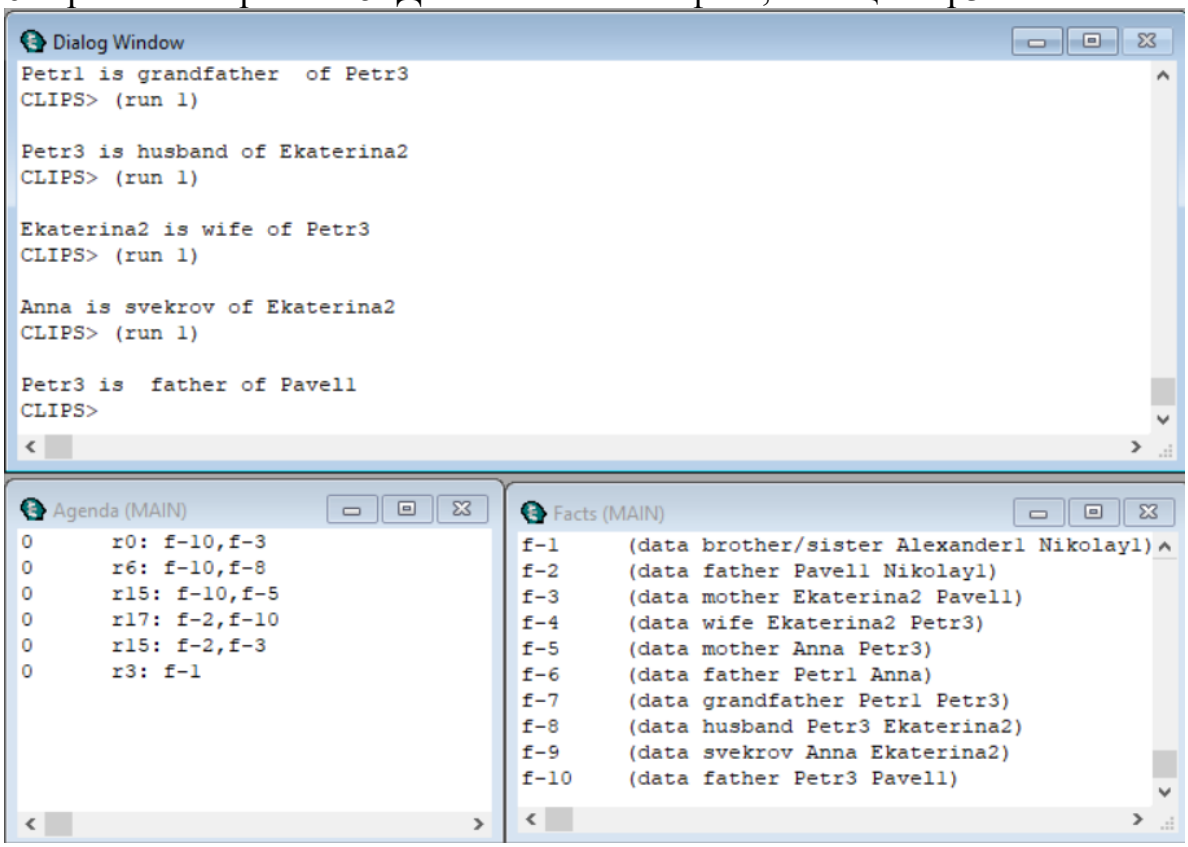
4. Сработало правило 2. Новый факт «Жена Екатерина2 Петр3» уже был выведен ранее – не добавляется



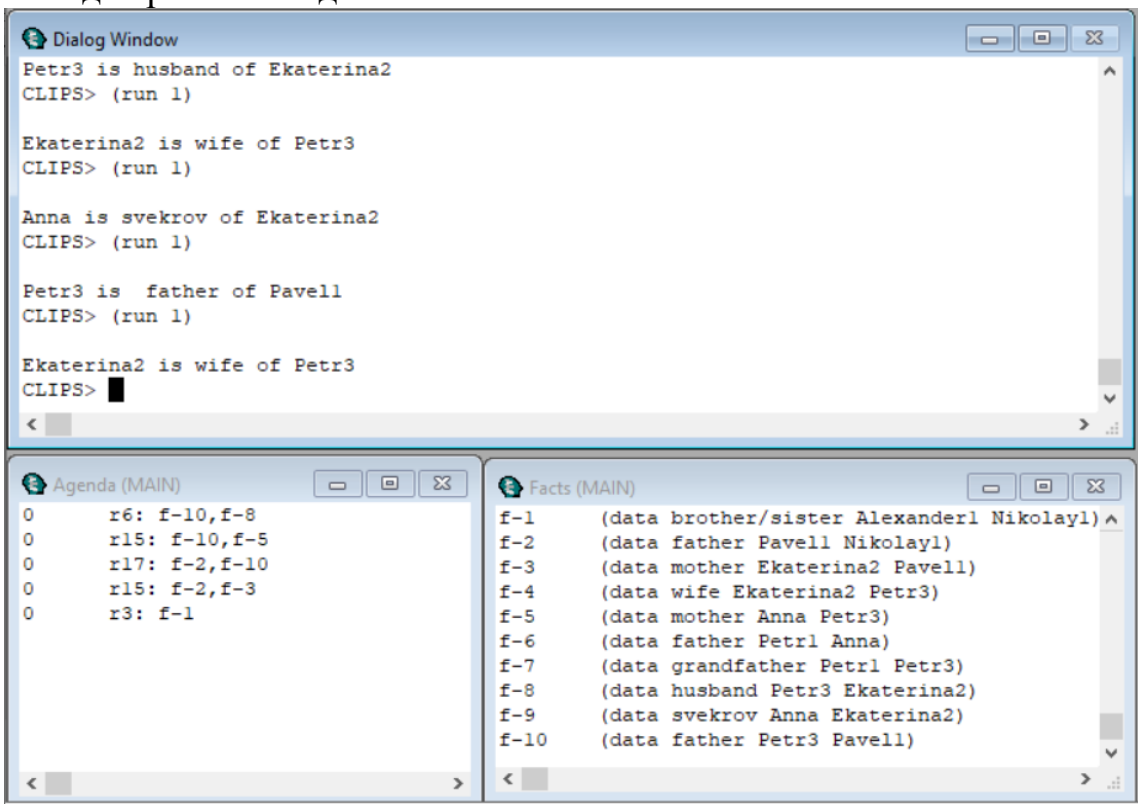
5. Сработало правило 18. Добавился новый факт, «Свекровь Анна Екатерина2»



