Продукционная модель знаний в среде CLIPS

Факты и правила

Состав продукционной модели системы CLIPS

CLIPS использует продукционную модель представления знаний и поэтому содержит три основных элемента:

1. базу фактов (fact base),
2. базу знаний (rule base),
3. Механизм логического вывода.

База фактов представляет исходное описание задачи. База правил содержит операторы, которые преобразуют состояния проблемы, приводя его к решению – целевому состоянию.

Механизм логического вывода

Механизм логического вывода CLIPS сопоставляет факты из базы фактов и правила из базы правил и выясняет, какие из правил можно активизировать. Это выполняется циклически, причем каждый цикл (так называемый продукционный цикл или цикл распознавания действия) состоит из трех основных фаз:

- сопоставление фактов и правил;
- выбор правила, подлежащего активизации;
- выполнение действий, предписанных активным («зажженным») правилом.

Что такое факты?

Факты – это одна из основных форм представления информации в системе CLIPS.

Каждый *факт* представляет фрагмент информации, который был помещен в текущий список фактов, называемый fact-list.

Факт представляет собой основную единицу данных, используемую правилами.

Размещение фактов в списке фактов

Количество фактов в списке и объем информации, который может быть сохранен в факте, ограничивается только размером памяти компьютера.

Если при добавлении нового факта к списку обнаруживается, что он полностью совпадает с одним из уже включенных в список фактов, то эта операция игнорируется (хотя такое поведение можно изменить).

Индекс (адрес) факта

Факт может описываться *индексом* или *адресом.* Всякий раз, когда факт добавляется (изменяется), ему присваивается уникальный целочисленный индекс.

Индексы фактов начинаются с нуля и для каждого нового или измененного факта увеличиваются на единицу.

Каждый раз после выполнения команд reset и clear выделение индексов начинается с нуля. Факт также может задаваться при помощи адреса.

Адрес факта может быть получен путем сохранения возвращаемого значения команд, которые возвращают в качестве результата адрес факта (таких как assert, modify и duplicate), или путем связывания переменной с адресом факта в левой части правила.

Идентификатор факта

Идентификатор факта - это короткая запись для отображения факта на экране. Она состоит из символа f и записанного через тире индекса факта. Например, запись f-10 служит для обозначения факта с индексом 10. Существует два формата представления фактов: позиционный и непозиционный.

Позиционные факты

Позиционные факты состоят из выражения символьного типа, за которым следует последовательность (возможно, пустая) из полей, разделенных пробелами. Вся запись заключается в скобки. Обычно первое поле определяет "отношение", которое применяется к оставшимся полям.

Примеры позиционных фактов

(Ivanova is student)

(Jonh has the son)

(Sidorov Ivan is an ingeneer)

(Weather is warm)

Требования к использованию позиционных фактов

Поля в позиционных фактах могут быть любого простого типа (за исключением первого поля, которое всегда должно быть типа symbol), на порядок полей также не накладывается никаких ограничений.

Следующие символьные выражения зарезервированы и не должны использоваться как первое поле любого факта (позиционного или нет).

Непозиционные факты

Для того чтобы обратиться к информации, содержащейся в позиционном факте, пользователь должен знать не только какие данные содержатся в факте, но и то, в каком поле они хранятся.

Непозиционные (шаблонные) факты дают возможность пользователю абстрагироваться от структуры факта, задавая имена каждому из полей факта.

Задание шаблонов

Для задания шаблона, который затем может использоваться при доступе к полям по именам, используется конструкция deftemplate. Эта конструкция подобна структуре или записи в языках программирования C и Паскаль.

Конструкция deftemplate позволяет наряду с определением *именованных полей,* или *слотов,* вводить имя шаблона. В отличие от позиционных фактов слоты шаблонного факта могут быть ограничены по типу, значению, числовому диапазону. Кроме того, для любого слота можно определить значения по умолчанию.

Слоты в шаблонных фактах

Слот состоит из открывающейся скобки, за которой следует имя слота, полей (могут отсутствовать) и закрывающейся скобки. Заметим, что слоты не могут использоваться в позиционных фактах, так же как позиционные поля не могут использоваться в шаблонных фактах.

Конструкция deftemplate

Общая структура конструкции def template такова:

(deftemplate <name>)

(slot-1)

(slot-2)

...

(slot-N)

)

Создание шаблонов

Создание шаблонов фактов и правил осуществляется с помощью конструктора **deftemplate:**

(deftemplate <имя объекта>

(slot <имя слота 1>)

(slot <имя слота 2>)

……

(slot <имя слота n>)

)

Как и все конструкторы в языке CLIPS, конструктор шаблонов не возвращает никакого значения.

