МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра вычислительной техники

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г. Локтионова
«14» 12 2017 г.

Создание продукционных моделей

Методические указания
по выполнению лабораторной работы по дисциплине
«Организация систем искусственного интеллекта»
для студентов направления подготовки 09.03.01
Информатика и вычислительная техника

Курск 2017 г.

УДК 004.89
Составители: С.В. Дегтярев, Е.Н. Иванова

Рецензент
Профессор кафедры биомедицинской инженерии,
доктор технических наук
С.А. Филист

Создание продукционных моделей: методические указания по выполнению лабораторной работы / Юго-Зап. гос. ун-т; сост.: С.В. Дегтярев, Е.Н. Иванова. — Курск, 2017. — 18 с.: Библиограф.: с. 18.

Рассматривается методика построения продукционных моделей для решения интеллектуальных задач в определенной предметной области. Приводится пример применения указанной методики для конкретной задачи.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по направлению Информатика и вычислительная техника.

Предназначены для студентов направления 09.03.01 Информатика и вычислительная техника очной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать 14.12.12. Формат 60x84 1/16.
Усл.печ.л. 0,9. Уч.-изд.л. 0,8. Тираж 20 экз. Заказ 2574. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

1. Цель

Приобретение умения самостоятельно строить продукционные модели для решения сложных интеллектуальных задач, связанных с анализом, диагностикой или поддержкой принятия решений в выбранной проблемной области; овладение навыками описания предметной области в виде продукционных правил; получение опыта формулировки запросов и оформления интерфейса задачи.

2. Введение

Продукционные модели (ПМ) представляют собой комбинацию элементов логических и сетевых моделей. Из логических моделей взята идея правил вывода, которые здесь называют продукциями, а из сетевых моделей — описание знаний в виде семантической сети. В результате применения правил вывода к фрагментам сетевого описания происходит трансформация семантической сети за счет замены ее фрагментов, наращивания сети и исключения из нее ненужных фрагментов. В продукциях отсутствуют жесткие ограничения, характерные для логических исчислений, что дает возможность изменять интерпретацию элементов продукции. Системы продукций (СП) очень широко распространены для построения баз знаний экспертных систем.

Продукционные модели базируются на понятии „формальная продукционная система“ и берут свое начало с работ Е. Поста, который в 1943 году ввел термины „продукция и каноническая (продукционная) система“. Пост доказал, что продукционная система является логической системой, эквивалентной машине Тьюринга. Продукционные системы универсальны, т.е. любая формальная система, оперирующая символами, может быть реализована в виде одной из

продукционных систем Поста.

Идея продукции используется в языках логического программирования, нашедших применение в системах искусственного интеллекта (СИИ). Среди языков подобного типа наиболее известен Пролог. Продукционный подход стал стилем нового этапа программирования в 90-е годы.

Продукционные модели в настоящее время широко используются в системах представления знаний и были впервые применены в СИИ в 1972 г.

3. Представление продукционных моделей

В том практическом смысле, который представляет для нас интерес, продукция есть правило–продукция, представляющая собой пару: ситуация $\rightarrow$ действие, посылка $\rightarrow$ заключение, причина $\rightarrow$ следствие и т.п. Подобного рода правила встречаются в различных областях знаний и видах деятельности; в повседневной жизни мы постоянно окружены различного рода правилами поведения, уличного движения, грамматическими правилами.

Если говорить о программировании, то продукция выступает как тройка: имя продукции, условие применимости, оператор. В некоторых случаях продукция близка по смыслу импликации „если–то“, так что можно принять для продукции обозначение в виде импликации:

$$\alpha \rightarrow \beta,$$

а если требуется раскрыть более подробно условие применимости, то можно использовать запись следующего вида:

$$P_1 \& P_2 \& \cdots \& P_n \rightarrow B,$$

где P_i, $i = 1, 2, \cdots, n$ — условия применимости, образующие конъюнктивную форму;

B — заключение, которое может трактоваться и как действие (что отличает продукции от импликаций).

