

Министерство образования Республики Беларусь

Учреждение образования
“Белорусский государственный университет
информатики и радиоэлектроники”

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине «Логические основы интеллектуальных систем»

на тему

«Логическое программирование»

Вариант 9

Выполнил студент гр. 121702

Заломов Р.А.

Проверил

Ивашенко В. П.

Минск 2023

Цель: приобрести навыки логического программирования поиска решения задачи.

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

Теоретические сведения

Грамматика языка PROLOG.

<ПРОЛОГ-предложение> ::= <правило> | <факт> | <запрос>

<правило> ::= <заголовок> ‘:-’<тело>

<факт> ::= <заголовок> ‘.’

<запрос> ::= <тело>‘.’

<тело> ::= <цель> /’,’<цель>/’,’

<заголовок> ::= <предикат>

<цель> ::= <предикат> | <выражение>

<предикат> ::= <имя> / ‘(’<терм> /’,’<терм>/ ‘)’/

<терм> ::= <атом> | <предикат> | <список>

<атом> ::= <переменная> | <число> | <строка> | <имя>

<список> ::= <список с заголовком> | <простой список>

<список с заголовком> ::= ‘[’<терм> /’,’<терм>/’,’<терм>’]

<простой список> ::= ‘[’<терм> /’,’<терм>/’,’[’]’

<выражение> ::= <терм> /<оператор><терм>/

<оператор> ::= ‘is’ | ‘=’ | ‘==’ | ‘\=’ | ‘>=’ | ‘<=’ | ‘=\=’ |

Описание программы и алгоритма

В рамках лабораторной работы на языке Prolog была реализована программа для поиска решения задачи о восьми ферзях. Данная версия программы может работать с более общим случаем, а именно N ферзей на шахматной доске NxN, где N – целое число. Так как для решения задачи в любом случае требуется занять все столбцы доски, то задача сводится к поиску конкретной позиции на каждом столбе для ферзей.

Листинг программы:

```
1  % Лабораторная работа №2 по дисциплине ЛОИС
2  % Выполнена студентом группы 121702 БГУИР Заломовым Романом Андреевичем
3  % 18.05.22
4  % Файл программы осуществляет решение задачи N ферзей, размещённых на шахматной доске NxN
5  % таким образом, что ни один ферзь не может бить любого другого
6  % 04.06.23 ver 1.1 Изменены имена термов
7
8  % Учитывая то, что все колонки шахматной доски должны быть заняты,
9  % суть задачи осуществляется в поиске клетки на соответствующей колонке.
10
11
12 % Главный предикат, отвечающий за решение задачи
13 queens(N, S) :-
14     length(S, N),
15     numlist(1, N, R),
16     permutation(R, S),
17     safe_all(S).
18
19 % Факт, указывающий на то, что доска 0x0 всегда имеет решение.
20 safe_all([]).
21
22 % Предикат, определяющий условия правильности решения
23 safe_all([Q|A]) :-
24     safe(A, 1, Q),
25     safe_all(A).
26
27 % Факт, указывающий на то, что на пустой доске никто никого бить не может
28 safe([], _, _).
29
30 % Предикат, определяющий условия безопасности какого-либо ферзя относительно остальных
31 safe([D|A], C, Q) :-
32     Q \= D,
33     Q + C \= D,
34     Q - C \= D,
35     N is C + 1,
36     safe(A, N, Q).
37
```

Рис 1. Листинг программы

Дерево вывода для данной программы (для случая, если требуется расставить 4 королей на доске 4×4):