Создадим два шаблона для фактов person и parent.

Шаблон person

```
(deftemplate person
 (slot name)
(slot gender)
(slot age)
)
```

где name – имя,

gender – пол,

age – возраст.

Шаблон parent

```
(deftemplate tparent
(slot parentname)
(slot childname)
)
```

где parentname – имя родителя,

childname – имя ребенка.

Конструктор deffacts

На основе шаблона факта можно создать список фактов с помощью конструктора фактов deffacts:

(deffacts <имя списка фактов>

(<имя_шаблона_факта 1 > (<имя_слота 1> <значение_слота 1>)

(<имя_слота 2> <значение_слота 2>)

…….

(<имя_слота n> <значение_слота n>))

……..

(<имя_шаблона_факта m> (<имя_слота 1> <значение_слота 1>)

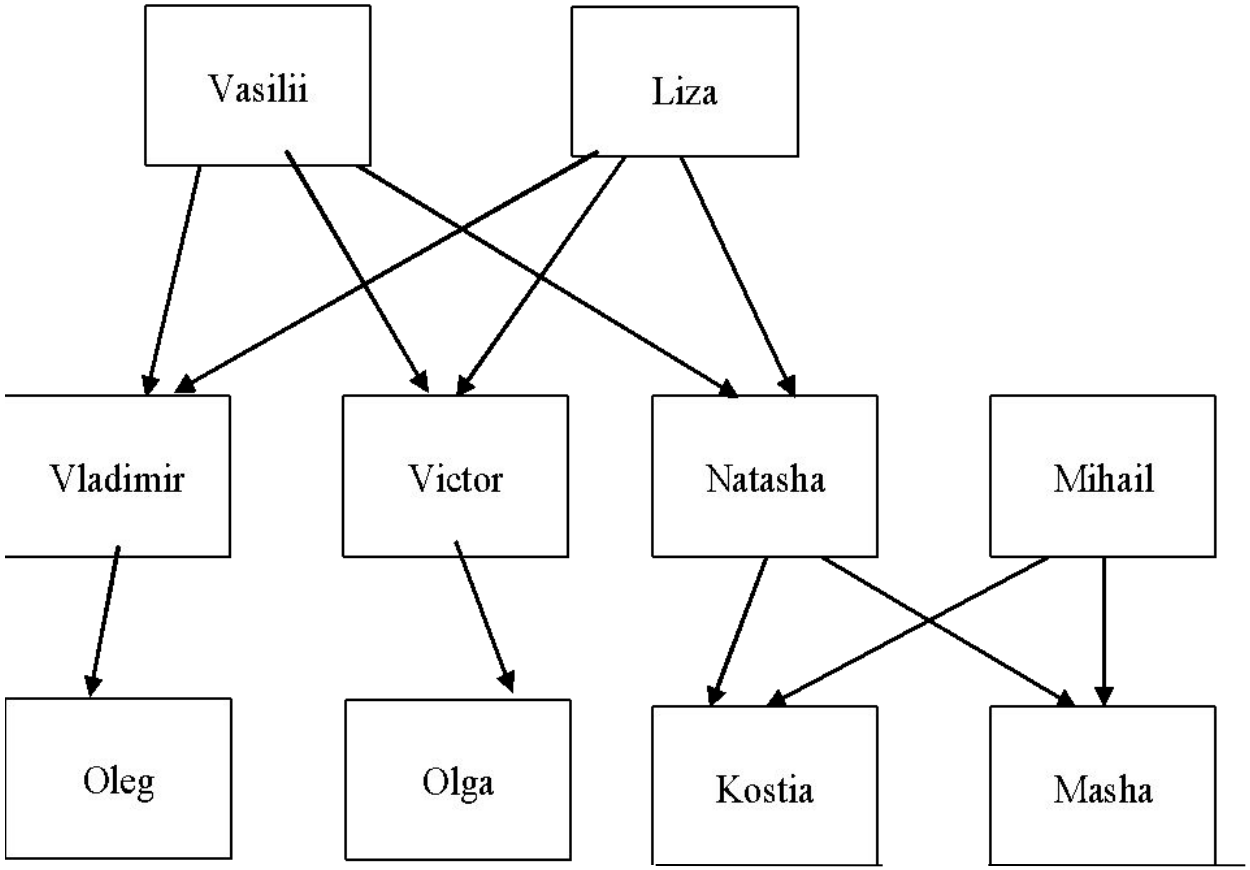
(<имя_слота 2> <значение_слота 2>)

…….

