

---

# Основы теории интеллектуального управления

---

## Лекция 2.4

Представление знаний в виде  
правил-продукций

---

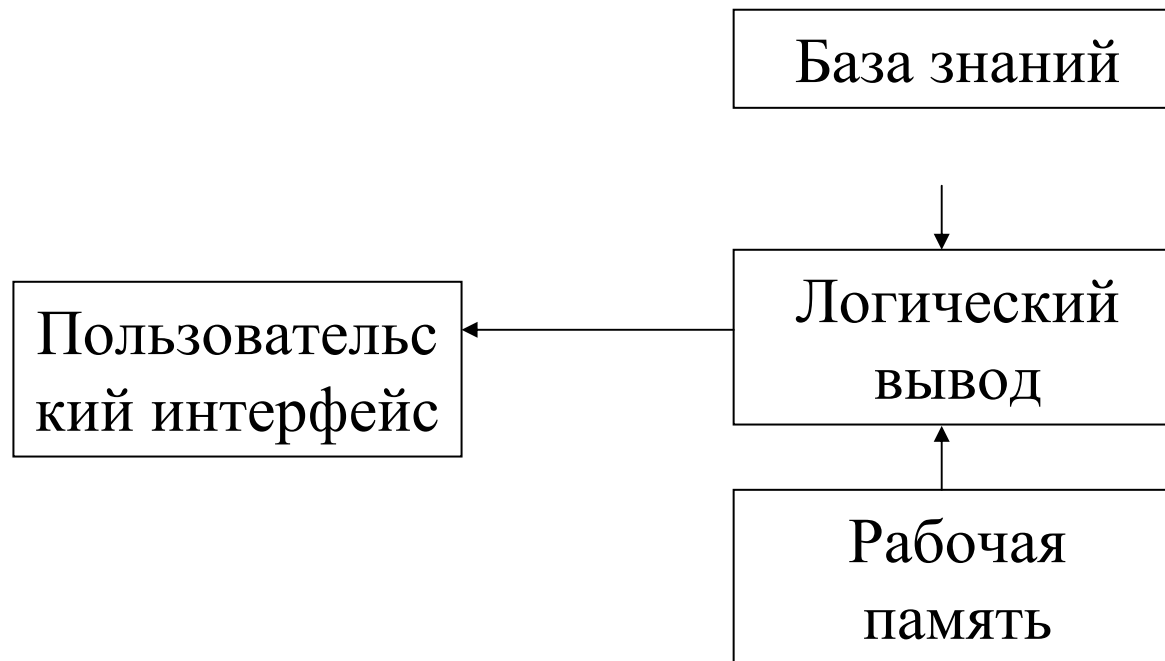
---

# Экспертные системы

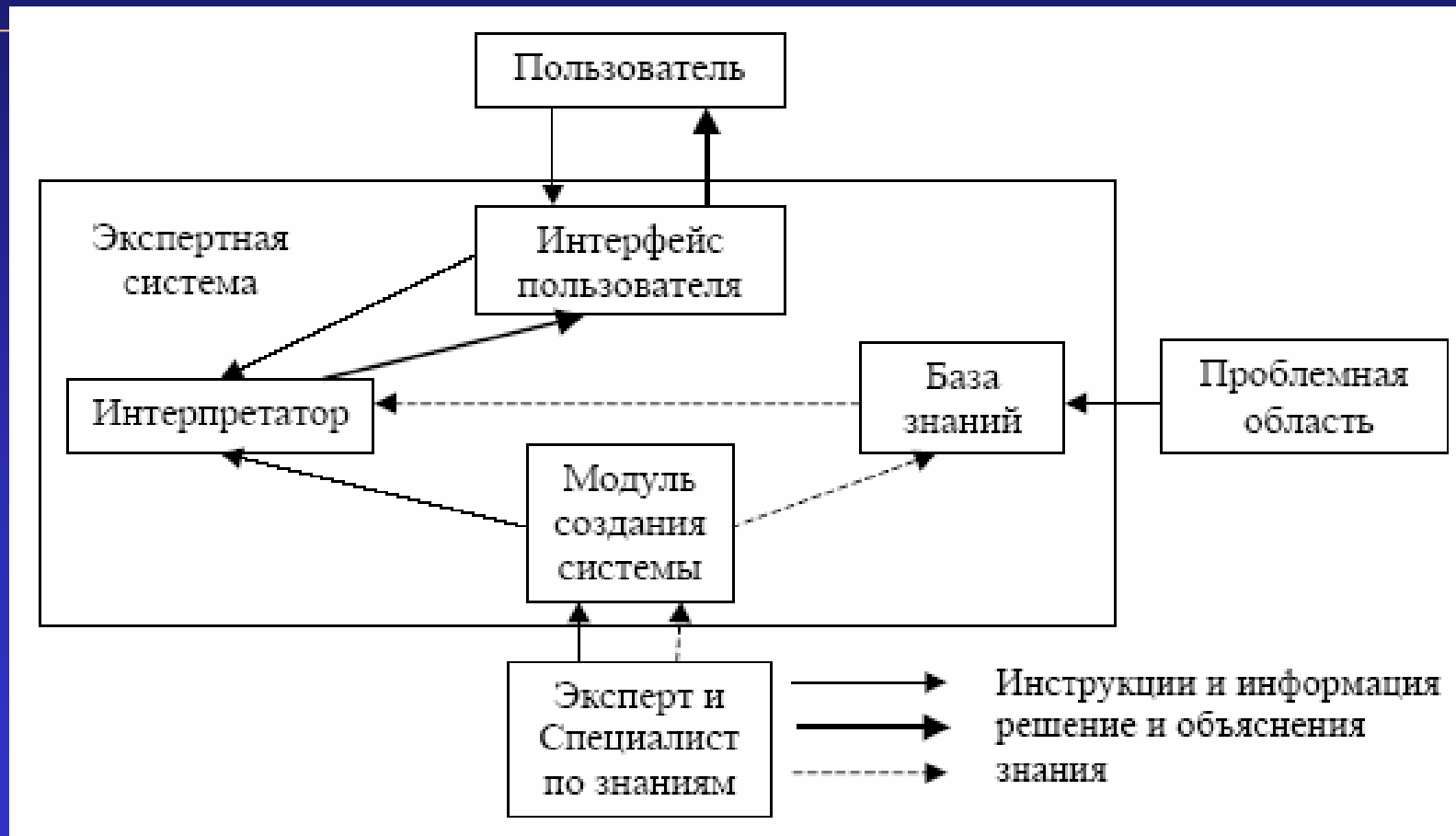
---

- Правила (rules) для представления знаний появились с появлением экспертных систем (1969). С этого начались системы, основанные на знаниях, и инженерия знаний
- Экспертные системы обеспечивают диагностику или выдачу рекомендаций для решения реальных задач на основе формализованных знаний экспертов
- Разрабатываются для замены реальных экспертов
- Примеры:
  - Медицинский диагноз – программа заменяет квалифицированного доктора для диагностики и выдачи рекомендаций по лечению
  - Диагностика неисправностей автомобиля – выдает на основе симптомов диагностику двигателя

# Структура экспертной системы.



## Структура экспертной системы (2)



---

# Системы, основанные на знаниях – введение в правила

---

- База знаний экспертной системы часто основана на правилах – система имеет список правил, которые определяют что нужно сделать в различных ситуациях
- Эти правила исходно формулируются экспертом
- Правила называются правилами-продукциями
- Каждое правило имеет две части – условие и заключение (действие)
  - Условие – что должно быть истинно чтобы правило сработало
  - Заключение – что происходит, когда правило срабатывает
- **Правило можно представить как IF-THEN правило**

---

# УСЛОВИЯ

---

- Условия состоят из двух частей:
  - Объекты – например, погода (weather)
  - Значение объектов – например, солнечная (sunny)
    - IF sunny(weather) THEN print “wear sunglasses”
    - Если погода солнечная то надень темные очки
- Может быть еще оператор типа «больше» и т.п. –
  - IF >30(temperature) THEN print “take some water”
  - Если температура больше 30 то возьми немного воды
- Простые условия могут объединяться логическими связками AND, OR, NOT
  - IF sunny(weather) AND outdoors(x) print “give your sunglasses x”
  - Если погода солнечная и X находится на улице то дай свои темные очку X

---

# Правило

---

- Правило - (I, A, P, A->B, F)
- I – идентификатор правила (номер или имя)
- O – область использования правила
- P – условие использования правила
- A – условие правила
- B – заключение правила
- F – постусловие (комментарии или дополнительные действия)
- A->B – ядро правила, может быть разная интерпретация ядра

