

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)	Программная инженерия
Дисциплина	Системы искусственного интеллекта

ЛАБОРАТОРНАЯ РАБОТА 1
ОТЧЕТ

Выполнил студент: Кулагин Вячеслав Дмитриевич (408946)
Группа: Р3309
Преподаватель: Болдырева Елена Александровна (157150)

г. Санкт-Петербург

2025

Содержание	
ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ	2
ОТЧЕТ О ХОДЕ ВЫПОЛНЕНИЯ	2
ЗАКЛЮЧЕНИЕ	9

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Тематика базы знаний: Настольная игра «Ticket to Ride Europe»

- **Создание базы знаний:**
Создайте базу знаний. База знаний должна включать в себя не менее 20 фактов с одним аргументом, 10–15 фактов с двумя аргументам, которые дополняют и показывают связь с другими фактами и 5–7 правил. Факты могут описывать объекты, их свойства и отношения между ними. Факты 2 и более аргументами могут описывать различные атрибуты объектов, а правила - логические законы и выводы, которые можно сделать на основе фактов и предикатов.
- **Выполнение запросов:**
Напишите несколько запросов для БЗ. Запросы должны быть разной сложности и включать в себя:
 - Простые запросы к базе знаний для поиска фактов.
 - Запросы, использующие логические операторы (и, или, не) для формулирования сложных условий (или использовать логические операторы в правилах).
 - Запросы, использующие переменные для поиска объектов с определенными характеристиками.
 - Запросы, которые требуют выполнения правил для получения результата.
- **Документация:**
В коде должны быть комментарии описания фактов, предикатов и правил.

ОТЧЕТ О ХОДЕ ВЫПОЛНЕНИЯ

Для упрощения восприятия интерпретации игры в базе знаний, приведена фотография игрового поля на рисунке 1.



Рисунок 1 - Игровое поле настольной игры "Ticket to Ride Europe"

Построим базу знаний. Она содержит факты в первую очередь о маршрутах и городах, отражая основную информацию. Кроме этого, реализованы правила для получения информации о маршрутах, уточнения некоторых моментов и более удобного поиска необходимой информации. Листинг базы знаний на языке Prolog приведен ниже.

% Виды карт в игре Ticket to Ride Europe
карта (вагон) .
карта (локомотив) .

% Цвета маршрутов
цвет (синий) .
цвет (зеленый) .
цвет (желтый) .
цвет (черный) .
цвет (оранжевый) .
цвет (белый) .
цвет (красный) .
цвет (любой) .

% Города на игровом поле (взята только левая часть)
город (москва) .
город (петроград) .
город (рига) .
город (смоленск) .

город(киев) .
город(харьков) .
город(ростов) .
город(севастополь) .
город(будапешт) .
город(бухарест) .
город(варшава) .
город(вена) .
город(вильнюс) .

% Типы городов по размеру
тип_города(москва, большой) .
тип_города(петроград, большой) .
тип_города(рига, обычный) .
тип_города(смоленск, обычный) .
тип_города(киев, большой) .
тип_города(харьков, большой) .
тип_города(ростов, обычный) .
тип_города(севастополь, обычный) .
тип_города(будапешт, большой) .
тип_города(бухарест, обычный) .
тип_города(варшава, большой) .
тип_города(вена, большой) .
тип_города(вильнюс, обычный) .

% Доступные маршруты между двумя городами, их длина (в клетках), цвет. Все маршруты двунаправленные
маршрут(москва, петроград, 4, белый) .
маршрут(петроград, москва, 4, белый) .

маршрут(москва, харьков, 4, любой) .
маршрут(харьков, москва, 4, любой) .

маршрут(москва, смоленск, 2, оранжевый) .
маршрут(смоленск, москва, 2, оранжевый) .

маршрут(петроград, рига, 4, любой) .
маршрут(рига, петроград, 4, любой) .

маршрут(рига, вильнюс, 4, зеленый) .
маршрут(вильнюс, рига, 4, зеленый) .

маршрут(петроград, вильнюс, 4, синий) .
маршрут(вильнюс, петроград, 4, синий) .

маршрут(вильнюс, смоленск, 3, желтый) .
маршрут(смоленск, вильнюс, 3, желтый) .

маршрут(смоленск, киев, 3, красный) .
маршрут(киев, смоленск, 3, красный) .

маршрут(вильнюс, киев, 2, любой) .
маршрут(киев, вильнюс, 2, любой) .

