Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления
Кафедра Интеллектуальных информационных технологий

ОТЧЁТ

по дисциплине «Логические основы интеллектуальных систем»
Лабораторная работа №2
Вариант 1

Выполнил: Самута Д. В.
гр. 221703

Проверил: Ивашенко В. П.

Минск 2024

Тема: Логическое программирование поиска решения задачи.

Цель: Приобрести навыки логического программирования поиска решения задачи.

Постановка задачи: Необходимо реализовать программу решающую поставленную задачу. В результате выполнения программы должен выводиться результат работы, описывающий решение задачи.

Условие задачи:
Два береги реки. На одном берегу есть 3 миссионера и 3 людоеда, требуется с помощью лодки вмещающей не более 2 человек, переправить всех на другой берег. Число присутствующих миссионеров на берегу и в лодке должно быть всегда не меньше числа людоедов.

Описание программы и алгоритма:
Код программы:

```
check(CL, ML, CR, MR) :-
   ML >= 0, CL >= 0, MR >= 0, CR >= 0, % Количество людей не может быть
отрицательным
   (ML >= CL; ML = 0), % На левом берегу должно быть больше миссионеров, чем
людоедов, либо не должно быть миссионеров вовсе
   (MR >= CR; MR = 0). % На правом берегу должно быть больше миссионеров, чем
людоедов, либо не должно быть миссионеров вовсе

% Возможные перемещения:
move([CL,ML,CR,MR,left],[CL,ML2,CR,MR2,right], 'Два миссионера переплывают реку слева
направо') :-
        % Два миссионера переплывают реку слева направо
        MR2 is MR+2,
        ML2 is ML-2,
        check(CL,ML2,CR,MR2).

move([CL,ML,CR,MR,left],[CL2,ML,CR2,MR,right], 'Два каннибала переплывают реку слева
направо') :-
        % Два каннибала переплывают реку слева направо
        CR2 is CR+2,
        CL2 is CL-2,
        check(CL2,ML,CR2,MR).

move([CL,ML,CR,MR,right],[CL,ML2,CR,MR2,left], 'Два миссионера переплывают реку
справа налево') :-
        % Два миссионера переплывают реку справа налево
        MR2 is MR-2,
        ML2 is ML+2,
        check(CL,ML2,CR,MR2).

move([CL,ML,CR,MR,right],[CL2,ML,CR2,MR,left], 'Два каннибала переплывают реку справа
налево') :-
        % Два каннибала переплывают реку справа налево
```

```
        CR2 is CR-2,
        CL2 is CL+2,
        check(CL2,ML,CR2,MR).

move([CL,ML,CR,MR,left],[CL2,ML2,CR2,MR2,right], 'Один миссионер и один каннибал переплывают реку слева направо') :-
        %  Один миссионер и один каннибал переплывают реку слева направо
        CR2 is CR+1,
        CL2 is CL-1,
        MR2 is MR+1,
        ML2 is ML-1,
        check(CL2,ML2,CR2,MR2).

move([CL,ML,CR,MR,right],[CL2,ML2,CR2,MR2,left], 'Один миссионер и один каннибал переплывают реку справа налево') :-
        %  Один миссионер и один каннибал переплывают реку справа налево
        CR2 is CR-1,
        CL2 is CL+1,
        MR2 is MR-1,
        ML2 is ML+1,
        check(CL2,ML2,CR2,MR2).

move([CL,ML,CR,MR,left],[CL,ML2,CR,MR2,right], 'Один миссионер переплывает реку слева направо') :-
        % Один миссионер переплывает реку слева направо
        MR2 is MR+1,
        ML2 is ML-1,
        check(CL,ML2,CR,MR2).

move([CL,ML,CR,MR,left],[CL2,ML,CR2,MR,right], 'Один каннибал переплывает реку слева направо') :-
        % Один каннибал переплывает реку слева направо
        CR2 is CR+1,
        CL2 is CL-1,
        check(CL2,ML,CR2,MR).

move([CL,ML,CR,MR,right],[CL,ML2,CR,MR2,left], 'Один миссионер переплывает реку справа налево') :-
        % Один миссионер переплывает реку справа налево
        MR2 is MR-1,
        ML2 is ML+1,
        check(CL,ML2,CR,MR2).

move([CL,ML,CR,MR,right],[CL2,ML,CR2,MR,left], 'Один каннибал переплывает реку справа налево') :-
        % Один каннибал переплывает реку справа налево
        CR2 is CR-1,
        CL2 is CL+1,
        check(CL2,ML,CR2,MR).
```

```
solve(FinalState, FinalState, _, Path) :-
    reverse(Path, Solution), % Переворачиваем Path, результат записываем в Solution
    print_solution(Solution). % Вывод Solution в консоль.

solve(CurrentState, FinalState, Visited, Path) :-
    move(CurrentState, NextState, Action), % переход от одного состояния к следующему
    \+ member(NextState, Visited), % проверка, что такое состояние ещё не было достигнуто,
предотвращает циклы в поиске решения.
    solve(NextState, FinalState, [NextState | Visited], [Action | Path]). % рекурсивно вызываем
для перехода в следующее состояние, добавляем NextState в Visited и Action в Path.

% Вывод решения в консоль.
print_solution([]). % Остановка вывода, когда T будет пустым списком.
print_solution([H | T]) :-
    write(H), nl, % записываем один шаг и вызываем
    print_solution(T).

% Стартовое положение миссионеров и каннибалов, запуск поиска и вывода.
start(InitialState, FinalState) :-
    solve(InitialState, FinalState, [InitialState], []). % вызываем поиск решения для текущего
начального состояния и конечного, сразу добавляя InitialState в Visited.
```

