

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет	Программной Инженерии и Компьютерной Техники
Направление подготовки (специальность)	
Дисциплина	Системы искусственного интеллекта

ЛАБОРАТОРНАЯ РАБОТА 1
ОТЧЕТ

Выполнил студент: Алхимовици Арсений (408138)
Группа: Р3210
Преподаватель: Болдырева Елена Александровна (157150)

г. Санкт-Петербург

2025

Содержание	
ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ	2
ОТЧЕТ О ХОДЕ ВЫПОЛНЕНИЯ	2
СПИСОК ФАКТОВ С ОПИСАНИЕМ АРГУМЕНТОВ:	3
СПИСОК ПРАВИЛ С КОММЕНТАРИЯМИ:	4
ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАПРОСОВ:	7
ЗАКЛЮЧЕНИЕ	13

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Создание базы знаний в среде Prolog

Предметная область – Ваше семейное дерево

1. Создать не менее 30 объектов – членов вашей семьи.
2. Факты, отражающие состояние – события, регистрируемые органами ЗАГС (рождение, смерть, заключение брака, расторжение брака).
3. Создать не менее 30 правил, определяющих на основании событий состояния членов семьи и отношения между ними.
4. Правила должны учитывать темпоральность состояний с точностью до года.

Содержание отчета:

1. Семейное дерево (можно хоть от руки).
2. Список фактов с описанием аргументов.
3. Список правил с комментариями.
4. Не менее 10 скриншотов с результатами выполнения запросов. Запросы могут быть примерно такими: На ком был ХХХ женат в 1997 году? Запросы НЕ на естественном языке. На Прологе.

Если у вас не набирается 30 человек, можете выдумать недостающих.

Если не хотите раскрывать сведения о своей семье, можете частично анонимизировать данные. Но когда станете знаменитостью, то можете сильно упростить работу своим биографам ☺. Мы же распространять ваши данные не будем.

.

ОТЧЕТ О ХОДЕ ВЫПОЛНЕНИЯ

Построенное семейное дерево (представлено на рисунке 1):

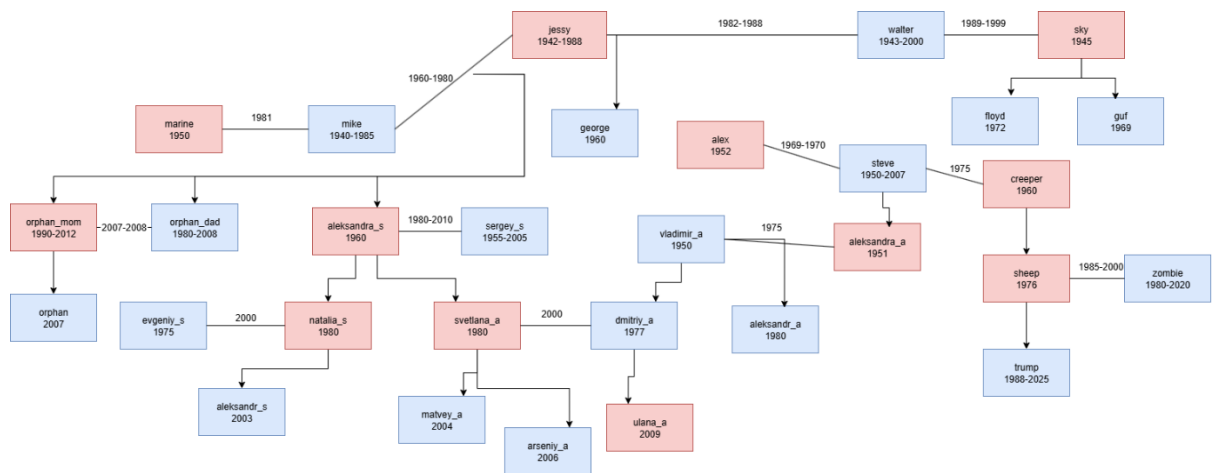


Рисунок 1 – Семейное дерево

СПИСОК ФАКТОВ С ОПИСАНИЕМ АРГУМЕНТОВ:

born/3: рождение.

