



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине:

Введение в искусственный интеллект

Студент	Абидоков Рашид Ширамбиевич
Группа	РК6-11М
Вариант	56
Тема лабораторной работы	Программирование на языке prolog

Студент	_____	<u>Абидоков Р. Ш.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Преподаватель	_____	<u>Федорук В. Г.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Оценка _____

Москва, 2020 г.

Оглавление

Задание на лабораторную работу	3
Программная реализация.....	3
Результат решения задачи	7

Задание на лабораторную работу

Ферзь находится на поле A7 шахматной доски. Необходимо найти последовательность из четырех ходов, обеспечивающую прохождение ферзем следующих девяти полей: A6, A7, A8, B6, B7, B8, C6, C7, C8.

Программная реализация

Все возможные клетки доски заданы в виде 64 фактов вида $\text{pos}([X_i, Y_i])$.

Принцип работы программы:

- 1) Непосредственно из терминала вызывается предикат $\text{moves}(\text{Pos}, \text{Points}, N, \text{Moves})$ с заданными аргументами Pos – текущая позиция, Points – требуемые клетки, N – количество ходов и незадаанным аргументом Moves – ходы, которые нужно совершить, являющийся оберткой над moves_sub (см. Рис 1 в последнем разделе)
- 2) Вызывается предикат $\text{moves_sub}(\text{Pos}, \text{Points}, N, \text{Moves})$
- 3) В случае, если оставшееся количество ходов $N \geq 0$ и Point и Moves – пустые списки, предикат удовлетворен
- 4) Иначе, если $N \geq 0$, находятся возможные из данной точки ходы и для каждого:
 - 1) Находятся клетки, которые будут пройдены
 - 2) Из списка требуемых клеток Points удаляются те, которые были пройдены
 - 3) Найденный ход становится головой списка Moves
 - 4) Рекурсивно вызывается предикат $\text{moves_sub}/4$, которому передается новая позиция, модифицированный список требуемых точек и уменьшенное на единицу количество ходов, а также незадаанный хвост списка Moves, который в результате унифицируется
 - 5) В случае, если удовлетворен рекурсивно вызванный предикат, текущий предикат удовлетворен

Исходный код программы на языке prolog вместе с комментариями приведен в Листинге 1.

Листинг 1.

```
67  % Пары буква-цифра для последующей замены
68  pair(a, 1).
69  pair(b, 2).
70  pair(c, 3).
71  pair(d, 4).
72  pair(e, 5).
73  pair(f, 6).
74  pair(g, 7).
75  pair(h, 8).
76
77  % Клетка левее
78  left_pos([CurrH|CurrT], [ResH,ResT]):-
79      ResH is CurrH - 1, ResT is CurrT, pos([ResH, ResT]),!.
80
81  % Клетка правее
82  right_pos([CurrH|CurrT], [ResH,ResT]):-
83      ResH is CurrH + 1, ResT is CurrT, pos([ResH, ResT]),!.
84
85  % Клетка ниже
86  lower_pos([CurrH|CurrT], [ResH,ResT]):-
87      ResH is CurrH, ResT is CurrT - 1, pos([ResH, ResT]),!.
88
89  % Клетка выше
90  higher_pos([CurrH|CurrT], [ResH,ResT]):-
91      ResH is CurrH, ResT is CurrT + 1, pos([ResH, ResT]),!.
```

```

93  % Возможный ход
94  ♡ possible_move([PosH|PosT], [MoveH|MoveT]):-
95      % По горизонтали
96      pos([MoveH|MoveT]), MoveH = PosH, not(MoveT = PosT);
97      % По вертикали
98      pos([MoveH|MoveT]), MoveT = PosT, not(MoveH = PosH);
99      % Диагональ влево вниз - вправо вверх
100     pos([MoveH|MoveT]), Y is MoveH - PosH, X is MoveT - PosT,
101     X = Y, not(MoveH = PosH);
102     % Диагональ влево вверх - вправо вниз
103     pos([MoveH|MoveT]), Y is MoveH - PosH, X is PosT - MoveT,
104     X = Y, not(MoveH = PosH).
105
106  % Клетки, которые пройдем, если сделаем по горизонтали
107  hor_passed_cells(A, A, [A]):-!.
108  ♡ hor_passed_cells([FromH|FromT], [ToH|ToT], [CellsH|CellsT]):-
109      FromH < ToH, right_pos([FromH|FromT], NewFrom),
110      CellsH = [FromH|FromT], hor_passed_cells(NewFrom, [ToH|ToT], CellsT), !;
111
112      FromH > ToH, left_pos([FromH|FromT], NewFrom),
113      CellsH = [FromH|FromT], hor_passed_cells(NewFrom, [ToH|ToT], CellsT), !.
114
115  % Клетки, которые пройдем, если сделаем по вертикали
116  ver_passed_cells(A, A, [A]):-!.
117  ♡ ver_passed_cells([FromH|FromT], [ToH|ToT], [CellsH|CellsT]):-
118      FromT = [FromTH|_], ToT = [ToTH|_],
119      FromTH < ToTH, higher_pos([FromH|FromT], NewFrom),
120      CellsH = [FromH|FromT], ver_passed_cells(NewFrom, [ToH|ToT], CellsT), !;
121
122      FromT = [FromTH|_], ToT = [ToTH|_],
123      FromTH > ToTH, lower_pos([FromH|FromT], NewFrom),
124      CellsH = [FromH|FromT],
125      ver_passed_cells(NewFrom, [ToH|ToT], CellsT), !.

