Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

**Лабораторная работа №2**

по дисциплине

*«Системы искусственного интеллекта»*

Вариант «Семейное древо»

**Выполнил:** Батаргин Егор Александрович

**Группа:** P3332

**ITMO.ID:** 335189

**г. Санкт-Петербург, 2024 г.**

Оглавление

[Задание 2](#_Toc185005029)

[Выполнение 2](#_Toc185005030)

[Вывод 5](#_Toc185005031)

# Задание

Преобразовать факты и отношения из Prolog в концепты и свойства в онтологии. Описать классы и свойства в онтологии, которые соответствуют объектам и отношениям из базы знаний. Сделать это на основе семейного древа.

Изначально на Prolog у меня были классы:

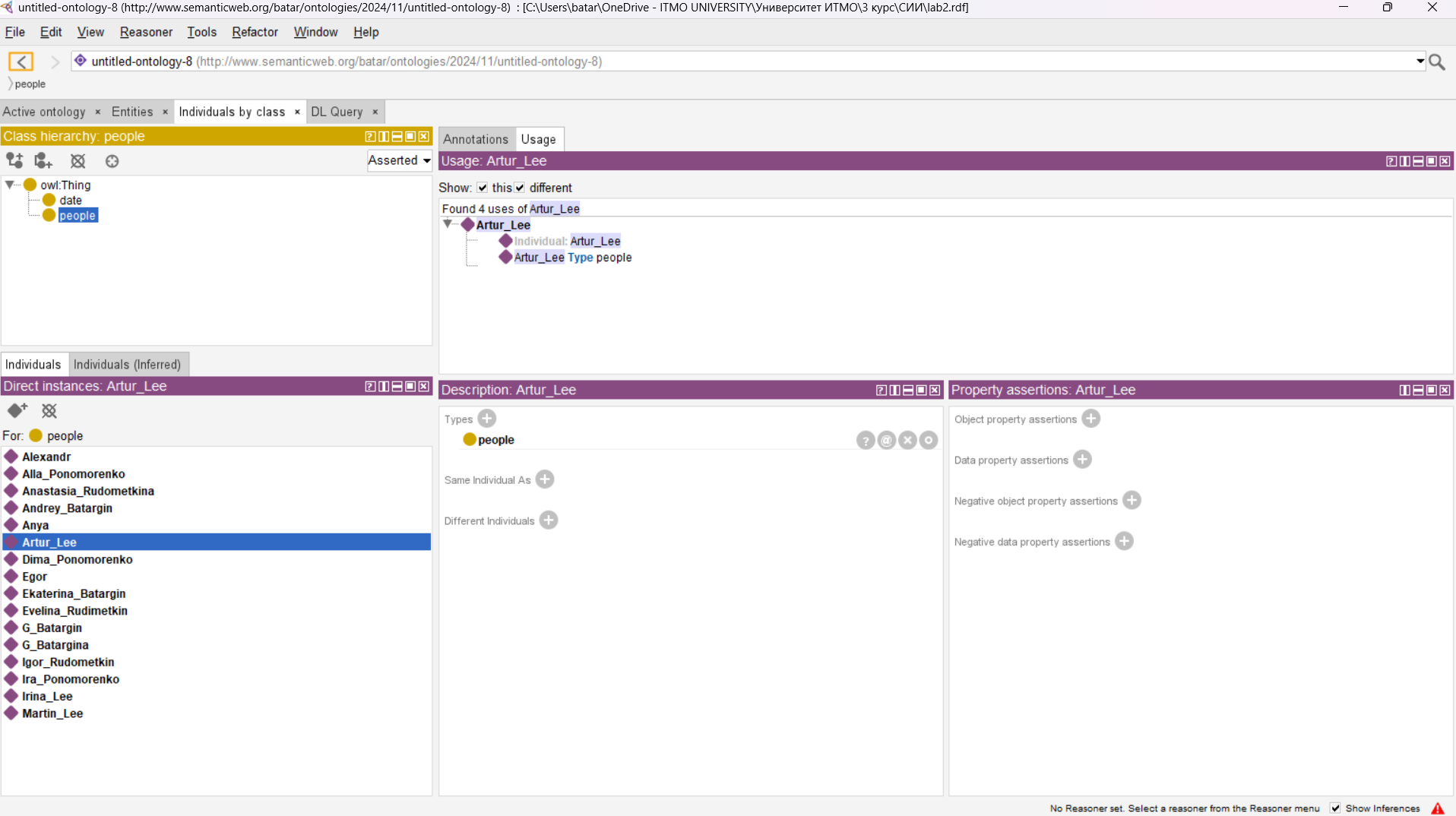
* people – база знаний с именами людей
* parent(name1, name2) – факт, что name1 родитель у name2
* birthday (name1, date) – факт, что у name1 дата рождения в date
* marriage(name1, name2) – факт, что name1 и name2 женаты