- Name: имя человека.
- Year: год рождения.
- Sex: пол человека.

died/2: смерть.

- Name: имя человека.
- Year: год смерти.

married/3: заключение брака.

- Name1: имя первого человека.
- Name2: имя второго человека.
- Year: год заключения брака.

divorced/3: расторжение брака.

- Name1: имя первого человека.
- Name2: имя второго человека.
- Year: год расторжения брака.

parents/3: родители ребенка.

- Parent1: имя первого родителя.
- Parent2: имя второго родителя.
- Child: имя ребенка.

СПИСОК ПРАВИЛ С КОММЕНТАРИЯМИ:

% жив ли такой то персонаж в таком то году

```
alive(Person, Year) :-  
    number(Year),  
    Year =< 2025,  
    born(Person, BirthYear, _),  
    BirthYear =< Year,  
    ( \+ died(Person, _) ; (died(Person, DiedYear), DiedYear > Year)).
```

% true, если Parent является одним из родителей Child

```
parent_of(P, C) :-  
    parents(P, _, C) ; parents(_, P, C).
```

% человек - мужчина?

```
male(Person) :-  
    born(Person, _, m).
```

% человек - женщина?

```
female(Person) :-  
    born(Person, _, f).
```

% нахождение отца ребенка

```
father(F, Child) :-  
    parent_of(F, Child),  
    male(F).
```

% нахождение матери ребенка

```
mother(M, Child) :-  
    parent_of(M, Child),  
    female(M).
```

% полные братья или сестры: (оба родителя)

```
full_sibling(X, Y) :-  
    X \= Y,  
    mother(M, X), mother(M, Y),  
    father(F, X), father(F, Y).
```

% один общий родитель

```
half_sibling(X, Y) :-  
    X \= Y,  
    (  
        (mother(M, X), mother(M, Y), \+ full_sibling(X, Y))  
        ;  
        (father(F, X), father(F, Y), \+ full_sibling(X, Y))  
    ).
```

% хотя бы 1 общий родитель

```
sibling(X, Y) :-  
    X \= Y,  
    parent_of(P, X),  
    parent_of(P, Y).
```

% нахождение сестер

```
sisters(X, Y):-  
    female(X),  
    female(Y),  
    full_sibling(X, Y).
```

% нахождение братьев

```
brothers(X, Y):-  
    male(X),  
    male(Y),  
    full_sibling(X, Y).
```

% женаты ли всю жизнь

```
is_married(Wife, Husband):-  
    married(Wife, Husband, _) ; married(Husband, Wife, _),  
    \+ divorced(Wife, Husband, _) ; \+ divorced(Husband, Wife, _).
```

% найти мужа

```
husband(Wife, Husband):-  
    female(Wife),  
    is_married(Wife, Husband),  
    male(Husband).
```

% найти жену

```
wife(Husband, Wife):-  
    male(Husband),  
    is_married(Husband, Wife),  
    female(Wife).
```

% двоюродные братья и сестры

```
cousins(X, Y):-  
    \+ sibling(X, Y),  
    parent_of(P1, X),  
    parent_of(P2, Y),  
    full_sibling(P1, P2).
```

% двоюродные сестры

```
cousins_sisters(X, Y):-  
    female(X),  
    female(Y),  
    cousins(X, Y).
```

% двоюродные братья

```
cousins_brothers(X, Y):-  
    male(X),  
    male(Y),  
    cousins(X, Y).
```