маршрут(вильнюс, варшава, 3, красный) .
маршрут(варшава, вильнюс, 3, красный) .

```
маршрут(варшава, киев, 4, любой) .  
маршрут(киев, варшава, 4, любой) .
```

```
маршрут(киев, харьков, 4, любой) .  
маршрут(харьков, киев, 4, любой) .
```

```
маршрут(киев, будапешт, 4, любой) .  
маршрут(будапешт, киев, 4, любой) .
```

```
маршрут(будапешт, севастополь, 4, белый) .  
маршрут(севастополь, будапешт, 4, белый) .
```

```
маршрут(ростов, севастополь, 4, любой) .  
маршрут(севастополь, ростов, 4, любой) .
```

```
маршрут(ростов, харьков, 2, зеленый) .  
маршрут(харьков, ростов, 2, зеленый) .
```

```
% Правило о существовании прямого маршрута между двумя городами  
существует_маршрут(X, Y) :- маршрут(X, Y, _, _).
```

```
% Правило для получения длины прямого маршрута между двумя городами (в  
клетках)  
длина_маршрута(X, Y, Длина) :- маршрут(X, Y, Длина, _).
```

```
% Правило для получения цвета прямого маршрута между двумя городами  
цвет_маршрута(X, Y, Цвет) :- маршрут(X, Y, _, Цвет).
```

```
% Правило для опеределения городов-соседей. Города являются соседями, если  
между ними есть прямой маршрут  
соседи(X, Y) :- существует_маршрут(X, Y).
```

```
% Правило для проверки, что маршрут может быть любого цвета  
маршрут_с_любым_цветом(X, Y) :- цвет_маршрута(X, Y, любой).
```

```
% Правило для проверки совместимости по цвету карты и маршрута между двумя  
городами  
совместима_по_цвету(Цвет_карты, X, Y) :-  
    маршрут_с_любым_цветом(X, Y) ;  
    цвет_маршрута(X, Y, Цвет_карты).
```

```
% Правило для проверки, большой ли город  
город_большой(X) :- тип_города(X, большой).
```

Далее выполним несколько запросов к созданной базе знаний.

Простые запросы к базе знаний для поиска фактов представлены на рисунке 2. Это запросы, использующие напрямую факты, в том числе, получая все их возможные параметры, а также использующие простые правила.

