

# ЛЕКЦИЯ 9

Продукционные модели

## Продукционная модель определение

**Продукционная модель**, или модель, основанная на правилах, позволяет представить знания в виде множества предложений типа: *Если* (условие), *то* (действие).

Под *условием* понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под *действием* – действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и терминальными или целевыми, завершающими работу системы).

Множество продукционных правил формируется на основе анализа опыта человека – специалиста в некоторой области деятельности.

# Продукционное правило

Продукционные правила представляются в виде **если <условие>, то <заключение>** (действие)

В продукционных системах в качестве условия (часто – множество условий) выступает образец для поиска, а заключением будет вывод о подтверждении некоей гипотезы, факта.

Пример такого продукционного правила:

**если** Сигнал\_датчика\_давления = «красный» **то** ситуация = ТРЕВОГА.

В экспертных системах часто используются правила, в которых посылкой является описание ситуации, а заключением – действия, которые необходимо предпринять в данной ситуации.

# Вывод на продукциях

Процедура логического вывода в системах, основанных на продукционных моделях, связана с работой специальной программы, управляющей перебором и проверкой правил. Такая программа называется *интерпретатором*, или *машиной вывода*.

Множество продукционных правил образует *базу знаний* содержащую все допустимые зависимости между фактами предметной области и хранящуюся в долговременной памяти.

Также необходимо использовать *рабочую память* – область для хранения истинных фактов, описывающих текущее состояние предметной области. Такая база фактов может пополняться новыми доказанными фактами в процессе работы с продукционными правилами.

# Стратегии вывода на продукциях

**Механизм логического вывода** обеспечивает формирование заключений, воспринимая вводимые факты как истинные, отыскивая правила, в состав которых входят введенные факты, и проверяя возможность выполнения этих правил.

Механизм логического вывода выполняет функции поиска в базе правил, последовательного выполнения операций над знаниями и получения заключений. Существует два способа вывода на продукциях – прямой и обратный вывод.

**Прямой вывод:** от имеющихся данных – к поиску цели.

**Обратный вывод:** от поставленной цели – к данным, необходимым для подтверждения цели.

# Пример прямого вывода

Продукции	Шаги вывода	Факты <b>H, K</b> даны
p1: $A \rightarrow E$	1) p3: <b>H</b> $\rightarrow A$	Доказан факт <b>A</b>
p2: $B \rightarrow D$	2) p1: $A \rightarrow E$	<b>E</b>
p3: $H \rightarrow A$	3) p5: $E \& K \rightarrow B$	<b>B</b>
p4: $E \& G \rightarrow C$	4) p2: $B \rightarrow D$	<b>D</b>
p5: $E \& K \rightarrow B$	5) p6: $D \& E \& K \rightarrow C$	<b>C</b>
p6: $D \& E \& K \rightarrow C$		
p7: $G \& K \& F \rightarrow A$		За 5 шагов доказан факт <b>C</b>

## Прямой вывод

Мы видим, что в процессе прямого вывода последовательно выводятся новые факты, начиная с уже известных.

Однако отсутствие связи между фактами может привести к обрыву процедуры и конечный результат не всегда может быть получен.

Кроме того, экспертная система может иметь сотни продукционных правил. Используя прямой вывод можно выполнить множество правил и получить факты, истинные сами по себе, но не имеющие никакого отношения к цели.

В случае, когда надо установить один, конкретный факт, обратный вывод может оказаться предпочтительнее.

## Обратный вывод

**Обратным выводам** (обратной цепочке рассуждений) соответствует движение от цели (факта, который требуется установить) к посылкам. В обратном механизме логического вывода работа начинается от поставленной цели. Если цель согласуется с заключением продукции, то посылка (условие) принимается за подцель и делается попытка подтверждения истинности этого факта. Процесс повторяется до тех пор, пока не будут просмотрены все правила, имеющие в качестве заключения требуемый факт.



# Пример обратного вывода

1. Проверяем наличие  $Z$  в списке известных фактов.  $Z$  отсутствует.

Ищем правило, которое может привести к  $Z$ . Это  $p1: F \& B \rightarrow Z$

Условия правила –  $B$  и  $F$ .  $B$  известно.

2. Пытаемся установить факт  $F$ .

Ищем правило, которое может привести к  $F$ . Это  $p2: C \& D \rightarrow F$

Условия правила  $C$  и  $D$ .  $C$  известно.

3. Пытаемся установить факт  $D$ .

Ищем правило, которое может привести к  $D$ . Это  $p3: A \rightarrow D$

Условие правила  $A$  – известно.