---

# Виды интерпретации ядра

---

- Логическая
  - $A$  – логическая функция с логическими связками
  - Если условие истинно, правило срабатывает
- Вероятностная
  - $A$  – логическая функция с логическими связками
  - Правило срабатывает с некоторой вероятностью
- Пороговая
  - $A$  – множество факторов, которые суммируются с учетом своих весов, и правило срабатывает, если сумма превышает определенный порог (модель нейрона)



---

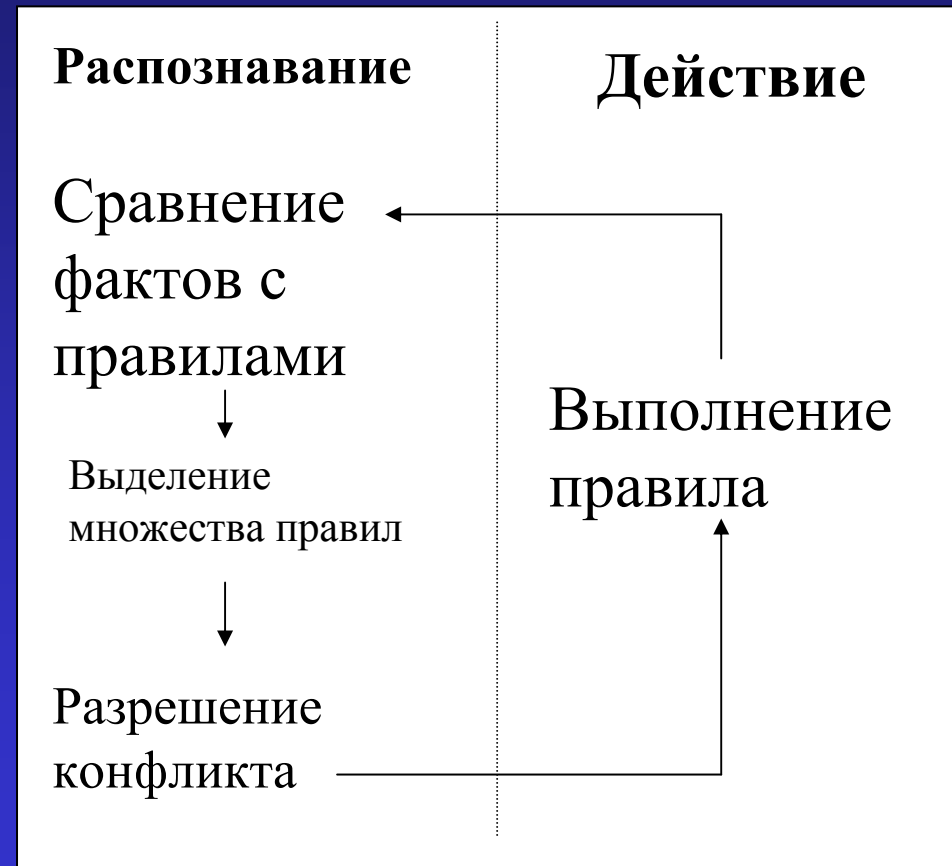
# Рабочая память

---

- Содержимое рабочей памяти постоянно сравнивается с правилами
- Когда факты из рабочей памяти совпадают с условием правила, правило срабатывает
- Более одного правила могут быть готовы к исполнению

# Цикл «распознавание-действие»

- Когда факт совпадает с условием правила, правило добавляется к множеству правил, которые могут выполняться
- Система должна решать какое из правил выбрать для выполнения (разрешить конфликт)



# Стратегии разрешения конфликтов

---

## Упорядоченные по специализации

Если условная часть правила является подмножеством условной части другого правила, то это другое обладает большим приоритетом, т.к. описывает более конкретный (специальный) случай

## В порядке расположения правил

Выбирается первое подходящее правило

## Упорядоченные данные

Данные ранжируются в соответствии с приоритетом. Выбирается правило, которое применяется для данных с высшим приоритетом

## Упорядоченные по размеру условия

Выбирается правило с самым длинным условием (количество элементарных условий)

## Упорядоченные по времени использования

Недавно использованное правило обладает большим приоритетом.

