

# Lecture 3. Goal Trees and Rule-Based Expert Systems

Коваленко Анастасия  
Рожков Игорь  
Степанова Полина



# Содержание лекции

- Понятие ЭС (история, наглядный пример)
- Архитектура ЭС
- Построение рассуждений в ЭС
  - Деревья решений
- Области применения и примеры ЭС
- Демонстрация работы ЭС на правилах

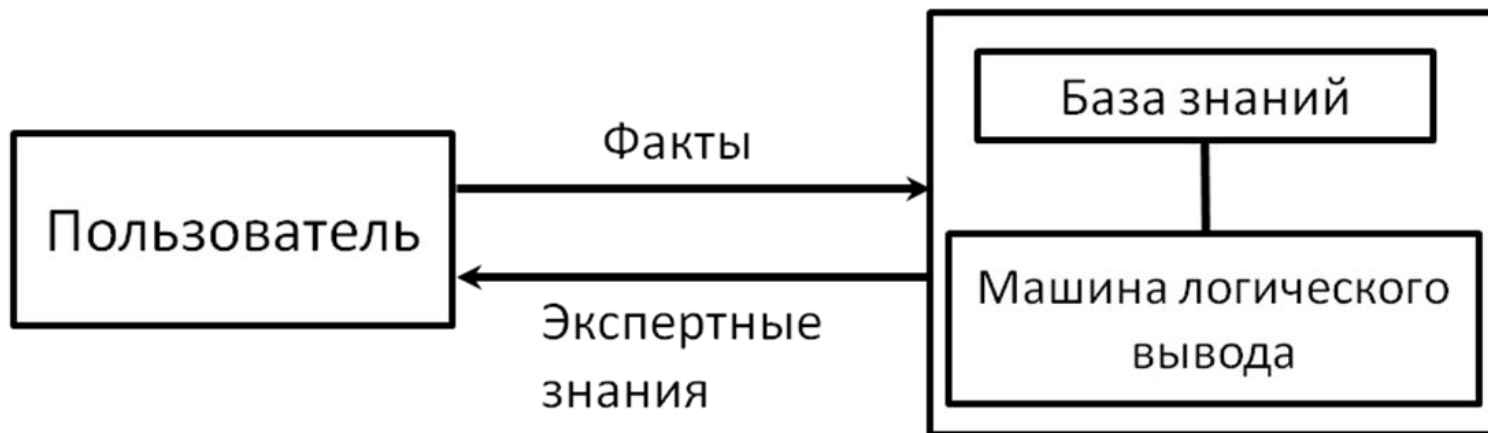
# История вопроса

- Появление компьютеров в 1940-х – 1950-х => желание создать машину, способную “мыслить”
- 1970-е: вместо поиска универсального алгоритма мышления и решения задач => моделирование **конкретных знаний специалистов-экспертов**
- Новый подход к решению задач ИИ - **представление знаний**
- Первые значительные ЭС появились в области медицины - могли принимать диагностические решения
  - mycin, caduceus, dendral

# Экспертная Система

(система основанная на знаниях)

- **Экспертная система** - программа, аккумулирующая знания специалистов (экспертов) в определенной предметной области и оперирующая ими с целью выработки рекомендаций или решения проблем.



# Особенности ЭС

- Доступнее и дешевле услуг эксперта-человека
- Может использоваться там, где жизнь эксперта была бы под угрозой
- Знания собраны из многих источников, не искажаются со временем
- Неизменно правильный, полный, обоснованный ответ в любых обстоятельствах
- Плохо соотносится с реляционными базами данных

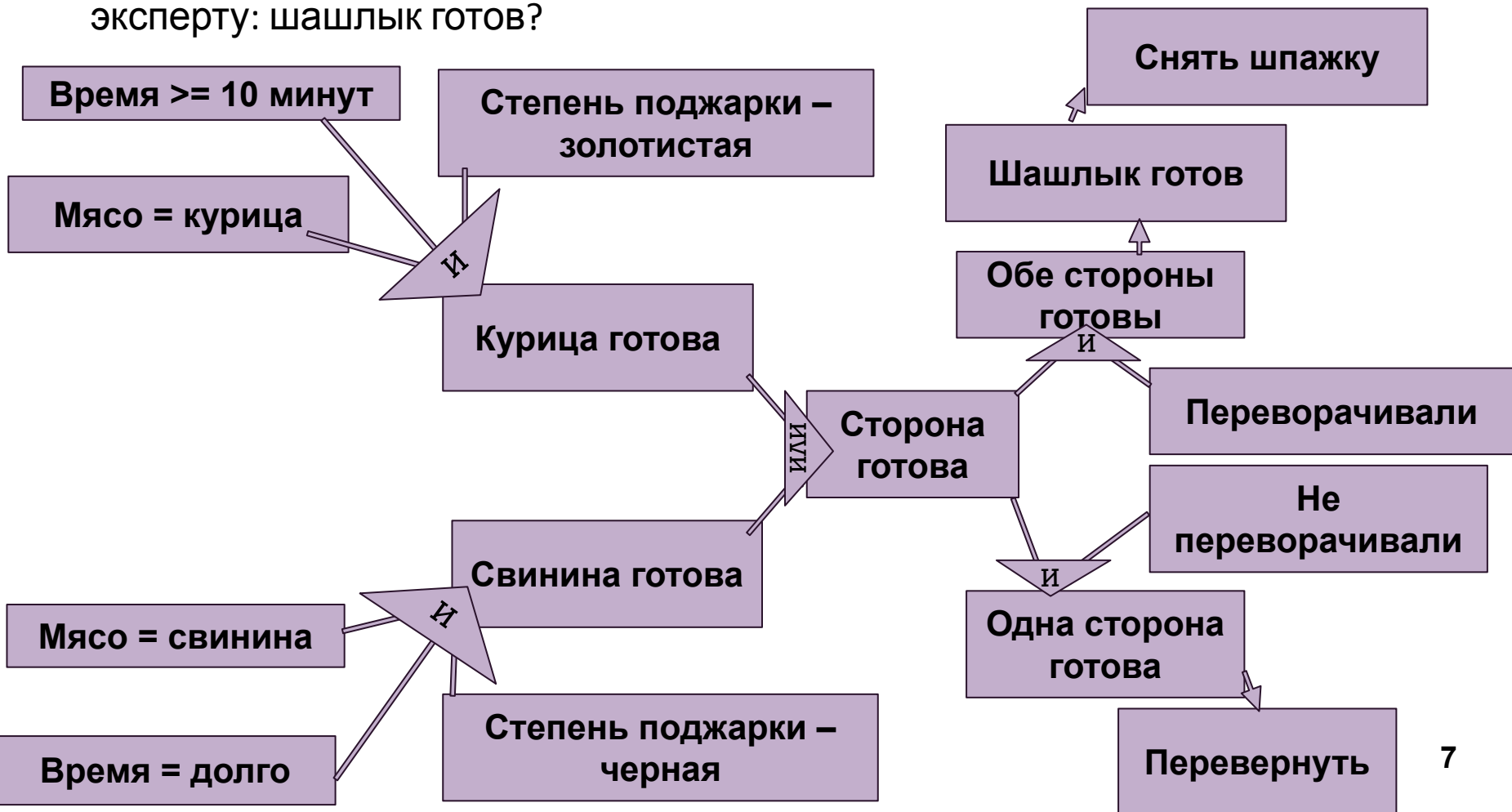
# ЭС наглядно

- Будем готовить шашлык
- Упрощенная схема. Пусть есть только курица или свинина, одна установленная шпалка с мясом и один готовый мангал. Как готовить?

