

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ по дисциплине «Прикладные
интеллектуальные системы и экспертные системы»

Исследование алгоритмов принятия решения при задании экспертной
информации в виде системы четких высказываний. Принятие решений на
основе четкого правила *modus ponens* и дедуктивной схемы вывода.

Алгоритмы прямой и обратной цепочки рассуждений

Студент

Глебов Д.А.

Группа М-ИАП-22

Руководитель

Кургасов В.В.

Липецк 2022 г.

Ход работы

Краткая теоретическая справка

Продукционный метод представления знаний. Понятие продукции. Различные формы записи продукции. Классификация продукций. Пример.

Продукция в общем виде – это выражение типа: $(i); P; Q; A \rightarrow B; N$.

(i) - имя продукции, номер

P – сфера применения, классификация, лексема

Q – предикат применимости, предварительные условия применимости, предусловие

$A \rightarrow B$ – ядро продукции

N – постусловие, предикат, который становится истинным после выполнения условия.

Продукция как система подстановок вида $X_i W \rightarrow W y_i$, где X_i, y_i - элементы алфавита $C = \{C_1, C_2, \dots, C_i\}$ цепочки и C_i - цепочка, W - неизменяемая часть слова, ввел в 1943 году Пост.

Пример: применить продукцию ψ к P_r ,

$P_r(\psi)$ значит $\psi = abc, aw \rightarrow wd$

$P_r(abc) = bcd$

Повторно применить продукцию не всегда возможно

$P_r(\psi_1, \psi_2), P = \text{true}$, если $P_r(\psi_1) = \psi_2$

Пост показал, что система продукций над словами некоторого алфавита эквивалентна машине Тьюринга. Решение задач в системе продукций заключается:

- фиксированный набор исходных данных (в виде набора слов в исходном алфавите)
- фиксированный набор продукций $P_{r1}, P_{r2} \dots$
- вся система продукций применяется поочередно к начальным словам, до тех пор, пока мы не получим набор слов, к которым нельзя применить ни одну продукцию.

Полученный набор конечных слов является результатом задачи. В 1972 году Ньюэлл обратил внимание, что система Поста эквивалентна некоторой системе правил, а Марков показал неявную запись алгоритма. Ядро продукции практически представляет собой подстановку.

\rightarrow - секвенция

A – условие

B – следствие

Стали считать запись вида : Если A, то B – правилом ($A \rightarrow B$)

B может быть условием или действием. Если A истинно, то B истинно.

Если A ложно, то о B ничего сказать нельзя.

Если A true, то выполняется последовательность действий B

Предусловие – это предикат применимости. Множество продукций представляет собой набор правил продукций. Таким образом схема решений по правилам аналогична решению по продукциям Поста. Исходные данные – это факты (36,6 - температура).

Продукции – правила (36,6 – нормальная, 39,1 - большая)

Множество фактов – описание предметной области.

Набор правил образует систему правил, каждое правило имеет имя: R1, R2...

Основные формы записи продукций:

- If – then
- If – then – else

Пример: If A then B

Сложность правила оценивается сложностью A.

Классификация ядер продукций.

Ядро:

- Детерминированное (с четкой логикой)
- Недетерминированные (с нечеткой логикой) – содержат лексемы неопределенности типа : Если A, то вероятно B; Если A, то с вероятностью γ B; Если A, то чаще B, иначе реже C.

Построение логического вывода в продукционной системе. Приоритеты и стоимость правил.

На первый взгляд процесс вывода кажется достаточно простым – выполняются однотипные операции по перебору записей БЗ и сравнении их с имеющимися фактами, пока не будет найдено решение или некий целевой факт. Однако, управление процессом вывода, независимое от контекста проблемы на практике мало эффективно. При решении реальных задач человек крайне редко прибегает к перебору данных. Вместо этого, люди пользуются эвристическими правилами, которые значительно ограничивают пространство поиска решения и позволяет быстро и эффективно решать задачи. Эвристические знания имеют эмпирическую природу, то есть формируются на базе опыта и интуиции эксперта.

Существует два основных типа логического вывода: прямой и обратный. Прямой вывод соответствует обычному ходу решения задачи – от исходных фактов к целевым. Примером прямого вывода является задача классификации. ЭС осуществляет постепенное обобщение исходных фактов, описывающих свойства исследуемого объекта, выявляя наиболее характерные признаки того или иного класса.

