Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Кафедра вычислительной математики и механики

Лабораторная работа № 2

по дисциплине: «Интеллектуальные ИСИТ»

Выполнила:

студентка группы ИСТ-19-1б

Дорогина И.В.

Проверил:

ассистент кафедры ВММБ

Нетбай Г. В.

Пермь, 2022

**Описание метода**

Продукционная экспертная система основана на правилах. В такой системе знания представлены в форме многочисленных правил, которые указывают, какие заключения должны быть сделаны или не сделаны в различных ситуациях.

Каждое правило обозначается именем (листинг 1). Вслед за именем находится часть IF правила. Участок правила между IF и THEN правила называют антецедентом. Далее следует часть THEN, в которой находится список действий, которые должны быть выполнены после запуска правила, - так называемый консеквент.

|  |
| --- |
| Правило: «Имя правила»  IF «условие» THEN «заключение» |
| Листинг 1. Общий вид правила |

Для более наглядного представления продукционных знаний можно использовать графы И/ИЛИ.

В системе, основанной на правилах, машина логического вывода определяет, какие антицеденты правил выполняются согласно фактам. В качестве стратегий решения задач в экспертных системах обычно используются два общих метода логического вывода: прямой логический вывод и обратный.

Прямой представляет собой метод формирования рассуждений от фактов к заключениям, которые следуют из этих фактов. Обратный логический вывод предусматривает формирование рассуждений в обратном направлении от гипотезы к фактам, которые ее подтверждают.

Для реализации базы знаний (БЗ) можно использовать существующие оболочки экспертных систем. В данной лабораторной работе – CLIPS.

**Постановка задания**

В рамках лабораторной работы необходимо разработать собственную оболочку для продукционной экспертной системы, аналогичную по своей работе оболочке CLIPS (C Language Integrate Production System) – программная среда для разработки экспертных систем. CLIPS является продукционной системой. Реализация вывода использует алгоритм Rete.

При использовании алгоритма Rete строится своего рода префиксное дерево, в котором узлы соответствуют части условий правил. Тогда путь от корня до листа дерева образует полное условие некоторой продукции. Алгоритм Rete жертвует объемом памяти ради скорости.

Основная идея CLIPS состоит в представлении знаний в виде формы, представленной в листинге 2.

|  |
| --- |
| Правило1:  IF  (выполняются условия1)  THEN  (выполнить действия1)  Правило2:  IF  (выполняются условия2)  THEN  (выполнить действия2)  ... |
| Листинг 2. Форма записи знаний |

**Описание задачи**

В рамках лабораторной работы по созданию продукционной экспертной системы для реализации разработана задача по выбору ноутбука.

Критерии, по которым будет осуществляться выбор:

1. Дизайн ноутбука, включая его цвет и наличие подсветки клавиатуры;

2. Предназначение устройства – ноутбук нужен для работы или нет;

3. Диагональ экрана;

4. Наличие сенсорного дисплея;

5. Бюджет.

В результате ранжирования важности выделенных критериев был построен граф И/ИЛИ (Приложение 1), описывающий процесс формирования заключения экспертной системы.

Первым этапом разработки программной оболочки для продукционной системы выделена разработка необходимых для обработки условий функций:

1. *set* – установка значения заданному факту;

2. *eq*  – сравнение двух поступивших значений (проверяет соответствие значения факта из рабочей памяти необходимому);

3. *andParams* – проверка, что все поступившие аргументы имеют значение *true*;

4. *yes\_or\_no* – вопрос пользователю с ответом «Да» или «Нет»;

5. *ask\_question* – вопрос пользователю с заданными вариантами ответа;

6. *printResult* – вывод заключения.

На листингах 3 и 4, соответственно, приведены примеры реализации универсальных функций *ask\_question* и *yes\_or\_no* для взаимодействия с пользователем.

На вход данные функции принимают вопрос, который необходимо задать пользователю, а функция *ask\_question* ещё и набор параметров, которым должен соответствовать ответ. На выход же функции выдают полученный ответ, который присваивая необходимому факту с помощью функции *set*.

|  |
| --- |
| def ask\_question(question, \*values):  answer = ""  while all([str(el) != answer for el in values]):  answer = input(question)  return answer |
| Листинг 3. Реализация функции *ask\_question* |
|  |
| def yes\_or\_no(question):  answer = ""  while(answer != "yes" and answer != "no"):  answer = input(question)  return answer |
| Листинг 4. Реализация функции *yes\_or\_no* |

Все правила для экспертной системы записаны в JSON-файл, что позволяет легко обращаться к ним по ходу программы.

