

Продукционные системы

Нотация процедур в виде последовательности правил (предложена математиком Постом)

Правило продукции - правило вида

ЕСЛИ < условие> **ТО** <действие>

Продукционная система состоит из 3-х элементов:

- 1) классов и отношений (рабочей области);
- 2) правил;
- 3) управляющей структуры

Классы и отношения трактуются как «база данных», которая, по существу, содержит декларативные знания. В некоторых ИС используются комбинации сетевых и продукционных моделей. В них декларативные знания описываются в сетевой части, а процедурные - в продукционной

Набор правил:

ЕСЛИ < условие> **ТО** <действие>

Управляющая структура определяет какое правило будет проверено следующим. Часто управляющую структуру называют **интерпретатором правил**, а базу данных - **рабочей памятью**. Имеет очень важное значение

<Условие> - это проверка состояния базы данных, а <действие> некоторым образом видоизменяет содержимое базы данных.

Пример

Определение максимума и минимума из некоторого набора чисел с использованием продукционной системы, чтобы проиллюстрировать разницу в подходах. Рабочая область содержит числа от $A(1)$ до $A(10)$; N - количество рассматриваемых элементов, I - текущее значение индекса проверяемого числа

2 варианта:

1. Порядок следования правил существенен.
2. Порядок следования правил несущественен.

Порядок следования правил существенен

ПРОДУКЦИОННАЯ СИСТЕМА

Рабочая область:

```
A(1) = 4;  A(2) = 7;  ... ;  A(10) = 21;  
N = 10;  I=1;  MAX= A(1) ;  MIN = A(1)
```

Правила:

1. IF I>N THEN writeln (MAX,MIN); STOP;
2. IF A(I) > MAX THEN MAX:= A(I);
3. IF A(I) < MIN THEN MIN:= A(I);
4. If i<=N then I:=I+1

Управляющая структура:

Поочередно пробовать правила до конца. Затем следовать к началу списка правил и повторить все заново. Закончить процесс, когда встретится оператор STOP. Правило срабатывает, если выполняется условие.

Порядок следования правил несущественен

ПРОДУКЦИОННАЯ СИСТЕМА

Рабочая область:

$A(1) = 4; A(2) = 7; \dots; A(10) = 21;$
 $N = 10; I=1; MAX = A(1); MIN = A(1)$

Правила:

1. IF $(I \leq N) \ \& \ (A(I) > MAX)$ THEN $MAX := A(I);$
2. IF $(I \leq N) \ \& \ (A(I) < MIN)$ THEN $MIN := A(I);$
3. IF $I > N$ THEN writeln (MAX,MIN); STOP;
4. IF $(I \leq N) \ \& \ (A(I) \leq MAX) \ \& \ (A(I) \geq MIN)$ THEN
 $I := I + 1$

Управляющая структура:

Поочередно пробовать правила до конца. Затем следовать к началу списка правил и повторить все заново. Закончить процесс, когда встретится оператор STOP. Правило срабатывает, если выполняется условие.

- Если так тщательно определять правила, то можно вводить новые функции в программу просто путем добавления соответствующих правил. Но сконструировать систему независимых от порядка правил не так-то просто. Для этого условие оператора ЕСЛИ должно быть единственным для каждого правила. При большом количестве правил это достаточно трудно.
- Сложность реальных систем значительно превосходит сложность программы, рассмотренной в качестве примера. В экспертных системах, построенных на базе правил продукций, нередко применяются свыше 800 правил. Таким образом, управление их выполнением становится солидной программой

Другой вариант управляющей структуры,
которая не подходит для данного решения

Управляющая структура:

Поочередно пробовать все правила, пока одно из них не сработает. Затем повторить процесс снова.
Закончить, когда встретится оператор STOP.

