НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет систем управления и робототехники

Системы искусственного интеллекта

Лабораторная работа № 1

Выполнил студент

Куприянов А.А.

Группа № P33113

Преподаватель: Болдырева Елена Александровна

г. Санкт-Петербург

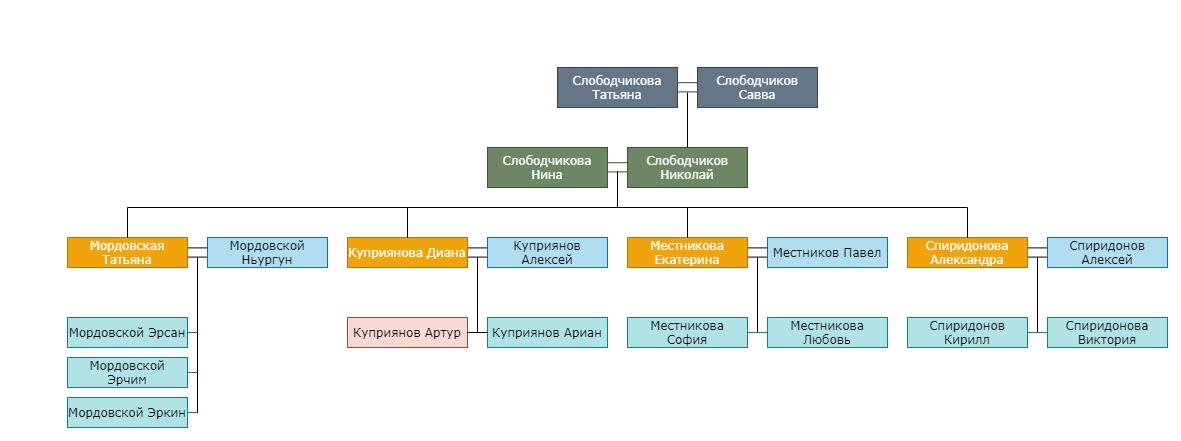
2020

**Вариант: Семейное древо**

**Задание:**

Необходимо составить семейное древо и составить программу (базу знаний) на языке Prolog. При выполнении лабораторной работы соблюдаются несколько ограничений: как минимум 20 элементов в дереве и минимум 10 правил.

**Семейное древо:**

****

Здесь представлен 21 элемент, состоящий из 4 поколений

**Отчет:**

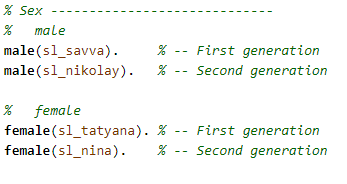
Сначала я построил семейное древо, чтобы легче было ориентироваться при написании базы знаний, так как программа (по сути просто набор определений) на прологе не является удобочитаемым для человека

Для начала дадим несколько простых определений, которые явно могут выражены из дерева:

* Родитель
* Пол
* Брак

Эти правила являются обязательными (фундаментальными) так как не могут быть выражены из других отношений (кроме инвертированных (сын/дочь))

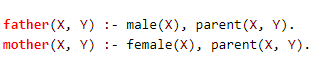
Рассмотрим для начала первое и второе поколения. Дадим определения пола и родительства



Зададим также родительство

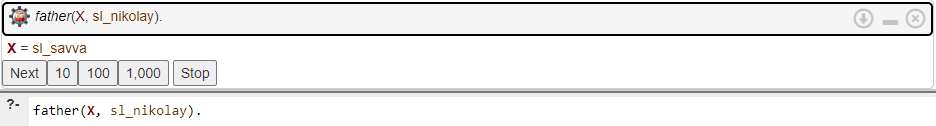


Теперь определим первое правило

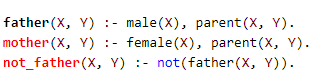


Следуя правилам пролога, можно сказать, что [X является father Y] тогда и только тогда, когда [X является male] и верно условие [X является parent Y]

Проверим вывод программы при вызове функции father с (x, sl\_nikolay):



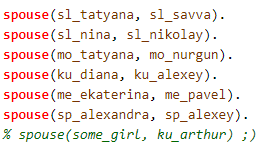
Также мы можем создать отрицания для каждого правила:





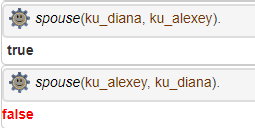
Но они довольно простые и писать для каждого правила их отрицания займет много времени и место в программе

Задав parent для остальных поколений, сделаем пару определений со spouse:



Все опредедения готовы. Давайте напишем пару функций:

Во-первых, например, функция spouse в предметной области имеет двунарпавленное отношение, то есть если X spouse Y, то Y spouse X

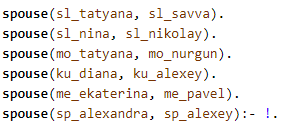


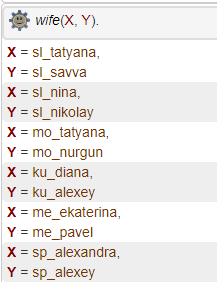
Чтобы Prolog понял что мы имеем в виду, мы можем написать:





Но в таком случае, здесь будет происходить бесконечное углубление в рекурсию. Например, вызвав wife(x, y) мы получим бесконечную рекурсию. Чтобы предотворатить это, сделаем красное отсечение:

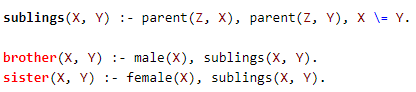




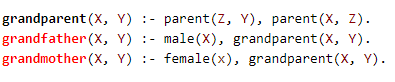


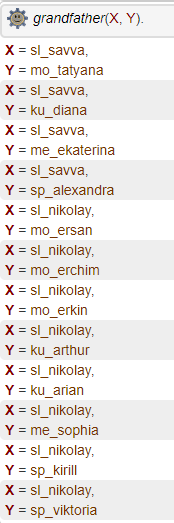
Братство возможно, если X и Y имеют хотя бы один Z, который является parent для обоих, при этом важно сказать, что X не есть Y

Для сестринства услвоие такое же, но вместо male нужно использовать female. В этих двух определниях мы видим сходства, поэтому сделаем отдельное определение для братьев/сестер



(сделано не для заполнения до 10 правил 😇 )





**Вывод:**

Из данной работы можно понять, что пролог является довольно специфичным языком программирования. На самом деле, мне даже трудно называть это программирование, так как он сильно отличается от привычных нам языков (декларативность react и рядом с прологом не стояла)

Тем не менее, язык богат своим синтаксисом и предлагает пользователю большой их набор для составления обширных баз знаний.

Из минусов я бы хотел выделить сложность разработки (если это так можно назвать). Потому что поддерживать код написанный на прологе мне кажется очень не просто. На примере, spouse отношения мы легко могли попасть в бесконечную рекурсию, а ведь мы писали очень простое задание (боюсь представить что будет в ИИ). Мне кажется работу с Prolog нужно вести не на прямую, а через прикладные программы.