Правило проверки состояния (1) учитывает, что кол-во людей на берегах не может быть отрицательным, а также кол-во миссионеров на берегах должно быть больше чем кол-во людоедов.

Правила перемещения людей (2) позволяют менять положение миссионеров и людоедов на берегах, проверяя с помощью правила (1) возможность такого перехода.

Правило (3) позволяет вывести состояния в консоль.

Правило поиска решения (4) позволяет найти решение от начального положения людей и конечного.

Пример вывода в консоль:
Начальное положение: 3 каннибала и 3 людоеда на левом берегу.
Конечное положение: 3 каннибала и 3 людоеда на правом берегу.
Цель:
?- start([3,3,0,0,left],[0,0,3,3,right]).
Вывод в консоль:
Два каннибала переплывают реку слева направо
Один каннибал переплывает реку справа налево
Два каннибала переплывают реку слева направо
Один каннибал переплывает реку справа налево
Два миссионера переплывают реку слева направо
Один миссионер и один каннибал переплывают реку справа налево
Два миссионера переплывают реку слева направо
Один каннибал переплывает реку справа налево
Два каннибала переплывают реку слева направо

Один миссионер переплывает реку справа налево
Один миссионер и один каннибал переплывают реку слева направо

Формализация на языке логики предикатов первого порядка:

∀X (non_negative(X))
∀X (zero(X))
∀X, ∀Y greater_or_equal(X, Y)
∀X, ∀Y, ∀S sum(A, B, S)

∀CL, ∀ML, ∀CR, ∀MR ((non_negative(CL)/\non_negative(ML)/\
non_negative(CR)/\non_negative(MR))/\(greater_or_equal(ML, CL)\/ (zero(ML))/\(
greater_or_equal(MR, CR), zero(MR) -> check(CL, ML, CR, MR))

∀CL, ∀ML, ∀CR, ∀MR, ∀ML2, ∀MR2 ((sum(MR, 2, MR2)/\sum(ML2, 2, ML)/\check(CL, ML2, CR, MR))->move([CL, ML, CR, MR, left], [CL, ML2, CR, MR2, right]), a)

∀CL, ∀ML, ∀CR, ∀MR, ∀CL2, ∀CR2 ((sum(CR, 2, CR2)/\sum(CL2, 2, CL)/\check(C2L, ML, CR2, MR))->move([CL, ML, CR, MR, left], [CL2, ML, CR2, MR, right]), a)

∀CL, ∀ML, ∀CR, ∀MR, ∀CL2, ∀ML2, ∀CR2, ∀MR2 ((sum(CR, 1, CR2)/\sum(CL2, 1, CL)/\ sum(MR, 1, MR2)/\ sum(ML2, 1, ML2)/\check(C2L, ML2, CR2, MR2))->move([CL, ML, CR, MR, left], [CL2, ML2, CR2, MR2, right]), a)

∀CL, ∀ML, ∀CR, ∀MR, ∀ML2, ∀MR2 ((sum(MR, 1, MR2)/\sum(ML2, 1, ML)/\check(CL, ML2, CR, MR))->move([CL, ML, CR, MR, left], [CL, ML2, CR, MR2, right]), a)

∀CL, ∀ML, ∀CR, ∀MR, ∀CL2, ∀CR2 ((sum(CR, 1, CR2)/\sum(CL2, 1, CL)/\check(C2L, ML, CR2, MR))->move([CL, ML, CR, MR, left], [CL2, ML, CR2, MR, right]), a)