4. Построение решателя продукционного типа

В состав продукционных моделей входят:

База Знаний (БЗ), записанная в виде продукционных правил (знания представлены в виде предложений типа: ЕСЛИ условие, ТО действие);

Рабочая Память (РП) и Интерпретатор Правил (Решатель), реализующий определенный механизм логического вывода.

Любое продукционное правило, содержащееся в базе знаний, состоит из двух частей: условие (антецедент — предшествующий) и заключение (консеквент — следствие, вывод). Антецедент представляет собой условную часть правила (посылку) и состоит из элементарных высказываний, соединенных логическими связками „И“(&), „ИЛИ“($\vee$), „НЕ“($\neg$). Консеквент может включать одно или несколько высказываний, которые выражают либо некоторый факт, либо указание на определенное действие, подлежащее исполнению.

Антецедент и консеквент связаны логической операцией „импликация“($\rightarrow$), означающей, что из данного условия (антецедент) следует определённое заключение (консеквент).

Рассмотрим, как будут выглядеть декларативные знания, записанные в виде продукционных правил:

1. Если Ваш бюджет ограничен (B), и стоимость фотоаппарата имеет для Вас существенное значение (Z), мы предлагаем вам простые фотоаппараты-мыльницы (M): $B\&Z \rightarrow M$.

2. Если ваш бюджет ограничен (B) и для вас важно качество фотографий (K), то Вам подойдут фотоаппараты мыльницы с отличным качеством снимков (MK), стоимостью выше 7-8 тыс. руб.: $B\&K \rightarrow MK$.
3. Если ваш бюджет ограничен (B) и для Вас важно наличие видео съемки (V), Вам подойдут такие фотоаппараты, как Nikon Coolpix S3100 Red и проч (N): $B\&V \rightarrow N$.
4. Если Вы не ограничены в средствах (B), то Вам могут быть доступны профессиональные фотоаппараты (P): $\neg B \rightarrow P$.

Все оставшиеся правила можно записать точно так же. В результате получится система продукционных правил, которые и составят Базу Знаний (БЗ) продукционной системы. В Рабочей Памяти (РП) систем, основанных на продукционных моделях, хранятся факты (в виде высказываний) истинность которых была задана исходно или была установлена к текущему моменту времени в процессе решения конкретной задачи.

Продукционная система должна найти решение поставленной задачи (идентифицировать объект, диагностировать поломку, дать прогноз и т.д.), сопоставляя имеющиеся факты, хранящиеся в рабочей памяти компьютера, и правила логического вывода, хранящиеся в базе знаний.

Если в рабочей памяти есть факты (образцы), соответствующие каждому из элементарных высказываний, входящих в условие (антецедент), то Правило считается выполненным (срабатывает). В этом случае заключение сработавшего правила заносится в рабочую память и считается подтвержденным фактом.

Таким образом, в процессе решения задачи содержимое рабочей памяти изменяется. Это происходит по мере срабатывания правил. Поэтому в процессе логического вывода объём фактов в рабочей памяти, как правило, увеличива-

ется (уменьшаться он может в том случае, если действие какого-нибудь правила состоит в удалении фактов из рабочей памяти).

В процессе логического вывода последовательно (или по определенному принципу) просматриваются все правила, записанные в базе знаний. Если в рабочей памяти нет фактов об истинности какого-то высказывания, входящего в условие (антецедент), то правило пропускают и переходят к проверке условия следующего правила. В процессе логического вывода каждое правило из базы знаний может сработать только один раз (потом оно уже не проверяется). Часто для получения решения бывает необходимо сделать несколько итераций (проверить соответствие имеющихся в рабочей памяти фактов продукционным правилам) и принять решение, используя правила из базы знаний, но уже с учётом новых фактов — тех заключений, которые были записаны в рабочую память на предыдущих итерациях.

Основные достоинства систем, основанных на продукционных моделях, связаны с простотой представления знаний и простой организации вывода решения.

5. Построение решателя (прямой и обратный вывод)

Существуют два типа продукционных моделей — с „прямыми“ и „обратными“ выводами. Прямые выводы реализуют стратегию „от фактов к заключениям“. При обратных выводах выдвигаются гипотезы вероятных заключений, которые могут быть подтверждены или опровергнуты на основании фактов, поступивших в рабочую память. Существуют также системы с двунаправленными выводами.