**Очень важно ПРАВИЛЬНО определять
управляющую структуру!**

Общий вид продукционного правила

(i); Q; P; $A \Rightarrow B$; N

$A \Rightarrow B$ - ядро продукции (основной элемент продукции). Это правило ЕСЛИ **A**, ТО **B**. Более сложные конструкции ядра допускают правила вида ЕСЛИ **A**, ТО **B1**, ИНАЧЕ **B2**. При этом секвенция (\Rightarrow) может трактоваться в обычном смысле (логическом) как знак логического следования **B** из истинного **A** (если **A** - ложь, то о **B** ничего сказать нельзя). Возможны и другие интерпретации ядра, например, **A** описывает некоторое условие, необходимое для совершения **B**.

i - имя продукции, с помощью которого данная продукция выделяется из всего множества продукции. Это может быть просто порядковый номер или лексема, отражающая суть продукции, например, «ПОКУПКА АВТОМОБИЛЯ», «ПЛАНЫ НА ВЕЧЕР» и т.д.

Q - характеризует сферу применения продукции. Разделение знаний на сферы применения позволяет эффективно организовать поиск нужных знаний.

P - условие применимости ядра продукции. Обычно это логическое выражение или предикат. Если **P** - истина, то ядро продукции активизируется, иначе оно не может быть использовано.

Пример: **P** – «НАЛИЧИЕ ДЕНЕГ»; если хочешь купить вещь **X**, то заплати за нее в кассу и отдай чек продавцу.

N - постусловия продукции. Могут активизироваться лишь при реализации ядра. Постусловия описывают действия и процедуры, которые необходимо выполнить после реализации **B**. Например, после покупки вещи в магазине уменьшить количество вещей такого типа на 1.

Как это работает

- **Рабочая память** (*working memory*) содержит описание *текущего состояния мира* в процессе рассуждений. Это описание является образцом, который сопоставляется с условной частью продукции с целью выбора соответствующих действий при решении задачи. Если условие некоторого правила соответствует содержимому рабочей памяти, то может выполняться действие, связанное с этим условием. Действия продукционных правил предназначены для изменения содержимого рабочей памяти.
- **Управляющая структура** продукционной системы: рабочая память инициализируется начальным состоянием задачи. Текущее состояние представляется набором образцов в рабочей памяти. Эти образцы сопоставляются с условиями продукционных правил, что порождает подмножество правил вывода, называемое конфликтным набором (*conflict set*). Продукции, содержащиеся в конфликтном наборе, называют допустимыми. Возникает проблема разрешения конфликта (*conflict resolution*).
- **Активизация правила** означает выполнение его действия. При этом изменяется содержимое рабочей памяти. После того как правило сработало, цикл управления повторяется для модифицированной рабочей памяти. Процесс заканчивается, если содержимое рабочей памяти не соответствует никаким условиям.

Прямой вывод

Правило 1 *ЕСЛИ Вася имеет шерсть*

и

Вася ест мясо,

ТО Вася - кот.

Правило 2 *ЕСЛИ Вася - хищник,*

ТО Вася ест мясо.

В рабочую область заносится «*Вася имеет шерсть*» и «*Вася – хищник*». Рассматривается возможность применения представленных выше правил.

Сначала механизм вывода сопоставляет образцы из условной части правил с образцами, хранимыми в рабочей памяти. Если образец присутствует в рабочей области, то условная часть истинна. Для правила 1 она - ложь, а для правила 2- истина. Следовательно, образец «*Вася ест мясо*» заносится в рабочую область. При попытке вторично применить правила 2 выбывает, потому что уже использовалось, Правило1 даст истинный результат, поскольку теперь в БД присутствует нужный образец. Вывод - *Вася - кот*. Останов из-за отсутствия правил.

Обратный вывод

Способ, при котором на основании фактов, требующих подтверждения, чтобы выступить в роли заключения, исследуется возможность применения правила, пригодного для подтверждения, называется **ОБРАТНЫМ** выводом.

Пусть цель - это то, что *«Вася – кот»*.

Можно ли использовать Правило1, чтобы подтвердить этот факт? *«Вася имеет шерсть»* существует в рабочей области, следовательно, достаточно подтвердить тот факт, что *«Вася ест мясо»*. Но если *«Вася ест мясо»* – новая цель, то ее тоже надо подтвердить правилом. Исследуется возможность применения Правила 2. Условная часть правила на данный момент истинна, следовательно, подтверждение получено.

