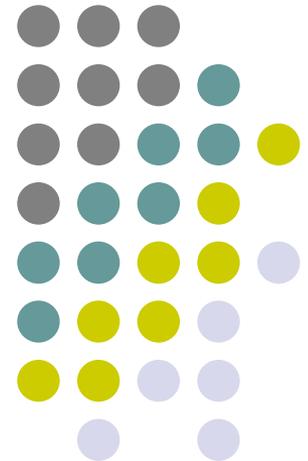
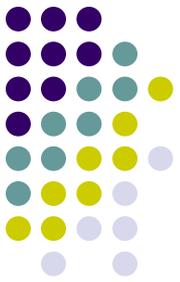


**ПРЕДСТАВЛЕНИЕ
ЗНАНИЙ:**

**ПРОДУКЦИОННАЯ
МОДЕЛЬ**

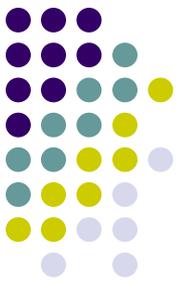


СОДЕРЖАНИЕ



1. Продукционная модель ПЗ
 - правила продукции и продукционные системы (ПС)
2. Основные компоненты и особенности ПС
3. Вывод в ПС
 - прямой и обратный
 - управление выводом
 - цикл работы ПС
4. Продукционные языки
5. Достоинства и недостатки ПС
6. Заключение

ПОНЯТИЕ ПРОДУКЦИИ



❖ **Продукция** – это правило (выражение) вида:
ситуация (условие) → действие

или *(пред)посылка → заключение*

$$p \rightarrow q$$

p – левая часть, *q* – правая часть

❖ Правила продукции удобны для представления поверхностных процедурных знаний, в частности, для эмпирических/эвристических знаний ПО:

медицина

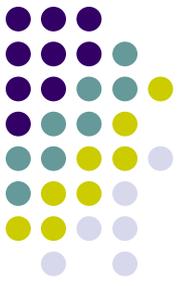
Если у вас $t^{\circ} > 38^{\circ}C$, то надо принять парацетамол

бытовая химия:

Если pH жидкости < 6 , то разлившееся вещество –
кислота

Если жидкость – кислота и пахнет уксусом, то это
уксусная кислота

СЕМАНТИКА ПРОДУКЦИИ



Если текущая ситуация – данные о решаемой задаче (*факты*) согласуются с левой, условной частью правила продукции, то выполняется правая часть, задающая:

- ✓ Получение определённого заключения о задаче: добавление к текущему состоянию решаемой задачи нового факта или гипотезы
- ✓ Внешнее воздействие: например, ввод/вывод информации
- ❖ Правила продукции часто используются в экспертных системах (ЭС)
- ❖ ЭС– специальный класс систем ИИ, основанных на знаниях экспертов

ПРОДУКЦИОННАЯ МОДЕЛЬ ПЗ



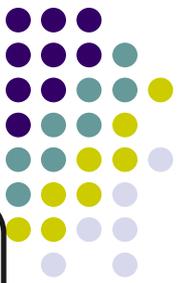
- ❖ Единица знаний – правило продукции
- ❖ Последовательное применение правила продукций к текущему состоянию задачи – порождение так называемой *цепочки вывода*

$$S_1 \rightarrow_{r_1} S_2 \rightarrow_{r_2} \dots \rightarrow_{r_i} S_i$$

где S_i – набор фактов, описывающих текущее состояние решаемой задачи

- ❖ Процесс решения задачи (*вывод*) – поиск решающей цепочки правил продукций для решения определённой задачи исходя из заданных фактов
- ❖ *Продукционная система* (ПС) – специальный вычислительный формализм

ОСНОВНЫЕ КОМПОНЕНТЫ ПС



Рабочая память:
База фактов

- предназначена для описания решаемой задачи
- в начальный момент – исходная информация

База знаний:
База правил

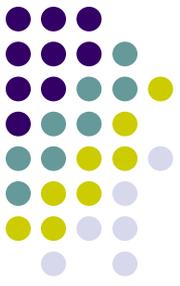
- набор правил продукций, выражающих в совокупности знания в некоторой ПО

Управляющая
стратегия
(Система
управления)

- стратегия выбора и применения продукций при поиске решения задачи (т.е. решающей цепочки продукций)

УС последовательно определяет правило, используемое в текущий момент, и прекращает вычисления в нужный момент.

