

Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Системы искусственного интеллекта

Лабораторная работа 1

Создание базы знаний и выполнение запросов в Prolog

Выполнил: Чан Дык Зюи

Группа: P33202

Преподаватель: Кугаевских Александр Владимирович

Санкт-Петербург

2023г.

1. Задание

Создайте базу знаний. База знаний должна включать в себя не менее 20 фактов с одним аргументом, 10–15 фактов с двумя аргументами, которые дополняют и показывают связь с другими фактами и 5–7 правил. Факты могут описывать объекты, их свойства и отношения между ними. Предикаты могут описывать различные атрибуты объектов, а правила – логические законы и выводы, которые можно сделать на основе фактов и предикатов.

2. Исходный код:

% Факты с одним аргументом

```
player(john).  
player(sarah).  
player(mike).  
player(lisa).  
player(alex).  
player(emily).  
player(mark).  
player(jessica).  
player(david).  
player(amy).  
player(brian).  
player(linda).  
player(steve).  
player(karen).  
player(peter).  
player(laura).  
player(chris).  
player(rachel).  
player(kevin).  
player(michelle).
```

% Факты с двумя аргументами

```
player_type(john, casual).  
player_type(sarah, hardcore).  
player_type(mike, casual).  
player_type(lisa, casual).  
player_type(alex, hardcore).  
player_type(emily, hardcore).  
player_type(mark, casual).  
player_type(jessica, casual).  
player_type(david, hardcore).  
player_type(amy, casual).  
player_type(brian, casual).
```

% Связь фактов с двумя аргументами с другими фактами

```
player_level(john, 5).  
player_level(sarah, 10).  
player_level(mike, 3).  
player_level(lisa, 7).  
player_level(alex, 12).  
player_level(emily, 8).  
player_level(mark, 2).
```

```
player_level(jessica, 5).
player_level(david, 15).
player_level(amy, 4).
```

% Правила для профилей игроков

```
casual_player(Player) :-
    player_type(Player, casual).
```

```
hardcore_player(Player) :-
    player_type(Player, hardcore).
```

```
higher_level(Player1, Player2) :-
    player_level(Player1, Level1),
    player_level(Player2, Level2),
    Level1 > Level2.
```

```
same_level(Player1, Player2) :-
    player_level(Player1, Level),
    player_level(Player2, Level),
    Player1 \= Player2.
```

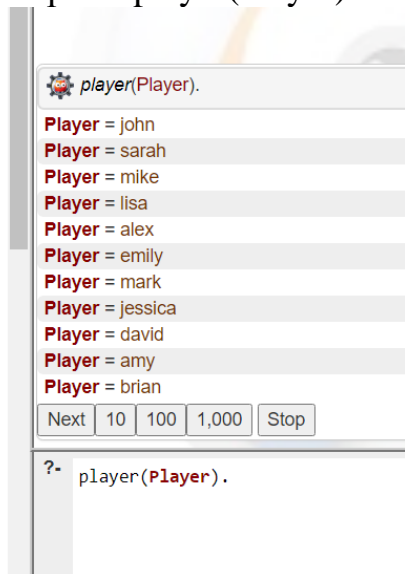
```
same_type(Player1, Player2) :-
    player_type(Player1, Type),
    player_type(Player2, Type),
    Player1 \= Player2.
```

3. Пример:

Простые запросы к базе знаний для поиска фактов:

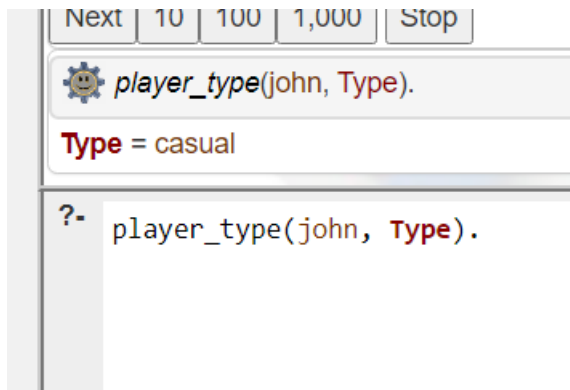
- "Какие игроки есть в базе знаний?"

Запрос: `player(Player).`



- "Какой тип игры у игрока John?"