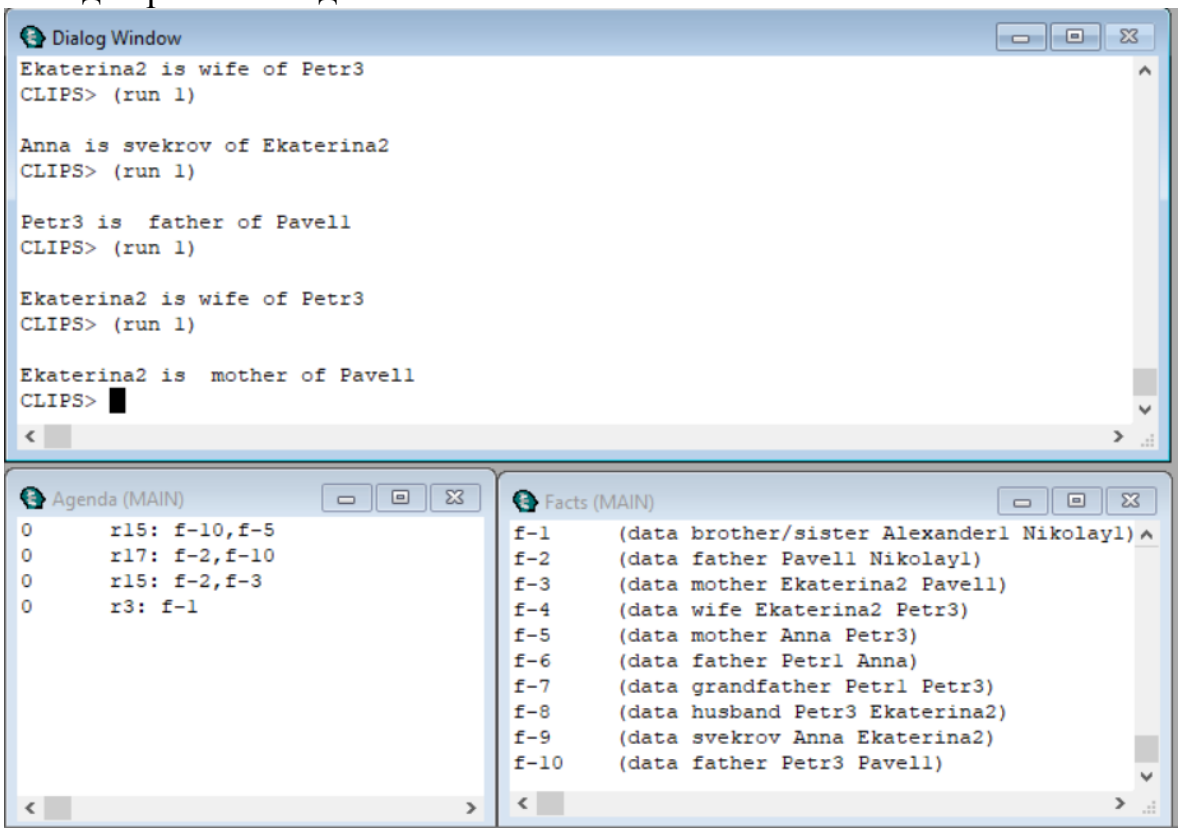
6. Сработало правило 5. Добавился новый факт, «Отец Петр3 Павел1»



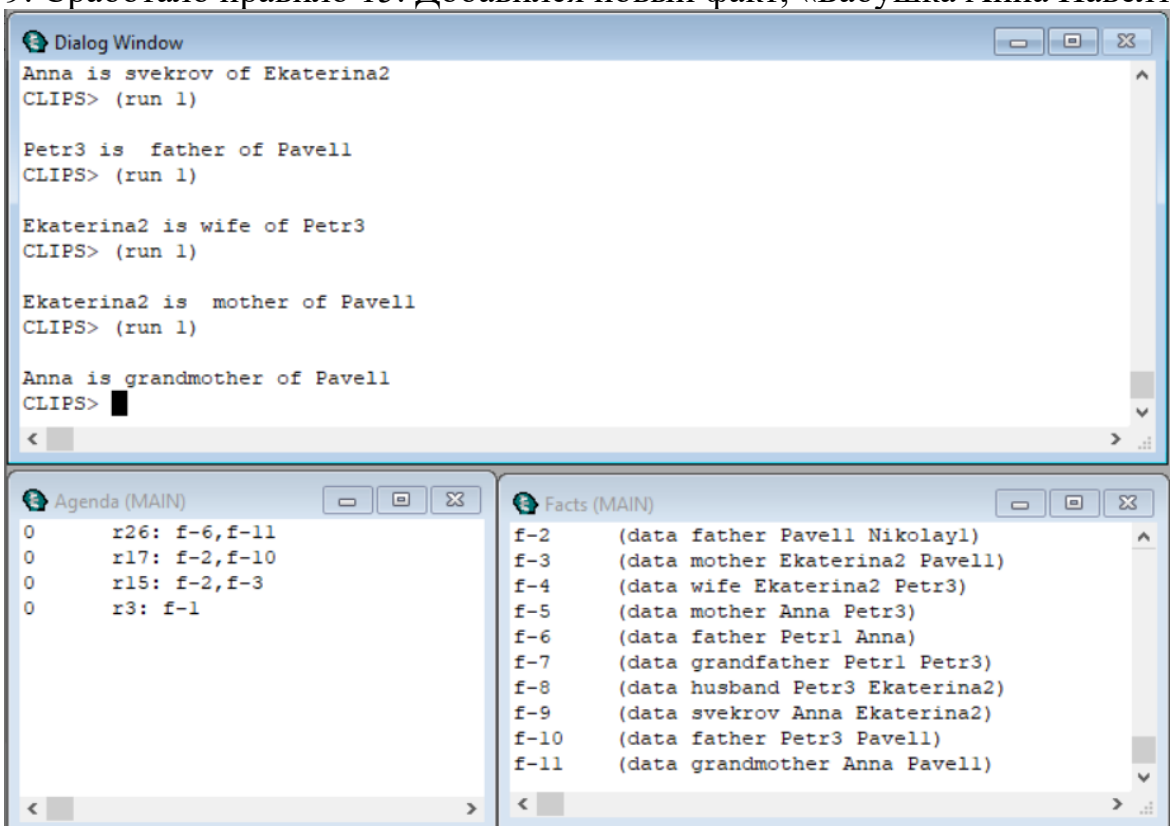
7. Сработало правило 0. Новый факт «Жена Екатерина2 Петр3» уже был выведен ранее – не добавляется



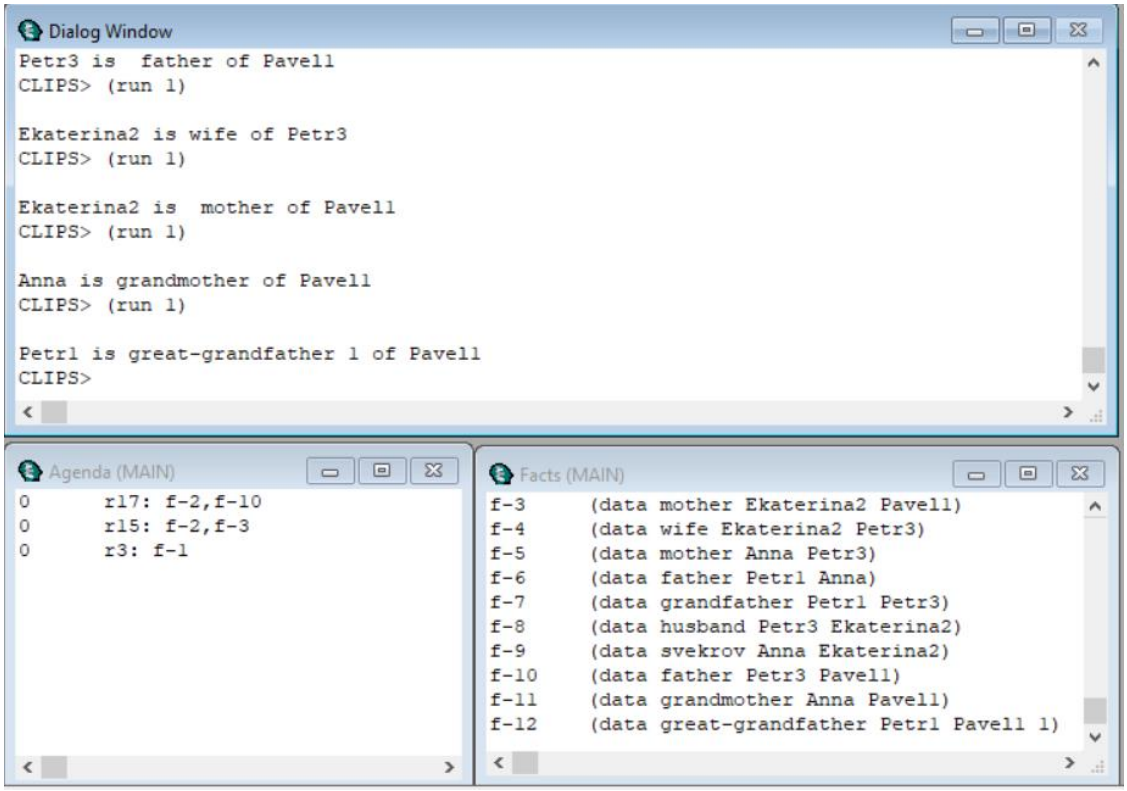
8. Сработало правило 6. Новый факт «Мать Екатерина2 Павел3» уже был выведен ранее – не добавляется



