

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Дисциплина «Интеллектуальные системы»

ОТЧЕТ

по лабораторной работе №6

на тему

«Разработка простой интеллектуальной системы на языке PROLOG»

Выполнил:

Студент группы 3540901/02001
Дроздов Н.Д.

«....» 2020г.,
(Подпись)

Проверил:

Бендерская Е.Н.

«....» 2020г.,
(Подпись)

Санкт-Петербург
2020 г.

Содержание

| | | |
|----------|-------------------------------|----------|
| 1 | Лабораторная работа №6 | 2 |
| 1.1 | Цель работы | 2 |
| 1.2 | Программа работы | 2 |
| 1.3 | Ход работы | 3 |
| 1.3.1 | Задание 1 | 3 |
| 1.3.2 | Задание 2 | 3 |
| 1.4 | Вывод | 5 |
| 1.5 | Список литературы | 6 |

Лабораторная работа №6

1.1 Цель работы

Разработка простой интеллектуальной системы на языке PROLOG.

1.2 Программа работы

1. Выполнить одно из индивидуальных заданий (см. задание 9-15 на стр. 32-34 из пособия Бураков С. В. «Язык логического программирования PROLOG», СПбГУАП, 2003).
2. Создайте проект в оболочке Visual Prolog 7.3., как это показано в примере [2]
3. Изучить 1-2 лаб.раб. по методич. указаниям - Середа С.Н. Представление знаний в информационных системах. Логическое программирование/ Методические указания к лабораторным работам. – Муром, МИ ВлГУ, 2005.
4. Согласно своему варианту решить с помощью PROLOG задачу описанные ниже: Продемонстрировать скриншотами и, при желании, нарисовать дерево решения.
5. Написать выводы. В выводах отразить, помимо своих мыслей, возникших в ходе работы, ответы на приведенные ниже вопросы:
 - В чем Плюсы и минусы языка Prolog?
 - Какие еще языки используются для разработки ИИ, приведите примеры (НЕ МЕНЕЕ 2-х) проектов, языков и краткое описание проектов. (Альтернативы PROLOG)
 - Решаема ли проблема комбинаторного взрыва, пути решения?
 - Корректно ли по-вашему в принципе разработка языка ИИ? Что он должен из себя представлять?
 - Можно ли разработать ИИ не понимая, как он работает, должны ли мы понимать, как он работает, думает, рассуждает?

1.3 Ход работы

1.3.1 Задание 1

Выполнить индивидуальное задание из пособия Буракова С.В.

Витя, Юра и Миша сидели на скамейке. В каком порядке они сидели, если известно, что Юра сидел слева от Миши и справа от Вити.

```
1 implement main
2     open core
3
4 domains
5     person = symbol.
6
7 class predicates
8     left : (person, person) nondeterm anyflow.
9     pos : (person [out], person [out], person [out]) nondeterm.
10
11 clauses
12
13     left("Vitya", "Ura").
14     left("Ura", "Misha").
15
16     pos(X, Y, Z) :-
17         left(X, Y),
18         left(Y, Z).
19
20 clauses
21     run() :-
22         console::init(),
23         pos(X, Y, Z),
24         stdIO::writef(X, " - ", Y, " - ", Z),
25         fail.
26
27     run().
28
29 end implement main
30
31 goal
32     console::runUtf8(main::run).
```

Результат выполнения задания:



```
Vitya - Ura - Misha
```

Рис. 1.1: Результаты выполнения индивидуального задания

1.3.2 Задание 2

Выполнить индивидуальное задание из пособия Седана С.Н.

Лабиринт представляет собой систему комнат, соединенных между собой переходами. В лабиринте имеется вход и выход, а также комната с золотым кладом. Кроме того, имеются комнаты, запрещенные для посещений: комната монстров и комната разбойников.

