

Министерство науки и Высшего образования Российской Федерации
Севастопольский государственный университет
Кафедра ИС

Отчет
по лабораторной работе №8
«Разработка и исследование экспертной системы»
по дисциплине
«МЕТОДЫ И СРЕДСТВА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

Выполнил студент группы ИС/б-17-2-о
Горбенко К. Н.
Проверил
Забаштанский А.К.

Севастополь
2021

1 ЦЕЛЬ РАБОТЫ

Разработка экспертной системы продукционного типа на Прологе, исследование базовых принципов организации экспертных систем.

2 ЗАДАНИЕ НА РАБОТУ

Реализовать продукционную экспертную систему в соответствии с номером варианта. При этом количество рассматриваемых объектов предметной области должно быть не менее 10 и характеризующих их атрибутов также — не менее 10. Система должна уметь давать объяснения вывода.

Предметная область – микропроцессоры.

3 ХОД РАБОТЫ

3.1 Разработка программы

Текст разработанной программы:

```

1 % База знаний ЭС классификация микропроцессоров.
2 % База знаний, необходимая для определения структуры микропроцессоров
3 % Имеем множество различных структур микропроцессоров
4
5 %Интерпретатор классифицирующей ЭС
6 % Интерпретатор (машина вывода) для ЭС продукционного типа
7 % Метод вывода: обратный вывод
8 % Вариант 2: интерпретатор обрабатывает правила, в которых
9 % предпосылки задаются в виде списка условий.
10 % Это позволяет в условной части правила, задавать произвольное
11 % количество условий.
12 :-dynamic
13 сообщено/2.
14
15 определить_операторы:-
16     op(950, xfx, то),
17     op(960, fx, если),
18     op(970, xfx, '::~').
19 :-определить_операторы.
20
21 %=====обратный вывод
22 %=====
22 % реализуется предикатом найти(S,Стек,Д), где S - список проверяемых гипотез,
23 % Стек - стек из имен доказываемых гипотез и правил (используется при ответе на

```

```

24 % вопросы "почему), Д - дерево вывода целевого утверждения (используется при от
    ве-
25 % те на вопросы "как"). Предикат получает на вход список [Н] и Стек=[Н] и в про
    -
26 % цессе обратного вывода строит дерево вывода Д.
27 % Предикат "найти" для доказательства отдельных гипотез из списка S
28 % использует предикат найти1(Н,Стек,Дерево).
29 %-----

30 %
31 % случай1:если цель Н была подтверждена пользователем,
32 % то дерево вывода Д=сообщено(Н).
33 найти1(Н,Стек,сообщено(Н)):-сообщено(Н,да).
34 найти1(Н,Стек,сообщено(Н)):-запрашиваемая(Н),
35                             not(сообщено(Н,_)),спроси(Н,Стек).
36
37 % случай2:если цель Н подтверждается фактом, уже известным системе,
38 % то дерево вывода Д=Факт :: Н
39 найти1(Н,Стек,Факт :: Н):-Факт :: Н.
40
41 % случай3: если цель Н соответствует следствию одного из
42 % правил -> Правило :: если Н1 то Н
43 % и если Д1 дерево вывода для подцели Н1,
44 % то Д= Правило :: если Д1 то Н и добавить¶ правила в Стек
45 найти1(Н,Стек,Правило :: если Д1 то Н):-
46                             Правило :: если Н1 то Н,
47                             найти(Н1,[Правило | Стек],Д1).
48
49 % случай4: если доказываемся конъюнкция гипотез, заданная списком гипотез,
50 % то найти доказательство первой гипотезы Н1 из списка
51 % с помощью найти1(Н1,Стек,Дерево1), а затем найти доказательство оставшихся
52 % гипотез Т с помощью найти(Т,Стек,Дерево) и
53 % объединить деревья вывода в общий список [Дерево1 | Дерево].
54 найти([],Стек,Дерево):-Дерево=[].
55 найти([Н1|Т],Стек,[Дерево1 | Дерево]):-
56                             найти1(Н1,Стек,Дерево1),найти(Т,Стек,Дерево).
57
58 % проверка: является ли гипотеза признаком, значение которого можно спросить
59 запрашиваемая(Н):-Факт :: признак(Н).
60
61 %=====вывод вопросов и обработка ответов "да, нет, почему"
    =====
62 %вывод вопроса и ввод ответа
63 спроси(Н,Стек):-write(Н),write('?''),nl,
64                 read(0),ответ(Н,0,Стек).
65
66 %обработка ответов: да, нет
67 ответ(Н,да,Стек):-assert(сообщено(Н,да)),!.