Выполняем правила  $p3$ ,  $p2$ ,  $p1$ .  $Z$  доказано

Продукции	Факты:	Цель
$p1: F \& B \rightarrow Z$	$A \ E \ H \ C \ B$	Доказать $Z$
$p2: C \& D \rightarrow F$		
$p3: A \rightarrow D$		

# Преимущества продукционной модели

Основные достоинства, благодаря которым продукционные правила получили широкое распространение, заключаются в следующем:

- продукционные правила легки для восприятия человеком;
- достоинством является однородность представления знаний и естественная модульность;
- отдельные продукционные правила могут быть независимо добавлены в базу знаний, исключены или изменены;
- продукционные правила допускают параллельную обработку;
- с помощью продукционных правил удобно представлять как декларативные, так и процедурные знания.

# Обработка не полностью достоверной информации.

Определим продукцию формально:

$$p_i(k_{pi}): A \rightarrow B$$

где  $A = a_1 \& a_2 \& \dots \& a_n$  – конъюнкция условий,

$B$  – заключение,  $k_{pi}$  – коэффициент достоверности продукции  $p_i$ .

Здесь  $B$  – конъюнктивная вершина.

Поставим цель – найти коэффициент достоверности заключения  $k_B$ , если известны коэффициенты достоверности посылок  $a_1, a_2, \dots, a_n$ . Обозначим  $k_1, k_2, \dots, k_n$  эти коэффициенты. Их возможные значения лежат в диапазоне от 1 (абсолютно достоверно) до 0 (абсолютно ложно).

Примем, что при  $k_{pi} = 1$  степень достоверности заключения не может превосходить степень достоверности самой слабой посылки:

$$k_B = \min(k_1, k_2, \dots, k_n).$$

$$\text{Если } k_{pi} < 1, k_B = \min(k_1, k_2, \dots, k_n) * k_{pi}.$$

## Дизъюнктивная вершина

Рассмотрим случай, когда к одному заключению (В) могут привести несколько наборов условий:

$p_i(k_{pi}): a_1 \& a_2 \& \dots \& a_n \rightarrow B$  (обозначим кратко  $A_i \rightarrow B$ ) и

$p_j(k_{pj}): b_1 \& b_2 \& \dots \& b_m \rightarrow B$  (обозначим кратко  $A_j \rightarrow B$ ).

Здесь В – дизъюнктивная вершина.

Поставим цель – найти  $k_B$ .

1. Пусть обе продукции достоверны полностью ( $k_{pi} = 1, k_{pj} = 1$ ).

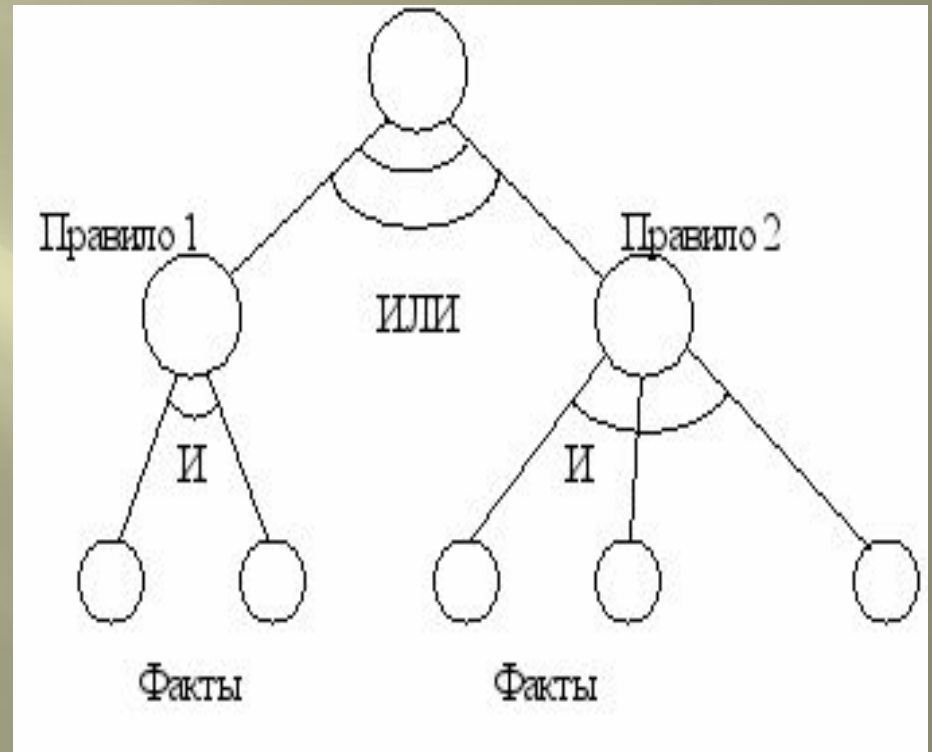
Для вершины ИЛИ считаем, что посылки подкрепляют друг друга при получении заключения. Тогда  $k_B = k_{Ai} + k_{Aj} (1 - k_{Ai})$

