

# 1 Введение

В первом блоке происходит знакомство с такими понятиями как база знаний и онтология. На их основе можно строить вопросно-ответные системы (экспертные), которые своим поведением могут напоминать современные LLM. Примером таких систем являются чатботы на сайтах банков, управляющих компаний etc. Для ознакомления с тем, на чем такие программы базируются, были сделаны лабораторные работы 1 и 2.

## 2 Анализ требований

### 2.1 Лабораторная работа №1, часть 1

Основными требованиями к базе знаний были характеристики содержащихся в ней фактов и правил, а именно:

- факты с 1 аргументом
- факты с 2 аргументами
- правила, содержащие логические операции (И/ИЛИ etc.)

На основе этих трех структур можно строить самые примитивные вопросно-ответные системы, способные отвечать на заданные вопросы да/нет или предоставлять список сущностей, подходящих под заранее определенный набор фильтров.

### 2.2 Лабораторная работа №1, часть 2

В данной лабораторной работе требовалось перевести ранее созданную базу знаний в онтологию Prolog. Онтологии отличаются от баз знаний устройством и процессом создания. Так объекты БЗ - классы онтологии, отношения БЗ - свойства онтологии. Соответственно, главным требованием является корректность переноса и способность выполнить те же запросы, что и в БЗ с тем же вариантом.

### 2.3 Лабораторная работа №2

Во второй лабораторной работе как раз предлагается создать систему на основе БЗ/онтологии - систему поддержки принятия решений, которая на основе данных из запроса будет выводить данные с рекомендациями. Соответственно главным требованием является корректный парсинг запроса и составление корректных запросов к БЗ/онтологии.

## 3 Изучение основных концепций и инструментов

### 3.1 Prolog

Пролог является логическим языком программирования. Основные концепции языка представлены ниже.

- Декларативный стиль программирования – следовательно программист описывает что должно быть сделано, а не как.
- Факты – описывают истины о некоторой теме. Например персонаж(иван) - значит иван является персонажем.
- Правила – логические зависимости между фактами. Например горожанин(X) :- персонаж(X) - значит любой X горожанин если он персонаж.
- Унификация – процесс сопоставления запроса с содержимым БЗ с целью найти данные, которые соответствуют запросу.
- Обратный поиск – в отличие от простых реализаций поиска решений, к которым может прийти самостоятельно, Prolog использует механизм обратного поиска, который позволяет ему возвращаться к чекпоинтам в случае неудачного поиска и начинать с конкретного места, а не с начала, что позволяет эффективно обходить все возможные варианты решений в поисках нужного.

### 3.2 Protégé

Данная программа используется для создания онтологий - позволяет моделировать и организовывать знания. Основные концепции ниже.

- Онтология – формальное описание знаний в определенной предметной области. Она включает в себя классы (понятия), свойства (отношения между понятиями) и индивиды (конкретные объекты).
- Классы и подклассы – представляют категории или типы объектов, а подклассы — это специфические категории, которые наследуют свойства от родительских классов.
- Свойства (отношения) – описывают связи между классами и индивидами. Свойства могут быть объектными (связь между двумя объектами) или датальными (связь объекта с данными, такими как числа или строки).
- Индивиды – конкретные экземпляры классов.
- Reasoners – возможность использовать механизмы вывода (reasoners), такие как Hermit или Pellet, для автоматической проверки логической целостности онтологии. Эти reasoners могут делать выводы на основе определенных фактов и правил, обнаруживать логические ошибки или находить скрытые связи между понятиями.

И Protégé и Prolog могут быть использованы для построения экспертных систем, которые в ограниченной области могут довольно эффективно находить ответы на заданные пользователем вопросы. Однако до современных LLM (большие языковые модели), если говорить о взаимодействии в формате чата, они не дотягивают ввиду узконаправленности БЗ/онтологий.

### 3.3 Библиотека JPL

В рамках лабораторных я ознакомился с библиотекой JPL для Java, которая позволяет взаимодействовать с базами знаний Prolog из java-программы. Основной командой для взаимодействия является:

```
Query facts = new Query(
    "consult",
    new Term[] { new Atom("path_to_*.pl_file") }
);
Query q = new Query("fact_or_rule", new Term[] { X })
q.allSolutions()
```

Через Query происходит подключение БЗ, обращение к БЗ и извлечение ответов.

## 4 Реализация системы искусственного интеллекта (системы поддержки принятия решений)

Для реализации СППР необходимо создать парсер, который будет вынимать из строки запроса параметры, на основе которых будет формироваться список обращений к БЗ. Так же необходимо создать метод, который будет обращаться к БЗ и записывать ее ответы. Это позволит создать простейшую вопросно-ответную систему.

## 5 Оценка и интерпретация результатов

В результате лабораторных работ первого блока была создана система помощи принятия решений соответствующая требованиям, были изучены алгоритмы создания баз знаний и онтологий. Для дальнейшего улучшения разработанной системы можно расширить базу знаний, добавив в нее больше фактов, а так же расширить количество шаблонов, которые принимает программа, чтобы общение с системой казалось более "натуральным".

## 6 Заключение

Системы помощи принятия решений довольно часто используются чтобы облегчить работу сотрудников службы поддержки или автоматизировать проверку гипотез, моделировать сценарии и проверять логическую корректность своих предположений.