В случае обратного вывода условием останова является либо достижение первоначальной цели, либо то, что правил больше нет.

Конфликтный набор

Пусть имеется еще одно правило:

Правило 3 **ЕСЛИ** *Вася имеет шерсть,*
 ТО *Вася - млекопитающее*

Условие останова – *«Вася – кот»*

Выбор применяемого правила оказывает прямое влияние на эффективность работы системы: и если применить ПЗ, то нужен дополнительный вывод.

В реальной продукционной системе это грандиозная проблема. Если на каждом этапе логического вывода существует множество применимых правил, то оно носит название **КОНФЛИКТНОГО НАБОРА**, а выбор одного из них называется **разрешением конфликта**. Эта же проблема характерна и для обратного вывода.

Правило 4 **ЕСЛИ** *Вася принадлежит семейству кошачьих,*
 ТО *Вася ест мясо.*
Цель - *«Вася – кот»*

Чтобы подтвердить, что *«Вася ест мясо»*, надо попробовать правила 2 или 4.

Если сразу обратиться к Правилу 2, то это удачный выбор, так как сразу можно исследовать Правило 1 и подтвердить цель.

Если сначала обратиться к Правилу 4, то в рабочей области отсутствует образец *«Вася принадлежит семейству кошачьих»* и не существует правила его подтверждающего. Следовательно, выбор неудачен. Только со второго захода через Правило 2 можно подтвердить цель *«Вася ест мясо»*.

Классификация ядер продукций

ядра продукций

```
graph TD; A[ядра продукций] --> B[детерминированные]; A --> C[недетерминированные]
```

детерминированные

При актуализации ядра и выполнимости **A** правая часть реализуется обязательно

недетерминированные

При актуализации ядра и выполнимости **A** правая часть может и не выполняться. Интерпретация ядра может выглядеть, например, так: ЕСЛИ **A**, ТО, возможно, **B**

Недетерминированные ядра

Возможность может определяться некоторыми ***оценками реализации ядра***.

Например,

ЕСЛИ **A**, ТО с вероятностью **p** РЕАЛИЗОВАТЬ **B**
(прогнозирующие продукции)

Лингвистическая оценка реализации:

ЕСЛИ **A**, ТО с *большой долей уверенности* **B**.

Существуют и другие способы оценки реализации.