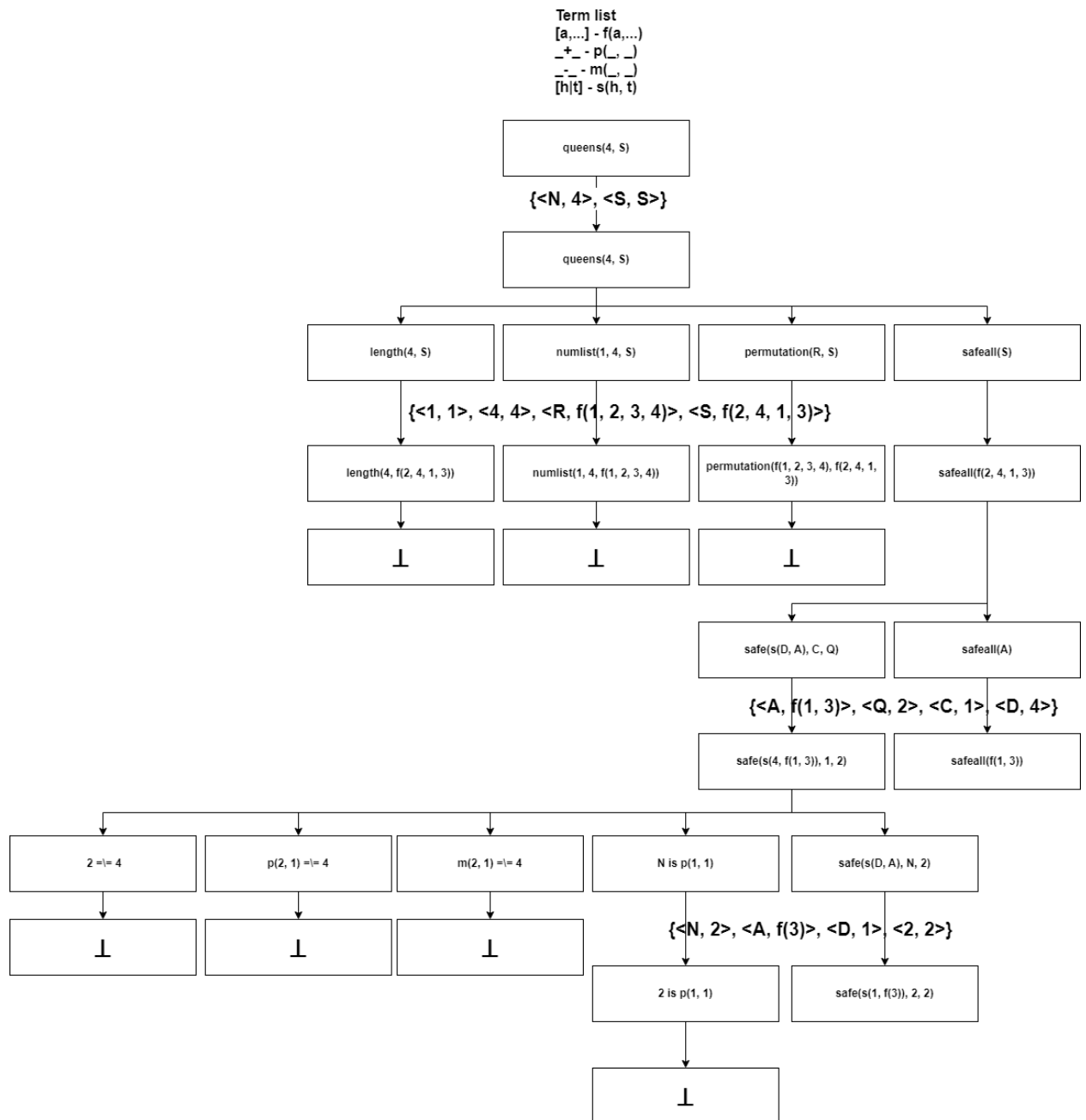


Рис 2. Дерево вывода программы

Пример выполнения программы:

```

S = [1, 5, 8, 6, 3, 7, 2, 4] ;
S = [1, 6, 8, 3, 7, 4, 2, 5] ;
S = [1, 7, 4, 6, 8, 2, 5, 3] ;
S = [1, 7, 5, 8, 2, 4, 6, 3] ;
S = [2, 4, 6, 8, 3, 1, 7, 5] ;
S = [2, 5, 7, 1, 3, 8, 6, 4] ;
S = [2, 5, 7, 4, 1, 8, 6, 3]

```

Рис 3. Пример выполнения программы

Встроенные предикаты

`length(?List, ?Length)` - истинно, если `Length` представляет собой количество элементов в `List`. Предикат может быть использован для нахождения длины списка или создания списка (содержащего переменные) длины `Length`.

Предикат является недетерминированным и производит списки возрастающей длины, если `List` - частичный список, а `Length` - переменная.

`numlist(+Low, +High, -List)` – заполняет список `List` целыми числами от `Low` до `High`.

`permutation(?Xs, ?Ys)` - истинно, если `Xs` является перестановкой `Ys`.

Предикат может решить задачу `Ys` по `Xs` или `Xs` по `Ys`, или даже перечислить `Xs` и `Ys` вместе. Предикат предназначен в первую очередь для генерации перестановок.

Формализация на языке логики предикатов первого порядка:

<code>queens</code>	<code>Q</code>
<code>length</code>	<code>L</code>
<code>safeall</code>	<code>A</code>
<code>safe</code>	<code>S</code>
<code>_=/=_</code>	<code>N</code>

Табл. 1 – Таблица предикатов

<code>numlist</code>	<code>n</code>
<code>permutation</code>	<code>p</code>
<code>[_ _]</code>	<code>h</code>
<code>_+_</code>	<code>f</code>
<code>_=_</code>	<code>m</code>
<code>1</code>	<code>a</code>

Табл. 2 – Таблица термов

Описание предикатов

1. Правило: queens(N, Solution) :-

length(Solution, N),
numlist(1, N, Rows),
permutation(Rows, Solution),
safe_all(Solution).

Результат: $(\neg x(\neg y((L(y) \wedge A(p(n(x), y))) \rightarrow Q(x, y))))$

2. Правило: safe_all([Queen|Queens]) :-

safe(Queens, 1, Queen),
safe_all(Queens).

Результат: $(\neg y((S(h(y), a) \wedge A(h(y))) \rightarrow A(y)))$

3. Правило: safe([OtherQueen|Queens], Offset, Queen) :-

Queen \neq OtherQueen,
Queen + Offset \neq OtherQueen,
Queen - Offset \neq OtherQueen,
NewOffset is Offset + 1,
safe(Queens, NewOffset, Queen).

Результат: $(\neg x(\neg y(\neg z(((N(z, h(x)) \wedge N(f(z, y), h(x))) \wedge N(m(z, y), h(x))) \wedge S(x, f(y, a), z)) \rightarrow S(x, y, z))))$

Вывод

В процессе выполнения лабораторной работы были приобретены навыки логического программирования на примере языка Prolog. На данном языке была реализована программа поиска решение задачи восьми ферзей.

Список использованных 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 Documentation [Электронный ресурс]. — Режим доступа: <https://www.swi-prolog.org> — Дата доступа: 15.05.2023