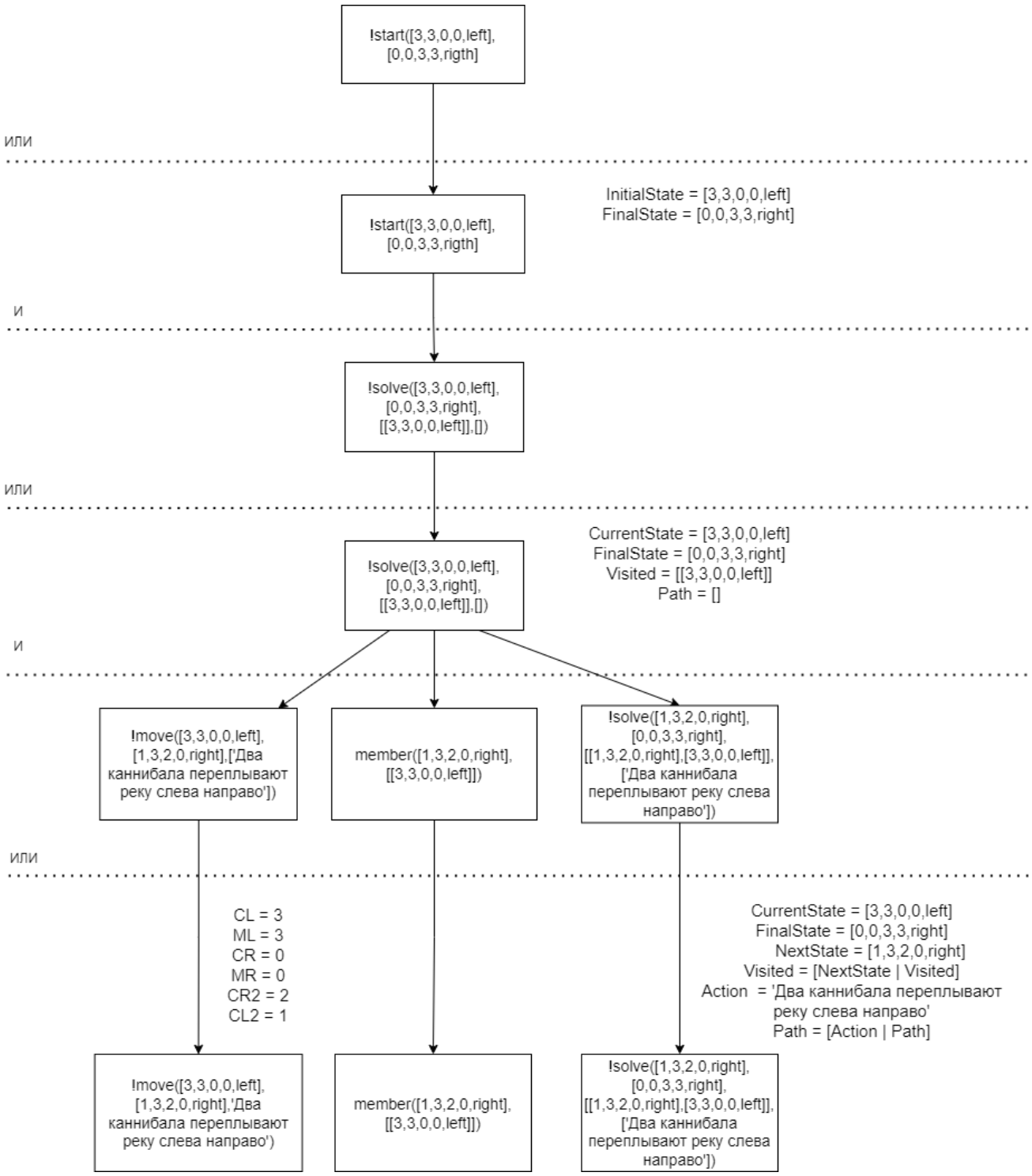
∀X, ∀Y, ∀Z, ∀W, ∀R, ∀T(move(X, Y, Z)/\ !member(Y, W)/\solve(Y, R, [Y|W], [Z,T])->solve(X, R, W, T))

∀X, ∀Y, ∀Z, ∀W((reverse(W, X)/\print_solution(X))->solve(Y, Y, Z, W))

∀X, ∀Y((write(X)/\print_solution(Y))->print_solution([H|T]))

print_solution([])

Дерево вывода:



Вывод:

В ходе выполнения лабораторной работы были приобретены навыки логического программирования поиска решения задачи.

Использованные источники:

1. Логические основы интеллектуальных систем. Практикум [Электронный ресурс]. Режим доступа:https://libeldoc.bsuir.by/bitstream/123456789/992/2/Log_Osn.pdf.
2. Prolog Search[Электронный ресурс]. Режим доступа: https://redirect.cs.umbc.edu/courses/771/current/presentations/prolog%20search.pdf
3. Введение в логическое программирование. SWI Prolog [Электронный ресурс]. https://www.youtube.com/watch?v=sw6JNBrkFNA&list=PLQJTi0aBG6_X4HEBRbddCCno8RCwTkX66&index=4