```

```

68 ответ(Н,нет,Стек):-assert(сообщено(Н,нет)),!,fail.
69
70 %обработка ответов - "почему"
71 % случай1: стек целей пустой
72 ответ(Н,почему,[]):-!,write(' Вы задаете слишком много вопросов'),nl,
73     спроси(Н,[]).
74
75 %случай2: в стеке осталась только первая введенная цель, т.е доказываемая гипот
    еза
76 ответ(Р,почему,[Н]):-!,write('моя гипотеза: '),
77     write(Н),nl,спроси(Р,[]).
78
79 %случай3: вывод заключения и номера правила для доказываемой текущей подцели Н
80 ответ(Н,почему,[Правило | Стек]):-!,
81     Правило :: если Н1 то Н2,
82     write('пытаюсь доказать '),
83     write(Н2),nl,
84     write('с помощью правила: '),
85     write(Правило),nl,
86     спроси(Н,Стек).
87
88 %неправильный ответ: повторяем вопрос
89 ответ(Н,_,Стек):-write(' правильный ответ: да, нет, почему'),nl,
90     спроси(Н,Стек).
91
92 %=====обработка ответов на вопросы "как
    ?"=====
93 % предикат как(Н,Д)- выполняет поиск подцели Н в построенном
94 % с помощью предиката "найти" дереве вывода Д и отображает соответствующий
95 % фрагмент дерева вывода, объясняя, как было получено доказательство Н.
96 % Дерево вывода Д представляет собой последовательность вложенных правил
97 % в виде списка, например:
98 % [правило5::если[правило1::если[сообщено(имеет(шерсть))]]то млекопитающее,
99 %     сообщено(ест_мясо)]то хищник,...]
100 %-----

101
102 % поиск целевого утверждения Н в дереве
103 как(Н,Дерево):-как1(Н,Дерево),!.
104
105 % вывод сообщения, если Н не найдено
106 как(Н,_-):write(Н),tab(2),write('не доказано'),nl.
107
108 % случай1: если Н сообщено пользователем,
109 % то вывести "Н было введено"
110 как1(Н,_-):-сообщено(Н,_-),!,
111     write(Н),write('было введено'),nl.
112

```

```

113 % случай2: если дерево вывода Д представлено фактом, подтверждающим Н
114 как1(Н,Факт :: Н):-!,
115     write(Н), write( 'является фактом'), write(Факт),nl.
116
117 % случай3: если дерево вывода Д - правило в заключение, которого есть Н,
118 % то отобразить это правило
119 как1(Н,[Правило :: если _ то Н]):-!,
120     write(Н),write(' было доказано с помощью'),nl,
121     Правило :: если Н1 то Н,
122     отобрази_правило(Правило :: если Н1 то Н).
123
124 % случай4: если в дереве Д нет правила с заключением Н,
125 %то поиск Н надо выполнять в дереве вывода предпосылок, т.е. в Дерево
126 как1(Н,[Правило :: если Дерево то _]):-как(Н,Дерево).
127
128 % случай5: если дерево вывода - список поддеревьев вывода
129 % каждой конъюнктивной подцели правила из БЗ,
130 % то поиск Н следует выполнять в каждом из поддеревьев;
131 % поиск Н следует выполнять сначала в поддереве [Д1], а
132 % если Н не найдено, то продолжить поиск в оставшихся поддеревьях
133 как1(Н,[ ]):-!.
134 как1(Н,[Д1|Д2]):-как(Н,[Д1]),!;
135     как1(Н,Д2).
136
137 %вывод правила на экран
138 отобрази_правило(Правило :: если Н1 то Н):-
139     write(Правило), write( ':' ),nl,
140     write('если '), write(Н1), nl,
141     write('то '), write(Н),nl.
142
143 /* Вызов интерпретатора*/
144 инициализация:-retractall(сообщено(_,_)).
145 start:-
146     /* Загрузка базы знаний из файла*/
147     reconsult('F:/base.pl'),
148     info, %отображение информации о базе знаний*
149     go_exp_sys.
150
151 go_exp_sys:- инициализация,
152     Факт :: гипотеза(Н),
153     найти([Н],[Н],Дерево),
154     write('решение:'),write(Н),nl,
155     объясни(Дерево),
156     возврат.
157
158 %объяснение вывода утверждения
159 объясни(Дерево):-write( 'объяснить ? [цель/нет]:'), nl,read(Н),
160     (Н\=нет,! ,как(Н,Дерево),объясни(Дерево));!.