(<имя_слота n> <значение_слота n>))

)

С помощью ранее созданных шаблонов создадим списки фактов для базы данных «Родственники»



Список фактов fperson

```
(deffacts fperson
(person (name Vasilii) (gender male)   (age 65))
(person (name Liza) (gender female)   (age 62))
(person (name Vladimir) (gender male)   (age 42))
(person (name Victor) (gender male)   (age 40))
(person (name Natasha) (gender female)   (age 37))
(person (name Oleg) (gender male)   (age 18))
(person (name Olga) (gender female)   (age 14))
(person (name Mihail) (gender male)   (age 41))
(person (name Kostia) (gender male)   (age 10))
(person (name Masha) (gender male)   (age 12))
)
```

Список фактов fparent

```
(deffacts fparent
(tparent (parentname Vasilii) (childname Vladimir))
(tparent (parentname Vasilii) (childname Victor))
(tparent (parentname Vasilii) (childname Natasha))
(tparent (parentname Liza) (childname Vladimir))
(tparent (parentname Liza) (childname Victor))
(tparent (parentname Liza) (childname Natasha))
(tparent (parentname Vladimir) (childname Oleg))
(tparent (parentname Victor) (childname Olga))
(tparent (parentname Natasha) (childname Kostia))
(tparent (parentname Natasha) (childname Masha))
(tparent (parentname Mihail) (childname Kostia))
(tparent (parentname Mihail) (childname Masha))
)
```

Правила

Правила определяют набор действий, которые будут выполнены для данной ситуации. Разработчик ЭС определяет набор правил, которые в совокупности используются для решения проблемы. Правило состоит из предусловия и следствия.

Предусловие правила – это левая сторона (LHS) правила. Предусловие правила – набор условий (состояний или условных элементов), которые должны выполниться для того, чтобы правило можно было активизировать.

Следствие правила – это правая сторона (RHS) правила. Следствие правила – это набор действий, которые будут выполнены в случае, если правило активировано. Если активировано более чем одно правило, механизм логического вывода использует стратегию разрешения противоречий.

Определение правила

Для определения правил используется конструкция defrule:

```
(defrule <имя правила> «необязательный комментарий"
(предпосылка_l)
(предпосылка_2)
.
.
.
(предпосылка _N)
=>
(действие_l)
(действие_2)
.
.
.
(действие_M)
)
```

Пример определения правила

1. Создадим шаблон нового факта, описывающего отношение “X является отцом Y“:

```
(deftemplate Father
(slot name1)
(slot name2)
)
```

