Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчёт по лабораторной работе №5 по дисциплине "Проектирование экспертных систем"

Тема Метод унификации

Студент Варламова Е. А.

Группа ИУ7-33М

Оценка (баллы) _____

Преподаватели Русакова З.Н.

Москва — 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель работы – реализовать метод унификации.
Для достижения поставленной цели потребуется:

— описать используемые структуры данных;

— описать алгоритм метода унификации;

— привести реализацию алгоритма;

— привести примеры работы алгоритма.

1 | Используемые структуры данных

Разработаем класс переменной. Поля класса:

— имя;

— тип (переменная).

Разработаем класс константы. Поля класса:

— значение;

— тип (не переменная).

Разработаем класс атома. Поля класса:

— имя;

— список термов (терм – константа или переменная).

Разработаем класс подстановок. Поля класса:

— словарь переменных (ключ – имя переменной, значение – связанная переменная или константа);

— словарь ссылок (ключ – значение константы, значение – список переменных, имющих знчение этой константы).

2 | Метод унификации

Вход:

— 2 атома;

— подстановка.

Выход:

— информация о возможности унификации.

Унифиация выполняется:

1. Если термы онстанты, то они унифицируемы если совпадают.
2. Если в первом атоме терм переменная, а во втором онстанта, то они унифицируемы и переменная получает значение онстанты.
3. Если терм в первом атоме переменная и во втором тоже, то они унифицируемы и становятся связанными.
4. Если в первом атоме терм переменная, а во втором фунция от переменных, то они унифицируемы и вместо переменной подставляется фунция (x и f(x) не унифицируемы).

3 | Реализация

Листинг 3.1: Структуры данных

```
class Atom:
    def __init__(self, name, terminals):
        self.name = name
        self.terminals = terminals

    def __str__(self):
        strterms = ""
        for term in self.terminals:
            strterms += str(term) + ", "
        return self.name + '(' + strterms.strip(", ") + ')'

    def __repr__(self):
        return self.__str__()


class Constant:
    def __init__(self, value):
        self.value = value
        self.variable = False

    def __str__(self):
        return self.value

    def __repr__(self):
        return self.__str__()


class Variable:
    def __init__(self, name):
        self.name = name
        self.variable = True

    def __str__(self):
        return self.name

    def __repr__(self):
        return self.__str__()

```

```
39
40 class Table:
41     def __init__(self):
42         self.variables = dict()
43         self.links = dict()
44
45     def reset(self, other):
46         self.variables = other.variables
47         self.links = other.links
48
49     def val(self, var):
50         return self.variables[var.name]
51
52     def var_links(self, var):
53         return self.links[self.variables[var.name]]
54
55     def __str__(self):
56         res = ""
57         for const in self.links.keys():
58             res += str(self.links[const]) + ": " + str(const) + "\n"
59         return res
```

Листинг 3.2: метод унификации

```
1 def unification(table, p1, p2):
2     if p1.name != p2.name:
3         print('                                    ')
4         return False
5
6     if len(p1.terminals) != len(p2.terminals):
7         print('                                    ')
8         return False
9
10    original = copy.deepcopy(table)
11    for t1, t2 in zip(p1.terminals, p2.terminals):
12        print(table.variables)
13        if t1.variable:
14            if t2.variable:
15                y = True
16
17                if t1.name not in table.variables and t2.name not in table.
                      variables:
18                    table.variables[t1.name] = t2.name
19                    table.variables[t2.name] = t1.name
20
21                elif t1.name not in table.variables:
22                    table.variables[t1.name] = table.variables[t2.name]
23
24                elif t2.name not in table.variables:
25                    table.variables[t2.name] = table.variables[t1.name]
```

```
26
27              elif set([t1.name, table.variables[t1.name]]) != set([table.
                    variables[t2.name], t2.name]):
28                  y = False
29
30              if y == False:
31                  print("                      ", t1.name, "
                                                ", t2.name, ": ", table.val(t1),
32                        " != ", table.val(t2), sep='')
33                  table.reset(original)
34                  return False
35
36          else:
37              y = True
38              if t1.name in table.variables and type(table.variables[t1.
                    name]) is not str:
39                  if table.variables[t1.name].value != t2.value:
40                      y = False
41
42              if t1.name not in table.variables:
43                  table.variables[t1.name] = t2
44
45              if type(table.variables[t1.name]) is str:
46                  k = table.variables[t1.name]
47                  table.variables[t1.name] = t2
48                  table.variables[k] = t2
49                  table.links[t2.value] = {k}
50
51              if t2.value not in table.links:
52                  table.links[t2.value] = {t1.name}
53              else:
54                  table.links[t2.value].add(t1.name)
55
56              if y == False:
57                  print("
                                              : ", t1.name, " = ", table.val(t1),
58                        "                       ", t2.value)
59                  table.reset(original)
60                  return False
61      else:
62          if t2.variable:
63              y = True
64
65              if t2.name in table.variables and type(table.variables[t2.
                    name]) is not str:
66                  if table.variables[t2.name].value != t1.value:
67                      y = False
68
```

```
69                  if t2.name not in table.variables:
70                      table.variables[t2.name] = t1
71
72                  if type(table.variables[t2.name]) is str:
73                      k = table.variables[t2.name]
74                      table.variables[t2.name] = t1
75                      table.variables[k] = t1
76                      table.links[t1.value] = {k}
77
78                  if t1.value not in table.links:
79                      table.links[t1.value] = {t2.name}
80                  else:
81                      table.links[t1.value].add(t2.name)
82
83                  if y == False:
84                      print("
                                                  :", t2.name, "=", table.val(t2),
85                            "                       ", t1.value)
86                      table.reset(original)
87                      return False
88              else:
89                  if t1.value != t2.value:
90                      print("
                                                       :", t1.value, "!=", t2.
                           value)
91                      table.reset(original)
92                      return False
93      return True
```