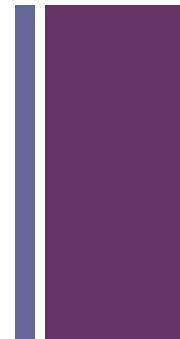


# ЭС по приготовлению шашлыка

Пусть на горящем мангале лежит шпажка с мясом. Вопрос к эксперту: шашлык готов?



# Архитектура ЭС

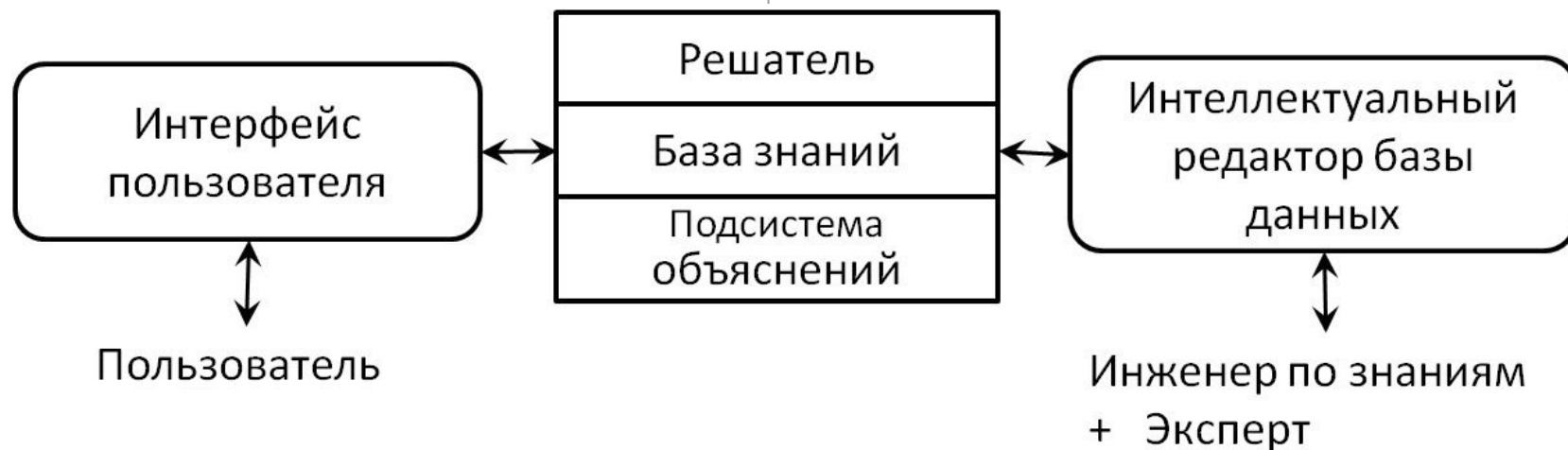


## Основные функции ЭС

- Приобретение знаний
- Представление знаний
- Управление процессом поиска решения
- Разъяснение принятого решения

## Элементы ЭС

- Подсистема усвоения знаний
- База знаний
- Решатель (машина вывода)
- Подсистема объяснения





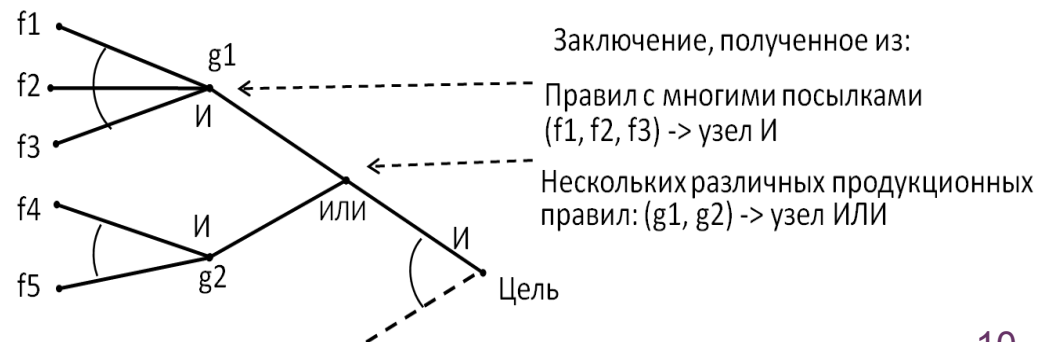
# Представление знаний

- **База знаний (БЗ)** – ядро ЭС, совокупность знаний предметной области, описанных с использованием выбранной модели их представления.
- ЭС по методу построения рассуждений на основе:
  - **правил ЕСЛИ-ТО (продукций)**
  - **моделей(семантические сети, фреймы)**
  - **опыта(базы решений, нейронные сети)** или комбинации этих способов
- **Получение знаний и выбор представления**  
- задача **инженерии знаний**



# Связь ЭС с деревьями целей

- Построение рассуждения в ЭС можно представить в виде **дерева целей**.
- **Дерево целей (И-ИЛИ дерево, дерево решений)** – ориентированный граф, отражающий разделение задачи на подзадачи.
  - исходная задача - корень дерева
  - подзадачи - поддеревья

