МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Новгородский государственный университет имени Ярослава Мудрого (НовГУ) Кафедра «Электронных и информационных систем»

**Продукционная модель представления знаний**

Лабораторная работа по дисциплине:

«Системы искусственного интеллекта»

Студент гр. 1095

Золотов Э. В.

(подпись)

«\_\_\_»\_ 2024 г.

Преподаватель

\_\_Михайлов Д.В

(подпись)

« » 2024 г.

г. Великий Новгород

-2024-

# ЦЕЛЬ РАБОТЫ

Целью данной работы является приобретение студентами умений и навыков реализации пополняемой динамической базы знаний, не включаемой непосредственно в текст программы.

Необходимые для достижения поставленной цели задачи состоят в следующем:

• научиться оценивать возможности применения современных языков высокого уровня для реализации баз знаний экспертных систем (ЭС);

• изучение технических аспектов реализации продукционной модели представления знаний.

# КОНФИГУРАЦИЯ СИСТЕМЫ ПРОДУКЦИИ

Предметная область для ЭС – Выбор профессии.

Профессии разбиты на группы, в зависимости от зарплаты

|  |  |  |
| --- | --- | --- |
| Группа | Профессия | Характеристики |
| Низкая | Физрук | Решительный 3  Информатика 1  Математика 1  Спортивный 2  Творческий 1  Общительный 2 |
| Тренер | Решительный 4  Информатика 1  Математика 1  Спортивный 3  Творческий 1  Общительный 3 |
| Средняя | Дизайнер | Решительный 1  Информатика 3  Математика 1  Спортивный 1  Творческий 3  Общительный 1 |
| Художник | Решительный 1  Информатика 1  Математика 1  Спортивный 1  Творческий 3  Общительный 1 |
| Разработчик\_UI | Решительный 1  Информатика 5  Математика 2  Спортивный 1  Творческий 3  Общительный 1 |
| Высокая | Проджект\_менеджер | Решительный 5  Информатика 5  Математика 3  Спортивный 1  Творческий 1  Общительный 3 |
| Аналитик\_данных | Решительный 1  Информатика 5  Математика 5  Спортивный 1  Творческий 1  Общительный 1 |

***Таблица 1***

Для описания объекта или его отдельных свойств выбираются некоторые характеристики – величины, которые могут принимать либо количественные, либо качественные значения.

В свою очередь, совокупность всех характеристик некоторого объекта образует так называемый список разрешенных характеристик данного объекта. Списки разрешенных характеристик и разрешенных значений этих характеристик охватывают множество всех имеющихся фактов, подлежащих хранению в базе знаний экспертной системы.

В экспертных системах правила, по которым решаются проблемы в конкретной предметной области, хранятся в базе знаний. Проблемная область ЭС описывается посредством фактов и правил. Проблемы ставятся перед системой в виде совокупности фактов, описывающих некоторую ситуацию, и система с помощью базы знаний пытается вывести заключение из этих фактов.

Для данной лабораторной работы в качестве базы знаний был выбран текстовый файл формата txt. В начале файла описываются все атрибуты и их значения. 1-ый атрибут на 1-ой строке.

Пример атрибутов:

|  |
| --- |
| Решительный\* = **1** | **2** | **3** | **4** | **5**  Информатика\* = **1** | **2** | **3** | **4** | **5**  Математика\* = **1** | **2** | **3** | **4** | **5**  Спортивный\* = **1** | **2** | **3**  Творческий\* = **1** | **2** | **3**  Общительный\* = **1** | **2** | **3**  Зарплата\* = Высокая | Средняя | Низкая |

В базе знаний хранятся правила, синтаксис которых “IF A THEN S”, где А – условие; S – заключение.

Пример правила:

|  |
| --- |
| IF Зарплата = Высокая & Решительный = **1** & Информатика = **5** & Математика = **5** & Спортивный = **1** & Творческий = **1** & Общительный = **1** THEN профессия = Аналитик\_данных |

# ОПИСАНИЕ СПОСОБА ОРГАНИЗАЦИИ ПОИСКА НА ГРАФЕ

Поиск на графе организован с помощью последовательного отброса заведомо ложных путей и проверки доступных с помощью перебора.