При условии  $k_{pi} \neq 1, k_{pj} \neq 1$  получим

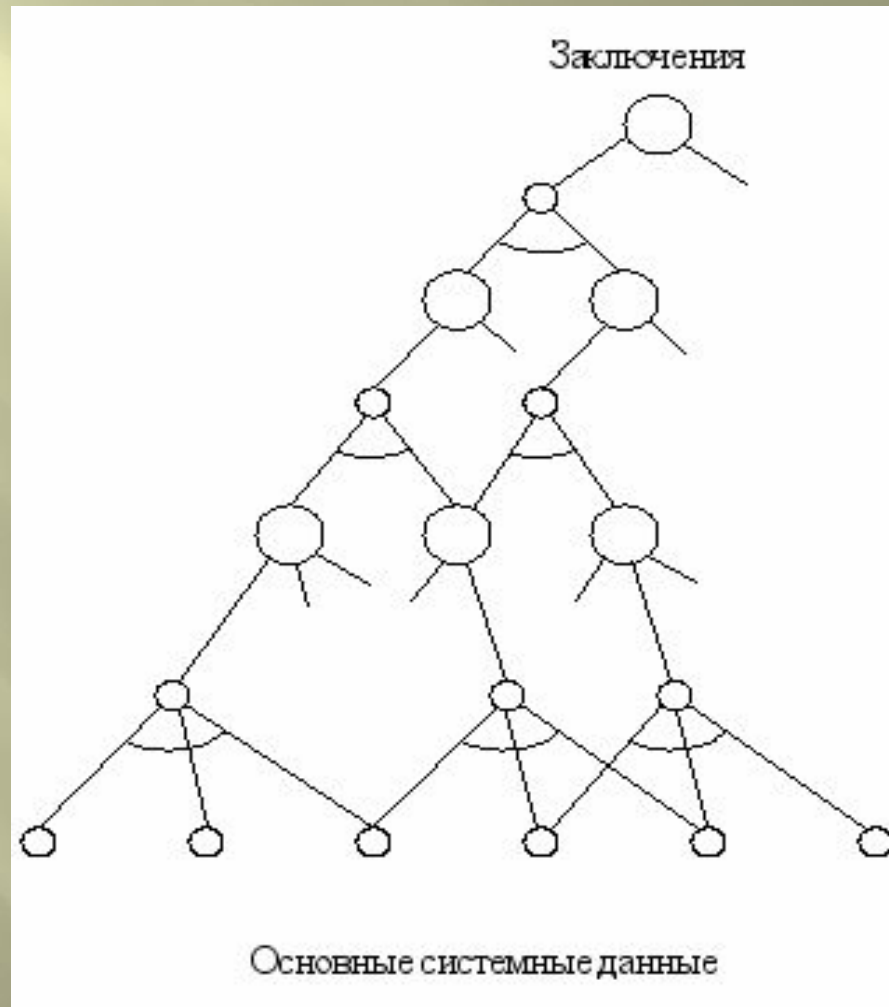
$$k_B = k_{pi} k_{Ai} + k_{pj} k_{Aj} (1 - k_{pi} k_{Ai})$$

# Простейший элемент структуры И –ИЛИ – дерева продукционной системы

Обычно в экспертной системе коэффициент достоверности заключения сравнивается с некоторым пороговым значением, определяемым эмпирически. Цель – исключить наименее достоверные выводы. Так, в экспертной системе MYCIN это значение равно 0.2).



# Структура И -ИЛИ – графа продукционной системы



# Проблемы формирования БЗ продукционной системы

При построении БЗ, содержащей множество продукций для сложных систем необходимо предусмотреть:

- проверку на полноту, то есть на способность системы правильно реагировать на любую возможную ситуацию в ПО;
- Проверку на непротиворечивость, т.е. отсутствие информации, применимой, например, к двум продукциям со взаимоисключающими гипотезами;
- устранение из БЗ избыточной информации и проведение допустимых обобщений;
- построение эффективного решателя, который за допустимое время выполняет поиск применимых правил и разрешает конфликты между продукциями.