Рассмотрим, как осуществляется логический вывод (решение) в продукционной модели.

5.1. Прямой вывод

Пусть в Базе Знаний имеются следующие правила:
Правило 1. „ЕСЛИ *A* И *B*, ТО *C*“: $A\&B \rightarrow C$
(если одновременно имеют место *A* и *B*, то из этого следует *C*).
Правило 2. „ЕСЛИ *D*, ТО *A*“: $D \rightarrow A$
(если имеет место *D*, то из этого следует *A*).

Предположим, что в Рабочую Память от пользователя поступили факты о *B* и *D* (РП: *B* и *D*).

Основные шаги алгоритма прямого вывода.

Правило срабатывает, если в РП есть образцы (факты), совпадающие с антецедентом (посылками, условиями) этого правила.

1. Сопоставим факты из РП с образцами правил из БЗ.

Правило 1 не может сработать, т.к. в РП нет данных об утверждении *A*.

Правило 2 срабатывает, так как в его антецеденте есть утверждение *D*, о котором есть данные в РП.

2. В РП заносится заключение сработавшего правила.

Правило 2 — имеет место утверждение *A*. Теперь содержимое рабочей памяти станет следующим:

РП: *B*, *D* и *A*.

3. Второй цикл сопоставления фактов в РП с образцами правил.

Теперь срабатывает Правило 1, так как конъюнкция условий (*A*&*B*) в его антецеденте становится истинной.

4. В РП заносится заключение сработавшего Правила 1 — имеет место *C*.

В результате действия Правила 1 пользователю выдается окончательный диагноз — из имеющихся и полученных в результате работы фактов следует утверждение *C*.

5. Конец работы (БП исчерпана).

5.2. Ппрямой вывод с конфликтным набором

Пусть в БЗ кроме Правила 1 и Правила 2 есть Правило 3:

Правило 1. „ЕСЛИ A И B, ТО C“: $A\&B \rightarrow C$.
Правило 2. „ЕСЛИ D, ТО A“: $D \rightarrow A$.
Правило 3. „ЕСЛИ D, ТО P“: $D \rightarrow P$.
РП: B и D.

В результате сопоставления в первом же цикле возможно применение двух правил — Правила 2 и Правила 3, т.е. возникает конфликтный набор и встает задача выбора: какое из этих правил применить первым. Если выберем Правило 2, то в РП добавится факт A, и на следующем шаге опять возникнет конфликтный набор, так как можно будет применить Правило 1 и Правило 3. Если будет выбрано Правило 1, то к заключению C мы придём за два шага. При любом другом выборе порядка применения правил к этому же заключению мы придём за три шага. Если завершение цикла работы наступает после просмотра всех правил, то число шагов будет равно трём, причем порядок применения правил не будет иметь какого-либо значения.

5.3. Обратный вывод

Пусть в БЗ имеется три правила (Правило 1, Правило 2 и Правило 3):

Правило 1. „ЕСЛИ A И B, ТО C“: $A\&B \rightarrow C$.
Правило 2. „ЕСЛИ D, ТО A“: $D \rightarrow A$.
Правило 3. „ЕСЛИ D, ТО P“: $D \rightarrow P$.
РП: B и D.

Основные шаги алгоритма обратного вывода.

1. Выдвигается гипотеза окончательного диагноза — (C?) — имеет место утверждение C?

2. Отыскиваем правило, заключение которого содержит выдвинутую гипотезу, в нашем примере — это Правило 1

($A\&B \rightarrow C$).

3. Исследуем возможность применения Правила 1, т.е. решается вопрос о том, может ли оно сработать. Для этого в рабочей памяти должны присутствовать факты, совпадающие с образцами антецедента этого правила. В рассматриваемом примере Правило 1 не может сработать из-за отсутствия в РП данных об A.

Проверка факта A становится новой целью на следующем шаге. Выдвигается гипотеза: (A?) — имеет место утверждение A?