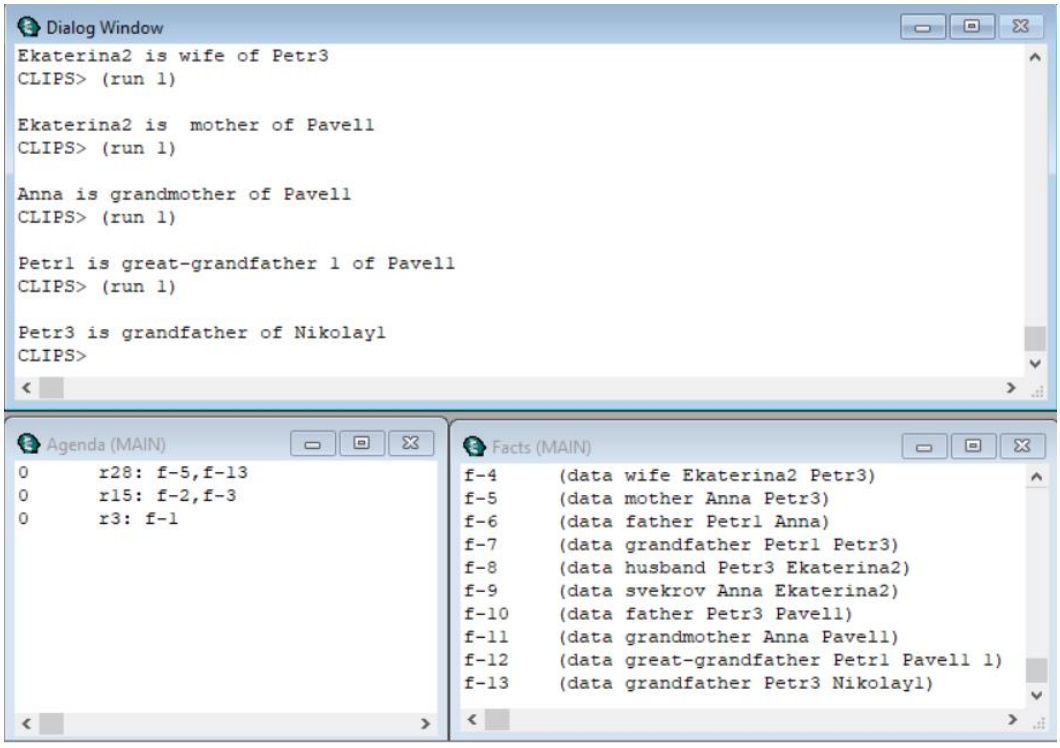
9. Сработало правило 15. Добавился новый факт, «Бабушка Анна Павел1»



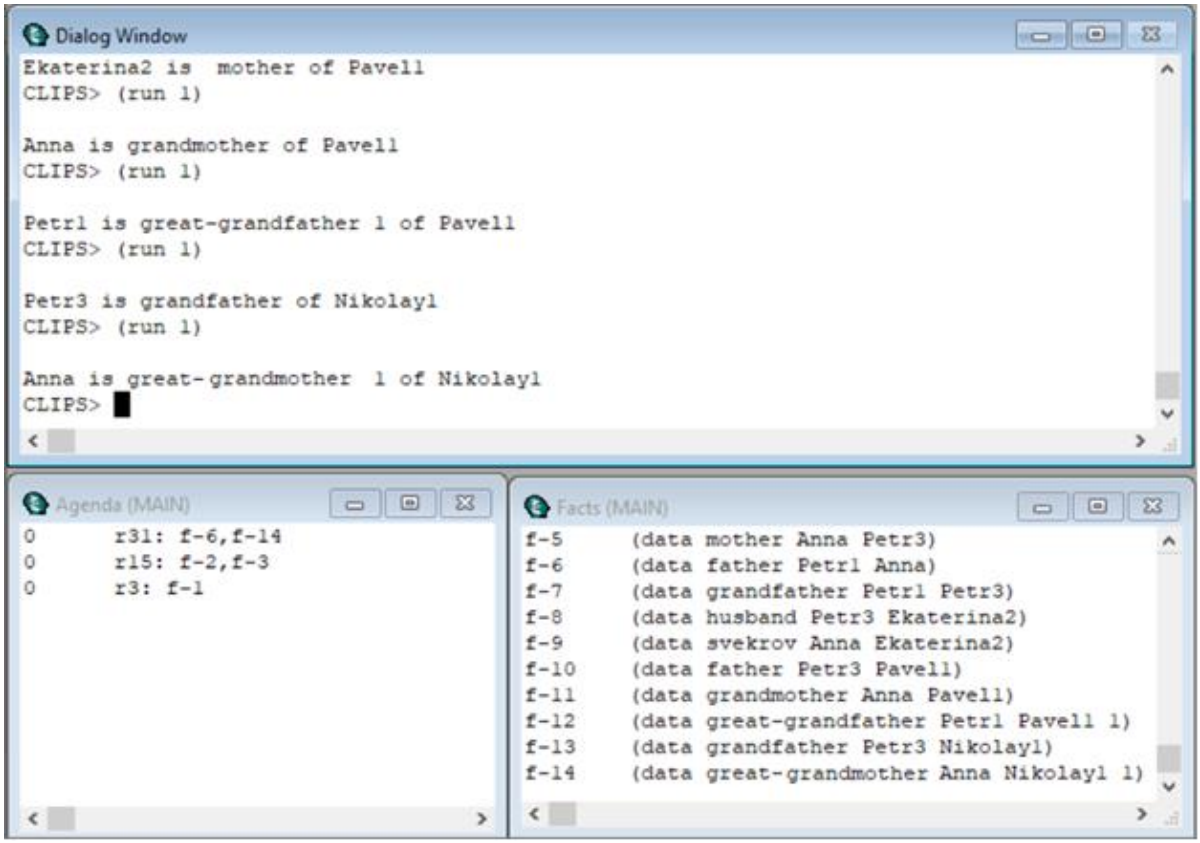
10. Сработало правило 26. Добавился новый факт, «Прадедушка Петр1 Павел1 1»



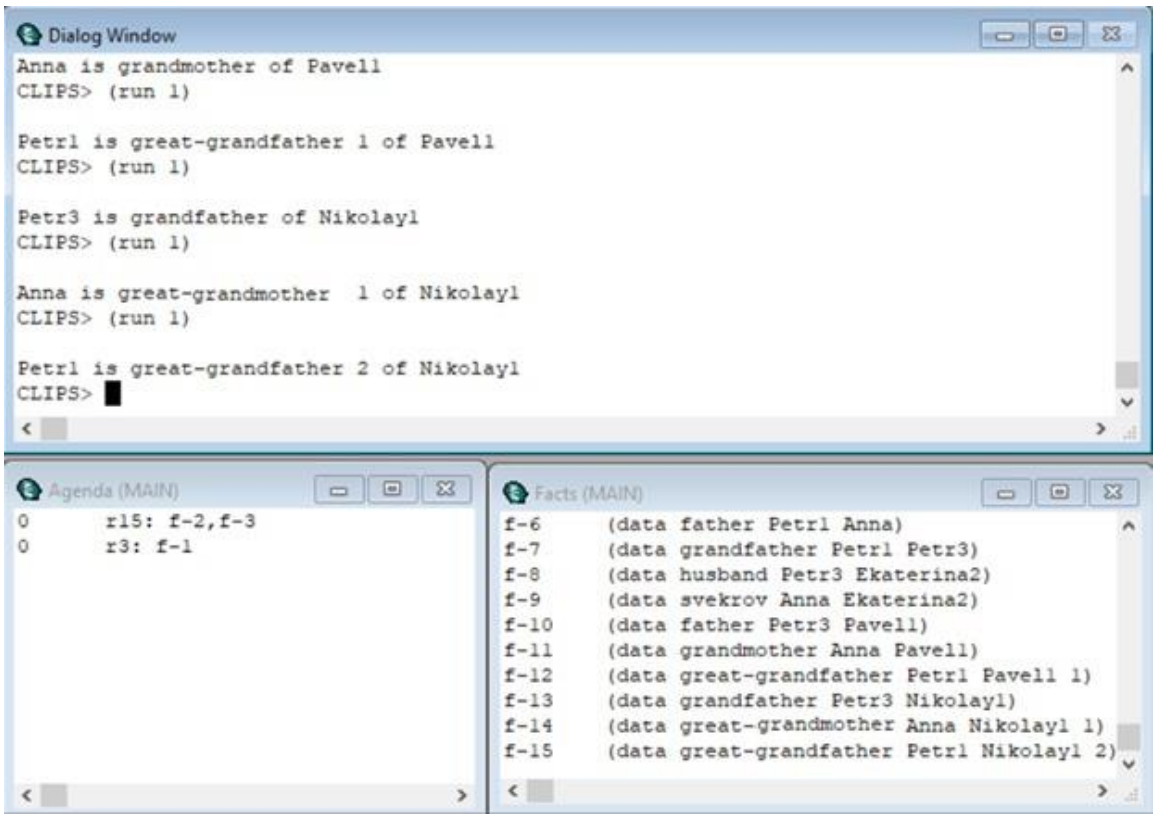
11. Сработало правило 17. Добавился новый факт, «Дедушка Петр3 Николай1»