% бабушки дедушки

```
grandparent(Child, Grandparent) :-  
    parent_of(Parent, Child),  
    parent_of(Grandparent, Parent).
```

```

% бабушки
grandmother(Child, Grandmother):-
    female(Grandmother),
    grandparent(Child, Grandmother).

% дедушки
grandfather(Child, Grandfather):-
    male(Grandfather),
    grandparent(Child, Grandfather).

% универсальная проверка на брак (без разницы кто идет первый)
married_univ(X, Y, Year):-
    married(X, Y, Year) ; married(Y, X, Year).

% женаты ли в такой год
married_in(X, Y, Year):-
    married_univ(X, Y, MarryYear),
    Year >= MarryYear,
    (+divorced(X, Y, _) ; (divorced(X, Y, DivYear), DivYear > Year)).

% дети в такой то год
childrens_in(Parent, Child, Year):-
    born(Child, BornYear, _),
    BornYear <= Year,
    parent_of(Parent, Child).

% сыновья в такой то год
sons_in(Parent, Boy, Year):-
    male(Boy),
    childrens_in(Parent, Boy, Year).

% дочери в такой то год
daughters_in(Parent, Girl, Year):-
    female(Girl),
    childrens_in(Parent, Girl, Year).

% сирота ли в такой то год
orphan_in(Child, Year):-
    born(Child, BornYear, _),
    BornYear <= Year,
    parent_of(P1, Child),
    parent_of(P2, Child),
    P1 \= P2,
    died(P1, P1Y),
    died(P2, P2Y),
    P1Y <= Year,
    P2Y <= Year.

% дяди
uncle(Uncle, Child) :-
    parent_of(Parent, Child),
    brothers(Uncle, Parent).

```

```

% тети
aunt(Aunt, Child) :-
    parent_of(Parent, Child),
    sisters(Aunt, Parent).

% внуки внучки
grandchild(Grandparent, Grandchild) :-
    parent_of(Parent, Grandchild),
    parent_of(Grandparent, Parent).

% кол-во детей к такому то году
children_count_in(Parent, Year, Count) :-
    findall(Child, childrens_in(Parent, Child, Year), Children),
    length(Children, Count).

% кол-во братьев к такому то году
brothers_count_in(X, Year, Count) :-
    findall(Bro, (brothers(X, Bro), born(Bro, BY, _), BY =< Year), Bros),
    length(Bros, Count).

% кол-во сестер к такому то году
sisters_count_in(X, Year, Count) :-
    findall(Sis, (sisters(X, Sis), born(Sis, BY, _), BY =< Year), Siss),
    length(Siss, Count).

```

ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАПРОСОВ:

```

?- sibling(arseniy_a, X).
X = matvey_a ;
X = ulana_a ;
false.

```

Рисунок 2 – Запрос “Братья или сестры Арсения”

```

?- brothers(arseniy_a, X).
X = matvey_a ;
false.

```

Рисунок 3 – Запрос “Братья Арсения”

```
?- married_in(svetlana_a, X, 2007).  
X = dmitriy_a ;  
false.  
  
?- married_in(svetlana_a, dmitriy_a, 2010).  
true .
```

Рисунок 4 – Запрос “С кем состояла в браке Светлана в 2007? Состояла ли она в браке в 2010?”

```
?- childrens_in(dmitriy_a, X, 2009).  
X = arseniy_a ;  
X = matvey_a ;  
X = ulana_a ;  
false.  
  
?- childrens_in(dmitriy_a, X, 2005).  
X = matvey_a ;  
false.
```

Рисунок 5 – Запрос “Дети Дмитрия в 2009 и 2005?”

```
?- orphan_in(orphan, 2015).  
true .
```

Рисунок 6 – Запрос “orphan сирота в 2015?”

```
?- cousins(matvey_a, X).  
X = aleksandr_s ;  
false.
```

Рисунок 7 – Запрос “Двоюродные братья/сестры Матвея”


```
?- grandchild(aleksandra_a, X).  
X = arseniy_a ;  
X = matvey_a ;  
X = ulana_a ;  
false.
```

Рисунок 8 – Запрос “Внуки Александры.”

```
?- children_count_in(svetlana_a, 2008, X).  
X = 2.  
  
?- ▲
```

Рисунок 9 – Запрос “Количество детей у светланы в 2008”

```
?- aunt(X, arseniy_a).  
X = natalia_s ;  
false.  
  
?- uncle(X, arseniy_a).  
X = aleksandr_a ;  
false.
```

Рисунок 10 – Запрос “Тетя и дядя Арсения?”

```
?- alive(trump, 2025).  
false.
```

Рисунок 11 – Запрос “Жив ли Трамп в 2025?”