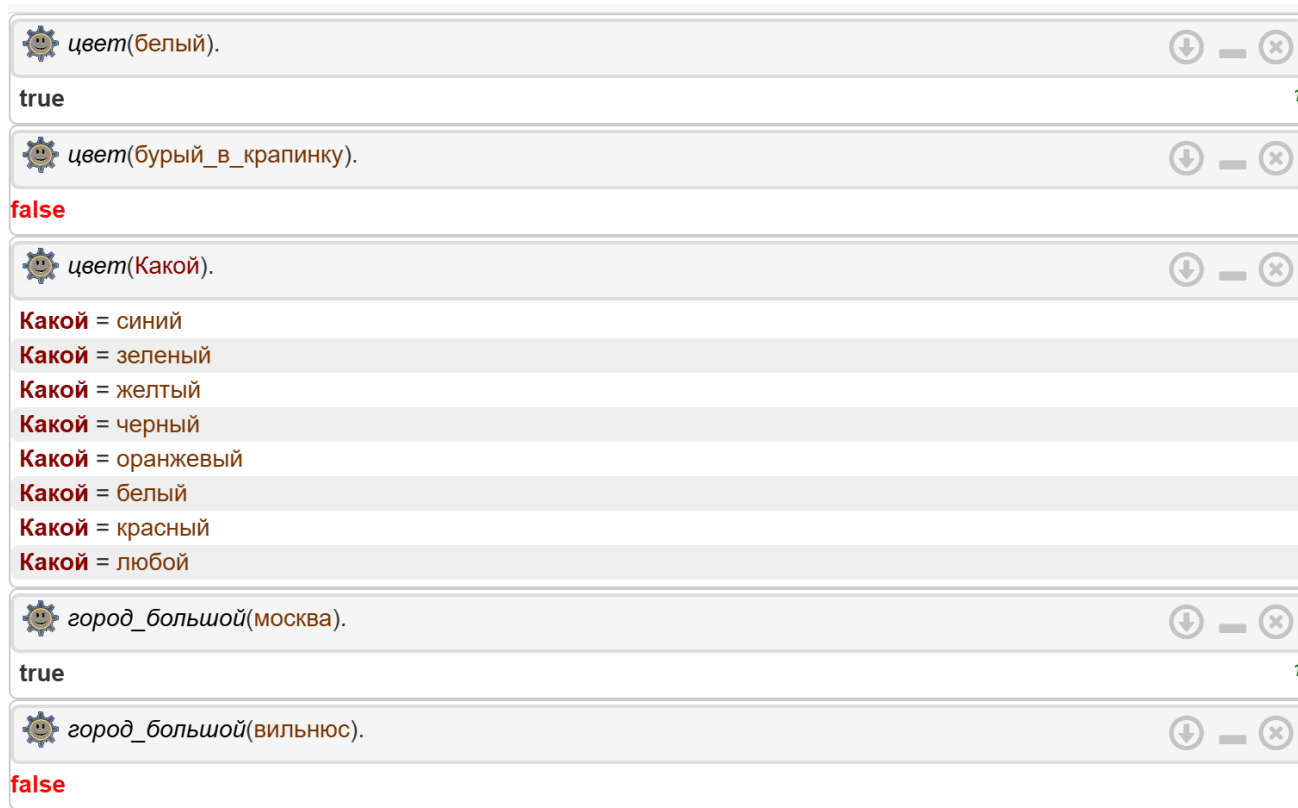


Рисунок 2 – Простые запросы к базе знаний

Далее приведу запросы с использованием логических операторов:

- Определим все возможные пункты назначения из Москвы, длина которых больше 3 клеток. Запрос и результат представлены на рисунке 3.

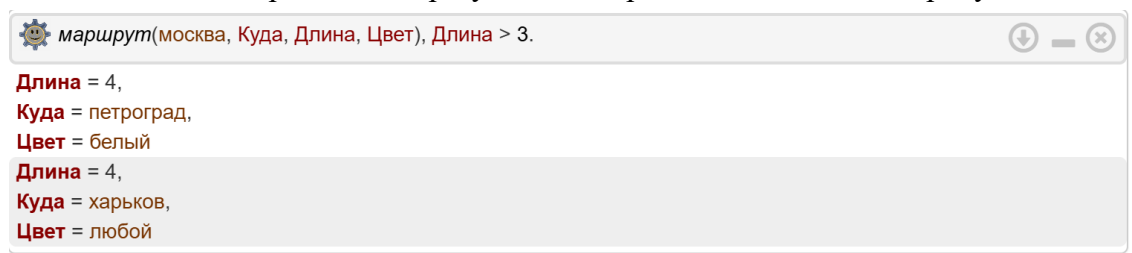


Рисунок 3 – Определение всех возможных пунктов назначения из Москвы, длина которых больше 3 клеток

- Определим куда можно уехать из Киева, используя маршрут красного цвета или маршрут белого цвета. Получу два результата: по красному маршруту можно уехать в Смоленск, а по белому маршруту невозможно уехать никуда (такого

маршрута не существует). Запрос и результат представлены на рисунке 4.

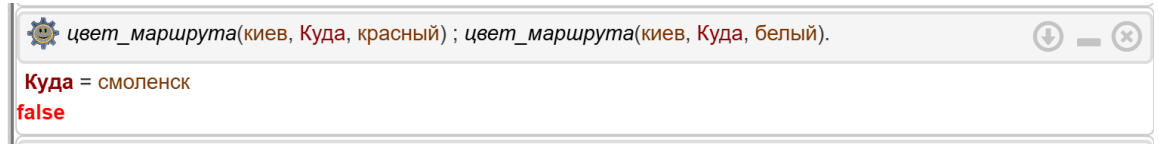


Рисунок 4 – Определение пунктов назначения из Киева по красным или белым маршрутам

- Определим откуда можно приехать в Москву по не цветным маршрутам, то есть по всем маршрутам, кроме бесцветного (любого цвета). Запрос и результат представлены на рисунке 5.

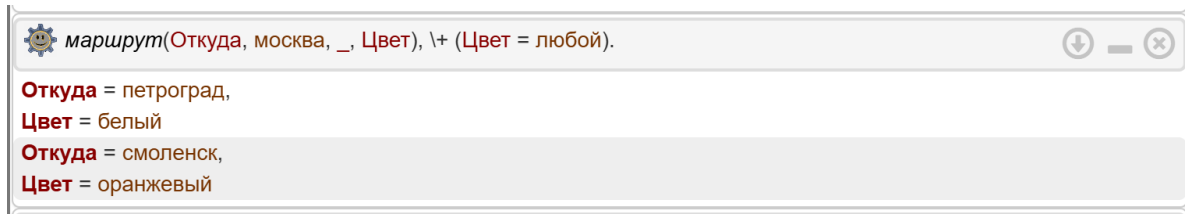


Рисунок 5 – Определение пунктов отправления в Москву по не цветным маршрутам

Далее приведу запросы, которые используют переменные для поиска объектов с определенными свойствами

- Определим все города, которые соединены между собой зелёными маршрутами. Запрос и результат представлены на рисунке 6.