Рассмотрим основные типы правил, реализованных в разработанной экспертной системе.

Первый тип правил – те, которые направлены на присвоение фактам системы значений посредством опроса пользователя. Например, правило *ask\_budget* (листинг 5) необходимо, чтобы узнать у пользователя предпочитает он более бюджетную модель ноутбука или нет.

|  |
| --- |
| {  "rule\_name" : "ask\_budget",  "left":{"func": "and",  "arg": [{"func": "eq", "arg": [ "Solution", 0]},  {"func": "eq", "arg": [ "Budget", 0]}]  },  "right": {"func": "set",  "arg": ["Budget",  {"func": "yes\_or\_no", "arg": "Do you  need a more budget laptop option?:"}]  }  } |
| Листинг 5. Правило *ask\_budget* |

Левая часть правила задана набором условных элементов, необходимых для сопоставления с уже имеющимися фактами. В данном примере – *solution* (наличие итогового заключения системы) и *budget* (наличие записанного ответа пользователя).

Правая часть правила содержит список действий, выполняемых при активизации правила механизмом логического вывода. В данном примере – при соблюдении всех условий левой части, пользователю будет задан вопрос, ответ на который будет записан в факт *budget*.

Аналогично устроены все правила, предназначенные для заполнения фактов базы знаний: *design* (наличие особого дизайна ноутбука), *unusual\_color* (наличие яркой расцветки), *keyboard\_backlight* (наличие подсветки клавиатуры), *budget* (предпочтение более бюджетной модели), *for\_work* (необходимость ноутбука для работы), *display\_size* (размер дисплея), *touchscreen* (наличие сенсорного дисплея).

Второй тип правил – предназначенные для вывода итогового заключения экспертной системы.

Рассмотрим пример с таким правилом *answer\_type\_1* (листинг 7).

|  |
| --- |
| {  "rule\_name" : "answer\_type\_1",  "left":{"func": "and",  "arg": [{"func": "eq",  "arg": ["Solution", 0]},  {"func": "eq",  "arg": ["Budget", "yes"]},  {"func": "eq",  "arg": ["Keyboard", "yes"]},  {"func": "eq",  "arg": ["Design", "yes"]},  {"func": "eq",  "arg": ["Color", "yes"]}]  },  "right": {"func": "and",  "arg": [{"func": "set",  "arg": ["Solution", 1]},  {"func": "print",  "arg": "A laptop will do for you HP  Pavilion 15-eh 1159ur 601D7EA"}]  }  } |
| Листинг 7. Правило *answer\_type\_1* |

В данном случае в левой части правила перечислены все факты и значения, которые они должны иметь, чтобы можно было вынести решение в пользу той или иной модели ноутбука.

В правую часть добавлены команды по выводу ответа пользователю и присвоению факту *solution* утвердительного значения, что позволяет остановить процесс логического вывода.

Аналогично прописаны все правила на формирование заключения системы. Полный набор правил представлен в Приложении 2.

Важным в разработанной экспертной системе является использование логического оператора *and*, работа которого реализована через функцию *andParams* (листинг 8).

|  |
| --- |
| def andParams(\*params):  return all([el == True for el in params]) |
| Листинг 8. Реализация функции *andParams* |

Для обработки правил предназначена функция *ruleHandler* (листинг 9), которая перебирает все имеющиеся правила и проверяет подходят ли они под уже имеющиеся в рабочей памяти факты или нет.

|  |
| --- |
| def ruleHandler():  i = 0;  while(fact["Solution"] == 0 and i < len(data)):  if (ruleManager(data[i]["left"])):  ruleManager(data[i]["right"])  i += 1 |
| Листинг 9 – Реализация функции *ruleHandler* |

Так с помощью функции *ruleManager* (листинг 10) проверяется сначала левая часть правила (условие). А затем, при соблюдении необходимых требований, той же функцией выполняется правая часть. *RuleManager* – использует рекурсию, чтобы последовательно обработать все перечисленное в той или иной части правила функции.

|  |
| --- |
| def ruleManager(part):  final\_arguments = []  if(isinstance(part["arg"], str)):  final\_arguments.append(part["arg"])  else:  for arg in part["arg"]:  if not(isinstance(arg, dict)):  final\_arguments.append(arg)  else:  final\_arguments.append(ruleManager(arg))  return functions.get(part["func"])(\*final\_arguments) |
| Листинг 10 – Реализация функции *ruleManager* |

Так, например, для правой части правила *ask\_budget* очередность действий имеет вид:

1. С помощью функции *yes\_or\_no* задать вопрос пользователю;

2. С помощью функции *set* занести полученный ответ в факт *Design*.