2 Часть: Protege

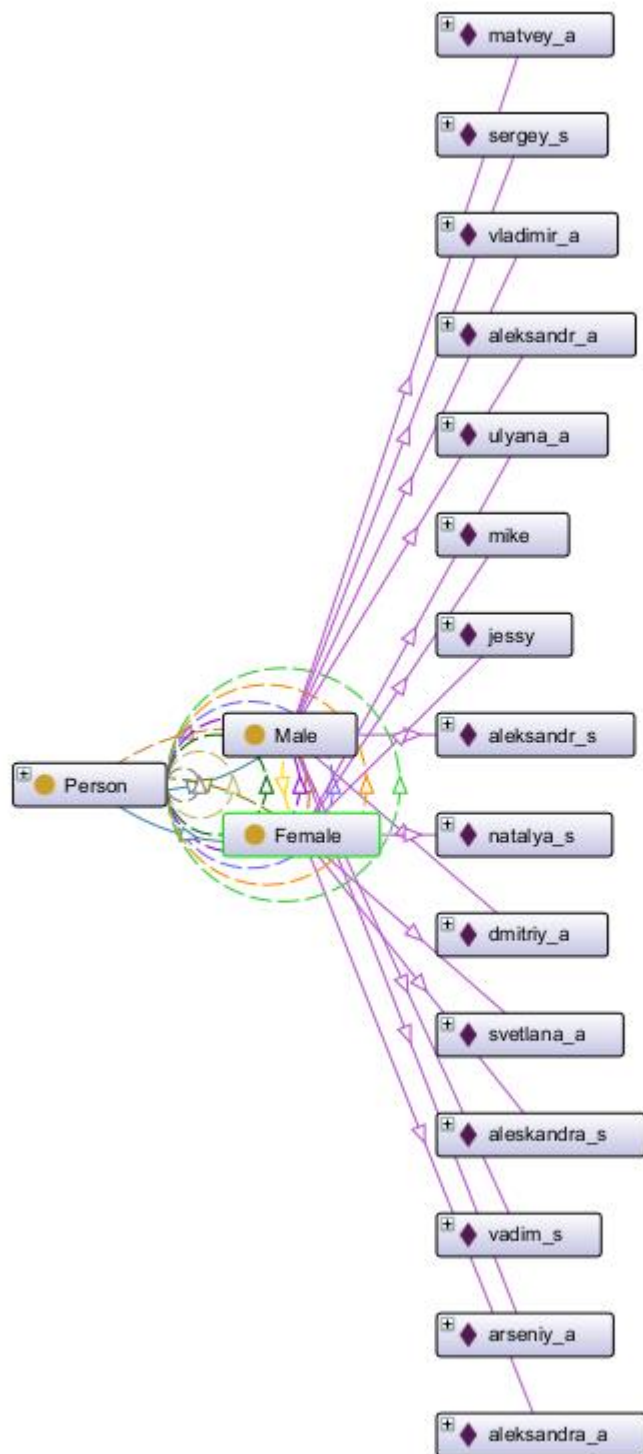


Рисунок 12 – Граф Protege.

Property assertions: ulyana_a

Object property assertions +

childOf dmitriy_a	?
childOf svetlana_a	?
grandsonOf aleksandra_a	?
grandsonOf aleskandra_s	?
grandsonOf sergey_s	?
grandsonOf vladimir_a	?
sibling arseniy_a	?
sibling matvey_a	?
sisterOf arseniy_a	?
sisterOf matvey_a	?

