

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления

Кафедра Интеллектуальных информационных технологий

Отчет по лабораторной работе №2

по дисциплине

Логические основы интеллектуальных систем

Выполнили:

И. В. Якимович

Студенты группы

121703

Проверил:

В. П. Ивашенко

Минск 2023

Тема: Логическое программирование поиска решения задачи.

Цель: Приобрести навыки логического программирования поиска решения задачи.

Задача: Требуется расставить на шахматной доске восемь ферзей так, чтобы ни один ферзь не находился под боем другого ферзя.

Дополнительные теоретические сведения:

Грамматика языка PROLOG.

$\langle \text{ПРОЛОГ-предложение} \rangle ::= \langle \text{правило} \rangle \mid \langle \text{факт} \rangle \mid \langle \text{запрос} \rangle$

$\langle \text{правило} \rangle ::= \langle \text{заголовок} \rangle \text{ :- } \langle \text{тело} \rangle$

$\langle \text{факт} \rangle ::= \langle \text{заголовок} \rangle \text{ '}'$

$\langle \text{запрос} \rangle ::= \langle \text{тело} \rangle \text{ '}'$

$\langle \text{тело} \rangle ::= \langle \text{цель} \rangle \text{ ',' } \langle \text{цель} \rangle \text{ '/' }$

$\langle \text{заголовок} \rangle ::= \langle \text{предикат} \rangle$

$\langle \text{цель} \rangle ::= \langle \text{предикат} \rangle \mid \langle \text{выражение} \rangle$

$\langle \text{предикат} \rangle ::= \langle \text{имя} \rangle \text{ / } (\langle \text{терм} \rangle \text{ ',' } \langle \text{терм} \rangle \text{ / }) \text{ '}'$

$\langle \text{терм} \rangle ::= \langle \text{атом} \rangle \mid \langle \text{предикат} \rangle \mid \langle \text{список} \rangle$

$\langle \text{атом} \rangle ::= \langle \text{переменная} \rangle \mid \langle \text{число} \rangle \mid \langle \text{строка} \rangle \mid \langle \text{имя} \rangle$

$\langle \text{список} \rangle ::= \langle \text{список с заголовком} \rangle \mid \langle \text{простой список} \rangle$

$\langle \text{список с заголовком} \rangle ::= [\langle \text{терм} \rangle \text{ ',' } \langle \text{терм} \rangle \text{ / } \mid \langle \text{терм} \rangle] \text{ '}'$

$\langle \text{простой список} \rangle ::= [\langle \text{терм} \rangle \text{ ',' } \langle \text{терм} \rangle \text{ / }] \text{ '}'$

$\langle \text{выражение} \rangle ::= \langle \text{терм} \rangle \text{ / } \langle \text{оператор} \rangle \langle \text{терм} \rangle \text{ / }$

$\langle \text{оператор} \rangle ::= \text{'is' } \mid \text{'=' } \mid \text{'\ = ' } \mid \text{'>= ' } \mid \text{'<= ' } \mid \text{'\ = \ = ' }$

Описание лабораторной работы:

В рамках лабораторной работы стандартными средствами языка PROLOG был реализован алгоритм, позволяющий найти расстановки на шахматной доске восьми ферзей так, чтобы ни один ферзь не находился под боем другого ферзя. Суть алгоритма заключается в сведении логической задачи к обходу дерева решения данной задачи.

Для решения данной задачи был использован ряд **встроенных правил**:

- **write(X)** - предикат, который выводит значение терма X на экран.
- **select(X, L, M)** - является истинным, если список M получается в результате удаления первого вхождения терма X из списка L.

Логические связки:

- ; - ИЛИ
- ,-И

Схемы использованных алгоритмов:

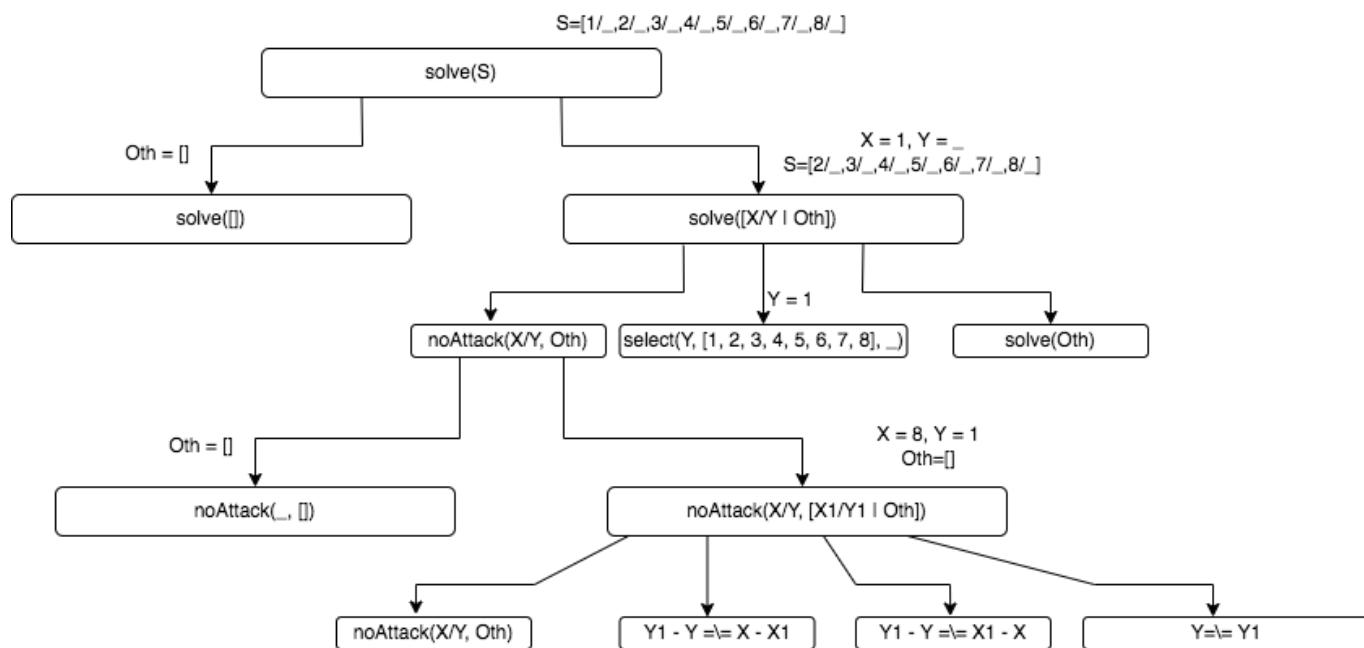


Рис.1 - Связанный фрагмент дерева реализованного алгоритма поиска

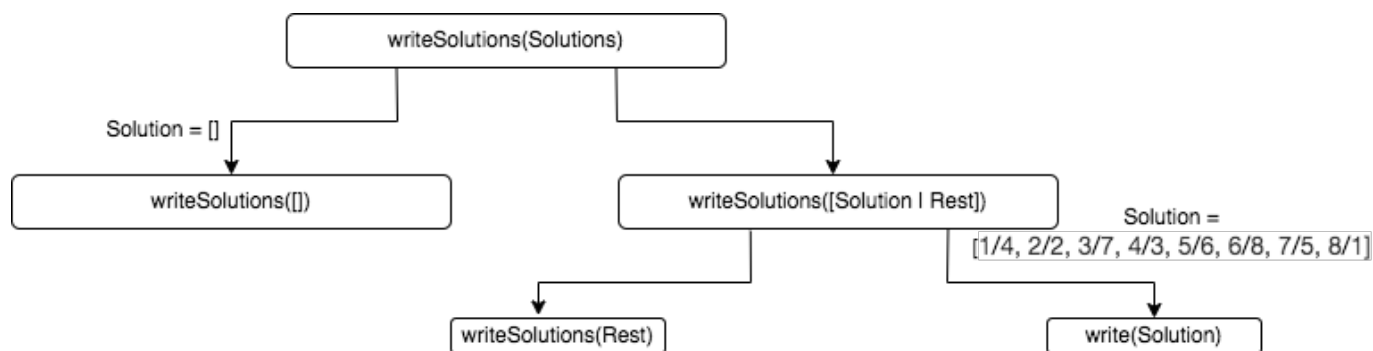


Рис.2 - Связанный фрагмент дерева реализованного алгоритма вывода

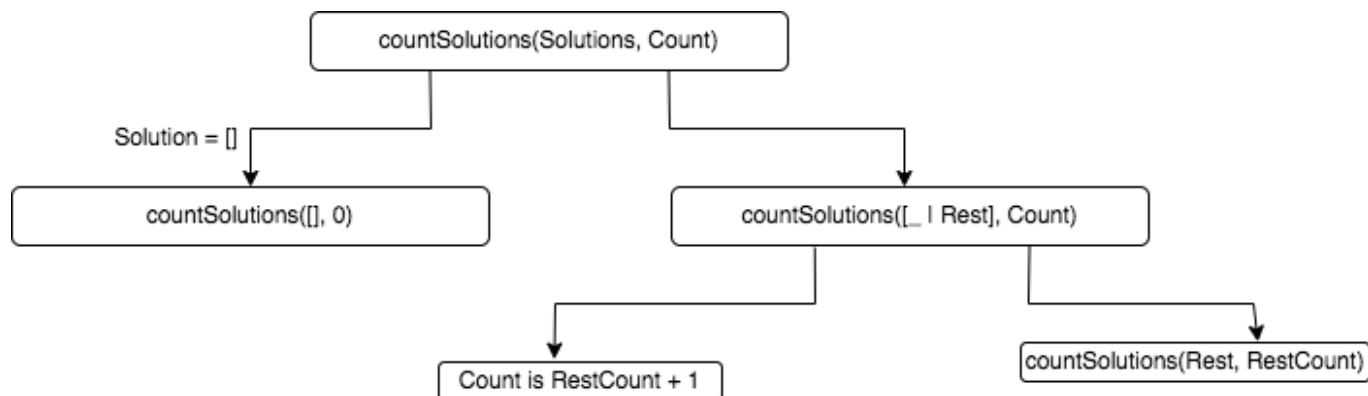


Рис.3 - Связанный фрагмент дерева реализованного алгоритма подсчета количество решении