Исполнение правил реализует *интерпретатор продукций*



ОСОБЕННОСТИ ПС

- При любом способе вычислений, в любом алгоритме можно выделить стандартные вычислительные компоненты:

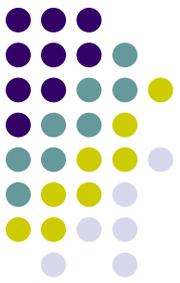
данные, операции, управление

ПС основаны на полном и чётком разделении (независимости) этих компонент.

РП + БЗ + управляющие знания

- Модульность БЗ (набора продукций) – возможность определять правила независимо
- «Недетерминированность»: выбор очередной продукции реализует УС, а не программист
- ПС – продукционная парадигма программирования:
rule-based programming

ВИДЫ УПРАВЛЯЮЩИХ СТРАТЕГИЙ



Классификация по различным признакам:

- возможность возврата в точках выбора применяемой продукции
- направление построения решающей цепочки продукции: **прямой** или **обратный вывод**
- ❖ *Прямой вывод в ПС*: поиск решения – в прямом направлении от исходного (начального) состояния к целевому состоянию (управляемый данными)
- ❖ *Обратный вывод в ПС*: формирование пробных гипотез и проверка их на соответствие текущим фактам (управляемый целями, целевое состояние – начальная гипотеза)

Пролог: режим возвратов (бэктрекинг) + обратный вывод

ПРИМЕР ПРЯМОГО ВЫВОДА



Рассмотрим различие двух видов УС на примере

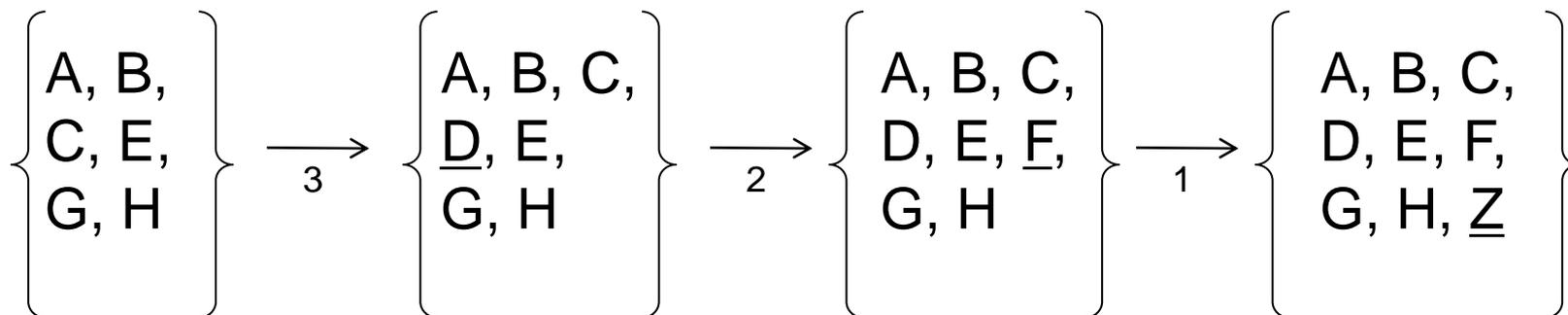
Пусть БЗ

$$\left\{ \begin{array}{ll} F \& B \rightarrow Z & (1) \\ C \& D \rightarrow F & (2) \\ A \rightarrow D & (3) \end{array} \right.$$

Начальное состояние РП – факты {A, B, C, E, G, H}

Целевое состояние РП – наличие факта Z (условие останова)

Тогда последовательные состояния РП:



Под стрелками стоит номер применённого правила продукций

Подчеркнут новый, добавляемый факт

ПРИМЕР ОБРАТНОГО ВЫВОДА



Та же самая БЗ:

$$\left\{ \begin{array}{ll} F \& B \rightarrow Z & (1) \\ C \& D \rightarrow F & (2) \\ A \rightarrow D & (3) \end{array} \right.$$

Начальное состояние РП – факты {A, B, C, E, G, H}

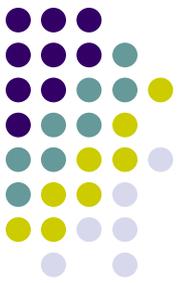
Целевое состояние РП – наличие факта Z (условие останова)