# ОПИСАНИЕ КОНФЛИКТНОГО НАБОРА И АЛГОРИТМ РАЗРЕШЕНИЯ КОНФЛИКТА ПРИ ЛОГИЧЕСКОМ ВЫВОДЕ

Прямой логический вывод начинается с предпосылок и идет к заключениям. В данной лабораторной работе, система начинает с фактов, доступных на текущем этапе, а затем применяет правила вывода, чтобы вывести новые факты или сделать выводы о значениях переменных.

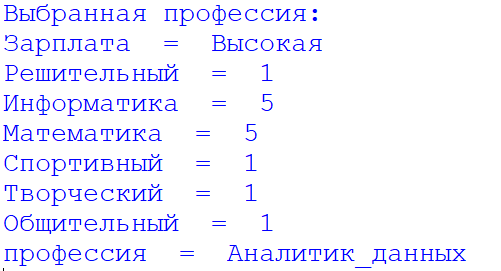
Алгоритм разрешения конфликта при логическом выводе определяет, какая из конфликтующих правил должна быть применена для выполнения следующего шага логического вывода. Система просматривает список конфликтующих правил в порядке их появления и применяет первое правило, для которого условия (LHS) выполняются на основе текущих фактов.

Процесс разрешения конфликта включает следующие шаги: Поиск конфликтующих правил для текущей цели; Для каждого конфликтующего правила проверка, выполняются ли его условия (LHS) на основе текущих фактов; Если условия (LHS) конкретного правила выполнены, то это правило считается активным для применения; Применение первого активного правила, для которого условия выполняются; Продолжение логического вывода на основе вывода (RHS) выбранного правила.

# ТЕСТОВЫЙ НАБОР ПРАВИЛ

Высокая->1->5->5->1->1->1

Ответ системы:



# ВЫВОД

В результате выполнения лабораторной работы была изучена реализация пополняемой динамической базы знаний, не включаемой непосредственно в текст программы. По мимо этого, была изучена реализация продукционной модели представления знаний и построена база знаний предметной области “Выбор профессии”.

# ПРИЛОЖЕНИЕ A

# Код программы

**import** **copy**

**def** **rule\_repr**(rule):

mas1 = []

**for** attr, values **in** rule['LHS'].items():

mas1.append(attr + " = " + "|".join(values))

(RHSkey, RHSvalue) = list(rule['RHS'].items())[**0**]

**return** "IF " + " & ".join(mas1) + " THEN " + RHSkey + " = " + RHSvalue

**def** **printResult**():

print("Выбранная профессия:")

**for** r, v **in** mas.items():

print(r, " = ", v)

**def** **getConflictRules**(rules, goal):

ruleset = []

**for** rule **in** rules:

attribute = list(rule['RHS'].keys())[**0**]

**if** attribute == goal:

ruleset.append(rule)

**return** ruleset

**def** **conflictRuleExists**(rules, goal):

**for** rule **in** rules:

attribute = list(rule['RHS'].keys())[**0**]

**if** attribute == goal:

**return** **True**

**return** **False**

**def** **ruleWorks**(rule, mas):

conditions = rule['LHS']

**for** param **in** conditions:

**if** param **in** mas:

**if** mas[param] **not** **in** conditions[param]:

**return** **False**

**else**:

**return** **False**

**return** **True**

**def** **parse**(path):

parameters = {}

rules = []

**try**:

**with** open(path, 'r', encoding="utf-8") **as** f:

lines = f.readlines()

**except**:

print("Проблема с открытием файла " + path)

**return**

**for** line **in** lines:

**if** **not** line **or** line.startswith('-') **or** line.startswith('#'):

**continue**

**elif** line.startswith('IF'):

current = {}

sides = line.replace('IF', '').split('THEN')

current['LHS'] = {}

conditions = sides[**0**].split('&')

**for** condition **in** conditions:

HS = list(map(str.strip, condition.split('=', **1**)))

current['LHS'][HS[**0**]] = [s.strip() **for** s **in** HS[**1**].split('|')]

action = sides[**1**]

HS = list(map(str.strip, action.split('=')))

current['RHS'] = {HS[**0**]: HS[**1**]}

rules.append(copy.deepcopy(current))

**else**:

splitLine = line.split('=', **1**)

parameters[splitLine[**0**].strip()] = [s.strip()

