
КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7

« Недетерминированное программирование »

по курсу: ФУНКЦИОНАЛЬНОЕ И ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2023

Цель: Ознакомиться с недетерминированным программированием
в языке Prolog.

Задание на лабораторную работу:

Дана карта автомобильных дорог, соединяющая N населенных пунктов, для каждой дороги задано расстояние и максимальная скорость. Построить путь из пункта А в пункт В по критерию: а] минимального времени; б] минимального расстояния.

Листинг программы:

```
% ?- min_time_path(a, b, Path, Time).
% ?- min_time_path(a, e, Path, Time).
% ?- min_distance_path(a, b, Path, Distance).
% ?- min_distance_path(a, e, Path, Distance).

:- dynamic(road/4).

% Примеры данных: дороги (откуда, куда, расстояние, максимальная скорость)
road(a, b, 100, 60).
road(a, c, 50, 70).
road(b, d, 80, 80).
road(c, d, 60, 50).
road(b, e, 70, 40).
road(d, e, 30, 70).

% Время на прохождение дороги
travel_time(From, To, Time) :-
    road(From, To, Distance, MaxSpeed),
    Time is Distance / MaxSpeed.

% Поиск кратчайшего пути по времени
shortest_time_path(From, To, Path, Time) :-
    shortest_time_path(From, To, [From], RevPath, Time),
    reverse(RevPath, Path).

shortest_time_path(From, To, Visited, [To | Visited], Time) :-
    travel_time(From, To, Time),
    \+ member(To, Visited).
shortest_time_path(From, To, Visited, Path, Time) :-
```

```
travel_time(From, Intermediate, T1),
\+ member(Intermediate, Visited),
shortest_time_path(Intermediate, To, [Intermediate | Visited], Path, T2),
Time is T1 + T2.
```

% Поиск кратчайшего пути по расстоянию

```
shortest_distance_path(From, To, Path, Distance) :-
    shortest_distance_path(From, To, [From], RevPath, Distance),
    reverse(RevPath, Path).
```

```
shortest_distance_path(From, To, Visited, [To | Visited], Distance) :-
    road(From, To, Distance, _),
    \+ member(To, Visited).
```

```
shortest_distance_path(From, To, Visited, Path, Distance) :-
    road(From, Intermediate, D1, _),
    \+ member(Intermediate, Visited),
    shortest_distance_path(Intermediate, To, [Intermediate | Visited], Path, D2),
    Distance is D1 + D2.
```

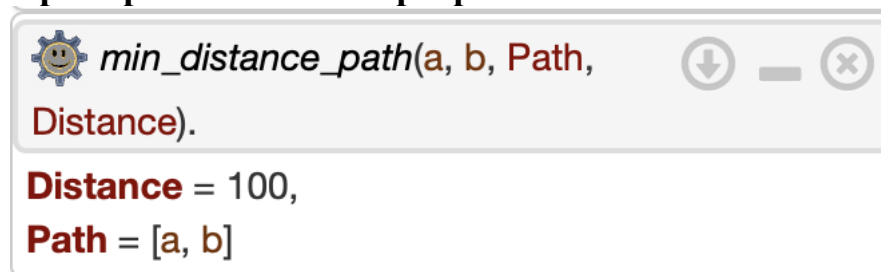
% Запрос для поиска пути с минимальным временем

```
min_time_path(From, To, Path, Time) :-
    setof((T, P), shortest_time_path(From, To, P, T), [(Time, Path) | _]).
```

% Запрос для поиска пути с минимальным расстоянием

```
min_distance_path(From, To, Path, Distance) :-
    setof((D, P), shortest_distance_path(From, To, P, D), [(Distance, Path) | _]).
```

Пример выполнения программы:



Вывод: В ходе лабораторной работы ознакомился с недетерминированным программированием в языке PROLOG