Тогда:

$$\left\{ \begin{array}{l} A, B, C \\ E, G, H \end{array} \right\} Z? \xrightarrow{1} F? \xrightarrow{2} D? \xrightarrow{3} t!$$

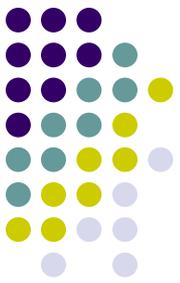
- Последовательный поиск правила, которое приводит к установлению нужного факта
- По окончании: инверсия найденной цепочки правил
- Изменяемая часть: Z, F, D

ПРЯМОЙ И ОБРАТНЫЙ ВЫВОД: СРАВНЕНИЕ



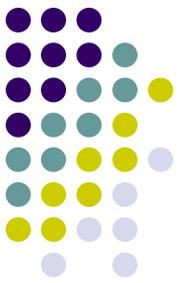
- Наиболее эффективное направление поиска решения определяется структурой задачи
- Если цели известны и их немного, то лучше обратный вывод
- Если цели неизвестны или их много, а исходных данных мало, то лучше прямой вывод
- В диагностических ЭС чаще применяется прямой вывод, а в ЭС планирования более эффективным оказывается обратный вывод
- Иногда эффективен (если возможен) двусторонний поиск: движение в прямом и обратном направлении одновременно
- При реализации ПС с прямым выводом обратный вывод может быть использован для её настройки/отладки

ПРЯМОЙ И ОБРАТНЫЙ ВЫВОД: ЗАМЕЧАНИЕ



- Н. Нильсон: с формальной точки зрения между ПС с прямым и обратным выводом различия нет, но есть содержательное различие:
- ❖ В ПС с прямым выводом РП содержит описания состояний (заключения, сделанные по известным фактам), а продукции преобразуют состояния в состояния (точнее, в их описания)
 - ❖ В ПС с обратным выводом РП содержит описание целей/подцелей (гипотез, которые надо подтвердить), а продукции преобразуют цели в подцели (точнее, в их описания)

МОНОТОННЫЙ И НЕМОНОТОННЫЙ ВЫВОД



Сравним вывод в ПС с выводом в логических системах:

- В логических системах – монотонный вывод, всё доказанное остаётся истинным, множество доказанных фактов монотонно возрастает
- Чистый Пролог – монотонный вывод
- В ПС в общем случае реализуем немонотонный и монотонный вывод – из-за возможности удаления фактов из РП
- В рассмотренных на предыдущем слайде примерах оба вывода (прямой и обратный) – монотонные (в правилах есть только добавление новых фактов, удаление фактов отсутствует)