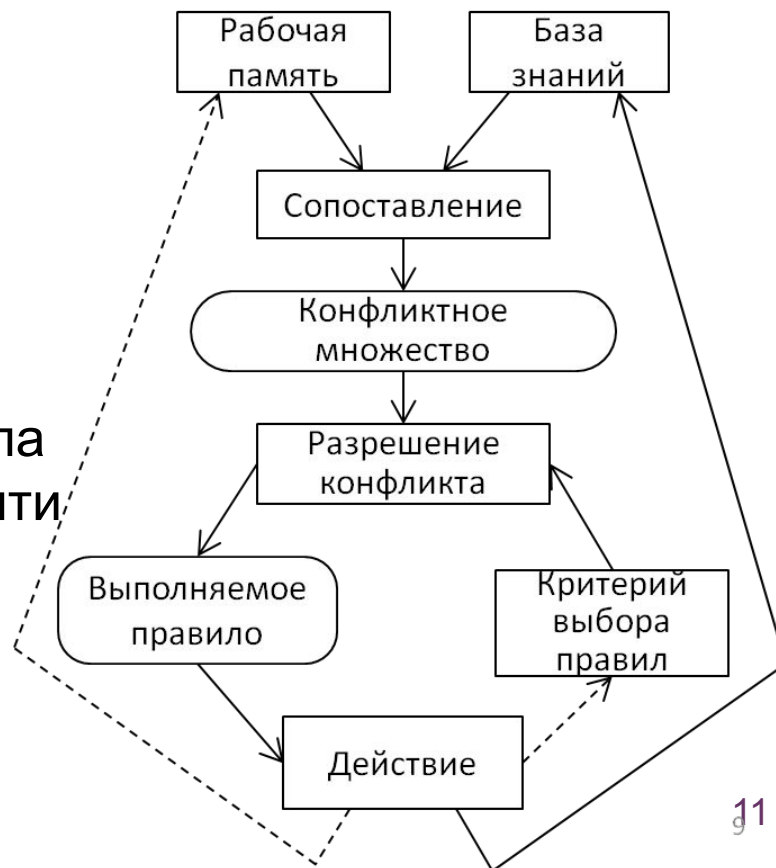


# Механизм вывода

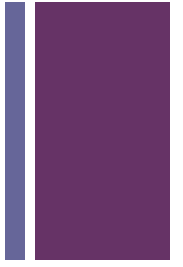
- **Машина вывода(решатель)**—программа, имитирующая логический вывод эксперта, пользующегося данной базой знаний для интерпретации поступивших в систему данных.
- Компоненты **вывода и управления**.

Цикл работы решателя:

- Выборка правил-кандидатов
- Разрешение конфликтов
- Срабатывание подходящего правила
- Действие, изменение рабочей памяти



# Управление выводом



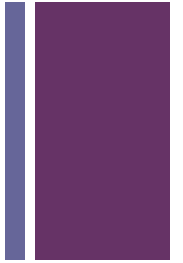
**Проблема управления** – разрешение конфликтов между правилами (приоритет выбора правила в узле ИЛИ):

- Исчерпывающим перебором
- Выбором правила с помощью оценки
- Управление с помощью метаправил

**Метазнания** – знания системы ИИ о ее собственных знаниях: как они структурированы, как и при каких условиях их можно менять.

**Метаправило** – правило, описывающее способы использования и взаимодействия других правил.

# Управление выводом



## Выбор правила с одинаковой посылкой:

П1: утечка серной кислоты -> использовать анион-обменник  
(дорого, опасность невелика)

П2: утечка серной кислоты -> использовать уксусную кислоту  
(дешево, опасность велика)

## Возможные метаправила:

*П3: прежде всего использовать правило, требующее минимальных затрат*

*П4: прежде всего использовать правило с минимальной степенью опасности*

# Направление вывода по дереву целей

- Порядок вывода в ЭС:
  - **Прямой** - от фактов к гипотезе (прогнозирование)
  - **Обратный** - от гипотезы к фактам (диагностика)
- Прямой - ЭС на основе правил с прямой связью (**forward-chaining rule-based expert system**)
  - движение по дереву целей от листьев к корню.
- Обратный - ЭС на основе правил с обратной связью (**backward-chaining rule-based expert system**)
  - движение по дереву целей от корня к листьям.