```

```

127 % Клетки, которые пройдем, если сделаем по диагонали
128 diag_passed_cells(A, A, [A]):-!.
129 diag_passed_cells([FromH|FromT], [ToH|ToT], [CellsH|CellsT]):-
130     FromT = [FromTH|_], ToT = [ToTH|_],
131     FromH < ToH, FromTH < ToTH,
132     higher_pos([FromH|FromT], TempFrom), right_pos(TempFrom, NewFrom),
133     CellsH = [FromH|FromT], diag_passed_cells(NewFrom, [ToH|ToT], CellsT), !;
134
135     FromT = [FromTH|_], ToT = [ToTH|_],
136     FromH < ToH, FromTH > ToTH,
137     lower_pos([FromH|FromT], TempFrom), right_pos(TempFrom, NewFrom),
138     %print('test'),
139     CellsH = [FromH|FromT], diag_passed_cells(NewFrom, [ToH|ToT], CellsT), !;
140
141     FromT = [FromTH|_], ToT = [ToTH|_],
142     FromH > ToH, FromTH < ToTH,
143     higher_pos([FromH|FromT], TempFrom), left_pos(TempFrom, NewFrom),
144     CellsH = [FromH|FromT], diag_passed_cells(NewFrom, [ToH|ToT], CellsT), !;
145
146     FromT = [FromTH|_], ToT = [ToTH|_],
147     FromH > ToH, FromTH > ToTH,
148     lower_pos([FromH|FromT], TempFrom), left_pos(TempFrom, NewFrom),
149     CellsH = [FromH|FromT], diag_passed_cells(NewFrom, [ToH|ToT], CellsT), !.
150
151 % Клетки, которые пройдем, если сделаем ход
152 passed_cells([FromH|FromT], [ToH|ToT], Cells):-
153     % По вертикали
154     FromH = ToH, not(FromT = ToT), ver_passed_cells([FromH|FromT], [ToH|ToT], Cells),!;
155     % По горизонтали
156     FromT = [FromTH|_], ToT = [ToTH|_], FromTH = ToTH, not(FromH = ToH),
157     hor_passed_cells([FromH|FromT], [ToH|ToT], Cells),!;
158     % По диагонали
159     diag_passed_cells([FromH|FromT], [ToH|ToT], Cells).
160
161 % Удаляет из листа элемент, если он там есть
162 remove_element([], _, []):-!.
163 remove_element([WhereH|WhereT], Elem, Res):-
164     %print(WhereH), nl(),
165     %print(Elem), nl(),
166     Res = [ResH|ResT], not(WhereH = Elem), ResH = WhereH, remove_element(WhereT, Elem, ResT),!;
167     WhereH = Elem, remove_element(WhereT, Elem, Res).
168
169 % Удаляет из листа элементы другого листа, если они в нем есть
170 remove_elements([], _, []):-!.
171 remove_elements(A, [], A):-!.
172 remove_elements(Where, [WhatH|WhatT], Res):-
173     remove_elements(Where, WhatT, ResNew), remove_element(ResNew, WhatH, Res).
174
175
176 % Заменяем в листе цифры на буквы и наоборот. Для красоты
177 pairs([], []):-!.
178 pairs([ListH|ListT], [ResH|ResT]):-
179     ListH = [ListHH|ListHT],
180     pair(ResHH, ListHH),
181     ResH = [ResHH|ListHT],
182     pairs(ListT, ResT),!.
183