Обратный вывод соответствует, как следует из названия, обратной задаче – определить какие именно факты требуются для подтверждения данной цели. Этот тип вывода соответствует противоположному ходу решения: сначала машина вывода рассматривает те правила БЗ, действием которых является вывод целевого факта. Затем выбираются новые подцели из условий этих правил, и процесс продолжается от целевых фактов к исходным. Можно сказать, что при обратном выводе происходит конкретизация свойств исследуемого объекта.

Этот вид логического вывода наделяет ЭС новым фундаментальным свойством – способностью объяснить, как было получено решение, или что требуется, для того, чтобы имел место тот или иной факт. В реальных

системах, как правило, используется комбинация из прямого и обратного вывода.

В процессе вывода решения все правила системы равнозначны и самодостаточны, то есть все необходимое для активизации правила содержится в его условии, и одни правила не могут непосредственно вызывать другие. Но иногда для получения решения требуется вмешательство в стандартный процесс вывода. Для этих целей некоторые продукционные системы позволяют вводить в базу знаний специальные правила для управления процессом вывода – метаправила. Метаправила не принимают непосредственного участия в процессе формирования рассуждений, а определяют приоритет выполнения или исключают из рассмотрения обычные правила и выполняются в первую очередь.

Прямая цепочка рассуждений и алгоритм ее реализации.

В простейшем случае применение правил сводится к попыткам поочередной инициализации правил в том порядке, в котором они записаны. Обычно вводится целевая переменная (GOAL, G). Считается, что применение продукций может быть закончено, если достигнуто значение целевой переменной. Идея прямой цепочки рассуждений заключается в последовательном применении правил по порядку, с возвратом после просмотра всех правил снова в начало, до тех пор, пока не будет получено значение целевой переменной. Мы можем определить для данного значения правила несколько переменных или переопределить. В любой продукционной системе можно получить значение переменной тремя способами:

1. Начальное присваивание, т.е. перед применением правил
2. Из правил – переменная стоит в следствии правила
3. Запрос у пользователя и ввод значения в диалоге

Недостатки: избыточность, попытка инициализировать ненужные правила

Достоинства: простота алгоритма, простота реализации.

Реализация алгоритма:

1. Завести массивы

Очередь вывода

Таблица переменных

Перем	№ правила	№ условия	Текущ.значение	Следствие

Текущее состояние системы

Переменная	Правило	№ условия	Назначение	Следствие

Алгоритм ПЦР:

1. Фиксируется исходное состояние системы, в очередь переменных вывода заносится целевая переменная, формируется таблица переменных с текущими значениями переменных.
2. Организуется цикл просмотра правил в том порядке, в котором они записаны, а внутри проверка условий в той последовательности, в которой они стоят в конъюнкциях. Осуществляется проверка первого правила, в нём первого условия, а в условии первой переменной. Если этой переменной нет в таблице переменных, то выдаётся сообщение об отсутствии решения. Если она имеется в таблице и у неё есть значение, то осуществляется переход к следующей переменной первой конъюнкции. Если значение переменной равно NIL, то проверяется можно ли её запросить у пользователя, если Да, то значение запрашивается, если Нет, то переменная помещается в очередь переменных вывода и осуществляется переход к проверке следующего правила. Если первая конъюнкция истинна, то переход к проверке второй конъюнкции, и она проводится аналогично. Если конъюнкция ложна, то осуществляется переход к проверке следующего правила.
3. Если посылка правила истинна, то реализуется действия части “То” правила (т.е действия, стоящего в следствие). В результате этих действий состояние системы меняется, что отображается в таблице текущего состояния системы (TSS), таблица переменных обновляется, а те переменные, которые получили значение, вычеркиваются из таблицы очереди вывода. Если значение