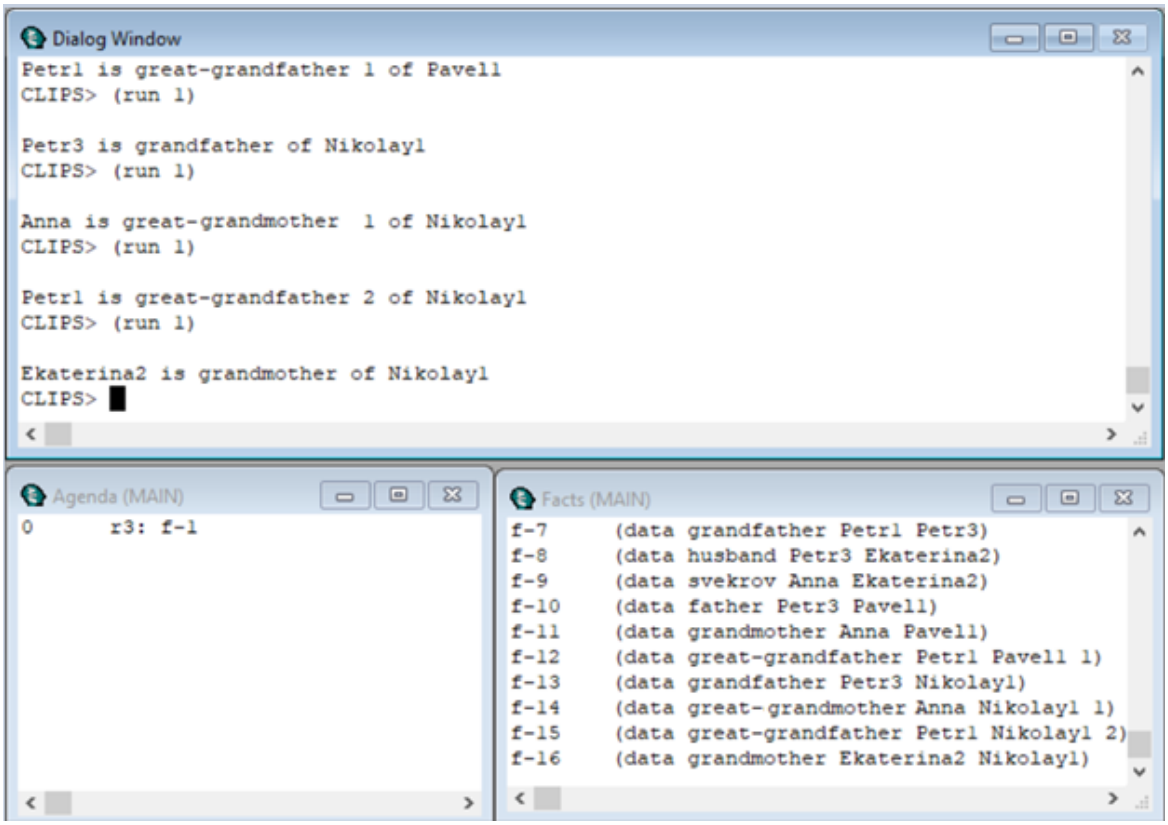
12. Сработало правило 28. Добавился новый факт, «Прабабушка Анна Николай1 1»



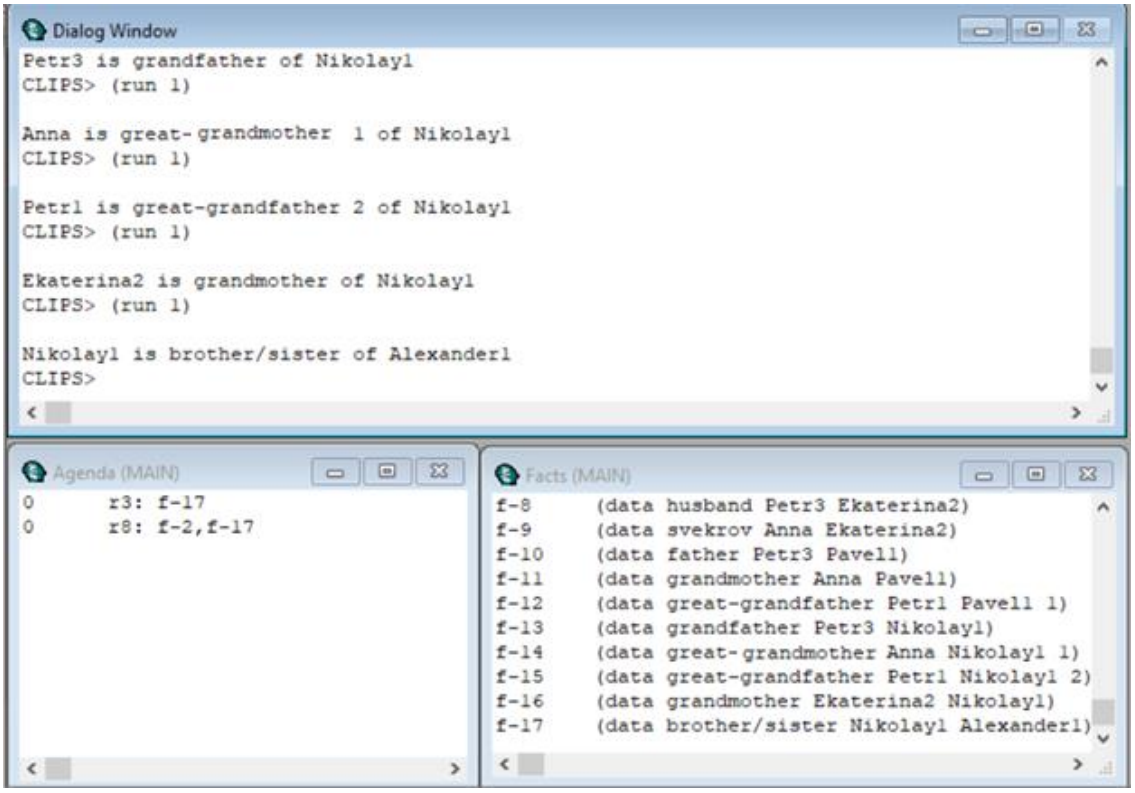
13. Сработало правило 31. Добавился новый факт, «Прадедушка Петр1 Николай1 2»