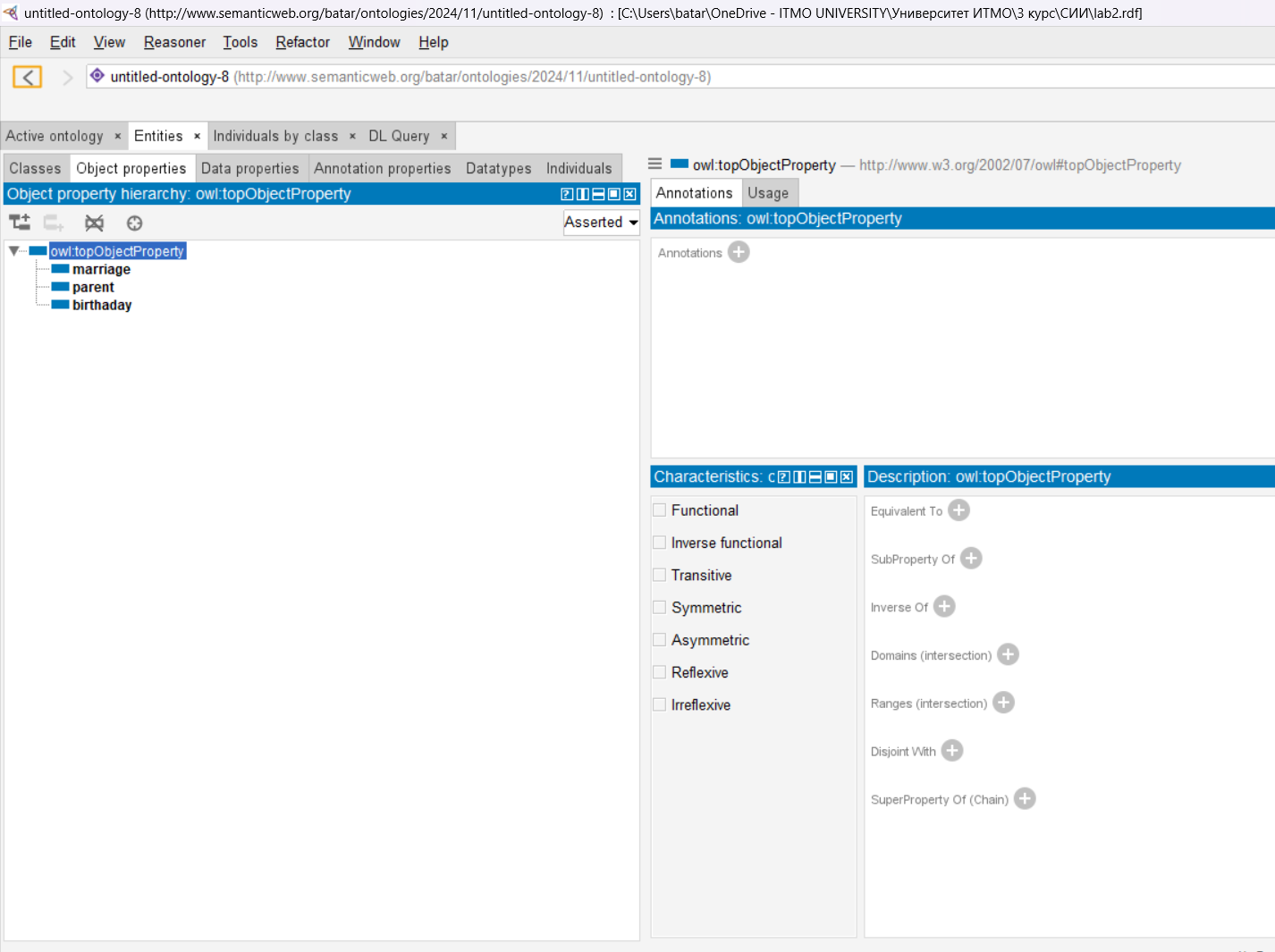
И есть три правила:

* is\_relatives (name1, name2) - правило, подтверждающее или не подтверждающее того, что name1 и name2 родные братья или сестры
* is\_grandparent (name1, name2) – правило, что name1 является прародителем name2
* is\_cousins(name1, name2) – правило, что name1 и name2 братья и сестры

# Выполнение

На фото нижу можно увидеть пример класса, который содержит имена людей. То есть это переделанная база знаний из Prolog



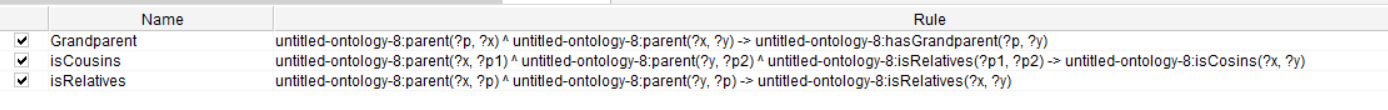
Так же на фото выше можно видеть класс date. Это нужно для дальнейшего установления дня рождения, чтобы создать аналог факту birthday. Такие связи можно установить в properties. Собственно на фото ниже можно увидеть три факта – marriage, parent, birthday. Вернее это аналоги фактов из Prolog  


Так же были реализованы правила, но под другими названиями

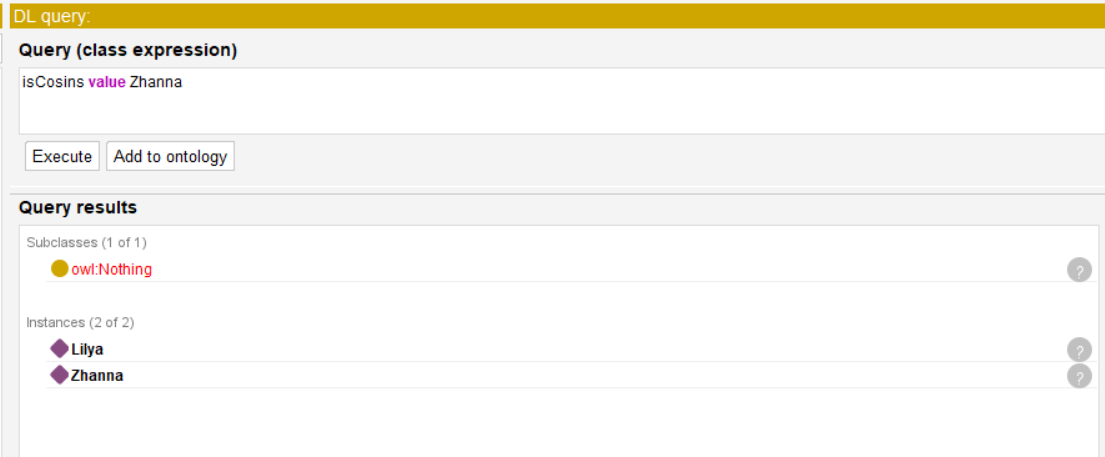
hasGrandparent – is\_grandparent из Prolog