# Экспертные системы на основе правил с прямой СВЯЗЬЮ



## ■ Известные факты

есть шерсть

вперед  
смотрящие глаза

есть когти

ест только мясо

острые зубы

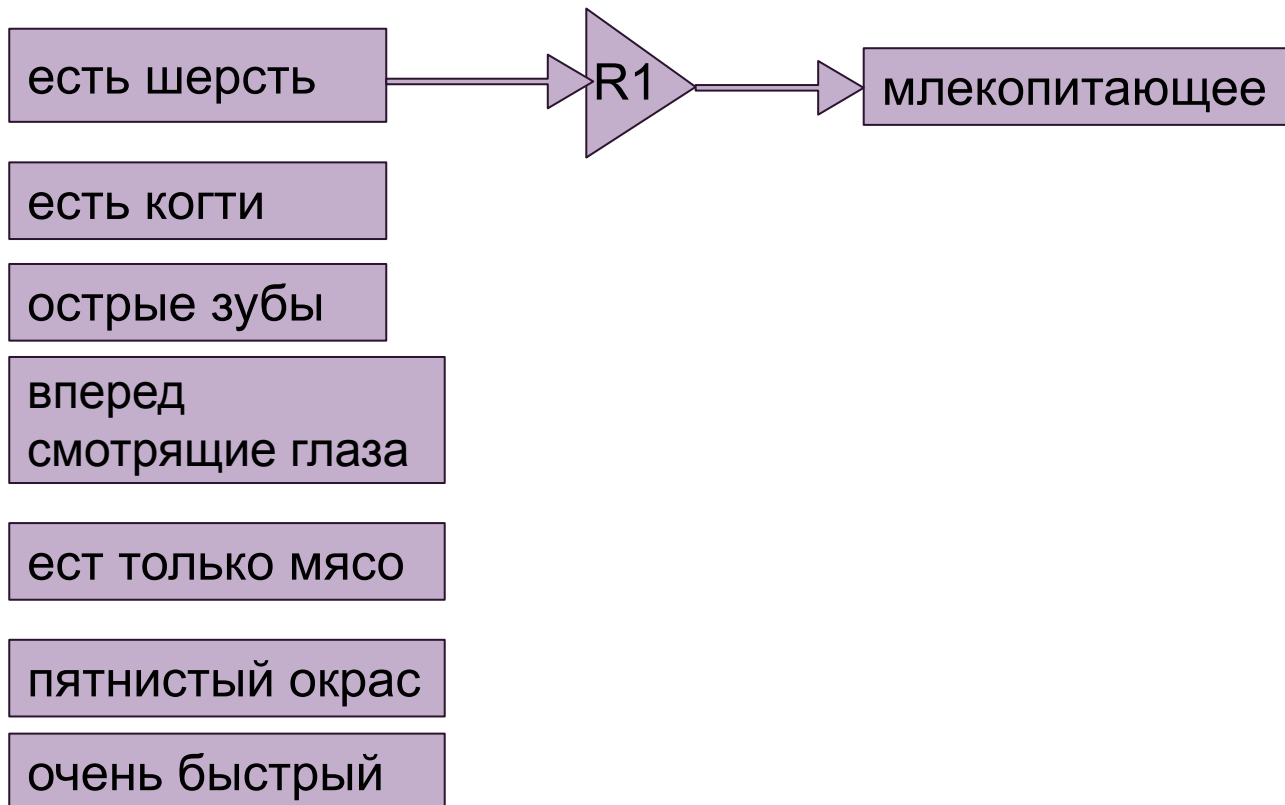
пятнистый окрас

очень быстрый

## ■ Какой вывод о животном может быть сделан?

# Экспертные системы на основе правил с прямой СВЯЗЬЮ

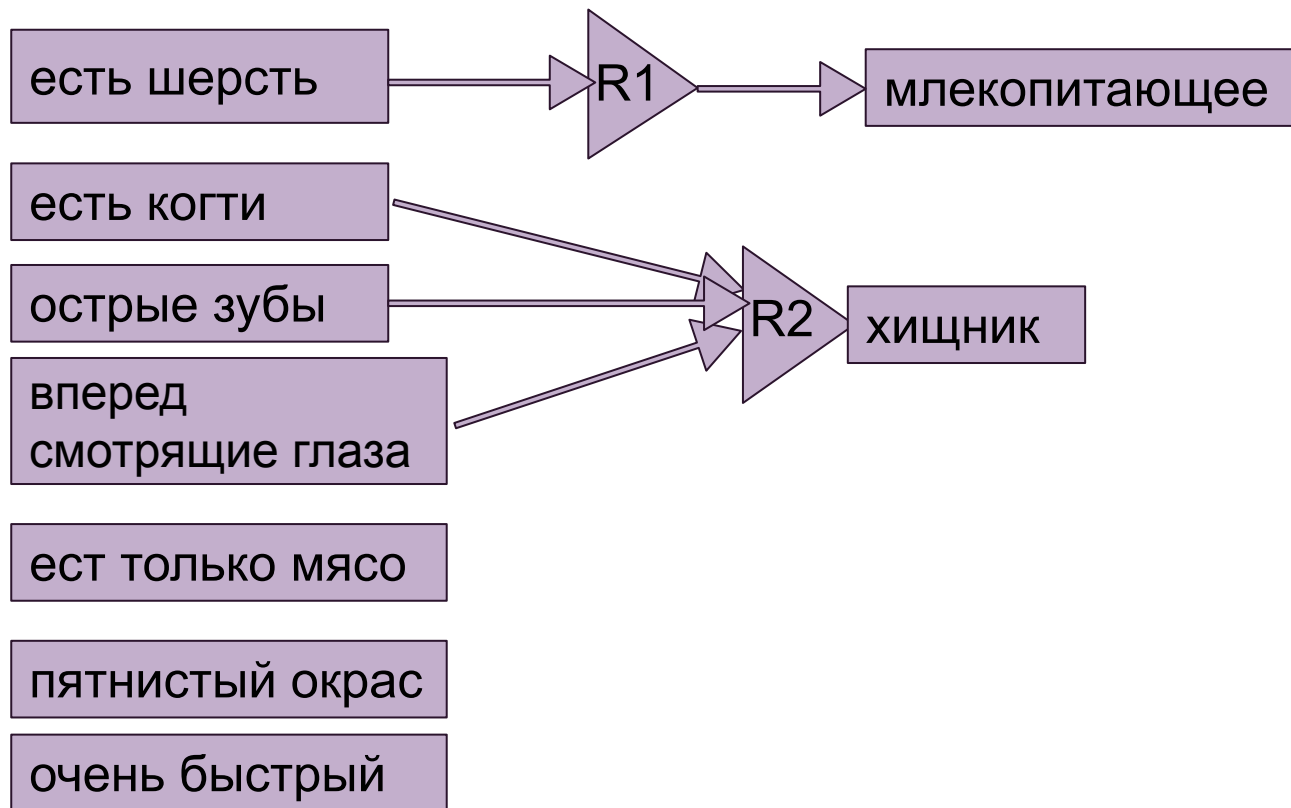
- Применяем известные правила к фактам





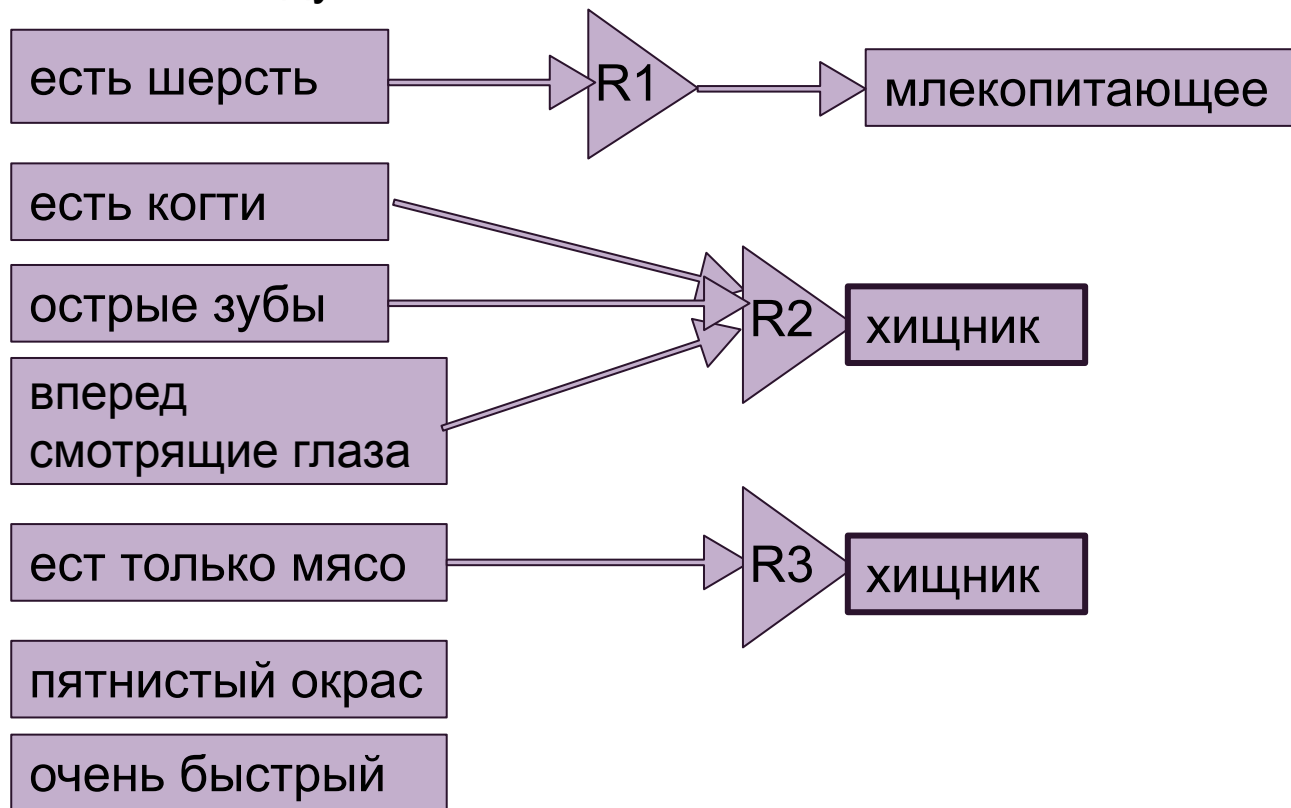
# Экспертные системы на основе правил с прямой СВЯЗЬЮ