Листинг разработанной продукционной экспертной системы представлен в Приложении 3.

**Примеры работы программы**

Ниже представлено несколько примеров работы программы и выводы, которые она сделала, опираясь на полученные от пользователя ответы:

|  |  |
| --- | --- |
| 1. |  |
| 2. |  |
| 3. |  |

**Заключение**

В рамках лабораторной работы были сформированы навыки проектирования и разработки экспертной системы, основанной на продукционной модели представления базы знаний.

Результатом работы являются:

* Описание базы знаний в нотации IF–THEN;
* Оформленный граф И/ИЛИ;
* Скрипт правил в формате JSON;
* Функционирующая продукционная экспертная система, основанная на составленной базе знаний.

Приложение 1

|  |
| --- |
| E:\PSTU\Интеллектуальные ИСИТ (Столбов)\lab_1\graf_2.png |
| Рисунок 1. Граф И/ИЛИ для задачи выбора ноутбука |

Приложение 2

**Скрипт правил в формате JSON**

[

{

"rule\_name" : "ask\_design",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Design", 0 ] } ] },

"right": { "func": "set", "arg": [ "Design", { "func": "yes\_or\_no", "arg": "Is laptop design important?: "} ] }

},

{

"rule\_name" : "ask\_unusual\_color",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": [ "Color", 0 ] } ] },

"right": { "func": "set", "arg": [ "Color", { "func": "yes\_or\_no", "arg": "Do you want a laptop with an unusual color?: "} ] }

},

{

"rule\_name" : "ask\_keyboard\_backlight",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": [ "Keyboard", 0 ] } ] },

"right": { "func": "set", "arg": [ "Keyboard", { "func": "yes\_or\_no", "arg": "Do you want a keyboard backlight?: "} ] } },

{

"rule\_name" : "ask\_for\_work",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", 0 ] } ] },

"right": { "func": "set", "arg": [ "Work", { "func": "yes\_or\_no", "arg": "Do you need a laptop for work?: " } ] }

},

{

"rule\_name" : "ask\_display\_size",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "no" ] }, { "func": "eq", "arg": ["Size", 0] } ] },

"right": { "func": "set", "arg": [ "Size", { "func": "ask\_question", "arg": ["Choose the display size (13.3 / 15.6 / 17.3): ", 13.3, 15.6, 17.3]} ] }

},

{

"rule\_name" : "ask\_touchscreen",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "yes" ] }, { "func": "eq", "arg": ["Touchscreen", 0] } ] },

"right": { "func": "set", "arg": [ "Touchscreen", { "func": "yes\_or\_no", "arg": "Need a touchscreen display?: "} ] }

},

{

"rule\_name" : "answer\_type\_1",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Keyboard", "yes" ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": ["Color", "yes"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you HP Pavilion 15-eh 1159ur 601D7EA" } ] }

},

{

"rule\_name" : "answer\_type\_2",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Keyboard", "yes" ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": ["Color", "yes"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Acer Swift 3 SF314-42-R7EN" } ] }

},

{

"rule\_name" : "answer\_type\_3",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Keyboard", "no" ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": ["Color", "yes"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Acer Aspire 1 A115-32-P7AU" } ] }

},

{

"rule\_name" : "answer\_type\_4",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Keyboard", "no" ] }, { "func": "eq", "arg": [ "Design", "yes" ]}, { "func": "eq", "arg": ["Color", "yes"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Acer Aspire 3 A315-56-33Z3" } ]}

},

{

"rule\_name" : "answer\_type\_5",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Keyboard", "yes" ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": ["Color", "no"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you HP Pavilion Aero 13-be0069ur" } ] }

},

{

"rule\_name" : "answer\_type\_6",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Keyboard", "yes" ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": ["Color", "no"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Acer Swift 3 SF314-43" } ] }