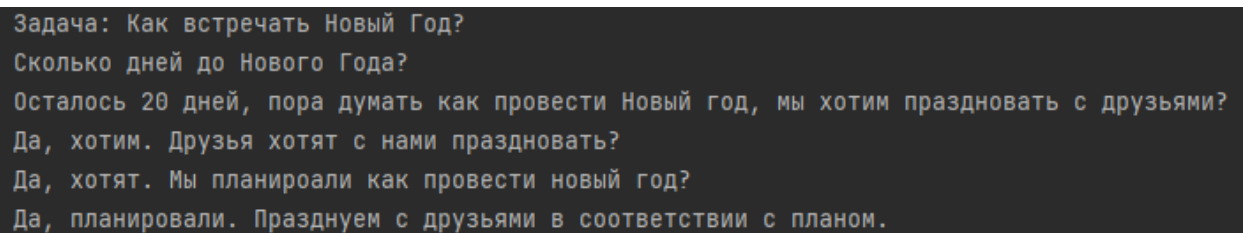
получила целевая переменная, то логический вывод завершается и осуществляется выход из цикла проверки правил (т.е конец – цель достигнута), иначе переход к проверке следующего правила.

Пример выполнения задания

Параметры №1:

```
parametr = {  
    'days_left' : 20,  
    'We_want_to_be_with_friends' : True,  
    'friends_answer' : True,  
    'planned' : True,  
    'Family' : True,  
}
```

Результат работы программы с приведёнными выше параметрами:



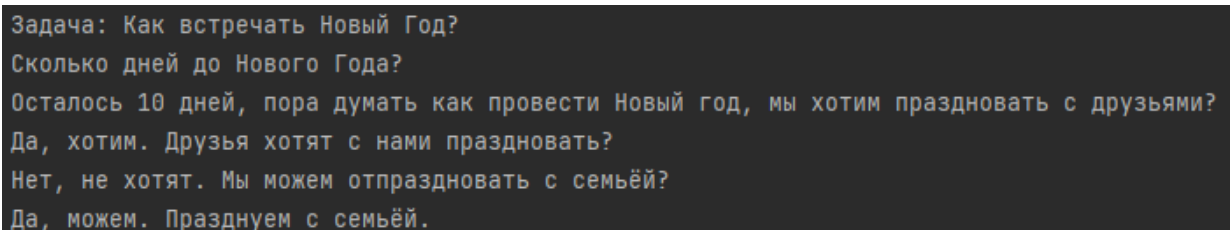
Задача: Как встречать Новый Год?
Сколько дней до Нового Года?
Осталось 20 дней, пора думать как провести Новый год, мы хотим праздновать с друзьями?
Да, хотим. Друзья хотят с нами праздновать?
Да, хотят. Мы планировали как провести новый год?
Да, планировали. Празднуем с друзьями в соответствии с планом.

Рисунок 1 – Результат работы программы

Параметры №2:

```
parametr = {  
    'days_left' : 10,  
    'We_want_to_be_with_friends' : True,  
    'friends_answer' : False,  
    'planned' : True,  
    'Family' : True,  
}
```

Результат работы программы с приведёнными выше параметрами:



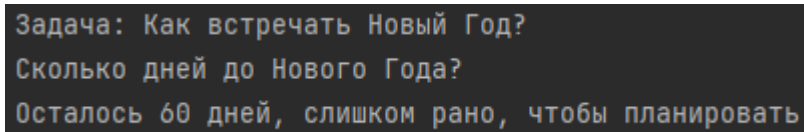
Задача: Как встречать Новый Год?
Сколько дней до Нового Года?
Осталось 10 дней, пора думать как провести Новый год, мы хотим праздновать с друзьями?
Да, хотим. Друзья хотят с нами праздновать?
Нет, не хотят. Мы можем отпраздновать с семьёй?
Да, можем. Празднуем с семьёй.

Рисунок 2 – Результат работы программы

Параметры №3:

```
params = {  
    'days_left' : 60,  
    'We_want_to_be_with_friends' : True,  
    'friends_answer' : False,  
    'planned' : True,  
    'Family' : True,  
}
```

Результат работы программы с приведёнными выше параметрами:

A screenshot of a terminal window with a dark background and light-colored text. It displays three lines of output: 'Задача: Как встречать Новый Год?', 'Сколько дней до Нового Года?', and 'Осталось 60 дней, слишком рано, чтобы планировать'.

```
Задача: Как встречать Новый Год?  
Сколько дней до Нового Года?  
Осталось 60 дней, слишком рано, чтобы планировать
```

Рисунок 3 – Результат работы программы