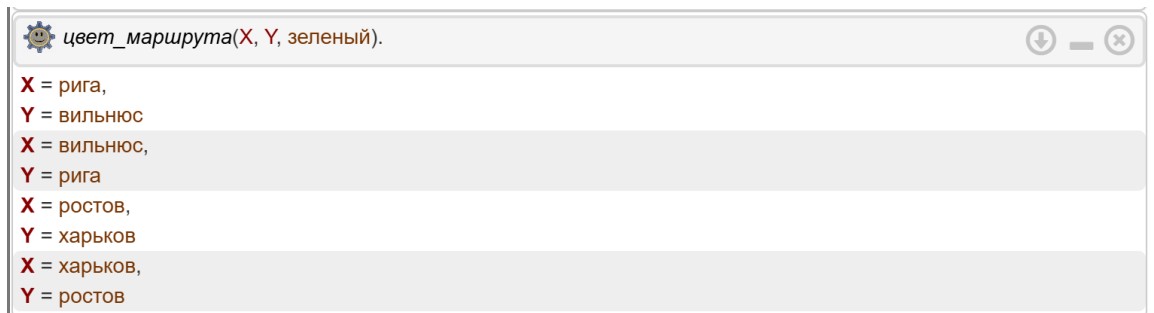


Рисунок 6 – Определение всех городов, соединенных зелеными маршрутами

- Определим все пары городов (откуда и куда можно доехать), а также длины этих маршрутов, которые могут быть любого цвета. Запрос и результат представлены на рисунке 7.

 маршрут(Откуда, Куда, Длина, любой).
Длина = 4, Куда = харьков, Откуда = москва
Длина = 4, Куда = москва, Откуда = харьков
Длина = 4, Куда = рига, Откуда = петроград
Длина = 4, Куда = петроград, Откуда = рига
Длина = 2, Куда = киев, Откуда = вильнюс
Длина = 2, Куда = вильнюс, Откуда = киев
Длина = 4, Куда = киев, Откуда = варшава
Длина = 4, Куда = варшава, Откуда = киев
Длина = 4, Куда = харьков, Откуда = киев
Длина = 4, Куда = киев, Откуда = харьков
Длина = 4, Куда = будапешт, Откуда = киев
Длина = 4, Куда = киев, Откуда = будапешт
Длина = 4, Куда = севастополь, Откуда = ростов
Длина = 4, Куда = ростов, Откуда = севастополь

Рисунок 7 – Определение пар городов и длины маршрутов между ними, которые не имеют определенного цвета

Далее приведу три запроса, которые используют только правила. Такие запросы и их результаты представлены на рисунке 8.

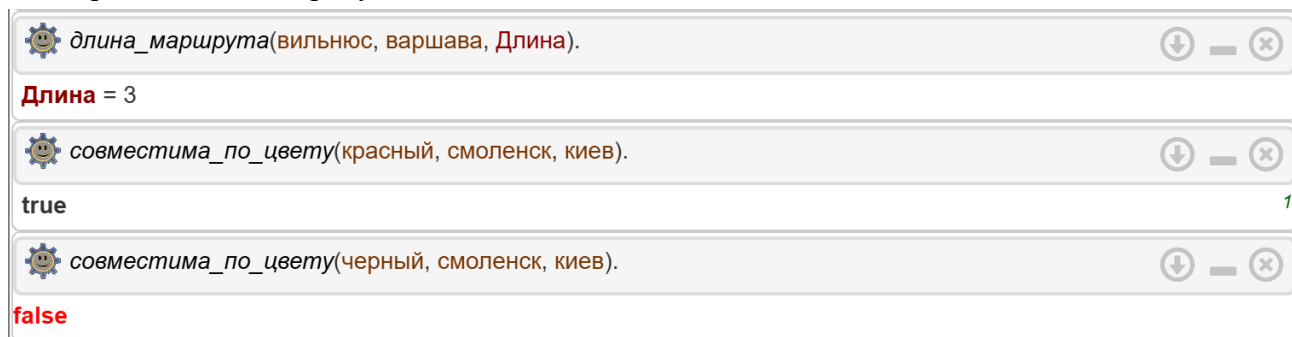


Рисунок 8 – Запросы, использующие правила

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы была создана база знаний на языке Prolog, описывающая настольную игру Ticket to Ride. Полученная база знаний содержит большое количество фактов и правил, отражающих концепцию игры и её правила.

На основе базы знаний были сформулированы и выполнены разнообразные запросы на получение простых данных из базы знаний, с использованием логических операторов, переменных, а также правил.

База знаний и запросы удовлетворяют требованиям задания: они корректно демонстрируют работу с фактами, предикатами, правилами и логическими операторами.