- Применяем известные правила к фактам



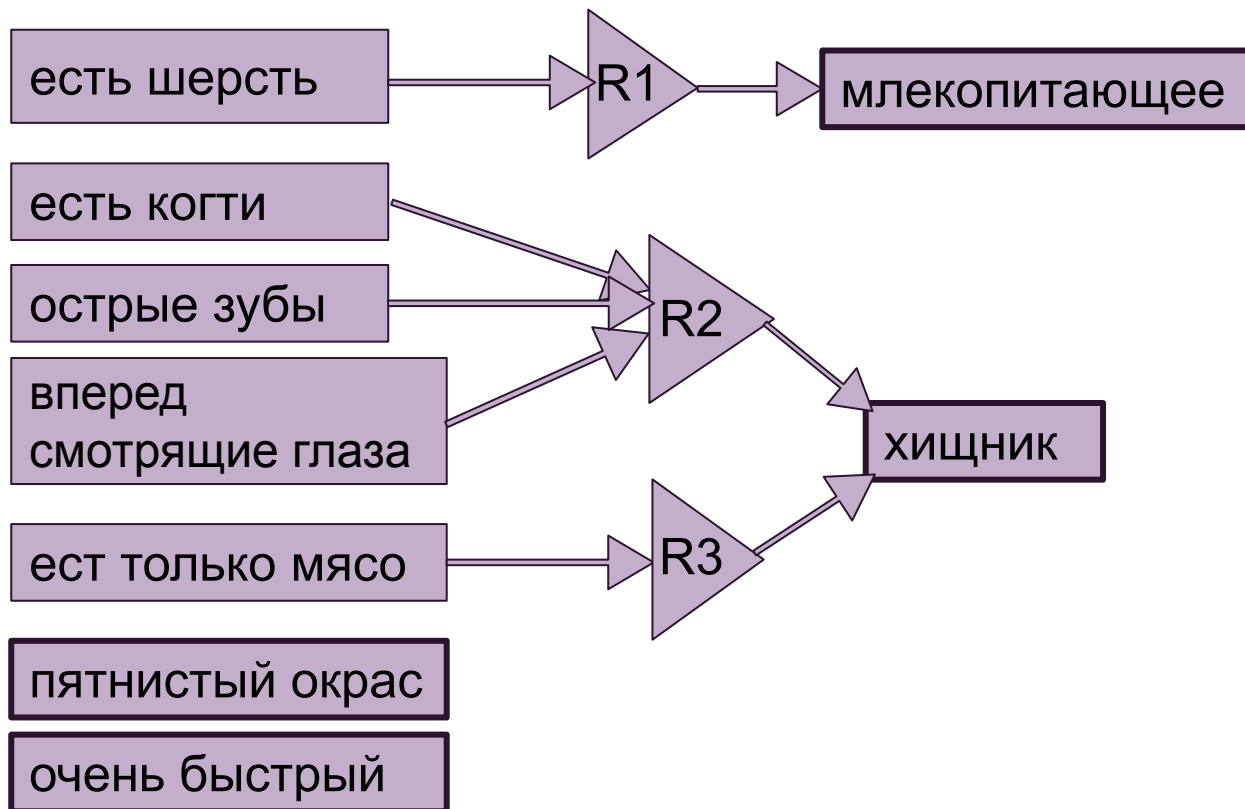
# Экспертные системы на основе правил с прямой СВЯЗЬЮ

- Несколько правил могут приводить к одинаковому выводу



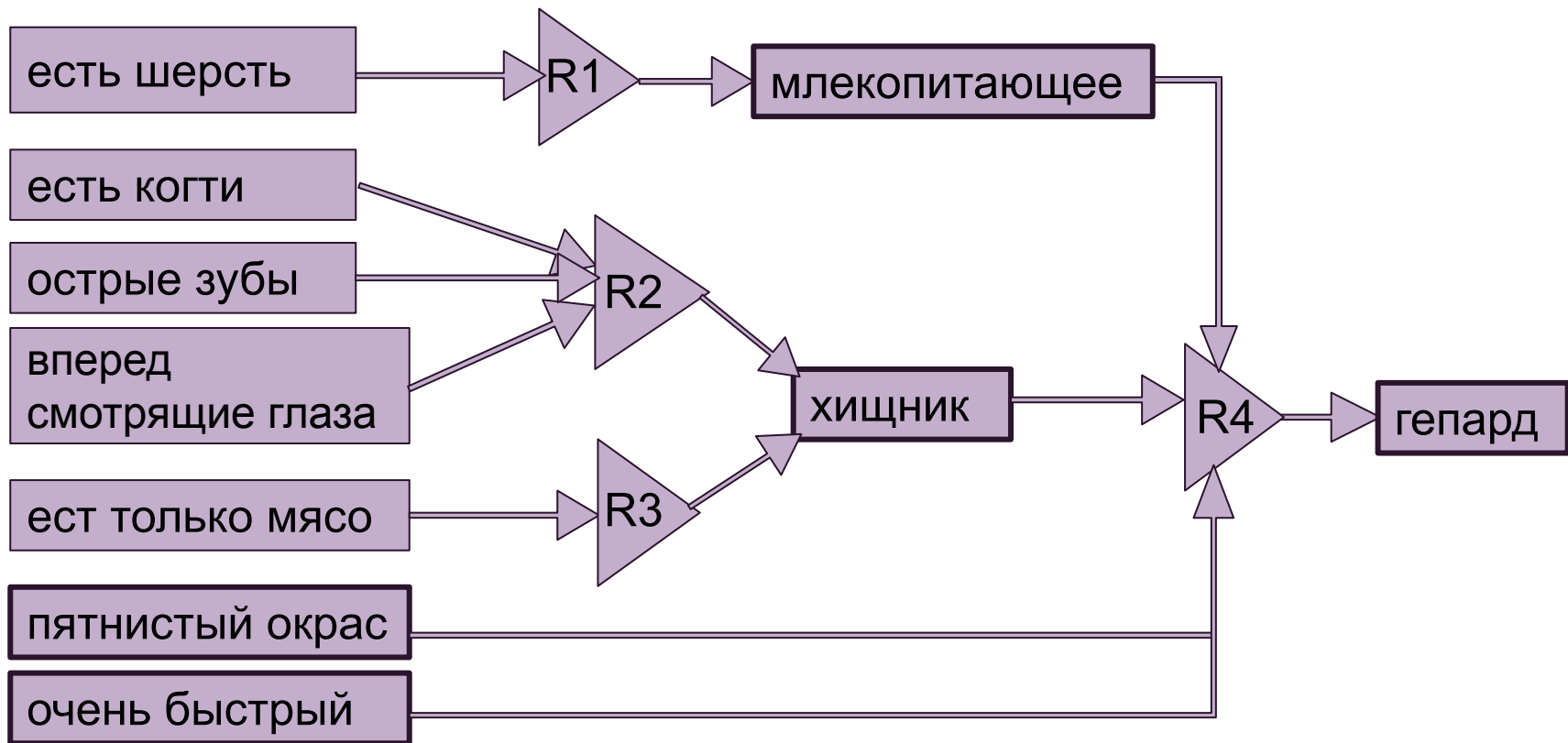
# Экспертные системы на основе правил с прямой СВЯЗЬЮ