**for** s **in** splitLine[**1**].split('|')]

**return** parameters,rules

**def** **parameterInput**(parameterinput1, mas):

value = input("Выбор-оценка '" + parameterinput1 +

"' " + str(parameters[parameterinput1+"\*"]) + ": ")

**while**(value **not** **in** parameters[parameterinput1+"\*"]):

value = input()

mas[parameterinput1] = value

**def** **knowledge\_base\_output**(parameters, rules):

print("Признаки:")

**for** attr, value **in** parameters.items():

print(attr + " = " + " | ".join(value))

print("**\n**Правила:")

**for** i, rule **in** enumerate(rules):

print(str(i+**1**) + ") " + rule\_repr(rule))

parameters, rules = parse('./globalBase.txt')

knowledge\_base\_output(parameters, rules)

mas = {}

goals = []

checked\_goals = []

goals.append('профессия')

**while**(**True**):

**if** len(goals) == **0**:

**break**

new\_goal = **False**

new\_parameter = **False**

goal = goals[-**1**]

conflictRules = getConflictRules(rules, goal)

remainingRules = len(conflictRules)

**for** cr **in** conflictRules:

**if** ruleWorks(cr, mas):

(RHSkey, RHSvalue) = list(cr['RHS'].items())[**0**]

mas[RHSkey] = RHSvalue

curr\_goal = goals.pop()

new\_goal = **True**

printResult()

**break**

**if** new\_goal:

**continue**

**for** cr **in** conflictRules:

**if** new\_goal:

**break**

remainingRules -= **1**

conditions = cr['LHS']

**for** parameterinput1 **in** conditions:

**if** parameterinput1 **in** checked\_goals:

**break**

**if** parameterinput1 **in** mas:

**if** mas[parameterinput1] **not** **in** conditions[parameterinput1]:

**break**

**else**:

**if** conflictRuleExists(rules, parameterinput1) **and** parameterinput1 **not** **in** goals:

goals.append(parameterinput1)

new\_goal = **True**

**break**

**elif** parameterinput1 + "\*" **in** parameters:

parameterInput(parameterinput1, mas)

new\_parameter = **True**

**break**

**else**:

checked\_goals.append(parameterinput1)

**if** new\_parameter:

**break**

**if** remainingRules == **0** **and** **not** new\_goal:

curr\_goal = goals.pop()

checked\_goals.append(curr\_goal)

print('В базе нет нужной информации о профессии')

**break**

# Файл globalBase.txt

Решительный\* = **1** | **2** | **3** | **4** | **5**

Информатика\* = **1** | **2** | **3** | **4** | **5**

Математика\* = **1** | **2** | **3** | **4** | **5**

Спортивный\* = **1** | **2** | **3**

Творческий\* = **1** | **2** | **3**

Общительный\* = **1** | **2** | **3**

Зарплата\* = Высокая | Средняя | Низкая

IF Зарплата = Высокая & Решительный = **1** & Информатика = **5** & Математика = **5** & Спортивный = **1** & Творческий = **1** & Общительный = **1** THEN профессия = Аналитик\_данных

IF Зарплата = Высокая & Решительный = **5** & Информатика = **5** & Математика = **3** & Спортивный = **1** & Творческий = **1** & Общительный = **3** THEN профессия = Проджект\_менеджер

IF Зарплата = Средняя & Решительный = **1** & Информатика = **5** & Математика = **2** & Спортивный = **1** & Творческий = **3** & Общительный = **1** THEN профессия = Разработчик\_UI

IF Зарплата = Средняя & Решительный = **1** & Информатика = **1** & Математика = **1** & Спортивный = **1** & Творческий = **3** & Общительный = **1** THEN профессия = Художник

IF Зарплата = Средняя & Решительный = **1** & Информатика = **3** & Математика = **1** & Спортивный = **1** & Творческий = **3** & Общительный = **1** THEN профессия = Дизайнер

IF Зарплата = Средняя & Решительный = **4** & Информатика = **1** & Математика = **1** & Спортивный = **3** & Творческий = **1** & Общительный = **3** THEN профессия = Тренер

IF Зарплата = Низкая & Решительный = **3** & Информатика = **1** & Математика = **1** & Спортивный = **2** & Творческий = **1** & Общительный = **2** THEN профессия = Физрук