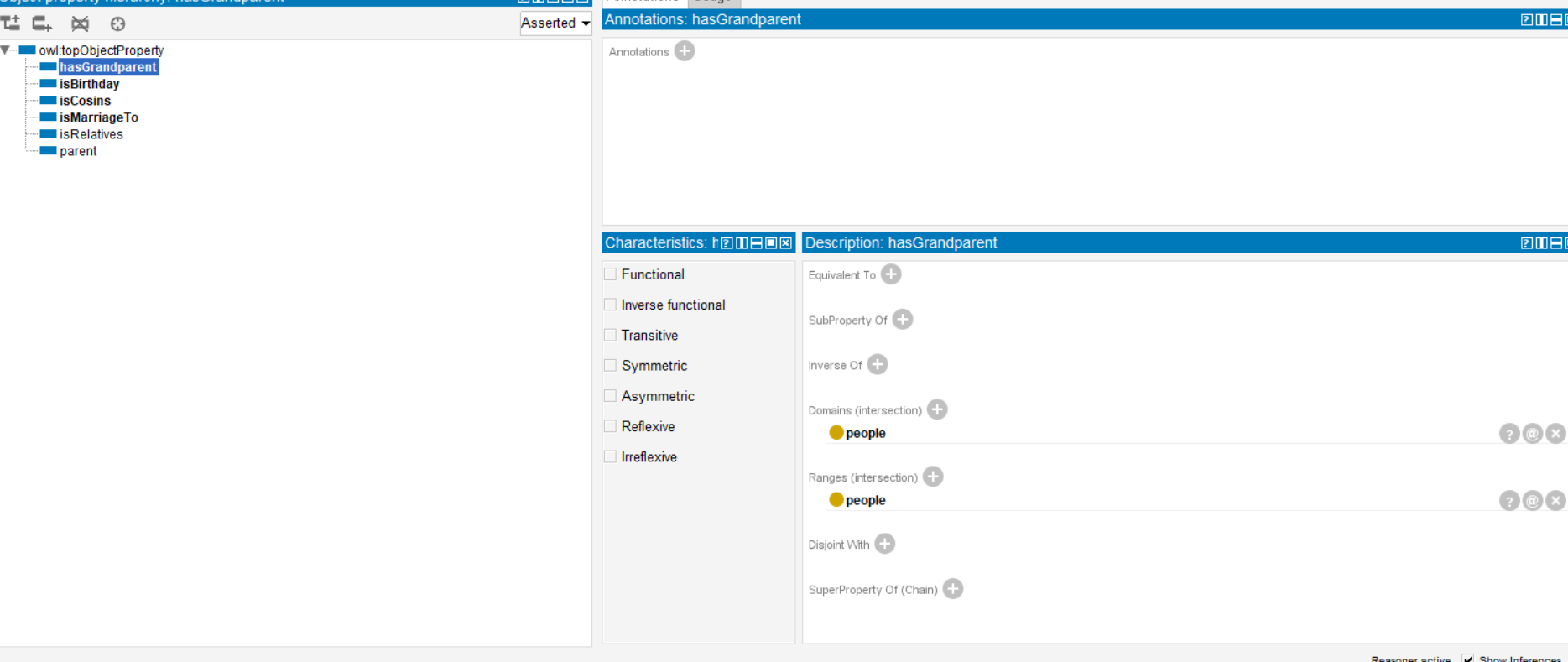
isRelatives – is\_relative из Prolog

isCousins – is\_cousins из Prolog



Теперь для работы с этими правилами мы должны запустить инструмент reasoner – инструмент для установки новых фактов на основе правил (логических выражений). Так же на вкладке DL Query мы можем писать запросы по конкретным людям. Пример работы ниже

К сожалению, но в правилах в Protégé тяжело избежать того, что выводиться само имя человека.



Выше пример того, что факты на основе правил появились автоматически.

# Вывод

В ходе лабораторной работы я познакомился с программной Protégé. На мое усмотрение эта программа не так удобна, как работа на Prolog, поскольку тяжелее разрабатывать правила и использовать их. Но в некоторых момент может и удобнее.