4. Отыскиваем правило, заключение которого соответствует новой цели: ищем правило, заключение которого содержит выдвинутую гипотезу (A?). Такое правило есть — Правило 2 ($D \rightarrow A$).

5. Исследуем возможность применения Правила 2. Проводим сопоставление с данными в РП. Правило срабатывает, так как в РП присутствует факт, совпадающий с его образцом (D).

6. Действие Правила 2 позволяет занести в рабочую память заключение: утверждение A имеет место.

РП: B, D и A.

7. Условная часть Правила 1 теперь подтверждена фактами (A и B есть в рабочей памяти), следовательно, это правило срабатывает, и выдвинутая начальная гипотеза (C?) подтверждается.

8. Конец работы.

При сравнении этого примера с примером прямого вывода нет заметных преимуществ обратного вывода перед прямым.

5.4. Обратный вывод с конфликтным набором

Предположим, в БЗ записаны Правило 1, Правило 2, Правило 3 и Правило 4:

Правило 1. „ЕСЛИ A И B, ТО C“: $A\&B \to C$.
Правило 2. „ЕСЛИ D, ТО A“: $D \to A$.
Правило 3. „ЕСЛИ D, ТО P“: $D \to P$.
Правило 4. „ЕСЛИ S, ТО A“: $S \to A$.
РП: B и D.

В данном случае алгоритм обратного вывода с конфликтным набором включает следующие шаги:

1. Выдвигается гипотеза: (C?).

2. Ищем правило, заключение которого совпадает с поставленной целью. Это Правило 1 — $A\&B \to C$

3. Исследуем возможность применения Правила 1.

Оно не может сработать, выдвигается новая подцель, соответствующая недостающему образцу: (A?).

4. Ищем правило, заключение которого совпадает с новой подцелью.

Таких правил два — Правило 2 ($D \to A$) и Правило 4 ($S \to A$). Если выберем Правило 2, то дальнейшие шаги совпадут с примером без конфликтного набора. Если выберем Правило 4, то оно не сработает, так как в РП нет данных об S. Поэтому будет применено Правило 2, что приведет к успеху, но путь окажется длиннее на один шаг.

Следует обратить внимание на то, что Правило 3, не связанное с поставленной целью, вообще не затрагивалось в процессе вывода. Этот факт свидетельствует о более высокой эффективности обратных выводов по сравнению с прямыми, так как при обратных выводах существует тенденция исключения из рассмотрения правил, не имеющих отношение к поставленной цели.

6. Исходные данные

Примерный круг предметных областей:
1) лекарственные растения (грибы, ягоды);

2) покупка компьютера (автомобиля, квартиры);
3) кулинария;
4) породы домашних (диких) животных;
5) починка телевизора (автомобиля, компьютера);
6) лечение собаки (кошки, человека).

Требование к Базе Знаний: количество правил должно быть не менее 30.

При построении продукционной модели важно правильно разработать структуру продукционного правила. Некоторые рекомендации:
— конструируйте правила, опираясь на структуру, присущую предметной области;
— используйте минимально достаточное количество условий при определении продукционного правила;
— избегайте противоречащих продукционных правил.

Каждое продукционное правило может быть независимым от других. Модульность правил позволяет легко модифицировать продукционную систему. Модификация заключается в добавлении, изменении, удалении правил и не затрагивает существующих процедур. Число правил в системе ограничено размерами памяти компьютера.

7. Порядок выполнения работы

1. Выбрать предметную область. Четко (письменное) сформулировать цель создания системы.

2. Формально описать предметную область. В качестве формализма для описания должны быть выбраны продукционные правила.

3. Представить предметную область на языке ПРОЛОГ.

4. Сформулировать запросы и оформить интерфейс.

5. Оформить отчет.

8. Пример выполнения работы

Рассмотрим семь животных распространенных пород. Ниже приведены продукционные правила, задающие описание животных. Здесь биологический класс — это птицы или млекопитающие.

Правило 1.
ЕСЛИ животное имеет волосы,
ТО это животное млекопитающее.