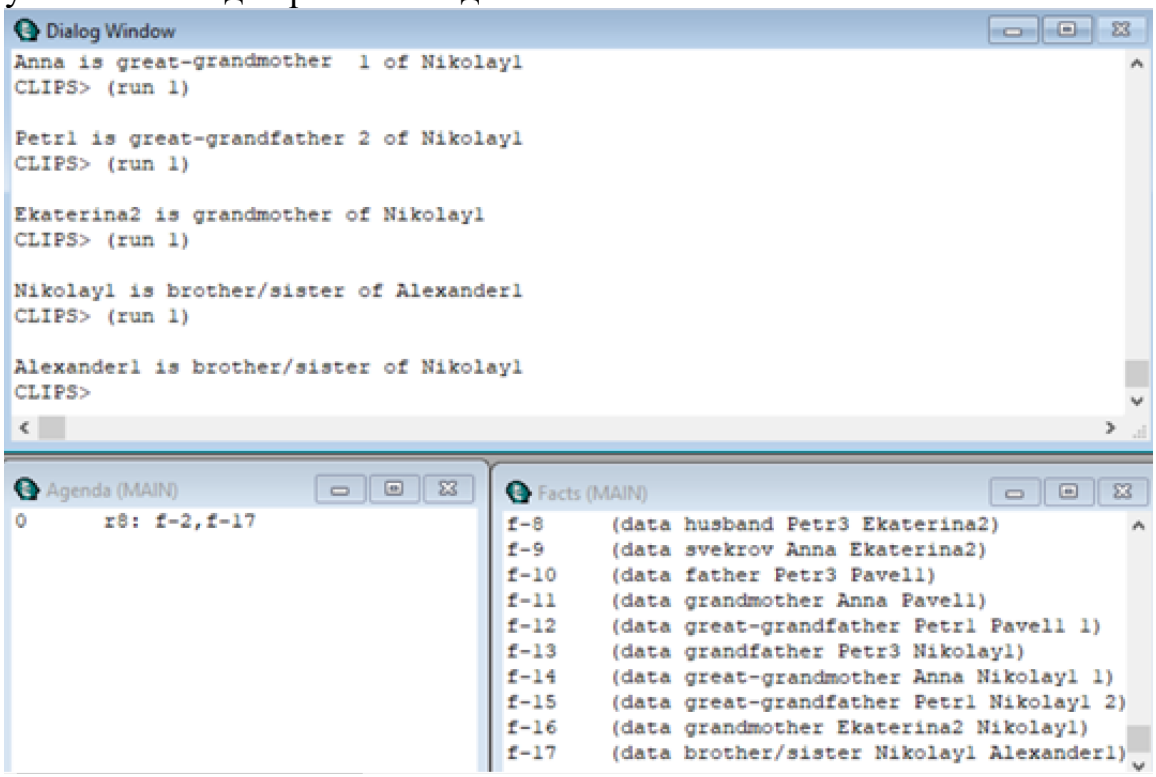
14. Сработало правило 15. Добавился новый факт, «Бабушка Екатерина Николай1»



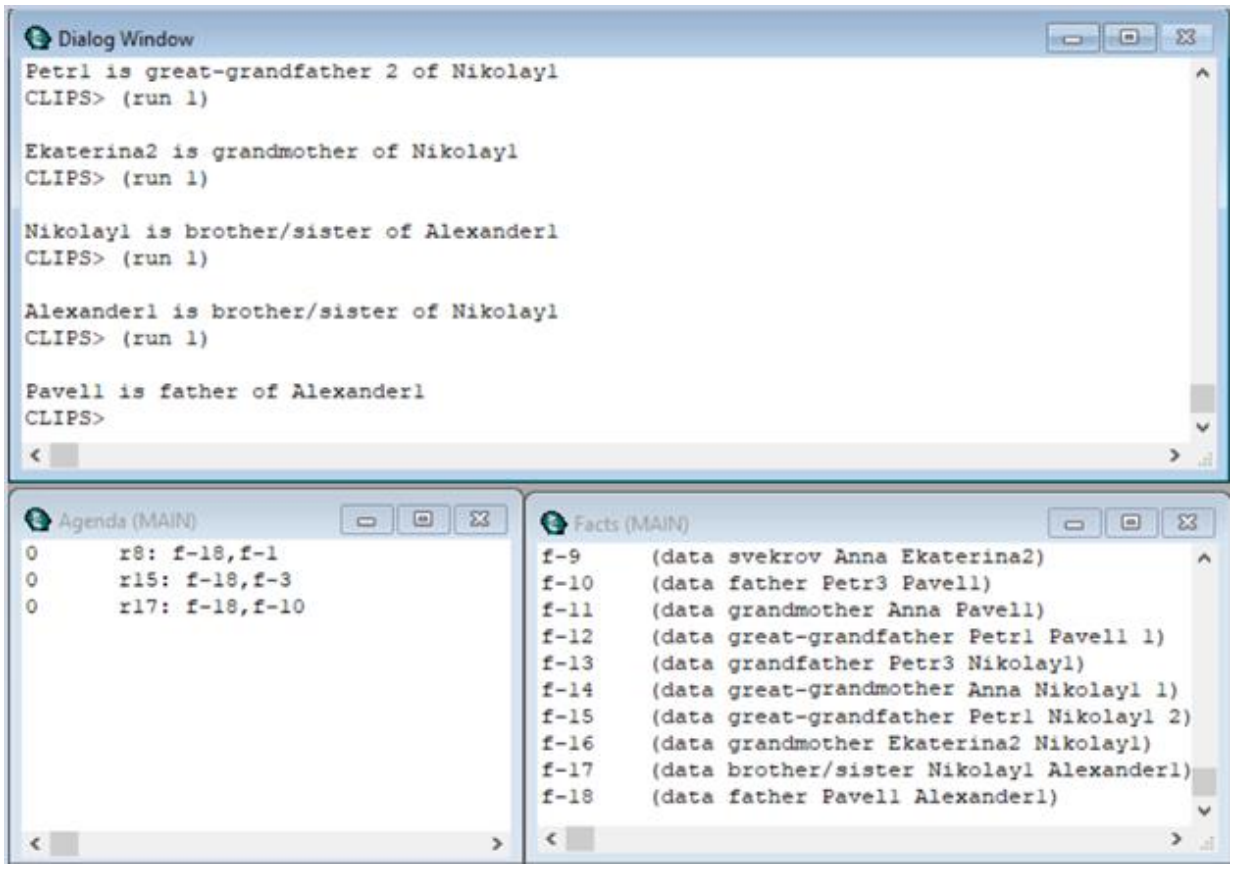
15. Сработало правило 3. Добавился новый факт, «Брат/сестра Николай1 Александр1»



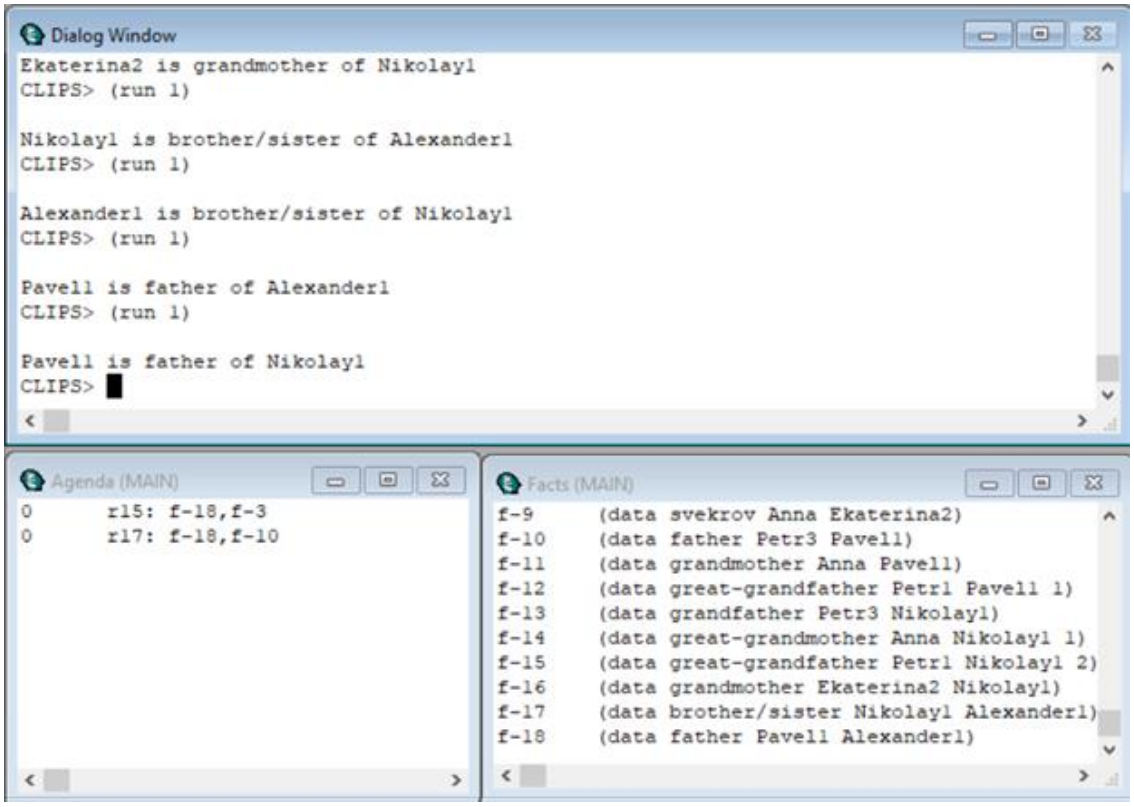
16. Сработало правило 3. Новый факт «Брат/сестра Александр1 Николай1» уже был выведен ранее – не добавляется