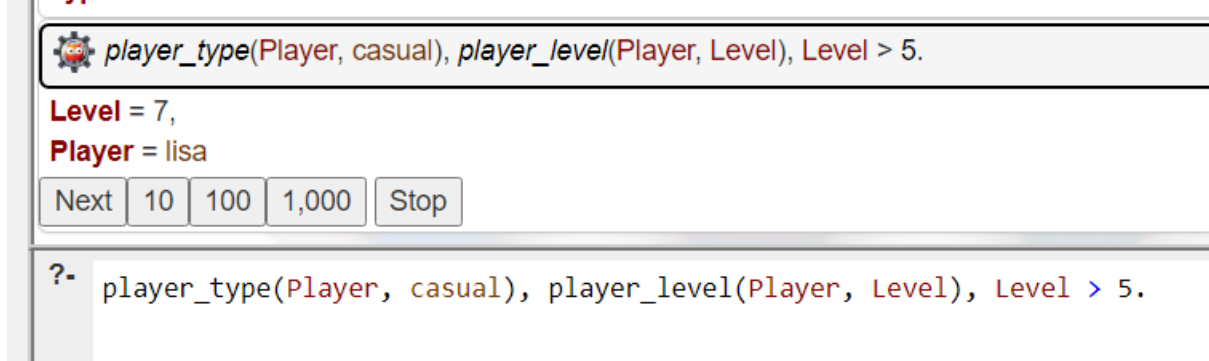
Запрос: `player_type(john, Type).`



Запросы, использующие логические операторы (и, или, не) для формулирования сложных условий:

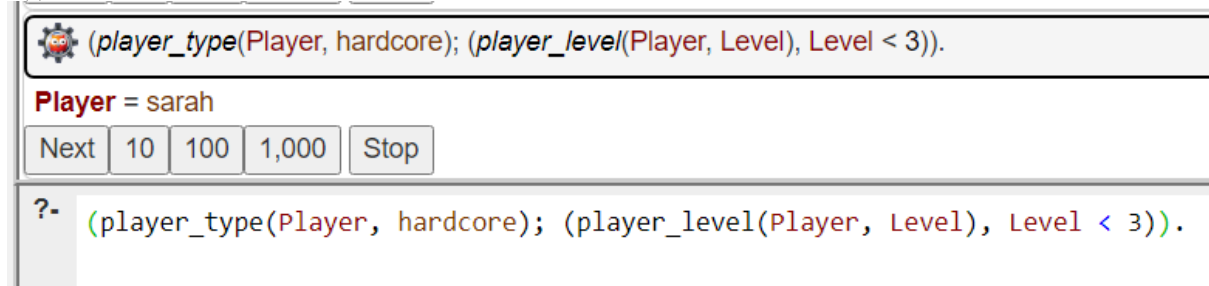
- "Какие игроки имеют тип игры casual и уровень больше 5?"

Запрос: ``player_type(Player, casual), player_level(Player, Level), Level > 5.``



- "Какие игроки имеют тип игры hardcore или уровень меньше 3?"

Запрос: ``(player_type(Player, hardcore); (player_level(Player, Level), Level < 3)).``



Запросы, использующие переменные для поиска объектов с определенными характеристиками:

- "Какие игроки имеют уровень больше 10?"

Запрос: ``player_level(Player, Level), Level > 10.``

`(player_level(Player, Level), Level < 3).`

Level = 2,
Player = mark

Next 10 100 1,000 Stop

?- `(player_level(Player, Level), Level < 3).`

- "Какие игроки имеют тип игры casual и уровень больше 5?"

Запрос: `player_type(Player, casual), player_level(Player, Level), Level > 5.`

`player_type(Player, casual), player_level(Player, Level), Level > 5.`

Level = 7,
Player = lisa

Next 10 100 1,000 Stop

?- `player_type(Player, casual), player_level(Player, Level), Level > 5.`

"Какие игроки имеют тот же уровень, что и игрок John?"

Запрос: `same_level(Player, john).`

`same_level(Player, john).`

Player = jessica

Next 10 100 1,000 Stop

?- `same_level(Player, john).`

"Какие игроки имеют тот же тип игры, что и игрок Sarah?"

Запрос: `same_type(Player, sarah).`

`same_type(Player, sarah).`

Player = alex


Next 10 100 1,000 Stop

?- `same_type(Player, sarah).`

Запросы, которые требуют выполнения правил для получения результата:

- "Какие игроки являются casual игроками?"

Запрос: `casual_player(Player).`

 `casual_player(Player).`

Player = john

Next

10

100

1,000

Stop

?-

casual_player(**Player**).

- "Какие игроки имеют более высокий уровень, чем игрок John?"
Запрос: `'higher_level(Player, john).'`

 `higher_level(Player, john).`

Player = sarah

Next

10

100

1,000

Stop

?-

higher_level(**Player**, john).

4. Вывод:

Я нахожу программирование очень интересным, все очень логично, а Пролог — простой и эффективный инструмент, я усвоил синтаксис и понимаю, как его использовать. Я научился создавать базу знаний с помощью Пролога.