```

```

161
162 %поиск следующих решений
163 возврат:-write('Искать ещё решение [да/нет] ?: '),nl, read(нет).
164
165 %-----
166 % База знаний ЭС
167 %-----
168
169 info:-
170     nl,
171     write('*****'),nl,
172     write('*      Экспертная система      *'),nl,
173     write('*      Микропроцессоры      *'),nl,
174     write('*      *'),nl,
175     write('*----- *'),nl,
176     write('*      Отвечайте на вопросы:      *'),nl,
177     write('*      да, нет, почему      *'),nl,
178     write('*      Для объяснения решения      *'),nl,
179     write('*      введите цель      *'),nl,
180     write('*****'), nl ,
181     write('Введите любой символ'),nl, %Ожидание ввода литеры
182     get0(_).
183
184 % база продукционных правил классификации микропроцессоров
185 % предпосылки правил задаются в виде списка условий
186 % [условие1,условие2,...] означает - условие1 и условие2 и...
187 правило1 :: если [вид(desktopный), разгонный_потенциал(большой)]
188             то amd_Zen.
189 правило2 :: если [вид(desktopный), разгонный_потенциал(небольшой), ядер(одно)]
190             то intel8080.
191 правило3 :: если [вид(desktopный), разгонный_потенциал(небольшой), ядер(два)]
192             то intel_pentium.
193 правило4 :: если [вид(desktopный), разгонный_потенциал(небольшой), ядер(четыре
194             )]
195             то intel_core_i7.
196 правило5 :: если [вид(мобильный), используется(nexus_4)]
197             то qualcomm_snapdragon.
198 правило6 :: если [вид(мобильный), используется(iphone_6)]
199             то apple_A9.
200 правило7 :: если [вид(мобильный), используется(lumia_920)]
201             то qualcomm_MSM8960.
202 правило8 :: если [вид(мобильный), используется(asus_zenfone)]
203             то intel_atom.
204 правило9 :: если [вид(мобильный), используется(samsung_galaxy_s6)]
205             то samsung_Exynos_7420.
206 правило10 :: если [вид(мобильный), используется(raspberry_pi)]
207             то arm.
208 % гипотезы

```

```

208 h1 :: гипотеза(intel8080).
209 h2 :: гипотеза(intel_core_i7).
210 h3 :: гипотеза(arm). %raspberrypi
211 h4 :: гипотеза(amd_Zen).
212 h5 :: гипотеза(qualcomm_snapdragon). %nexus 4
213 h6 :: гипотеза(intel_pentium).
214 h7 :: гипотеза(intel_atom).
215 h8 :: гипотеза(qualcomm_MSM8960). %lumia 920
216 h9 :: гипотеза(apple_A9). %iphone 6
217 h10 :: гипотеза(samsung_Exynos_7420).
218
219 % признаки микропроцессоров, истинность которых можно выяснить у пользователя
220 q1 :: признак(вид(мобильный)).
221 q2 :: признак(вид(десктопный)).
222 q3 :: признак(разгонный_потенциал(большой)).
223 q4 :: признак(разгонный_потенциал(небольшой)).
224 q5 :: признак(используется(nexus_4)).
225 q6 :: признак(используется(iphone_6)).
226 q7 :: признак(используется(lumia_920)).
227 q8 :: признак(используется(asus_zenfone)).
228 q9 :: признак(используется(samsung_galaxy_s6)).
229 q10 :: признак(ядер(два)).
230 q11 :: признак(ядер(четыре)).
231 q12 :: признак(ядер(одно)).
232 q13 :: признак(используется(raspberrypi)).

```

3.2 Проверка работы программы

```

1 вид(десктопный)?
2 |да.
3 разгонный_потенциал(небольшой)?
4 :-нет.
5 вид(мобильный)?
6 :-нет.
7 разгонный_потенциал(большой)?
8 :-да.
9 решение:amd_Zen
10 объяснить ? [цель/нет]:
11 :-нет.
12 Искать ещё решение [да/нет]?:
13 :-да.
14 false.
15
16
17 :-
18 вид(десктопный)?
19 |да.
20 разгонный_потенциал(небольшой)?

```

21 :-да.
22 ядер(одно)?
23 :-почему.
24 пытаюсь доказать intel8080
25 с помощью правила: правило2
26 ядер(одно)?
27 :-нет.
28 ядер(четыре)?
29 :-нет.
30 вид(мобильный)?
31 :-нет.
32 разгонный_потенциал(большой)?
33 :-нет.
34 ядер(два)?
35 :-да.
36 решение:intel_pentium
37 объяснить ? [цель/нет]:

ВЫВОДЫ

В ходе выполнения лабораторной работы была разработана экспертная система продукционного типа на Прологе, а также исследованы базовые принципы организации экспертных систем.