детерминированные продукции

однозначные

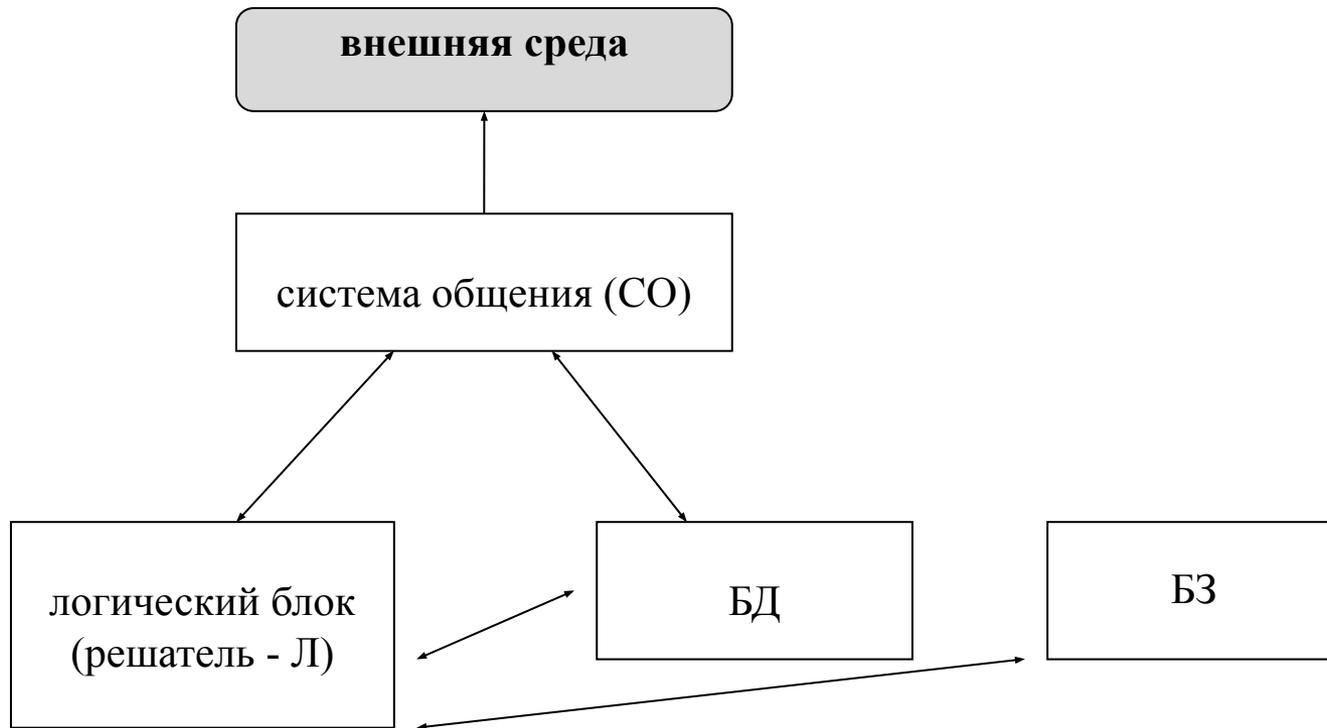
альтернативные

Указываются альтернативные возможности выбора, которые оцениваются специальными весаами выбора.

В качестве весов могут быть использованы вероятностные, лингвистические, экспертные оценки.

Например, ЕСЛИ А, ТО *чаще всего* надо делать V_1 , *реже* V_2 .

Другая классификация ядер



$$Ax \Rightarrow By$$

будет означать, что информация об **A** берется из блока **x**, а результат срабатывания **B** помещается в блок **y**

$A \backslash B$	CO	БД	БЗ	Л
CO		+		+
БД	+	+	+	+
БЗ	+		+	+
Л	+		+	+

ПРИМЕР1

1) $A_{БЗ} \Rightarrow B_{БЗ}$.

В этом случае **A** и **B** - некоторые фрагменты информации из БЗ.

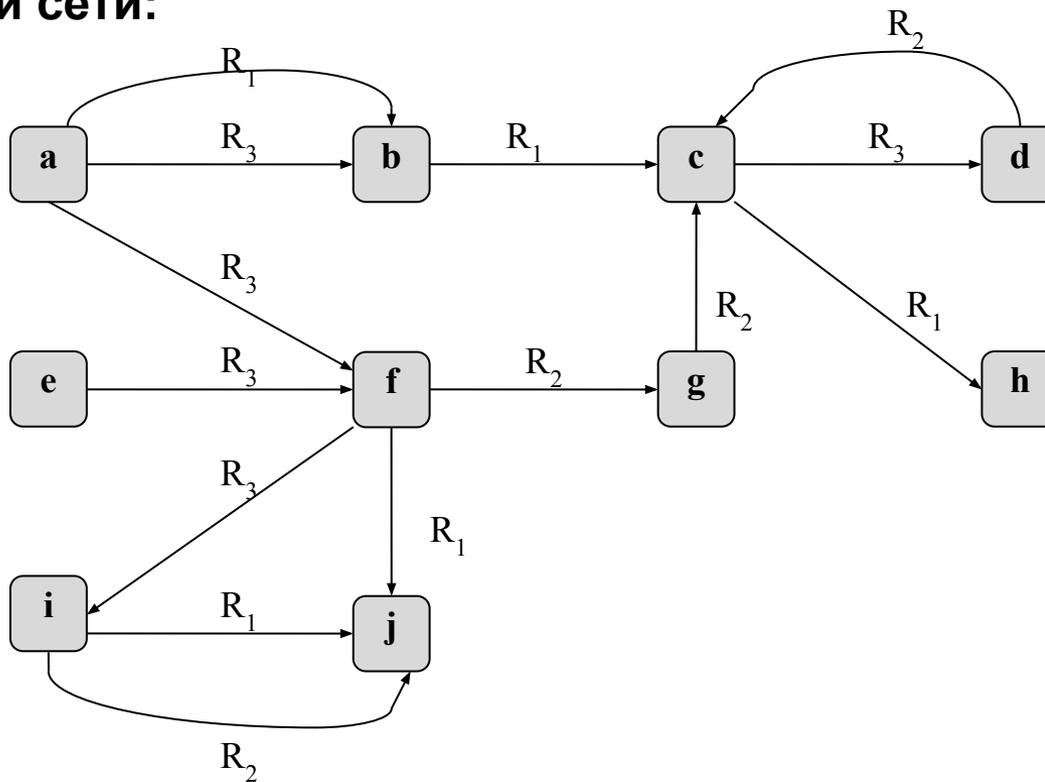
При сетевом представлении это могут быть фрагменты семантической сети, в случае логического представления - формулами того или иного исчисления.