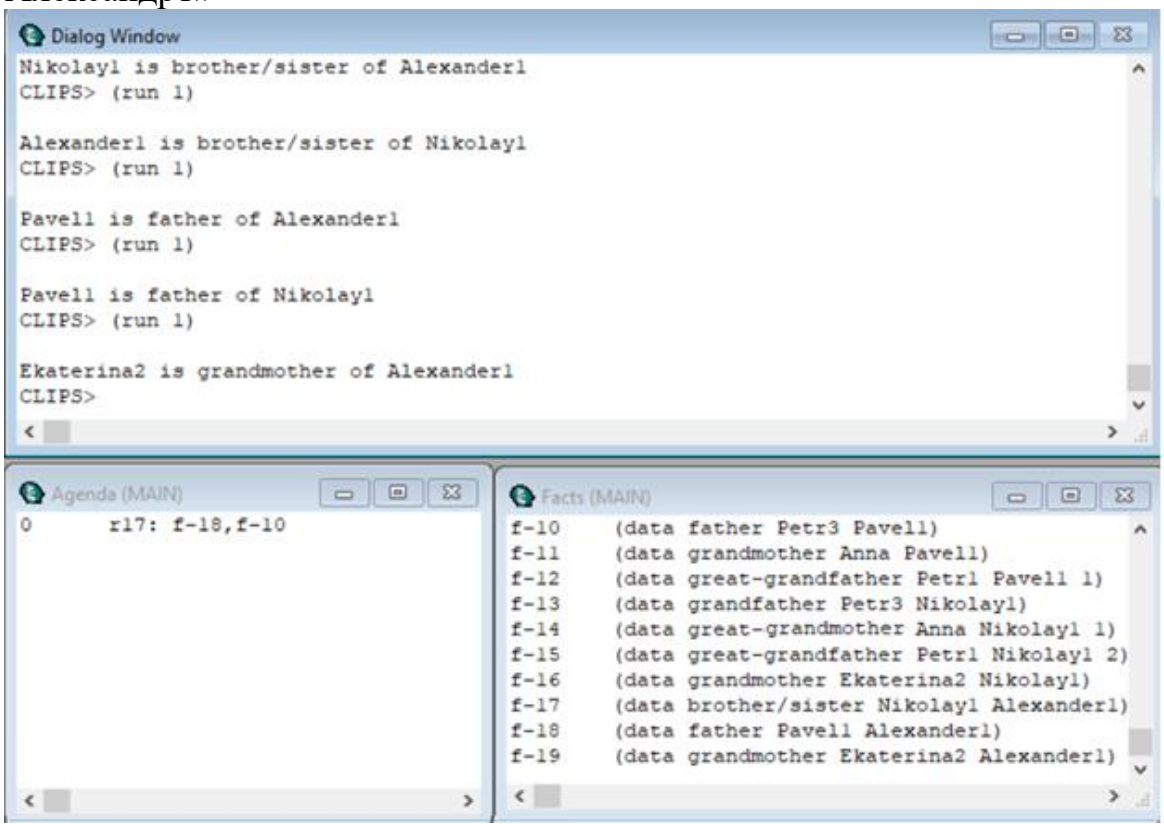
17. Сработало правило 8. Добавился новый факт, «Отец Павел1 Александр1»



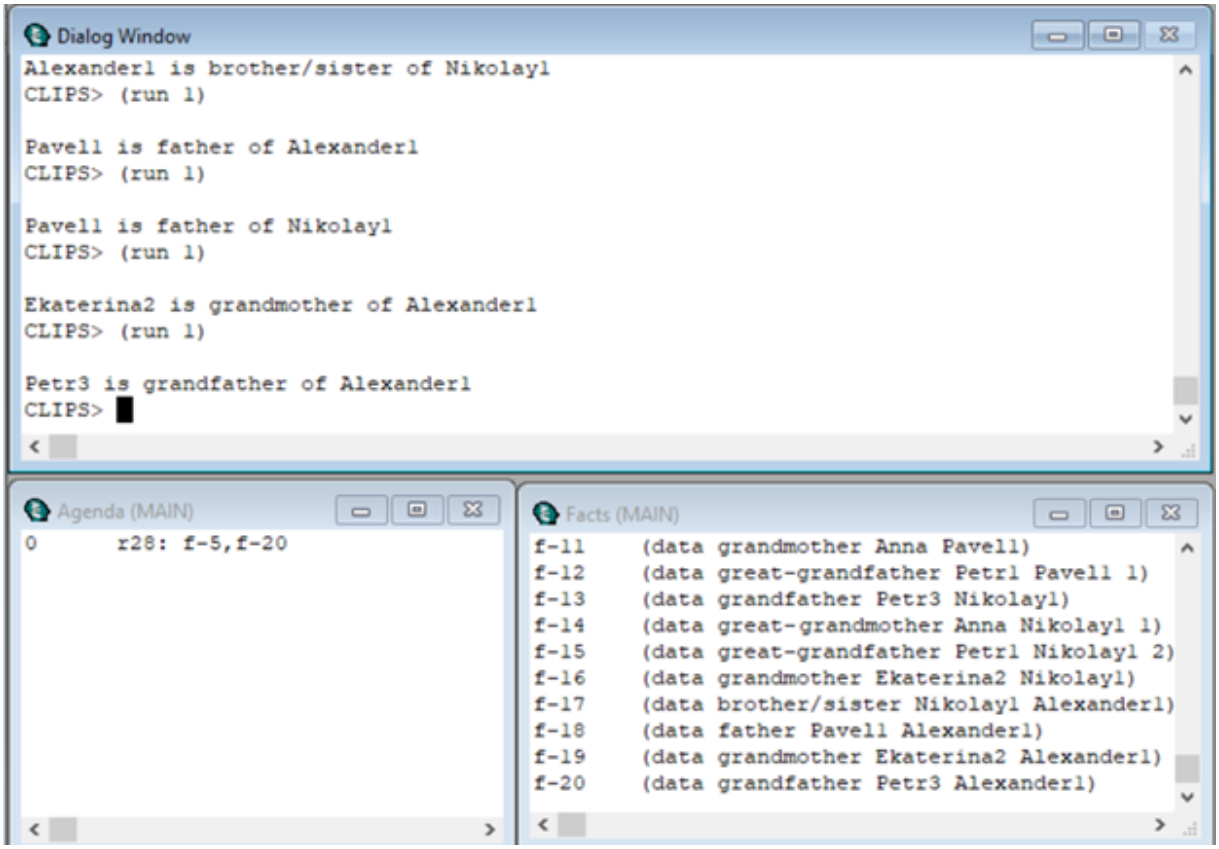
18. Сработало правило 8. Новый факт «Отец Павел1 Николай1» уже был выведен ранее – не добавляется