```

```

184 % Так, поехали
185 % (текущая позиция, что надо посетить, оставшееся количество ходов, сами ходы)
186 moves_sub(_, [], N, []):-
187     N >= 0, !.
188 moves_sub(Pos, Points, N, [MovesH|MovesT]):-
189     N >= 0,
190     possible_move(Pos, Move),
191     passed_cells(Pos, Move, PassCells),
192     remove_elements(Points, PassCells, PointsNew),
193     NNew is N - 1,
194     MovesH = Move,
195     moves_sub(Move, PointsNew, NNew, MovesT).
196
197 moves(Pos, Points, N, Moves):-
198     pairs([PosNum], [Pos]),
199     pairs(PointsNum, Points),
200     moves_sub(PosNum, PointsNum, N, MovesNum),
201     pairs(MovesNum, Moves).

```

Результат решения задачи

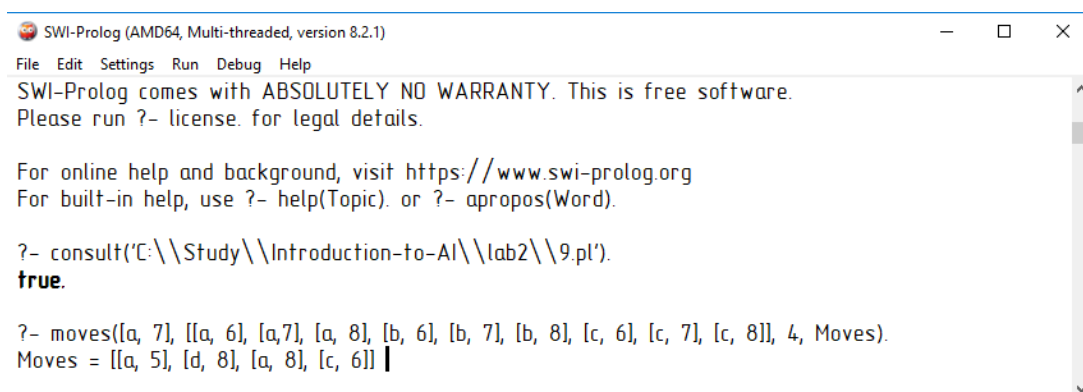


Рис. 1 Первое найденное решение

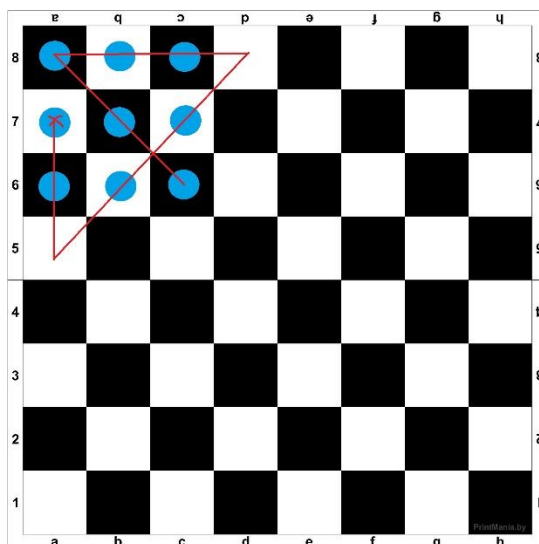


Рис. 2