При этом смысл $A_{БЗ} \Rightarrow B_{БЗ}$ заключается в замене одного фрагмента базы знаний другим.

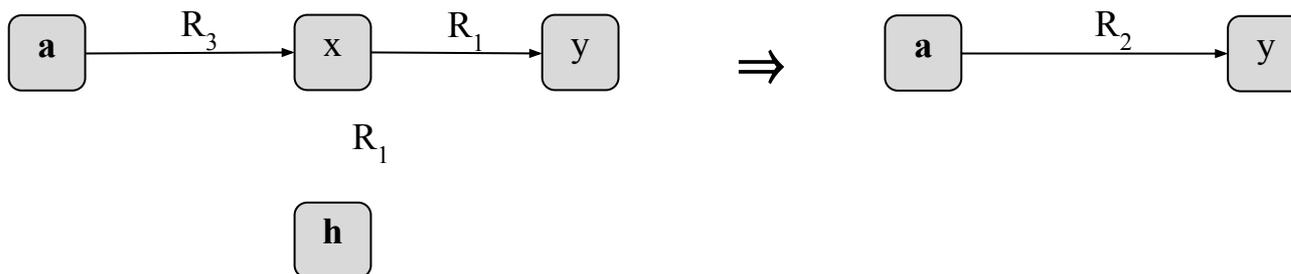
Для активизации этой продукции необходимо, чтобы в базе знаний существовал фрагмент, совпадающий с **A**. **A** играет роль образца, мы имеем дело с процедурой поиска по образцу (*pattern matching - сопоставление с образцом*).

Иллюстрация поиска по образцу в базе знаний

Фрагмент семантической сети:



Продукция:



$$A_{Бд} \Rightarrow B_{Бз}$$

Может соответствовать процедуре нахождения закономерностей по эмпирическим данным

Логический блок на основании просмотра и анализа данных выдвигает гипотезу о наличии закономерностей и, убедившись в их приемлемости и логической обоснованности, записывает их в базу знаний

Управление системой продукций

Проблема: какая из продукций должна быть активизирована?

Этим вопросом и должно заниматься система управления.

Если ИС реализована на ЭВМ с параллельной архитектурой, то из множества готовых продукций (фронта готовых продукций) может выбираться не одна продукция, а столько, сколько параллельных ветвей может выполнять ЭВМ. Но независимо от количества актуализированных продукций, проблема выбора остается.

2 пути решения задачи управления:

- 1) **централизованный:** решение об актуализации принимает специальная система управления;
- 2) **децентрализованный:** решение об актуализации продукции определяется складывающейся в этот момент ситуацией

Если порядок выполнения продукций важен, то в продукциях должна содержаться информация о требованиях к этому порядку. Если в постусловии продукции указывается имя той продукции, которая должна выполняться следующей, производственная система превращается в обычную программу.