# Пример построения обобщенного правила

Пусть мы построили правило:

**если** Сигнал\_датчика\_давления = «красный» **то** ситуация = ТРЕВОГА

На объекте расположены 25 различных датчиков, контролирующих давление, тогда нужно включить в БЗ 25 различных правил, имеющих одинаковую структуру

**p1:**если Сигнал\_датчика\_давления\_1 = «красный» **то** ситуация = ТРЕВОГА

**p2:**если Сигнал\_датчика\_давления\_2 = «красный» **то** ситуация = ТРЕВОГА...

**p25:** если Сигнал\_датчика\_давления\_25 = «красный» **то** ситуация = ТРЕВОГА

Можно построить *одно* обобщенное правило, например

**rgen:** **если** Сигнал\_датчика\_давления\_x  $\in [1, 25]$  = «красный» **то** ситуация = ТРЕВОГА



# Общая структура продукционной системы

## Основные компоненты.

1. База знаний. Содержит множество продукций. В состав продукций могут входить продукции, содержащие переменные, предикаты. Включает управление активными фактами и правилами.
2. Рабочая область (или база фактов). Хранит факты, текущие гипотезы, данные (символьные, числовые, логические). В начале работы рабочее пространство содержит формулировку поставленной задачи; затем добавляются факты, которые программа-интерпретатор установила к текущему моменту.
3. Программа – интерпретатор решает ряд важных задач: определяет правила, а также относящиеся к ним факты путём унификации; выбирает те правила, которые нужно выполнить (либо все пригодные, либо по оценкам важности); выполняет правило (продукцию) и пополняет базу фактов.

## Дополнительные компоненты

Для организации эффективного вывода необходимы:

- стратегия снятия конфликтов между правилами;
- стратегия обобщения правил;
- стратегия вычисления коэффициентов уверенности.

Необходимо хранить дополнительную информацию:

- правила, ожидающие очереди на выполнение;
- текущий план;
- список подзадач.

Продукции по сравнению с другими формами представления знаний имеют следующие **преимущества**:

- модульность;
- наглядность;
- единообразие структуры (основные компоненты продукционной системы могут применяться для построения интеллектуальных систем с различной проблемной ориентацией);
- естественность (вывод заключения в продукционной системе во многом аналогичен процессу рассуждений эксперта);
  - легкость внесения дополнений и простота механизма логического вывода.

### **Недостатки:**

- процесс вывода менее эффективен, чем в других системах, поскольку большая часть времени при выводе затрачивается на непроизводительную проверку применимости правил;
- этот процесс трудно поддается управлению;
- сложно представить родовидовую иерархию понятий.

# ЛЕКЦИЯ 10

Логические модели. Дедуктивные  
базы данных

# Понятие логической модели

Важным блоком интеллектуальной системы является система представления знаний. Такая система предназначена для хранения совокупности знаний о предметной области (ПО); при этом необходимо максимально использовать специфику ПО, её семантику.

Одним из наиболее мощных средств здесь являются логические модели. Такие модели используют логику предикатов 1 порядка и способны хранить как факты, так и общие законы (аксиомы).

*Факты* – это, чаще всего, утверждения о текущем состоянии ПО. Такие знания представляются в виде фундаментальных предикатов (вместо переменных подставлены константы).

Аксиомы представляют общие законы функционирования ПО, поэтому обычно содержат только переменные.

# Экстенсионал и интенционал

Конкретные факты, касающиеся ПО, могут быть представлены в обычной БД в виде таблиц, задающих отношения между объектами. БД, в которой хранятся факты, называется экстенсиональной.

Базу данных для хранения общих законов (аксиом) назовём интенциональной.

Рассмотрим два способа представить отношение «предшествует» ( $\leq$ ).

В интенциональном представлении определяем:

1. Если  $x \leq y$  и  $y \leq z$ , то  $x \leq z$ .
2. Не существует таких  $x_1, x_2, \dots, x_m$ , что
$$x_1 \leq x_2, x_2 \leq x_3, \dots, x_{m-1} \leq x_m, x_m \leq x_1.$$

В экстенсиональном представлении мы перечисляем все пары объектов, для которых это отношение истинно. Например, пусть  $x, y$  – бинарные векторы длины 3. Тогда экстенсионал отношения включает пары:  $\langle 0, 0, 0 \rangle \leq \langle 0, 0, 1 \rangle$

$$\langle 0, 0, 1 \rangle \leq \langle 1, 0, 1 \rangle, \dots, \langle 1, 1, 0 \rangle \leq \langle 1, 1, 1 \rangle$$

# Дедуктивные базы данных

Очевидно, интенциональное представление полезно для бесконечных областей. Однако в таком представлении сложно задавать интерпретации.