Листинг программы:

```

%Лабораторная работа №3 по дисциплине ЛОИС
%Выполнена студентом группы 121703 БГУИР Якимович Илья Викторович
%Программа содержит описание предикатов, позволяющих расставить на шахматной доске 8 ферзей
%так, чтобы ни один ферзь не находился под боем другого ферзя
%29.05.2023

getSolution(S) :-
    findall(S, solve(S), Solutions),
    writeSolutions(Solutions),
    countSolutions(Solutions, Count),
    write('Number of solutions: '), write(Count), nl.

writeSolutions([]).
writeSolutions([Solution | Rest]) :-
    write(Solution), nl,
    writeSolutions(Rest).

solve([]).
solve([X/Y | 0th]) :-
    solve(0th),
    select(Y, [1, 2, 3, 4, 5, 6, 7, 8], _),
    noAttack(X/Y, 0th).

noAttack(_, []).
noAttack(X/Y, [X1/Y1 | 0th]) :-
    Y \= Y1,
    Y1 - Y \= X1 - X,
    Y1 - Y \= X - X1,
    noAttack(X/Y, 0th).

countSolutions([], 0).
countSolutions([_ | Rest], Count) :-
    countSolutions(Rest, RestCount),
    Count is RestCount + 1.

```

Рис.4 - Листинг программы

Примеры выполнения

```
[1/5,2/1,3/8,4/4,5/2,6/7,7/3,8/6]
[1/4,2/1,3/5,4/8,5/2,6/7,7/3,8/6]
[1/5,2/2,3/8,4/1,5/4,6/7,7/3,8/6]
[1/3,2/7,3/2,4/8,5/5,6/1,7/4,8/6]
[1/3,2/1,3/7,4/5,5/8,6/2,7/4,8/6]
[1/8,2/2,3/5,4/3,5/1,6/7,7/4,8/6]
[1/3,2/5,3/2,4/8,5/1,6/7,7/4,8/6]
[1/3,2/5,3/7,4/1,5/4,6/2,7/8,8/6]
[1/5,2/2,3/4,4/6,5/8,6/3,7/1,8/7]
[1/6,2/3,3/5,4/8,5/1,6/4,7/2,8/7]
[1/5,2/8,3/4,4/1,5/3,6/6,7/2,8/7]
[1/4,2/2,3/5,4/8,5/6,6/1,7/3,8/7]
[1/4,2/6,3/1,4/5,5/2,6/8,7/3,8/7]
[1/6,2/3,3/1,4/8,5/5,6/2,7/4,8/7]
[1/5,2/3,3/1,4/6,5/8,6/2,7/4,8/7]
[1/4,2/2,3/8,4/6,5/1,6/3,7/5,8/7]
[1/6,2/3,3/5,4/7,5/1,6/4,7/2,8/8]
[1/6,2/4,3/7,4/1,5/3,6/5,7/2,8/8]
[1/4,2/7,3/5,4/2,5/6,6/1,7/3,8/8]
[1/5,2/7,3/2,4/6,5/3,6/1,7/4,8/8]
Number of solutions: 92
```

Рис.5 - Пример выполнения алгоритма с верным запросом без начального состояния для 8 ферзеи

```
?- getSolution([1/1,2/_ ,3/_ ,4/_ ,5/_ ,6/_ ,7/_ ,8/_ ]).
[1/1,2/7,3/4,4/6,5/8,6/2,7/5,8/3]
[1/1,2/7,3/5,4/8,5/2,6/4,7/6,8/3]
[1/1,2/5,3/8,4/6,5/3,6/7,7/2,8/4]
[1/1,2/6,3/8,4/3,5/7,6/4,7/2,8/5]
Number of solutions: 4
```

Рис.6 - Пример выполнения алгоритма с верным запросом с начальным состоянием для 1 ферзя

```
?- getSolution([1/1,2/7,3/_ ,4/_ ,5/_ ,6/_ ,7/_ ,8/_ ]).
[1/1,2/7,3/4,4/6,5/8,6/2,7/5,8/3]
[1/1,2/7,3/5,4/8,5/2,6/4,7/6,8/3]
Number of solutions: 2
```

Рис.7 - Пример выполнения алгоритма с верным запросом с начальным состоянием для 2 ферзеи

Вывод: В ходе выполнения лабораторной работы были приобретены навыки логического программирования поиска решения задачи; была разработана программа, позволяющая найти расстановки на шахматной доске восьми ферзей так, чтобы ни один ферзь не находился под боем другого ферзя.

Список используемых источников

1. Логические основы интеллектуальных систем. Практикум : учеб.- метод. пособие / В. В. Голенков [и др.]. – Минск : БГУИР, 2011. – 70 с. : ил. ISBN 978-985-488-487-5.
2. SWI Prolog [Электронный ресурс]. — Режим доступа: <https://www.swi-prolog.org/>.