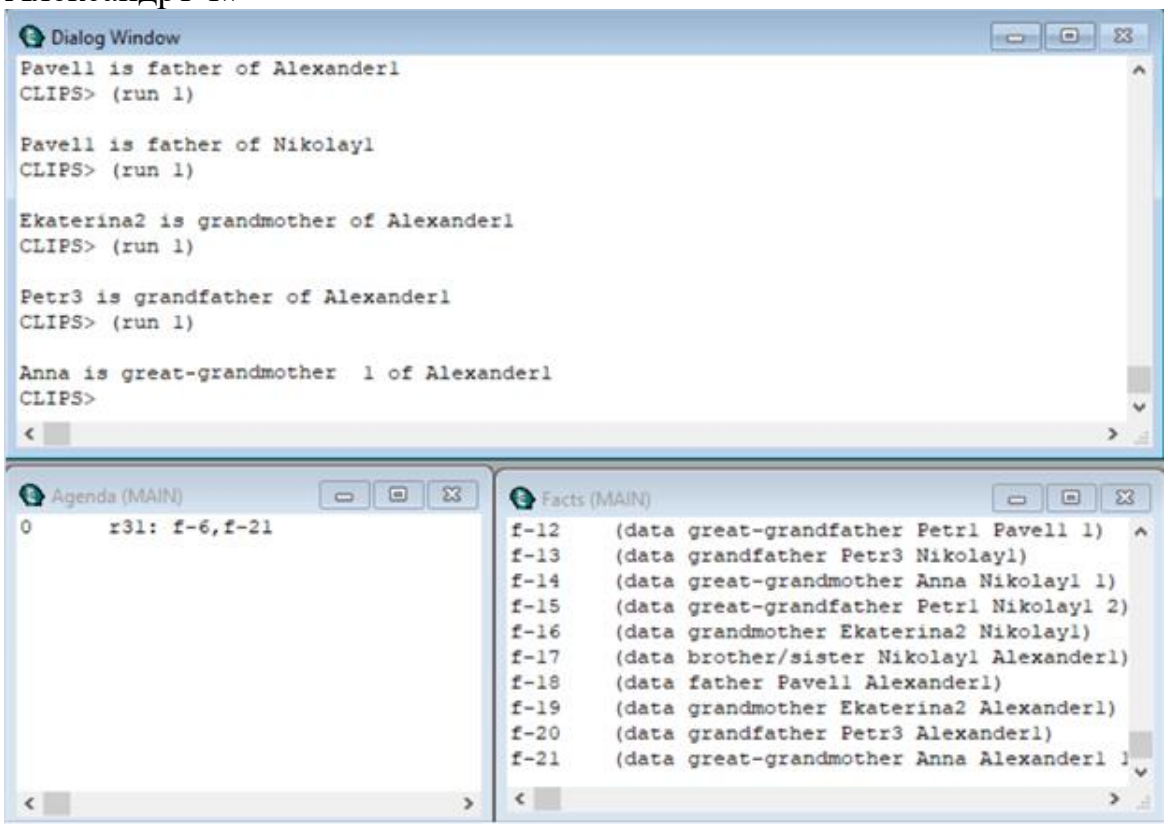
19. Сработало правило 15. Добавился новый факт, «Бабушка Екатирина2 Александр1»



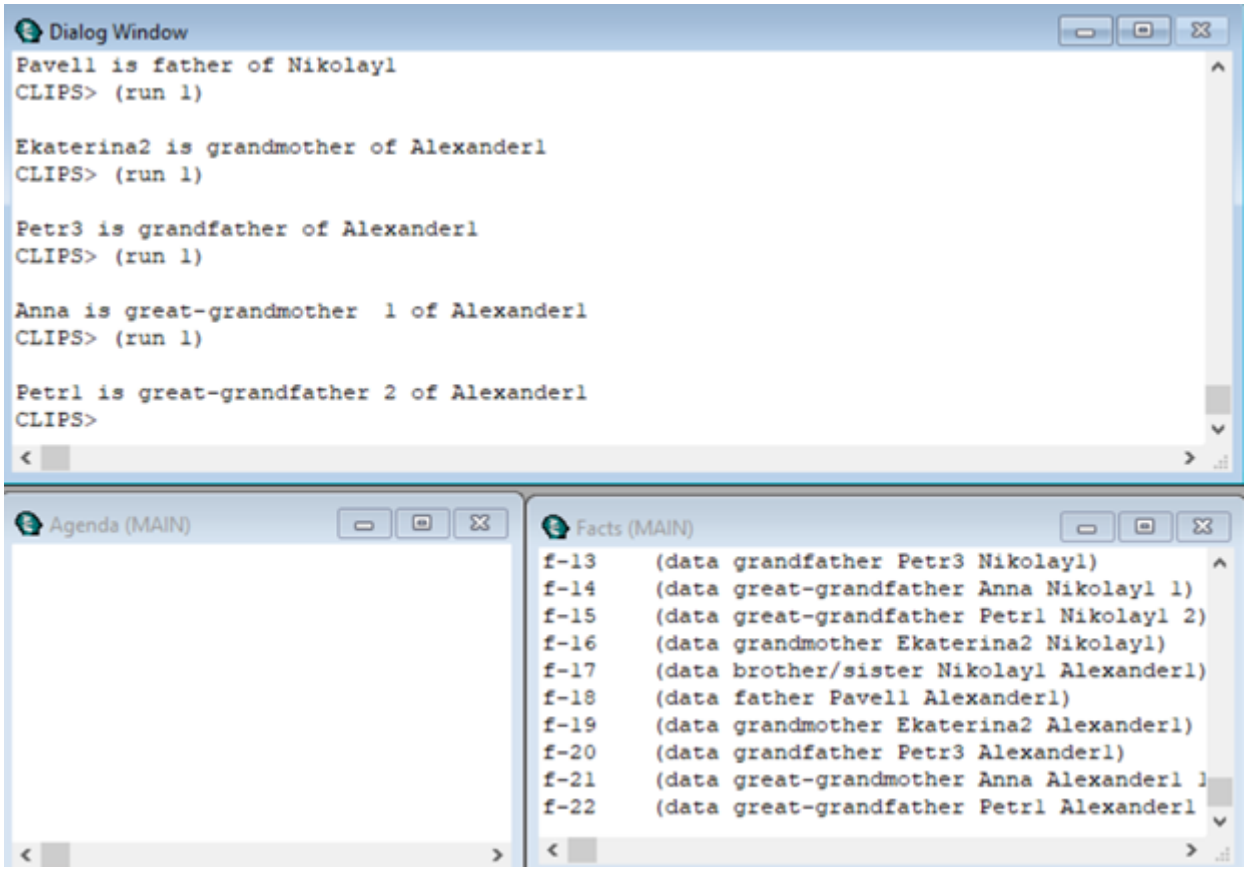
20. Сработало правило 17. Добавился новый факт, «Дедушка Петр3 Александр1»