Экстенциональное представление обеспечит быстрый поиск в небольших областях, но перечисление всех возможных конкретных случаев может быть очень громоздким.

Следовательно, полезно рассмотреть систему представления знаний как совокупность интенциональных и экстенциональных отношений. Такое представление называется *дедуктивной базой данных*. С помощью интенциональной части дедуктивной БД можно делать дедуктивные выводы, используя, например, средства языка ПРОЛОГ.

# Дедуктивные базы данных

По определению, дедуктивная БД состоит из двух частей: экстенциональной, содержащей факты, и интенциональной, содержащей правила для логического вывода новых фактов на основе экстенциональной части и запроса пользователя.

Легко видеть, что при таком общем определении SQL-ориентированную реляционную СУБД можно отнести к дедуктивным системам. Действительно, что есть определенные в схеме реляционной БД представления, как не интенциональная часть БД. В конце концов не так уж важно, какой конкретный механизм используется для вывода новых фактов на основе существующих. В случае SQL основным элементом определения представления является оператор выборки языка SQL, что вполне естественно, поскольку результатом оператора выборки является порождаемая таблица.



# Дизъюнкт Хорна

Дедуктивные БД работают с множеством дизъюнктов специального типа, называемых **дизъюнктами Хорна**.

Дизъюнкт Хорна – это дизъюнкт, имеющий не более чем одну положительную литеру. Его вид:  $\neg A_1 \vee \neg A_2 \vee \dots \vee \neg A_n \vee B$ ,

что соответствует формуле

$$A_1 \& A_2 \& \dots \& A_n \rightarrow B \text{ в обычной записи.}$$

Дизъюнкты Хорна без условий – это факты (используются в экстенциональном представлении).

Дизъюнкты Хорна, имеющие как условие, так и заключение, это аксиомы (общие законы), соответствующие интенциональному представлению знаний.

Дизъюнкты Хорна без заключения – это отрицания фактов, или цели, которые нужно доказать.

# Проблема представления негативной информации

Мы рассматривали данные, которые понимались как истинные. Назовём данные, принимающие значение «ложь», негативной информацией.

Негативная информация может быть выражена различными способами.

1. Явно, с помощью негативной фундаментальной литеры (например,  $\neg Q(a, b, c)$  ).
2. Неявно. Путём вывода при использовании некоторых законов.

Например, закон однозначности позволяет вывести факт:

$\neg$ Год рождения(Иванов, 1981) из конкретного факта

Год рождения(Иванов, 1979) и аксиомы «Каждый человек имеет единственный год рождения».

Негативная информация создаёт трудности при её обработке.

# Открытый и замкнутый миры

Р. Рейтер ввёл предположение об *открытом и замкнутом* мирах.

В предположении о *замкнутом* мире мы считаем, что располагаем *полным* знанием о предметной области.

В предположении об *открытом* мире как позитивная, так и негативная информация должна быть представлена явно, таким образом допускаются «бреши» в знании о ПО.

Рассмотрим **пример**. Пусть имеется множество {Амосов, Дубинский, Григорьев, Перескоков} – лекторы; {Смирнов, Петров, Парамонов, Галкин} – студенты.

Представлено отношение «Обучает»:

---

**Обучает:** лектор студент

Амосов	Смирнов
--------	---------

Дубинский	Петров
-----------	--------

Григорьев	Галкин
-----------	--------

Амосов	Петров
--------	--------

---

## Пример

Пусть есть запрос: «Кто **не** обучает студента Петрова?»

В предположении о замкнутом мире мы получим ответ «Лекторы, не обучающие студента Петрова, это {Григорьев, Перескоков}».

В предположении об открытом мире ответ будет пуст {}, поскольку в логике предикатов при выводе мы имеем дело с полной информацией о фактах, которые считаются истинными. Факт же

«Не обучает(x, **Петров**)»

не представлен в БД явно, следовательно, нельзя говорить об истинности этого факта. Для получения правильного ответа в БД надо в

Не обучает: лектор	студент
Амосов	Галкин
Дубинский	Смирнов
Григорьев	Петров
Перескоков	Петров

таблицу

# Основные черты представления знаний в дедуктивной БД

В реальных Дедуктивных БД число негативных фактов огромно, и перечислять их бессмысленно. Поэтому явно представляются позитивные факты; представление негативных фактов полагается неявным. Тогда в предположении о замкнутом мире можно получить ответ на любой запрос.

Для данной модели представления знаний характерно:

- универсальный характер процедуры поиска решений;
- полное описание состояний системы средствами логики предикатов первого порядка;
- оптимизация процесса поиска решений путём использования эвристик.