- Применяем известные правила к фактам



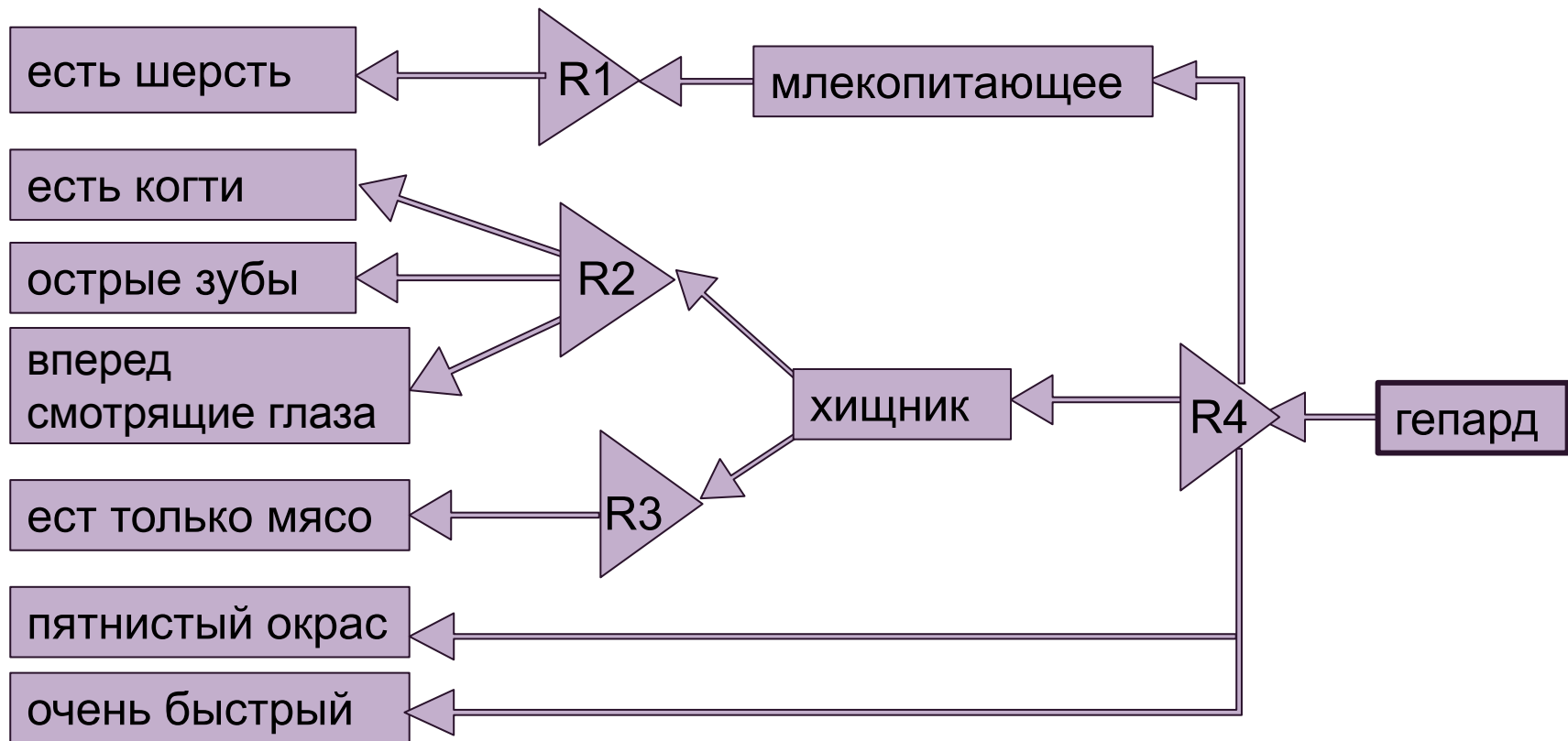
# Экспертные системы на основе правил с прямой СВЯЗЬЮ

- Применение правила приводит к получению финального вывода
- Построено И-ИЛИ дерево (дерево целей)