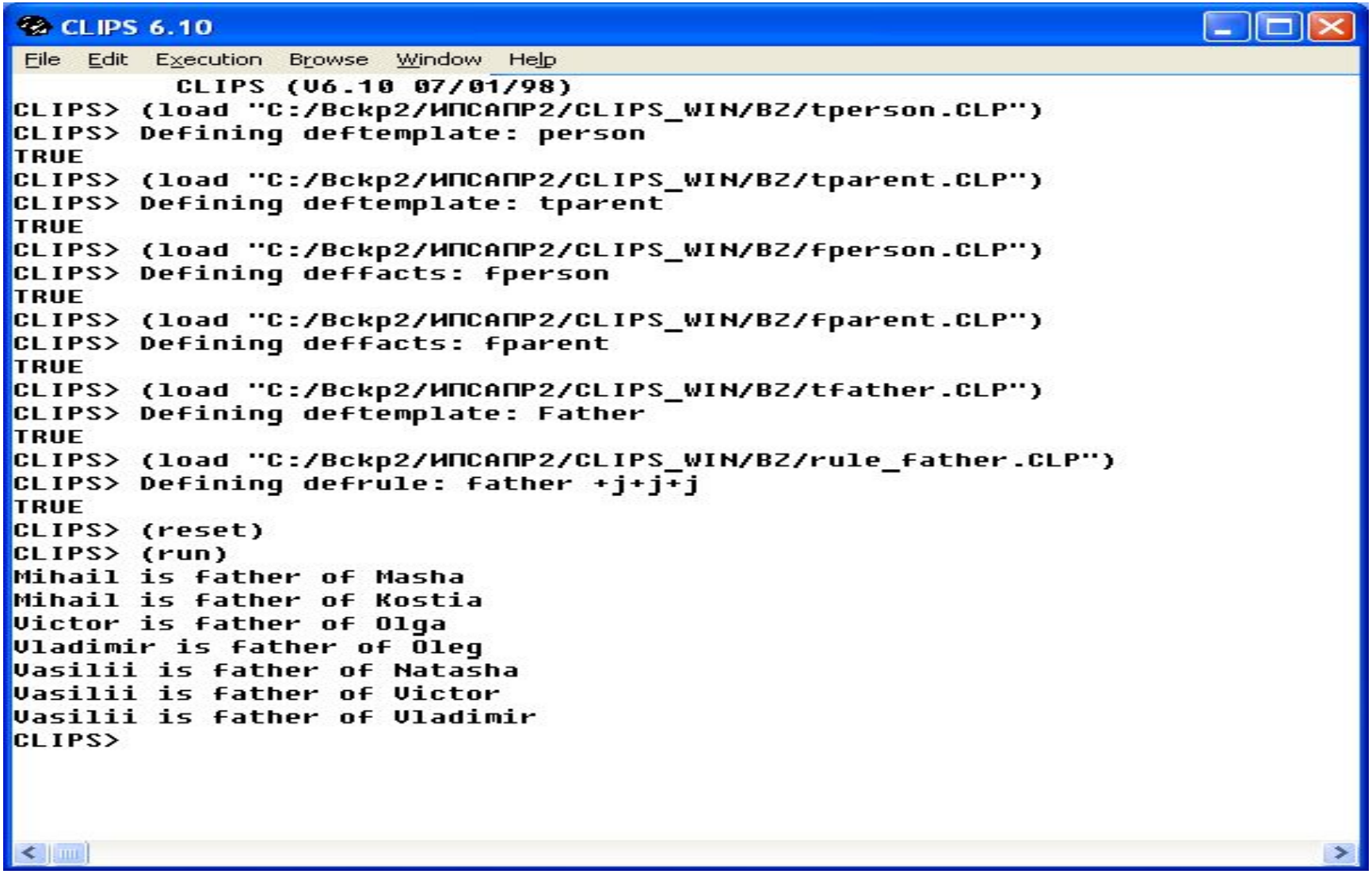
2. Определим правило, которое создает и добавляет факты по шаблону Father:

```
(defrule father
(person (name ?x) (gender male))
(person (name ?y))
(tparent (parentname ?x) (childname ?y))
=>
(printout t ?x " is father of " ?y crlf)
(assert (Father (name1 ?x) (name2 ?y)))
)
```

Создание новых фактов с помощью правил

Для того, чтобы проверить работу правила, необходимо загрузить созданные шаблоны фактов и правила, загрузить списки фактов, затем в главном меню выбрать пункт Execution (Выполнение) и последовательно выполнить команды reset и run. Результаты будут выданы на экран.