Стратегии управления производственной системой

1. Принцип «стопки книг»

Особенно хорош, когда частота использования подсчитывалась с учетом некоторой ситуации, в которой исполнялась продукция, и это исполнение имело положительную оценку.

Имеет смысл применять, если продукции относительно независимы друг от друга, например, когда каждая из них есть правило вида

$$\text{ситуация}(A) \Rightarrow \text{действие}(B).$$

Именно такой случай имеет место в планирующих системах для роботов.

Стратегии управления производственной системой

2. Принцип наиболее длинного условия

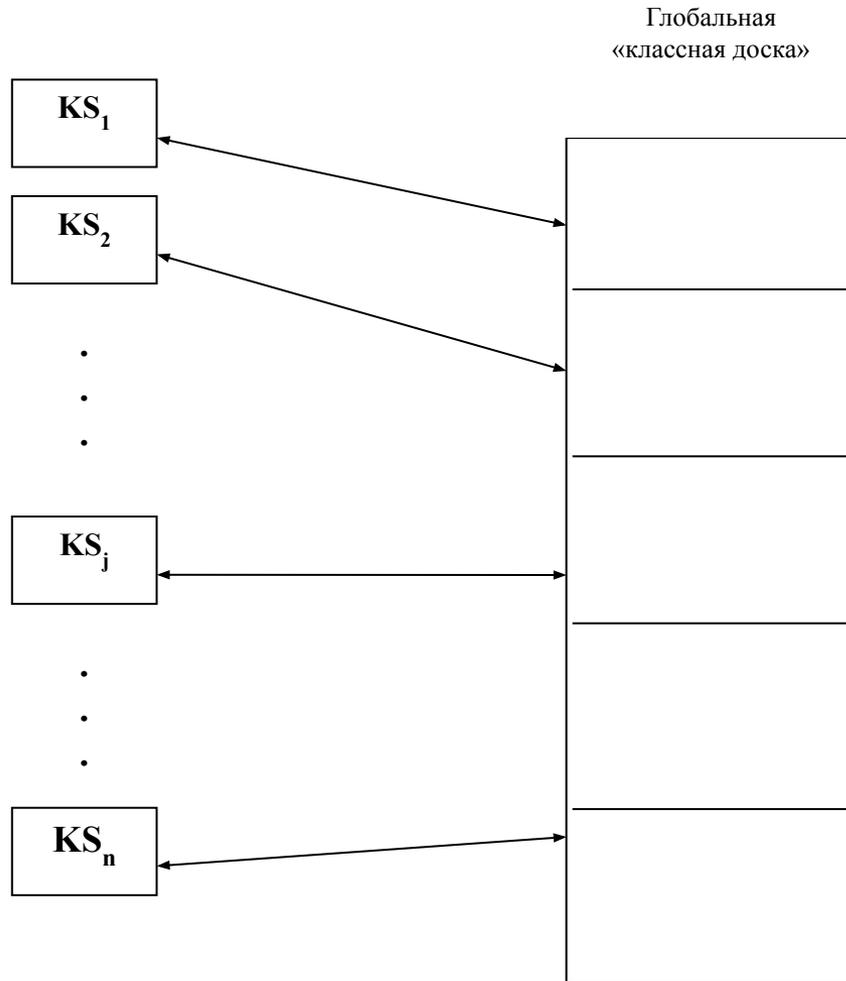
Целесообразно применять в тех случаях, когда знания и сами продукты хорошо структурированы привязкой к типовым ситуациям, на которых задано отношение «частное-общее».