ЦИКЛ РАБОТЫ ПС С ПРЯМЫМ ВЫВОДОМ

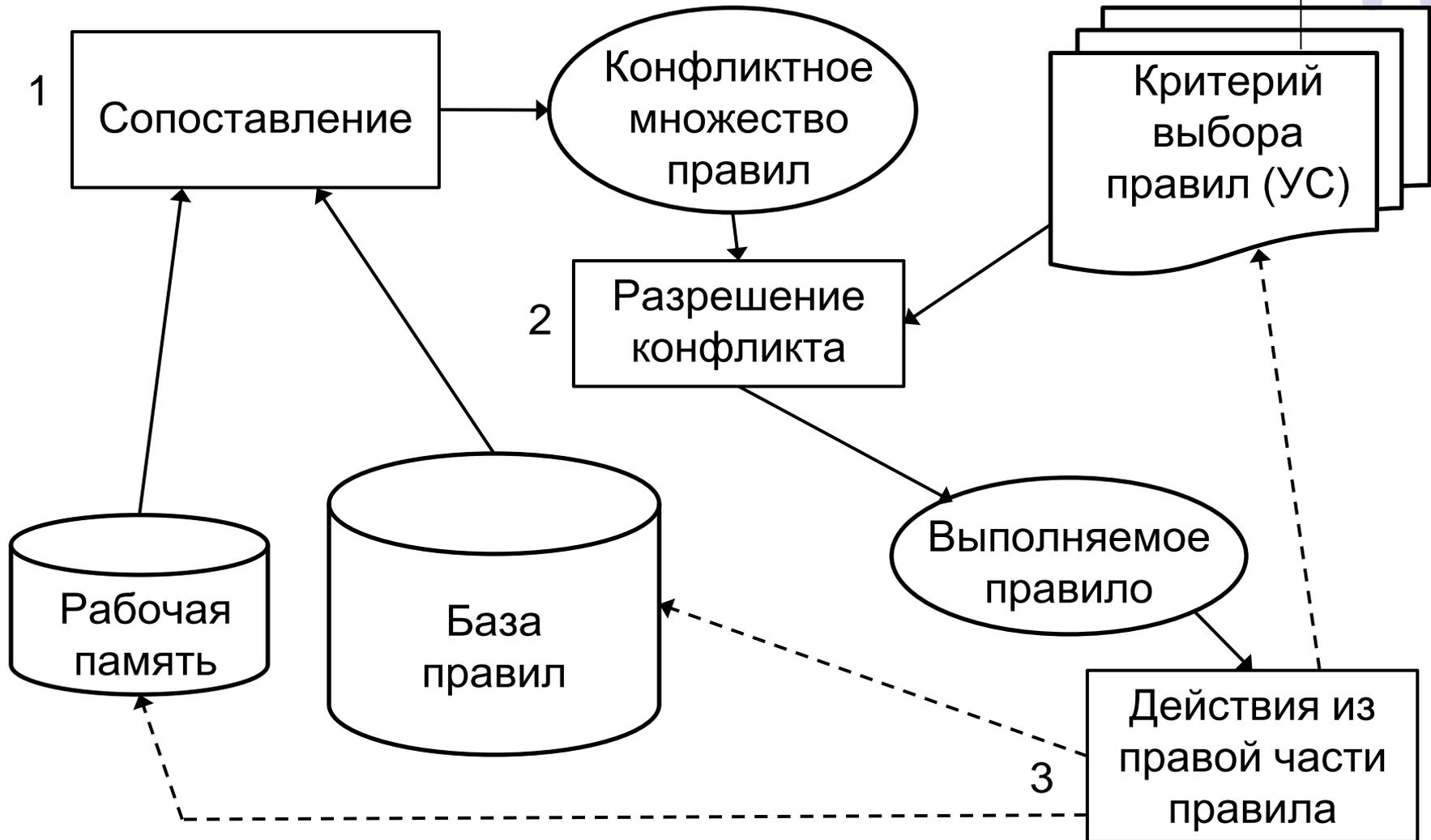


1. Сопоставление левых частей правил с фактами РП:
поиск применимых правил продукций
2. Разрешение конфликта: выбор одного правила согласно управляющей стратегии
3. Выполнение выбранного правила, при этом возможны:
 - изменение РП (множества фактов)
 - изменение БЗ
 - изменение управляющей стратегии

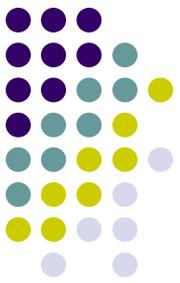
Повторение этого цикла, пока есть применимые правила

- ❖ Левая часть правил продукций обычно играет роль *образца данных*, поэтому поиск часто называется *поиском по образцу*
- ❖ *Конфликтное множество* – это множество найденных правил, применимых в данный момент

СХЕМА РАБОТЫ ПС



УПРАВЛЕНИЕ ВЫВОДОМ В ПС



Успешность вывода в ПС зависит от
процедуры разрешения конфликта

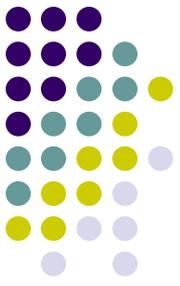
Возможные принципы/стратегии:

- Упорядочение элементов данных в РП: использование «возраста» – времени записи данных, тогда выбирается продукция, левая часть которой удовлетворяется на самых «свежих» данных, т.е. записанных последними
- Упорядочение продукций в конфликтном наборе
- Применение метаправил, организующих управление выбором

Например: мета-правило может быть таким (ЭС MYCIN)

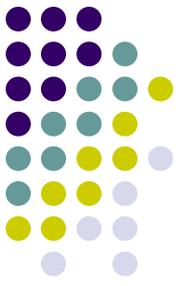
- ▶ Если «инфекция» есть *pelvic_abcess* (условие на РП)
- ▶ и в конфликтном наборе есть продукции, в левой части которых упоминается *grampos_rods* (условие на левую часть продукции)
- ▶ То продукции, в левой части которых используется *enterobacteriaceal* имеют больший приоритет