4 | Пример работы

4.1 Тест№1: константы

Вызов:

```
c_N = Constant('N')
c_M1 = Constant('M1')
c_W = Constant('W')
c_A1 = Constant('A1')

node1 = Atom("A", [c_M1, c_W])
node2 = Atom("A", [c_M1, c_W])
table = Table()
print("Результат: ", unification(table, node1, node2))
print("----------")

node1 = Atom("R", [c_M1, c_W])
node2 = Atom("A", [c_M1, c_W])
table = Table()
print("Результат: ", unification(table, node1, node2))
print("----------")

node1 = Atom("A", [c_M1, c_N])
node2 = Atom("A", [c_M1, c_W])
table = Table()
print("Результат: ", unification(table, node1, node2))
print("----------")
```

Результат:

```
A(M1, W)  and  A(M1, W)
Результат:  True
----------
R(M1, W)  and  A(M1, W)
Имена не совпадают
Результат:  False
----------
A(M1, N)  and  A(M1, W)
Константы не соответствуют: N != W
Результат:  False
----------
```

4.2 Тест№2: переменная и константа

Вызов:

```
c_N = Constant('N')
c_M1 = Constant('M1')
c_W = Constant('W')
c_A1 = Constant('A1')

v_x = Variable("x")
v_y1 = Variable("y1")
v_y3 = Variable("y3")
v_y4 = Variable("y4")
v_y2 = Variable("y2")
v_z = Variable("z")
v_x1 = Variable("x1")
v_x2 = Variable("x2")
v_x3 = Variable("x3")
v_x4 = Variable("x4")

node1 = Atom("A", [v_x3, v_y2, v_y2])
node2 = Atom("A", [v_x1, v_y1, v_y1])
table = Table()
```

```
print("Результат: ", unification(table, node1, node2))
print("Ссылки:" , table.links)
print("Переменные: ", table.variables)
print("----------")

node1 = Atom("A", [v_x4, v_y3, v_y3])
node2 = Atom("A", [v_x2, v_y4, v_y4])
table = Table()
print("Результат: ", unification(table, node1, node2))
print("Ссылки:" , table.links)
print("Переменные: ", table.variables)
print("----------")

node1 = Atom("A", [v_x1, v_y3, v_y3])
node2 = Atom("A", [v_x2, c_A1, c_A1])
table = Table()
print("Результат: ", unification(table, node1, node2))
print("Ссылки:" , table.links)
print("Переменные: ", table.variables)
print("----------")

node1 = Atom("A", [v_x1, v_y3, v_y3])
node2 = Atom("A", [v_x2, c_A1, c_N])
table = Table()
print("Результат: ", unification(table, node1, node2))
print("Ссылки:" , table.links)
print("Переменные: ", table.variables)
print("----------")

node1 = Atom("A", [v_x1, v_y3, v_y3])
node2 = Atom("A", [v_x2, c_A1, v_y4])
table = Table()
print("Результат: ", unification(table, node1, node2))
print("Ссылки:" , table.links)
print("Переменные: ", table.variables)
print("----------")
```

Результат:

```
A(x3, y2, y2)  and  A(x1, y1, y1)
Результат:  True
Ссылки: {}
Переменные:  {'x3': 'x1', 'x1': 'x3', 'y2': 'y1', 'y1': 'y2'}
----------
A(x4, y3, y3)  and  A(x2, y4, y4)
Результат:  True
Ссылки: {}
Переменные:  {'x4': 'x2', 'x2': 'x4', 'y3': 'y4', 'y4': 'y3'}
----------
A(x1, y3, y3)  and  A(x2, A1, A1)
Результат:  True
Ссылки: {'A1': {'y3'}}
Переменные:  {'x1': 'x2', 'x2': 'x1', 'y3': A1}
----------
A(x1, y3, y3)  and  A(x2, A1, N)
Несоответствующее значение переменной константе:  y3  =  A1 константа N
Результат:  False
Ссылки: {}
Переменные:  {}
----------
A(x1, y3, y3)  and  A(x2, A1, y4)
Результат:  True
Ссылки: {'A1': {'y3'}}
Переменные:  {'x1': 'x2', 'x2': 'x1', 'y3': A1, 'y4': A1}
----------
```

ЗАКЛЮЧЕНИЕ

В результаты работы цель была достигнута.
Для достижения поставленной цели потребовалось:

— описать используемые структуры данных;

— описать алгоритм метода унификации;

— привести реализацию алгоритма;

— привести примеры работы алгоритма.