},

{

"rule\_name" : "answer\_type\_7",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Keyboard", "no" ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": ["Color", "no"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Acer Aspire 3 A315-34-C9WH"} ] }

},

{

"rule\_name" : "answer\_type\_8",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Keyboard", "no" ] }, { "func": "eq", "arg": [ "Design", "yes" ] }, { "func": "eq", "arg": ["Color", "no"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Acer Aspire 3 A315-34" } ] }

},

{

"rule\_name" : "answer\_type\_9",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "no" ] }, { "func": "eq", "arg": ["Size", "13.3"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Lenovo ThinkBook 13s G2 ITL" } ] }

},

{

"rule\_name" : "answer\_type\_10",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "no" ] }, { "func": "eq", "arg": ["Size", "13.3"] }]},

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you ASUS ZenBook Flip 13 UX363EA-HP785W" } ] }

},

{

"rule\_name" : "answer\_type\_11",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "no" ] }, { "func": "eq", "arg": ["Size", "15.6"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you ASUS R522MA-BQ862W" } ] }

},

{

"rule\_name" : "answer\_type\_12",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "no" ] }, { "func": "eq", "arg": ["Size", "15.6"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you HUAWEI MateBook D 15" } ] }

},

{

"rule\_name" : "answer\_type\_13",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "no" ] }, { "func": "eq", "arg": ["Size", "17.3"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you HP 17-cp0125ur" } ] }

},

{

"rule\_name" : "answer\_type\_14",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "no" ] }, { "func": "eq", "arg": ["Size", "17.3"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you HP 17-CP0138UR" } ] }

},

{

"rule\_name" : "answer\_type\_15",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "yes" ] }, { "func": "eq", "arg": ["Touchscreen", "yes"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Lenovo IdeaPad Flex 5 14ITL05" } ] }

},

{

"rule\_name" : "answer\_type\_16",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "yes" ] }, { "func": "eq", "arg": ["Touchscreen", "yes"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you MSI Summit E16 Flip Evo A11MT-204RU" } ] }

},

{

"rule\_name" : "answer\_type\_17",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "yes" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "yes" ] }, { "func": "eq", "arg": ["Touchscreen", "no"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Acer Aspire 3 A315-56" } ] }

},

{

"rule\_name" : "answer\_type\_18",

"left": { "func": "and", "arg": [ { "func": "eq", "arg": [ "Solution", 0 ] }, { "func": "eq", "arg": [ "Budget", "no" ] }, { "func": "eq", "arg": [ "Design", "no" ] }, { "func": "eq", "arg": [ "Work", "yes" ] }, { "func": "eq", "arg": ["Touchscreen", "no"] } ] },

"right": { "func": "and", "arg": [ { "func": "set", "arg": [ "Solution", 1] }, { "func": "print", "arg": "A laptop will do for you Acer Aspire 3 A315-23-R384" } ] }

}

]

Приложение 3

**Листинг продукционной экспертной системы**

import json

def andParams(\*params):

return all([el == True for el in params])

def eq(param, value):

if not(param in fact):

fact[param] = 0

return fact[param] == value

def set(param, value):

fact[param] = value

return fact[param]

def yes\_or\_no(question):

answer = ""

while(answer != "yes" and answer != "no"):

answer = input(question)

return answer

def ask\_question(question, \*values):

answer = ""

while all([str(el) != answer for el in values]):

answer = input(question)

return answer

def printResult(text):

print(text)

functions = {"and": andParams, "eq": eq, "set": set, "yes\_or\_no": yes\_or\_no, "ask\_question": ask\_question, "print": printResult}

fact = {"Solution": 0}

def ruleManager(part):

final\_arguments = []

if(isinstance(part["arg"], str)):

final\_arguments.append(part["arg"])

else:

for arg in part["arg"]:

if not(isinstance(arg, dict)):

final\_arguments.append(arg)

else:

final\_arguments.append(ruleManager(arg))

return functions.get(part["func"])(\*final\_arguments)

def ruleHandler():

i = 0;

while(fact["Solution"] == 0 and i < len(data)):

if (ruleManager(data[i]["left"])):

ruleManager(data[i]["right"])

i += 1

with open("Base.json", "r", encoding="utf-8") as rules:

data = json.load(rules)

ruleHandler()