Создание новых фактов с помощью правил

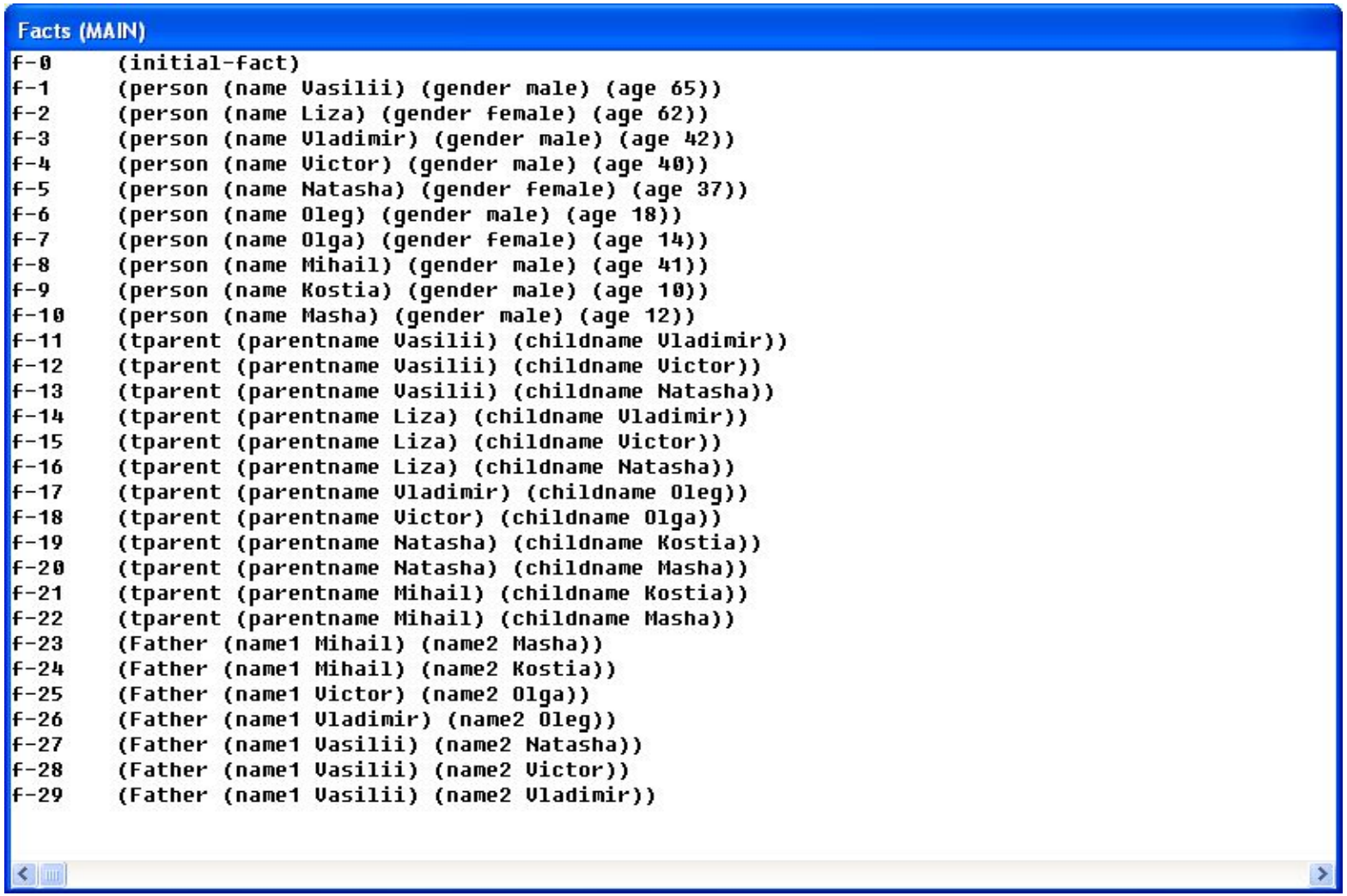


Просмотр новых фактов в окне фактов

Факты, созданные во время данного сеанса работы с системой CLIPS, можно просмотреть в окне фактов, вызвав в пункте Windows главного меню команду Facts Window. Кроме того созданные факты можно сохранить в текстовом файле с помощью команды

(save-facts "имя файла").

Окно фактов


Facts (MAIN)
f-0 (initial-fact)
f-1 (person (name Vasilii) (gender male) (age 65))
f-2 (person (name Liza) (gender female) (age 62))
f-3 (person (name Vladimir) (gender male) (age 42))
f-4 (person (name Victor) (gender male) (age 40))
f-5 (person (name Natasha) (gender female) (age 37))
f-6 (person (name Oleg) (gender male) (age 18))
f-7 (person (name Olga) (gender female) (age 14))
f-8 (person (name Mihail) (gender male) (age 41))
f-9 (person (name Kostia) (gender male) (age 10))
f-10 (person (name Masha) (gender male) (age 12))
f-11 (tparent (parentname Vasilii) (childname Vladimir))
f-12 (tparent (parentname Vasilii) (childname Victor))
f-13 (tparent (parentname Vasilii) (childname Natasha))
f-14 (tparent (parentname Liza) (childname Vladimir))
f-15 (tparent (parentname Liza) (childname Victor))
f-16 (tparent (parentname Liza) (childname Natasha))
f-17 (tparent (parentname Vladimir) (childname Oleg))
f-18 (tparent (parentname Victor) (childname Olga))
f-19 (tparent (parentname Natasha) (childname Kostia))
f-20 (tparent (parentname Natasha) (childname Masha))
f-21 (tparent (parentname Mihail) (childname Kostia))
f-22 (tparent (parentname Mihail) (childname Masha))
f-23 (Father (name1 Mihail) (name2 Masha))
f-24 (Father (name1 Mihail) (name2 Kostia))
f-25 (Father (name1 Victor) (name2 Olga))
f-26 (Father (name1 Vladimir) (name2 Oleg))
f-27 (Father (name1 Vasilii) (name2 Natasha))
f-28 (Father (name1 Vasilii) (name2 Victor))
f-29 (Father (name1 Vasilii) (name2 Vladimir))

Лабораторная работа № 5.

Задание.

Создайте шаблон и правило для отношений:

X является мамой Y,

X является братом Y,

X является сыном Y,

X является тётей Y.