Давно использованное правило обладает меньшим приоритетом. Недавно использованные данные обладают большим приоритетом.

Давно использованные данные обладают меньшим приоритетом

---

# Виды логического вывода

---

- Обратный
  - От цели к фактам (как в Prolog или как в ESWin)
- Прямой
  - От фактов к цели

---

# Прямой логический вывод

---

## Алгоритм match-resolve-act cycle:

### loop

1. Сравнение условий правил с фактами в рабочей памяти и создание множества подходящих правил;
  2. **если** в множестве более одного правила, *разрешение конфликта*;
  3. выполнение выбранного правила
- until** действие - STOP **or** нет подходящих правил

# Решение задач в ЭС. Прямой вывод

Правило 1: Если Коэффициент рентабельности  $> 0.2$

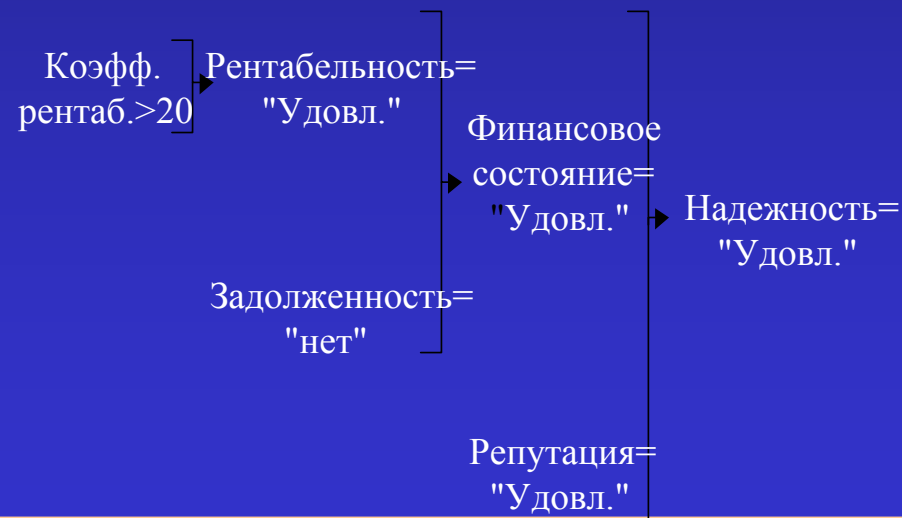
То Рентабельность = "удовл." CF 100

Правило 2: Если Задолженность = "нет" и Рентабельность = "удовл."

То Финансовое\_сост. = "удовл." CF 80

Правило 3: Если Финансовое\_сост. = "удовл." и Репутация="удовл."