Правило 2.
ЕСЛИ животное дает молоко,
ТО это животное млекопитающее.

Правило 3.
ЕСЛИ животное имеет перья,
ТО это животное птица.

Правило 4.
ЕСЛИ животное умеет летать И несет яйца,
ТО это животное птица.

Правило 5.
ЕСЛИ животное — млекопитающее И ест мясо,
ТО это хищник.

Правило 6.
ЕСЛИ животное — млекопитающее И имеет острые зубы, И имеет когти, И имеет посаженные впереди глаза,
ТО это хищник.

Правило 7.
ЕСЛИ животное — млекопитающее И имеет копыта,
ТО это копытное.

Правило 8.
ЕСЛИ животное — млекопитающее И жует жвачку,
ТО это копытное.

Правило 9.
ЕСЛИ животное — хищник И имеет рыжевато-коричневую

окраску, И имеет темные пятна,
ТО это леопард.

Правило 10.
ЕСЛИ животное — хищник И имеет рыжевато-коричневую окраску, И имеет черные полосы,
ТО это тигр.

Правило 11.
ЕСЛИ животное — копытное И имеет длинные ноги, И имеет длинную шею, И имеет рыжевато-коричневую окраску, И имеет темные пятна,
ТО это жираф.

Правило 12.
ЕСЛИ животное — копытное И имеет белый цвет, И имеет черные полосы,
ТО это зебра.

Правило 13.
ЕСЛИ животное — птица И не умеет летать И имеет длинные ноги, И имеет длинную шею, И имеет бело-черную окраску,
ТО это страус.

Правило 14.
ЕСЛИ животное — птица И не умеет летать, И умеет плавать, И имеет бело-черную окраску,
ТО это пингвин.

Правило 15.
ЕСЛИ животное — птица И умеет очень хорошо летать,
ТО это альбатрос.

Работа системы распознавания сводится к генерации гипотезы о принадлежности животного к тому или иному классу и к попытке подтвердить эту гипотезу. В нашем случае генерируется первая гипотеза: „распознаваемое животное — это млекопитающее“. Для подтверждения данной гипотезы необходимо, чтобы пользователь утвердительно отве-

тил хотя бы на один из вопросов: „имеет ли животное волосы“ или „дает ли животное молоко“. Если положительный ответ получен уже на первый вопрос, то система генерирует следующую гипотезу „это млекопитающее — хищник“. Если же положительный ответ не получен на первый вопрос, то система задает второй вопрос. Если на него получен положительный ответ, генерируется гипотеза „млекопитающее — хищник“, если получен отрицательный ответ, генерируется гипотеза: „распознаваемое животное — птица“. Процесс порождения гипотез и их проверки длится до тех пор, пока есть подходящие для этого правила. Описание таких правил приведено ниже:

```
rule(1,"животное","млекопитающее",[1]).
rule(2,"животное","млекопитающее",[2]).
rule(3,"животное","птица",[3]).
rule(4,"животное","птица",[4, 5]).
rule(5,"млекопитающее","хищник",[6]).
rule(6,"млекопитающее","хищник",[7, 8, 9]).
rule(7,"млекопитающее","копытное",[10]).
rule(8,"млекопитающее","копытное ",[11]).
rule(9,"хищник","леопард",[12, 13]).
rule(10,"хищник","тигр",[12, 14]).
rule(11,"копытное","жираф",[15, 16, 12, 13]).
rule(12,"копытное","зебра",[17, 14]).
rule(13,"птица","страус",[18, 15, 16, 19]).
rule(14,"птица","пингвин",[18, 20, 19]).
rule(15,"птица","альбатрос",[21]).
```

Первый аргумент в предикате rule — это номер правила, второй — род, третий — вид, четвертый — список вопросов, подтверждающий отношение род–вид.

