Федеральное государственное автономное образовательноеучреждение

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 аргументам, 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:

```
% Список игроков
player(john).     % Джон
player(sarah).    % Сара
player(michael).  % Майкл
player(lisa).     % Лиза
player(david).    % Дэвид
player(emily).    % Эмили
player(james).    % Джеймс
player(olivia).   % Оливия
player(alex).     % Алекс
player(sophia).   % София
player(chris).    % Крис
player(ava).      % Ава
player(ryan).     % Райан
player(natalie).  % Натали
player(ethan).    % Итан
player(lily).     % Лили
player(ian).      % Иан
player(hannah).   % Ханна
player(owen).     % Оуэн

% Список ролей
role(john, village).     % Джон - житель деревни
role(sarah, werewolf).   % Сара - оборотень
role(michael, village).  % Майкл - житель деревни
role(lisa, seer).        % Лиза - предсказатель
role(david, village).    % Дэвид - житель деревни
role(emily, werewolf).   % Эмили - оборотень
role(james, village).    % Джеймс - житель деревни
role(olivia, hunter).    % Оливия - охотник
role(alex, village).     % Алекс - житель деревни
role(sophia, witch).     % София - ведьма
role(chris, village).    % Крис - житель деревни
role(ava, guardian).     % Ава - страж
role(ryan, werewolf).    % Райан - оборотень
role(natalie, village).  % Натали - житель деревни
role(ethan, village).    % Итан - житель деревни
role(lily, fool).        % Лили - дурак
role(ian, village).      % Иан - житель деревни
role(hannah, werewolf).  % Ханна - оборотень
role(owen, village).     % Оуэн - житель деревни


% ОТНОШЕНИЯ
```

```
% Отношение "является оборотнем"
is_werewolf(Player) :- role(Player, werewolf).  % является оборотнем

% Отношение "является жителем деревни"
is_villager(Player) :- role(Player, village).  % является жителем деревни

% Отношение "является предсказателем"
is_seer(Player) :- role(Player, seer).  % является предсказателем

% Отношение "является охотником"
is_hunter(Player) :- role(Player, hunter).  % является охотником

% Отношение "является ведьмой"
is_witch(Player) :- role(Player, witch).  % является ведьмой

% Отношение "является стражем"
is_guardian(Player) :- role(Player, guardian).  % является стражем

% Отношение "является дураком"
is_fool(Player) :- role(Player, fool).  % является дураком

% Отношение "являются товарищами"
is_teammate(Player1, Player2) :-
    role(Player1, Role),
    role(Player2, Role),
    Player1 \= Player2.  % являются товарищами

% Отношение "являются врагами"
is_enemy(Player1, Player2) :-
    role(Player1, Role1),
role(Player2, Role2),
Role1 = Role2. % являются врагами
```

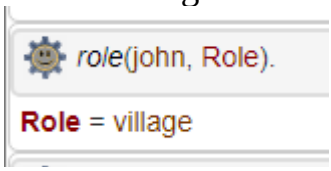
Правилы:

```
% Правило "игрок убит оборотнями"
killed_by_werewolf(Player) :-
player(Player),
is_werewolf(Werewolf),
is_enemy(Werewolf, Player). % игрок убит оборотнями

% Правило "игрок виден предсказателем как оборотень"
seen_as_werewolf(Player) :-
player(Player),
is_seer(Seer),
is_werewolf(Werewolf),
is_enemy(Seer, Werewolf). % игрок виден предсказателем как оборотень

% Правило "игрок жив"
alive(Player) :-
player(Player),
\+ killed_by_werewolf(Player). % игрок жив
```

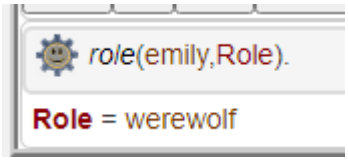
```
% Правило "игрок побеждает"
victory(Player) :-
is_villager(Player),
\+ killed_by_werewolf(Player),
\+ seen_as_werewolf(Player). % игрок побеждает
```

3. Пример:

Простые запросы к базе знаний для поиска факто:
?- role(john, Role).
Role = village

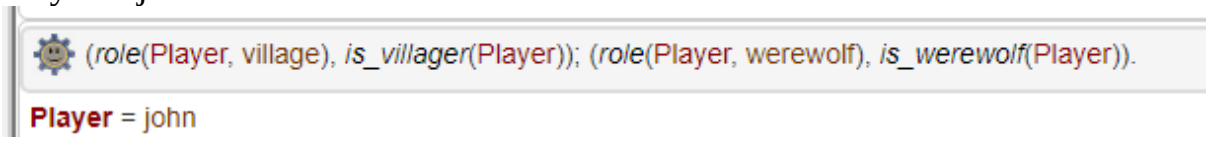


?- role(emily,Role).
Role = werewolf

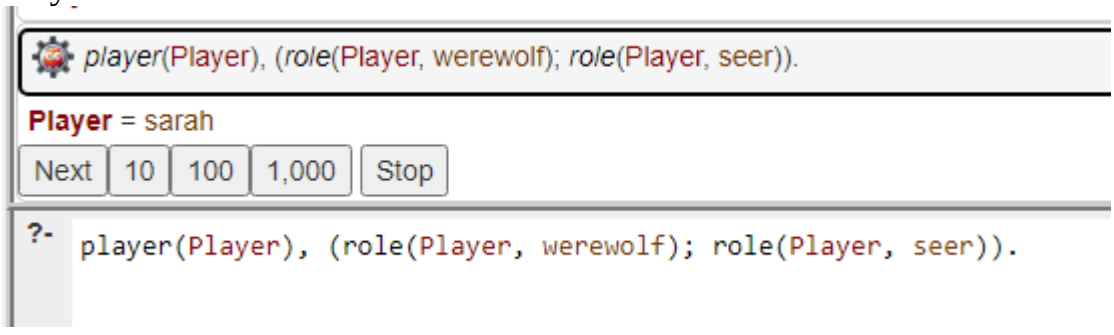


Запросы, использующие логические операторы (и, или, не) для формулирования сложных условий:

?- (role(Player, village), is_villager(Player)); (role(Player, werewolf), is_werewolf(Player)).
Player = john



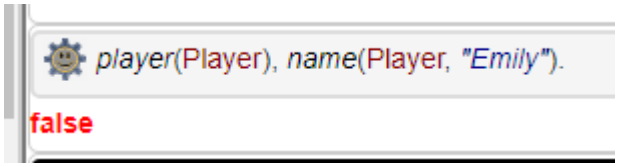
?- player(Player), (role(Player, werewolf); role(Player, seer)).
Player = sarah



Запросы, использующие переменные для поиска объектов с определенными характеристиками:

?- role(Player, village).
Player = john

?- player(Player), name(Player, "Emily").
false



?- player(Player), name(Player, "emily").
Player = emily.

Запросы, которые требуют выполнения правил для получения результата:
?- killed_by_werewolf(Player).
Player = sarah

?- victory(lisa).
False

4. Вывод:
Я нахожу программирование очень интересным, все очень логично, а Пролог — простой и эффективный инструмент, я усвоил синтаксис и понимаю, как его использовать.