То Надежность предприятия = "удовл." CF 90



---

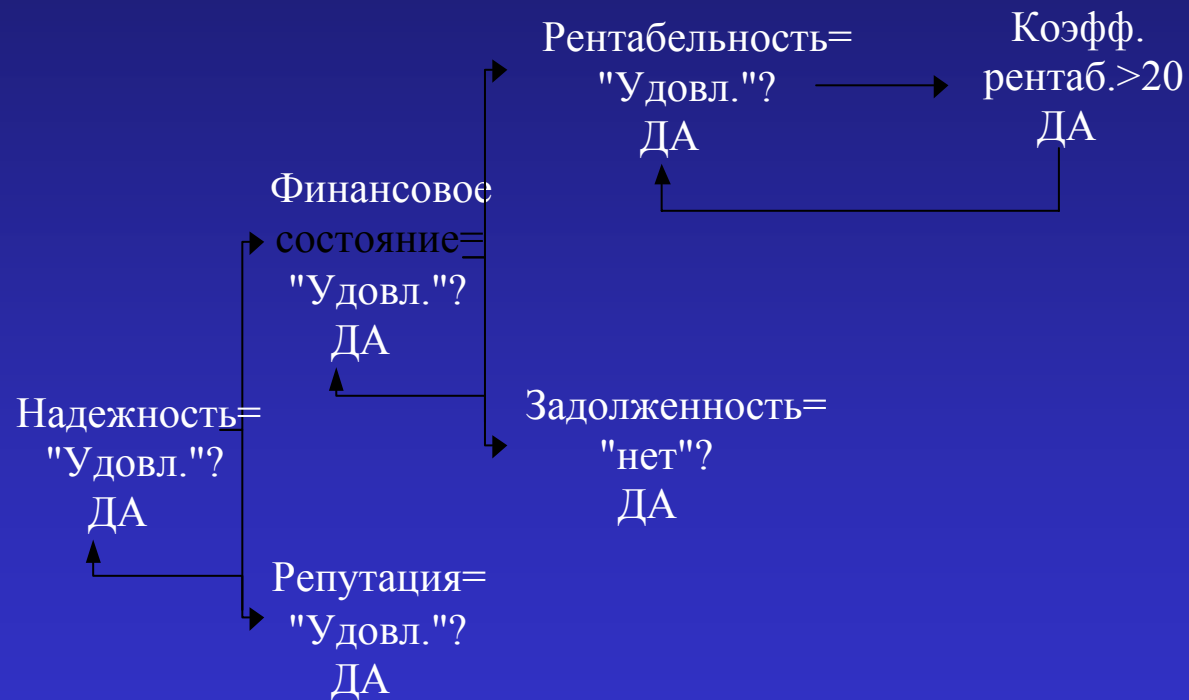
# Обратный логический вывод

В обратном выводе, мы работаем от возможного заключения (цели) к фактам.

Таким образом, обратный логический вывод управляется целью, как в Прологе.

Обратный логический вывод использует стек для запоминания текущей цели в порядке просмотра дерева правил сверху вниз для того, чтобы иметь возможность возвратиться в случае неудачи

# Решение задач в ЭС. Обратный вывод





# Представление неопределенности в правилах

- Факты с коэффициентом достоверности (confidence, certainty)
  - Confidence может быть в диапазоне  $(0,1)$ ,  $(-1,1)$ ,  $(0,100)$ ,  $(0,10)$
  - Обрабатывается в процессе проверки условий в соответствии с формулами нечеткой логики
- Правила с коэффициентом достоверности
  - Confidence (Conf) связано с заключением правила
  - Это означает, что если confidence условия is 1 (100%), то факт-заключение появляется в базе фактов с коэффициентом достоверности *Conf*

---

# Пример правила

---

ПРАВИЛО 1:

ЕСЛИ

Образование=Высшее И

Возраст=Молодой И

Коммуникабельность=Высокая

ТО

Шансы найти работу=Высокие КД=0.9.

---

- 
- Факты хранятся в базе данных  
производственной системы в форме  
(Объект, значение, КД)

или

(Объект, атрибут, значение, КД).

## Расчет коэффициента достоверности нового факта в ЭС МУСIN

$$КД = \begin{cases} ИП + РП(1 - ИП); ИП, РП > 0 \\ -(|ИП| + |РП|(1 - |ИП|)); ИП, РП < 0 \\ \frac{ИП + РП}{1 - \min(|ИП|, |РП|)}; ИП * РП < 0, \end{cases}$$

- КД – новое значение факта (-1, 1),
- ИП – показатель истинности уже существующего факта (исходный показатель),
- РП – показатель факта, формируемый исходя из истинности условия и заключения правила (результатирующий показатель).

---

## Расчет коэффициента достоверности нового факта в ЭС МУСИН (2)

---

- Легко проверить, что получающиеся значения не входят в противоречие с интуитивным представлением, о том, как должна меняться истинность факта при срабатывании правила, подтверждающего или опровергающего его.
- ИП=0.2; РП=0.6; КД=0.68
- ИП=0.2; РП=-0.6; КД=-0.5
- ИП=-0.2; РП=-0.6; КД=-0.68

---

# Метаправила

---

- Правила, которые управляют другими правилами

Пример метаправила:

ЕСЛИ

Экономика = развивается

ТО

Увеличить приоритет правила 1

- Достоинства правил как метода представления знаний
  - Наглядность и понятность знаний (по крайней мере, на уровне одного правила).
  - Возможность реализации немонотонного логического вывода и обработки противоречивых фактов.
  - Возможность введения различных модификаций в интерпретацию правил в соответствии с особенностями решаемых системой задач.
  - Возможность легкого наращивания базы знаний путем добавления новых правил.
- Недостатки
  - Необозримость большой базы знаний и ее структуры.
  - Возможность легкого внесения серьезных искажений в базу знаний, приводящих к неправильному функционированию системы (если в системе нет развитых средств проверки целостности базы знаний).
  - Ориентация на последовательную обработку правил