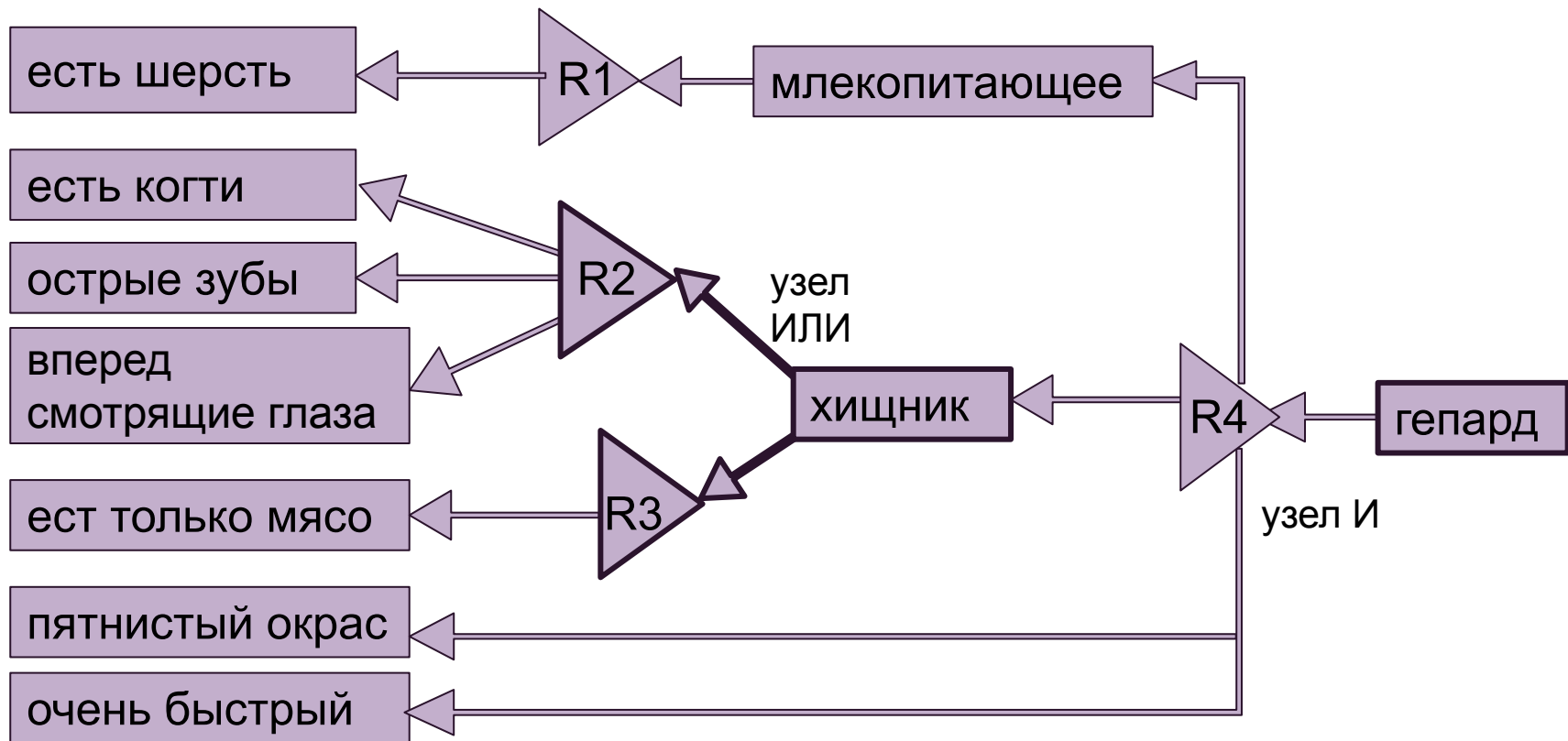
# Экспертные системы на основе правил с обратной СВЯЗЬЮ

- Рассуждение от гипотезы к попыткам доказательства



# Экспертные системы на основе правил с обратной СВЯЗЬЮ

- Рассуждение от гипотезы к попыткам доказательства



# Преимущества и недостатки ЭС с прямой связью



## Преимущества:

- ★ Получение новых знаний
- ★ Возможность сделать сразу несколько выводов

## Недостатки:

- Полный процесс вывода цепочки рассуждений может занять много времени
- Объяснения фактов или наблюдений не очень прозрачны

# Преимущества и недостатки ЭС с обратной связью



## Преимущества:

- ★ Эффективный вывод без анализа альтернативных ветвей
- ★ Более быстрый метод рассуждения