1. Найди путь в лабиринте от входа до выхода, не посещая дважды одной и той же комнаты;
2. Найти путь с посещением золотой комнаты;
3. Найти путь, избегающий запрещенных к посещению комнат.

```
1 implement main
2     open core
3
4 class predicates
```

```

5      connection : (string [out], string [out]) multi.
6      connection : (string, string [out]) nondeterm.
7      connection : (string [out], string) nondeterm.
8      path : (string, string, string*) nondeterm.
9      member : (string, string*) nondeterm.
10     reverse : (string*, string* [out], string*) nondeterm anyflow.
11
12 clauses
13     connection("entrance", "room 1").
14
15     connection("entrance", "room 4").
16
17     connection("room 1", "gold").
18
19     connection("room 1", "robbers").
20
21     connection("room 1", "room2").
22
23     connection("room2", "exit").
24
25     connection("room 3", "exit").
26
27     connection("room 4", "room2").
28
29     connection("gold", "monster").
30
31     connection("gold", "room 3").
32
33     connection("monster", "exit").
34
35     connection("robbers", "exit").
36
37     member(X, [Y | T]) :-
38         X = Y
39         or
40         member(X, T).
41
42     reverse([], Z, Z).
43
44     reverse([H | T], Z, A) :-
45         reverse(T, Z, [H | A]).
46
47     path(Y, Y, _).
48
49     path(X, Y, T) :-
50         connection(X, Z),
51         not(member(Z, T)),
52         path(Z, Y, [Z | T]).
53
54     path(X, Y, T) :-
55         connection(Z, X),
56         not(member(Z, T)),
57         path(Z, Y, [Z | T]).
58
59     path(_, _, [F | O]) :-
60         reverse([F | O], [L | V], []),
61         if F = "exit" and L = "entrance" and not(member("robbers", O)) and not(member("
monster", O)) and member("gold", O) then
62             stdIO::writef("%\n", [L | V])
63         else
64             end if.
65
66     run() :-
67         console::init(),
68         connection(X, Y),
69         path(X, Y, []),

```

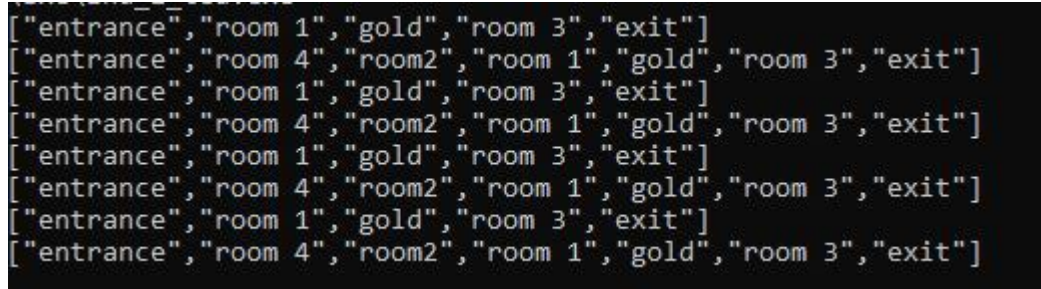
```

70         fail .
71
72     run () .
73
74 end implement main
75
76 goal
77     console :: runUtf8 (main :: run) .

```

В данном алгоритме указываются связи между соседними комнатами, после чего подбираются различные варианты маршрутов, с помощью path. Кроме того, на path установлен фильтр, который выводит в консоль все варианты пути, удовлетворяющие фильтру. Таким образом выполняются условия с типами комнат.

Результат выполнения задания:



```

["entrance","room 1","gold","room 3","exit"]
["entrance","room 4","room2","room 1","gold","room 3","exit"]
["entrance","room 1","gold","room 3","exit"]
["entrance","room 4","room2","room 1","gold","room 3","exit"]
["entrance","room 1","gold","room 3","exit"]
["entrance","room 4","room2","room 1","gold","room 3","exit"]
["entrance","room 1","gold","room 3","exit"]
["entrance","room 4","room2","room 1","gold","room 3","exit"]

```

Рис. 1.2: Результаты выполнения индивидуального задания

Варианты правильные, однако, повторяются несколько раз. Этого можно избежать, если заменить writef внутри функции на возвращаемый список, который впоследствии будет отфильтрован на предмет одинаковых вариантов.

1.4 Вывод

В результате работы были изучены основные возможности языка Prolog, а также среды разработки Visual Prolog.

Плюсы и минусы языка Prolog

К плюсам языка Prolog относится тот факт, что это декларативный язык: описывается логическая модель предметной области (ПО) в терминах этой ПО, их свойства и отношения между ними, а не детали программной реализации. Таким образом, мы описываем «что» хотим получить, а не «как» мы хотим это получить. Это позволяет сильно уменьшить время разработки приложений и размеры исходного кода.

Из минусов этого языка можно выделить его узкую специализацию. Язык хорошо показывает себя в своей области применения, однако, для написания чего-то универсального и производительного он не годится.

Аналоги языка Prolog

Существует множество аналогов языка PROLOG, например:

- Python – язык, который содержит огромный набор научных библиотек, в том числе и для ИИ.
- CLIPS – программная среда и язык для разработки экспертных систем.
- Planner – функционально-логический язык программирования, схожий по своему синтаксису с LISP.
- Linden Scripting Language (или LSL) – скриптовый язык программирования.
- LISP – функциональный язык программирования.

Проблема комбинаторного взрыва

Проблема комбинаторного взрыва в общем случае не решена, однако, использование метода наискорейшего подъема помогает значительно ускорить поиск.