Для того чтобы применить ту или иную продукцию, необходимо собрать факты, задав пользователю вопросы. Однако, прежде чем задавать вопрос, необходимо быть уве-

ренным в том, что этот вопрос уже не был задан ранее при подтверждении других промежуточных гипотез. Информация о заданном вопросе и полученном на него ответе хранится в отношении fact(*X*, *Y*) динамической базы данных, где *X* — номер вопроса, *Y* — ответ на этот вопрос („да“, „нет“). Если вопрос был уже задан и на него получен положительный ответ, то вывод успешно продолжается, если же получен отрицательный ответ, то система сообщает о неуспехе. Множество задаваемых вопросов приведено ниже.

```
ask(X):- fact(X, "да"),!.
ask(X):- fact(X, "нет"),!,fail.
ask(1):- write("оно имеет волосы?"), !, complete(1).
ask(2):- write("оно дает молоко?"), !, complete(2).
ask(3):- write("оно имеет перья?"), !, complete(3).
ask(4):- write("оно умеет летать?"), !, complete(4).
ask(5):- write("оно несет яйца?"), !, complete(5).
ask(6):- write("оно ест мясо?"), !, complete(6).
ask(7):- write("оно имеет острые зубы?"), !, complete(7).
ask(8):- write("оно имеет когти?"), !, complete(8).
ask(9):- write("оно имеет посаженные впереди глаза?"), !,
complete(9).
ask(10):- write("оно имеет копыта?"), !, complete(10).
ask(11):- write("оно жует жвачку?"), !, complete(11).
ask(12):- write("оно имеет рыжевато-коричневую окраску?"),
!, complete(12).
ask(13):- write("оно имеет темные пятна?"), !, complete(13).
ask(14):- write("оно имеет черные полосы?"), !, complete(14).
ask(15):- write("оно имеет длинные ноги?"), !, complete(15).
ask(16):- write("оно имеет длинную шею?"), !, complete(16).
ask(17):- write("оно имеет белый цвет?"), !, complete(17).
ask(18):- write("оно не умеет летать?"), !, complete(18).
ask(19):- write("оно имеет бело-черную окраску?"), !,
complete(19).
```

```
ask(20):- write("оно умеет плавать?"), !, complete(20).
ask(21):- write("оно умеет очень хорошо летать?"), !,
complete(21).
```

Процедура recognition(*X*) занимает центральное место в программной реализации продукционной системы. Процедура состоит из трех предложений. В первом предложении генерируется гипотеза (rule(*N*, *X*, *Y*, *Z*)) и ищется ее подтверждение (discover(*Z*)); если гипотеза подтверждается, то выдается соответствующее сообщение, если выдвинутая гипотеза не подтверждается, то генерируется следующая. Второе предложение процедуры описывает ситуацию, когда пользователь на все вопросы ответил отрицательно и системе не удалось выдвинуть ни одной гипотезы. И наконец, третье предложение задает успешное окончание работы, когда была подтверждена хотя бы одна гипотеза.

```
recognition(X):- rule(N, X, Y, Z), discover(Z), !, write(„___“,
X, „–“, Y, „по правилу “, N), nl, recognition(Y).
recognition("животное"):- write("это животное мне неизвест-
но"),!.
recognition(_).
discover([]).
discover([X|Y]):- ask(X), discover(Y).
complete(X):- nl, read(Y), assert(fact(X, Y)), Y="да".
```

Рассмотрим пример работы системы.

```
?- retractall(_), recognition("животное").
"оно имеет волосы?"
да
___ животное - млекопитающее по правилу 1
"оно ест мясо?"
нет
"оно имеет острые зубы?"
да
```

"оно имеет когти?"
да
"оно имеет посаженные впереди глаза?"
да
___ млекопитающее - хищник по правилу 6
"оно имеет рыжевато-коричневую окраску?"
да
"оно имеет темные пятна?"
нет
"оно имеет черные полосы?"
да
___ хищник - тигр по правилу 10

Пример показывает полное распознавание животного.

9. Литература

1. Джарратано, Джозеф. Экспертные системы: принципы разработки и программирование [Текст] / Райли, Гарри. - М.: Вильямс, 2005. – 1152 с.
2. Братко, Иван. Язык PROLOG (Пролог): алгоритмы искусственного интеллекта [Текст] / И. Братко. - М.: Вильямс, 2000. – 640 с.