РАЗРЕШЕНИЕ КОНФЛИКТА: УПОРЯДОЧЕНИЕ ПРОДУКЦИЙ



- По степени используемости (частоте): выбирается продукция с максимальной частотой использования (принцип «стопки книг» – наиболее часто используемая продукция считается наиболее полезной)
- По времени последнего использования: выбирается продукция, которая была использована последней
- По длине («жёсткости») условной части продукции: выбирается продукция с наиболее «длинным» условием (т.е. выражающая специальные, частные случаи)
- Посредством введения приоритета правил:
 - ❖ Статический приоритет задается априори, на основании сведений о важности продукции
 - ❖ Динамический приоритет формируется в процессе функционирования ПС (может учитывать время нахождения продукции в конфликтном множестве)

ПРОДУКЦИОННЫЕ ЯЗЫКИ



- ПС: определённая парадигма программирования, способ организации вычислений –
rule-based programming,
ситуационное, продукционное программирование
- OPS-5 и CLIPS – продукционные языки, в обоих:
 - безвозвратный прямой вывод
 - немонотонный вывод
 - 2 вида фактов:
 - факты с упорядоченными атрибутами
 - факты с неупорядоченными именованными атрибутами
- ❖ CLIPS – больше встроенных средств и возможностей

OPS-5: ПРИМЕРЫ ПРАВИЛ



ПО – музыкальные инструменты:

```
(р струнный? (инструмент ^струны да) ->
                    (modify 1 ^тип струнный))
(р контрабас? (инструмент ^тип смычковый
    ^размер большой) -> (modify 1 ^имя контрабас))
```

ПО – химия:

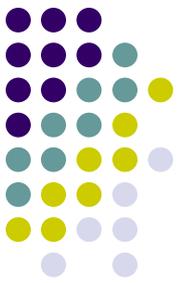
```
(р реакция (вещество ^класс кислота)
    (вещество ^класс щелочь) -> (remove 1) (remove 2)
                    (make (вещество ^класс соль)))
(р опр_хлорида (вещество ^имя AgNO3)
    (вещество ^цвет бесцветный ^класс соль
    ^вид хлорид) -> (remove 1) (remove 2)
                    (make (вещество ^класс соль ^имя AgCl
    ^цвет белый ^осадок да)
                    (make (вещество ^класс соль ^вид нитрат
    ^цвет бесцветный))
```

ДОСТОИНСТВА ПОДУКЦИОННОЙ МОДЕЛИ ПЗ



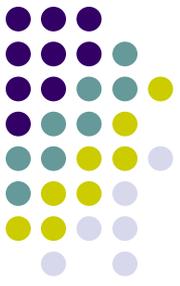
- Простота создания и понимания правил продукций, выражающих самостоятельные фрагменты знаний
- Удобство и простота пополнения и модификации БЗ, т.е. легкость построения продукционной БЗ
- Возможность изменения УС без изменения БЗ и БД
- Естественная возможность асинхронного и параллельного выполнения продукций (применимых в текущий момент)
- Возможность программирования немонотонных выводов

НЕДОСТАТКИ ПРОДУКЦИОННОЙ МОДЕЛИ ПЗ



- Эвристический метод: слабый теоретический базис, не хватает строгой теории
- Неэффективность вывода в ПС: с увеличением числа продукций существенно замедляется скорость вывода
- Сложность управления выводом в ПС: метаправила повышают гибкость, но ещё больше снижают эффективность
- Сложность проверки непротиворечивости БЗ с ростом числа продукций (из-за их независимости)
При уверенности в правильности любой продукции всё же априори нельзя быть уверенным в *полноте и непротиворечивости модели* ПО в виде ПС

ЗАКЛЮЧЕНИЕ



- ПС – модель для представления процедурных, поверхностных знаний
- Поверхностные знания получаются из описания результатов наблюдений за внешним видом/поведением объектов ПО
- ПС – самая распространённая к началу 2000 г. модель ПЗ для построения БЗ (более 80% систем ИИ, основанных на знаниях)
- Большинство диагностических ЭС ограничиваются использованием поверхностных знаний
- Одна из причин высокой популярности ПС как модели ПЗ – совпадение схемы функционирования с формой процессов человеческого мышления при решении задач (Ньюэлл, Саймон, 1972-73 гг.)



СПАСИБО ЗА ВНИМАНИЕ!