Data property assertions +

birthYear 2009	?	@	x
----------------	---	---	---

Negative object property assertions +

Рисунок 13 – Полученные данные для ulyana_a после “start Reasoner”

Property assertions: sergey_s

Object property assertions +

marriedTo aleskandra_s	?	@	x	o
fatherOf natalya_s	?	@		
fatherOf svetlana_a	?	@		
grandparentOf aleksandr_s	?	@		
grandparentOf arseniy_a	?	@		
grandparentOf matvey_a	?	@		
grandparentOf ulyana_a	?	@		
parentOf natalya_s	?	@		
parentOf svetlana_a	?	@		

Data property assertions +

birthYear 1960	?	@	x	o
diedYear 2007	?	@	x	o

Negative object property assertions +

Рисунок 14 – Полученные данные для sergey_s после “start Reasoner”

Query (class expression)

Person

Execute Add to ontology

Query results

Instances (15 of 15)

aleksandr_a	?	<input type="checkbox"/>
aleksandr_s	?	<input type="checkbox"/>
aleksandra_a	?	<input type="checkbox"/>
aleksandra_s	?	<input type="checkbox"/>
arseniy_a	?	<input type="checkbox"/>
dmitriy_a	?	<input type="checkbox"/>
jessy	?	<input checked="" type="checkbox"/>
matvey_a	?	<input type="checkbox"/>
mike	?	<input type="checkbox"/>
natalya_s	?	<input type="checkbox"/>
sergey_s	?	<input type="checkbox"/>
svetlana_a	?	<input type="checkbox"/>
ulyana_a	?	<input type="checkbox"/>
vadim_s	?	<input checked="" type="checkbox"/>
vladimir_a	?	<input checked="" type="checkbox"/>

Рисунок 15 – Запрос “Person”

Query (class expression)

childOf some and parentOf some

Execute Add to ontology

Query results

Instances (4 of 4)

aleksandr_a
dmitriy_a
natalya_s
svetlana_a

Рисунок 16 – Запрос “дети у которых есть дети?”

Query (class expression)

Male and sibling some Person

Execute Add to ontology

Query results

Instances (4 of 4)

aleksandr_a	?
arseniy_a	?
dmitriy_a	?
matvey_a	?

Рисунок 17 – Запрос “Мужчины у которых есть братья/сестры”

Query (class expression)

Person and (brotherOf some Person or sisterOf some Person) and motherOf some Person

Execute

Add to ontology

Query results

Instances (2 of 2)

natalya_s

svetlana_a

Query for

☐ Direct superclasses
 ☐ Superclasses
 ☐ Equivalent classes

Рисунок 18 – Запрос “Человек, который брат или сестра кого-то при этом мать кого-то.”

Name	Rule
brotherOf	untitled-ontology-3:sibling(?X, ?Y) ^ untitled-ontology-3:Male(?X) -> untitled-ontology-3:brotherOf(?X, ?Y)
fatherOf	untitled-ontology-3:parentOf(?X, ?Y) ^ untitled-ontology-3:Male(?X) -> untitled-ontology-3:fatherOf(?X, ?Y)
motherOf	untitled-ontology-3:parentOf(?X, ?Y) ^ untitled-ontology-3:Female(?X) -> untitled-ontology-3:motherOf(?X, ?Y)
siblings	untitled-ontology-3:childOf(?X, ?P) ^ untitled-ontology-3:childOf(?Y, ?P) ^ differentFrom(?X, ?Y) -> untitled-ontology-3:sibling(?X, ?Y)
sisterOf	untitled-ontology-3:sibling(?X, ?Y) ^ untitled-ontology-3:Female(?X) -> untitled-ontology-3:sisterOf(?X, ?Y)

Рисунок 19 – SWRL правила

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы была реализована база знаний на Prolog, моделирующая семейное дерево с учётом жизненных событий (рождения, браки, разводы, смерти). На основе фактов и правил стало возможным формализованно определять родственные отношения и отслеживать изменения статуса членов семьи во времени.

Практическим результатом стало то, что система не только хранит структурированные данные о семье, но и делает выводы, недоступные при простом перечислении. Работа продемонстрировала ценность логического программирования для представления знаний и рассуждений в задачах, где важна причинно-временная последовательность.

2 Часть:

В ходе лабораторной работы в среде Protege была построена онтология семейного дерева с использованием классов, свойств и правил SWRL. Reasoner позволил автоматически выводиться новые родственные отношения (брат, сестра, мать, дядя и др.), а не только хранить исходные факты. Практическим результатом стало получение системы, которая проверяет согласованность данных и расширяет знания о семье за счёт логического вывода.