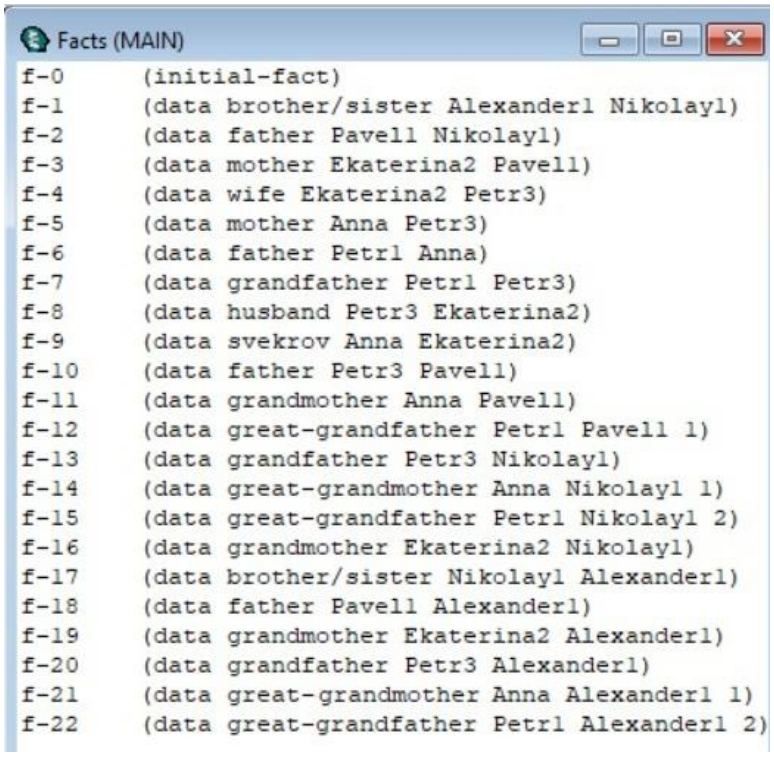
21. Сработало правило 28. Добавился новый факт, «Прабабушка Анна Александр1 1»



22. Сработало правило 31. Добавился новый факт, «Прадед Петр1Александр1 1»

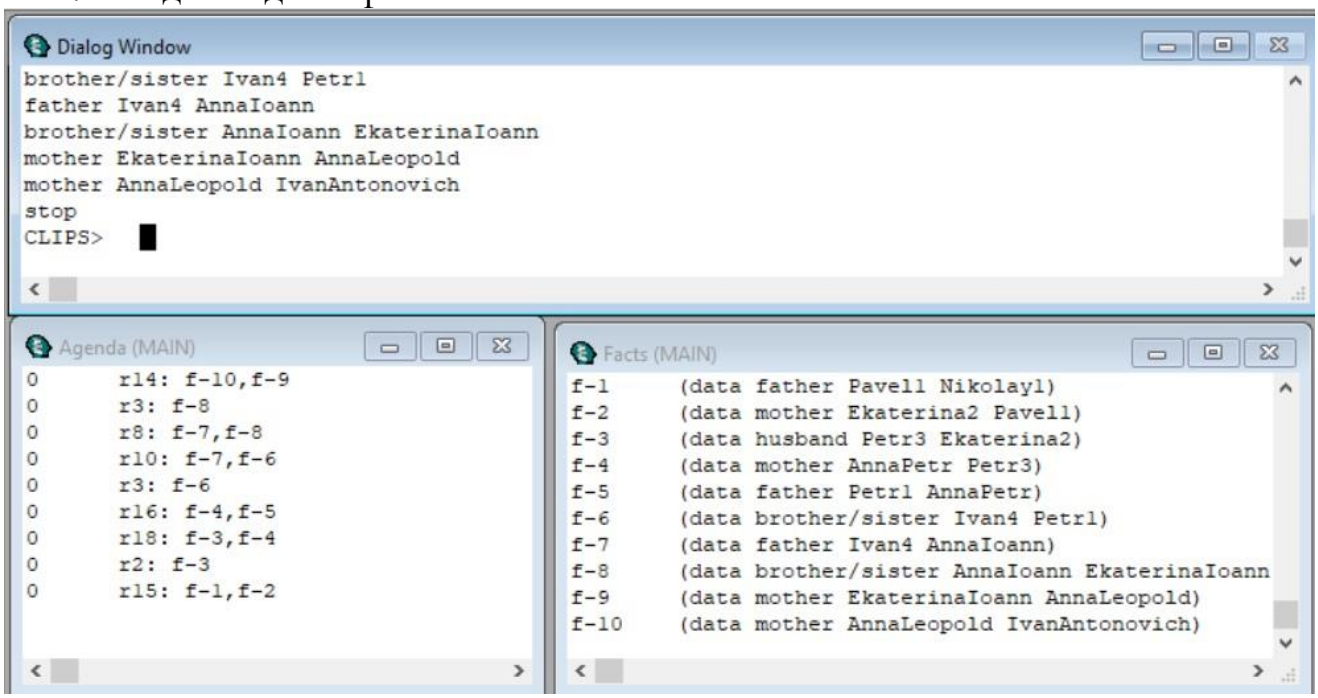


Список всех фактов

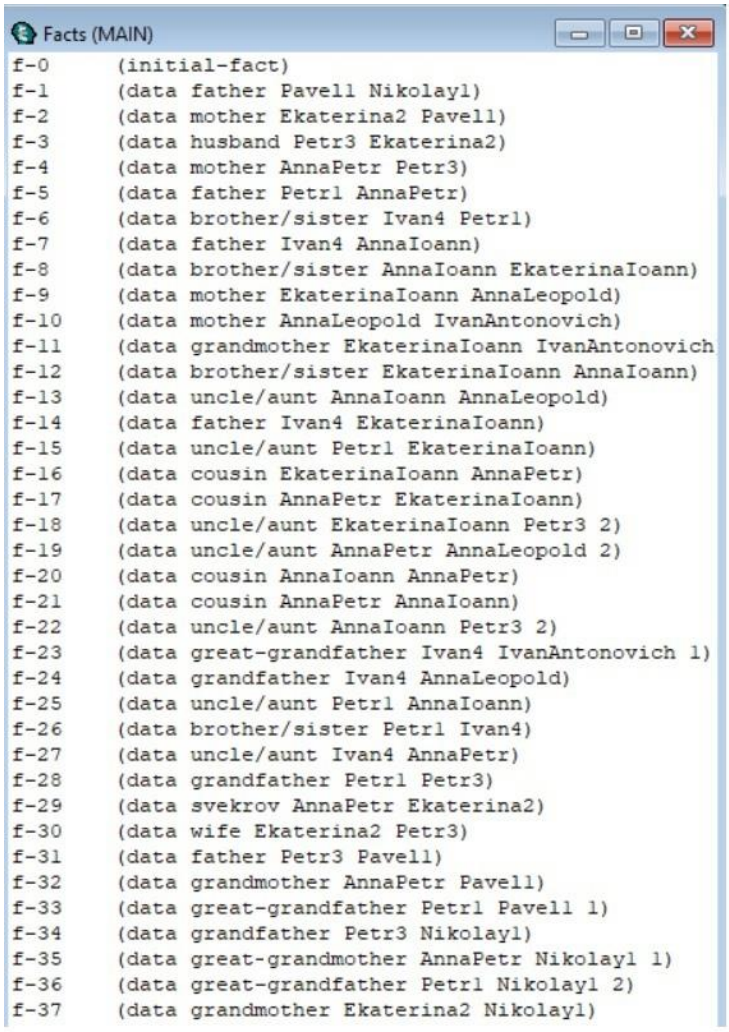


Без пошагового режима

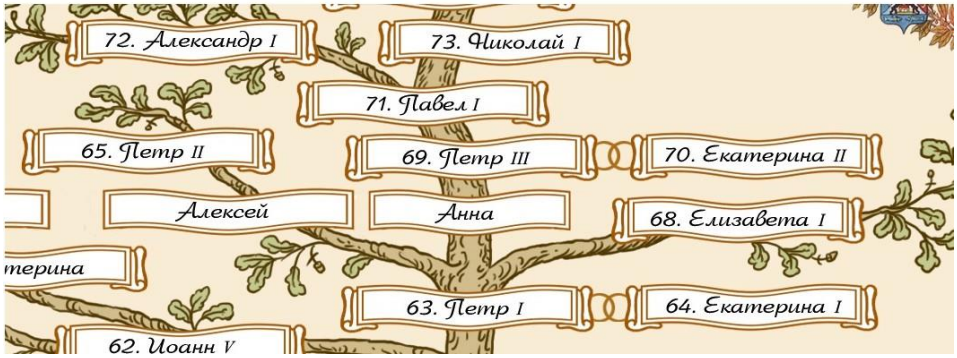
1. Ввод исходных фактов



2. Результат



Генеалогическое древо Романовых



Вывод

В результате выполнения была разработана экспертная система, которая устанавливает тип отношений между членами семьи (кто кому в семье приходится).