3. Принцип метапродукций

4. Принцип «классной доски»

«Классная доска» - центральная глобальная база данных, предназначенная для связи независимых асинхронных источников знаний». Архитектура «классной доски» (стратегия решения сложных системных задач, взаимодействующих через общее информационное поле) – модель управления, которая применяется для задач, требующих координации различных процессов или источников знания

KS_j -
knowledge source



HEARSAY-II

- **KS₁** - Форма волны (временная диаграмма) акустического сигнала
- **KS₂** - Фонемы или возможные звуковые сегменты акустического сигнала
- **KS₃** - Слоги, которые могут быть составлены из фонем
- **KS₄** - Возможные слова, результат анализа первого **KS_i**
- **KS₅** - Возможные слова, результат анализа второго **KS_i** - (обычно рассматриваются слова из различных частей данных)
- **KS₆** - Генерация возможных последовательностей слов
- **KS₇** - Связывание последовательности слов в фразы

Стратегии управления производственной системой

5. Принцип приоритетного выбора

Этот принцип связан с введением статических и динамических приоритетов на производстве.

Статические приоритеты могут формироваться на основании сведений о важности производственных правил в данной проблемной области. Эти сведения, как правило, - информация, полученная от эксперта.

Динамические приоритеты вырабатываются в процессе функционирования системы производств могут отражать, например, такой параметр, как время нахождения продукции среди готовых производств.

Стратегии управления производственной системой

6. Управление по именам

Пусть СП - простейшая, состоит только из ядер.

(1) $A \Rightarrow B$;

(2) $B \ \& \ D \Rightarrow A$

(3) $A \ \vee \ B \Rightarrow D$

(4) $D \Rightarrow C$

Система productions недетерминирована. Если выполняется A, то фронт готовых productions включает productions с именами (1) и (3), а если B и D, то (2), (3), (4).

Для устранения недетерминированности может быть введена грамматика для имен:

(1) \Rightarrow (2);

(2) \Rightarrow (2);

(3) \Rightarrow (4)

Тогда если в некоторый момент будет выполнена production с именем (2), то новые productions выполняться не будут.

Если же выполнялась (1), то после нее смогут выполняться (2) и (4). Но левые части таковы, что неоднозначность возникает лишь тогда, когда B и D одновременно выполнимы. С помощью грамматики можно задать однозначный алгоритмический процесс.

Факторы популярности продукционных моделей

1. Подавляющая часть человеческих знаний может быть записана в виде СП.
2. Разделение знания и управления. Преимущество такого разделения заключается в простоте изменения базы знаний, при котором не надо менять код программы управления. И, наоборот, можно менять код управляющей программы, не трогая набор правил вывода.
3. Управление на основе образцов (pattern-directed control). Правила в продукционной системе могут запускаться в любой последовательности. Описание задачи, представляющее текущее состояние мира, определяет конфликтное множество, и, следовательно, конкретный путь поиска и решения. Существуют возможности эвристического управления поиском.
4. СП являются модульными. За небольшим исключением, удаление или добавление продукций не приведет к изменениям в остальных продукциях. Между продукционными правилами отсутствует синтаксическое взаимодействие. Правила могут только влиять на активизацию других правил, изменяя образец в рабочей памяти. Правила не могут «вызывать» другие правила непосредственно, как подпрограммы. При этом они не могут устанавливать значения переменных в других продукционных правилах. Область действия переменных этих правил ограничена отдельным правилом. Эта синтаксическая независимость позволяет инкрементно разрабатывать экспертные системы путем последовательного добавления, удаления или изменения знаний (правил) системы.

Факторы популярности продукционных моделей

5. При необходимости СП могут реализовать любые алгоритмы и, следовательно, способны отражать любое процедурное знание, доступное ЭВМ.
6. Наличие в продукциях указателей на сферу применения продукции позволяет эффективно организовать память, сократив время поиска в ней необходимой информации. Классификация сфер может быть многоуровневой, что еще более повышает эффективность поиска знаний. так как позволяет наследовать информацию в БЗ.
7. При объединении СП и сетевых представлений получают средства, обладающие большой вычислительной мощностью.
8. Естественный параллелизм в СП, асинхронность их реализации делают СП удобной моделью вычислений для ЭВМ новой архитектуры, в которой идея параллелизма и асинхронности является центральной.
9. Независимость от выбора языка. Модель управления продукционной системой не зависит от представления правил в рабочей памяти, если это представление поддерживает сопоставление с образцом.