## Недостатки:

- Необходимость использования метаправил для выбора среди альтернатив



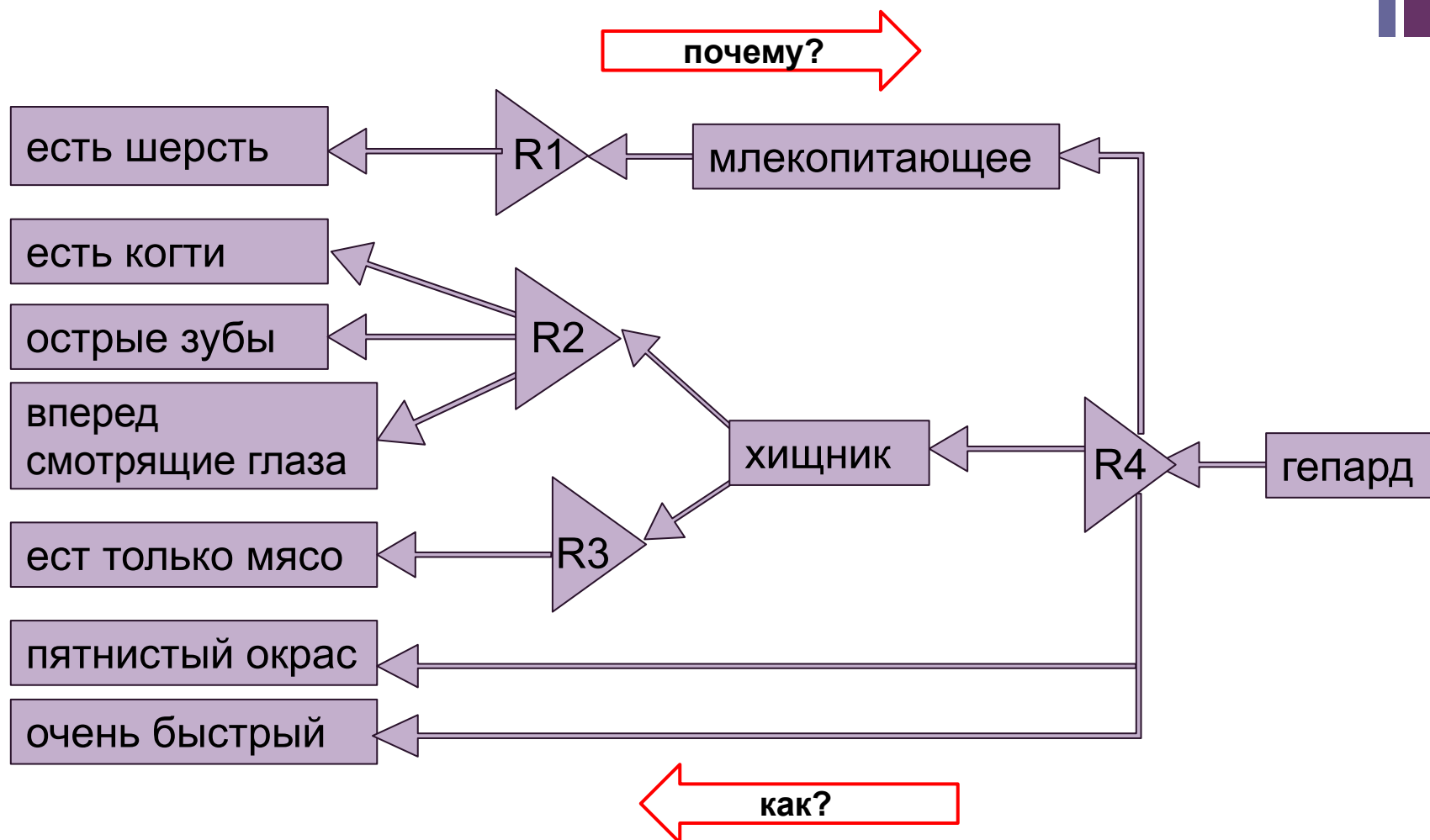
# Подсистема объяснений

*“ЭС - программа, которая может ответить на вопросы о своем поведении”*

## Что объяснять?

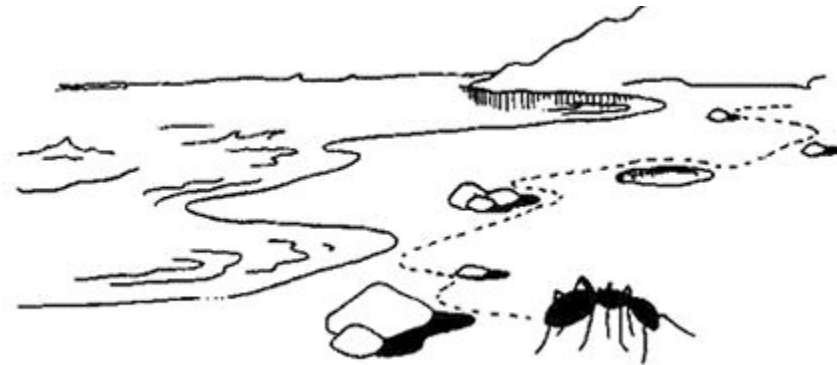
- как получено решение
- как использована некоторая информация (факты, правила)
- **почему** использована или нет некоторая информация
- что использовано в целом при решении задачи

# Подсистема объяснений



# Сложность поведения

Герберт Саймон: Муравей Саймона (1969 г.)



Сложность поведения =  
 $\text{MAX}(\text{Сложность окружения}, \text{Сложность программы})$

# Области применения

ЭС применяются при решении **задач анализа и синтеза**:

- **Анализ**: задана модель исследуемого объекта, требуется определить ее неизвестные характеристики.
- **Синтез**: заданы условия, которым удовлетворяют характеристики модели объекта, требуется построить эту модель.
- Комбинированные задачи.

ЭС полезны для решения **слабо структурированных задач**: нет алгоритмического решения из-за слишком большого количества параметров и

# Классификация ЭС: задачи

## ■ Анализ

- *интерпретация данных*
- *диагностика (медицинская/технологическая)*

## ■ Синтез

- *проектирование*
- *планирование*

## ■ Комбинированные

- *обучение*
- *мониторинг*
- *прогнозирование*
- *управление*
- *поддержка принятия решения*

# Примеры ЭС: **DENDRAL**

- Разработка: 1960е, Стенфордский университет, Lisp
  - Heuristic Dendral - вывод
  - Meta-Dendral - приобретение знаний (правил)
- Задача: проектирование - определение возможной молекулярной структуры химического соединения по его спектру масс
- Считается первой программно реализованной ЭС

# Примеры ЭС:MYCIN

- Разработка: 1970е, Стенфордский Университет, Lisp
  - Вопросно-ответная система, ~600 правил
- Задача: диагностика бактериальных инфекций, рекомендация лекарства
- Результат: 69% приемлемых предложений лечения
- Не применялась на практике из этических соображений и неприемлемо долгого сеанса работы
  - 1980е - пустые “оболочки” ЭС на основе MYCIN

# Примеры ЭС: **PROSPECTOR**

- Разработка: 1970е, Стенфордский исследовательский институт
  - Совмещение правил и организации БЗ в виде семантической сети
  - Предложение и доказательство гипотезы
- Задача: интерпретация данных - помощь в геологической разведке потенциальных месторождений полезных ископаемых

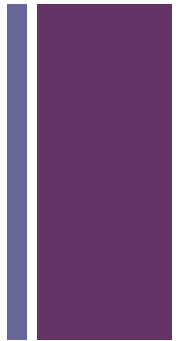


# Примеры ЭС:PROUST

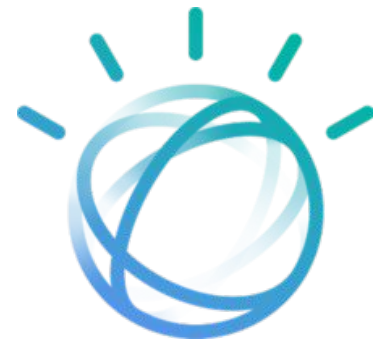


- Разработка: 1985, Lisp
  - Фреймовое представление знаний о программировании
  - Библиотека описаний задач
  - База знаний распространенных ошибок
- Задача: обучение - обнаружение несинтаксических ошибок в программах на языке Паскаль у начинающих программистов
- Принцип работы:
  - Формирование гипотез о используемых методах
  - Синтез возможных верных реализаций
  - Сравнение с представленным решением

# Примеры ЭС: **IBM WATSON**



- Разработка: 2006, Проект IBM DeepQA, Java, C++ и Prolog
  - Распределенная система, параллельность, машинное обучение
  - Вопросно-ответная система общего назначения
- Исходная задача: отвечать на вопросы викторины Jeopardy! соревнуясь с живыми игроками
- Реальные применения: поддержка меддиагностики, торговая рекомендательная система, реклама, ассистент преподавателя, бизнес аналитика и т.д.



# Демонстрация работы ЭС на правилах



## ПРАВИЛА:

0) ЕСЛИ ?x амбициозен

И ?x сквиб

ТО ?x не успеваает в семестре

1) ЕСЛИ ?x живет в башне

Гриффиндор

ТО ?x протагонист

2) ЕСЛИ ?x живет в подземелье

Слизерин

ТО ?x антагонист

И ?x амбициозен

3) ЕСЛИ ?x протагонист ИЛИ ?x

антагонист

И ?x амбициозен

ТО ?x усердно занимается

4) ЕСЛИ ?x усердно занимается

И ?x протагонист

ТО ?x дружится с Гермионой

5) ЕСЛИ ?x целует ?y

И ?x живет в башне Гриффиндор

И ?y живет в подземелье

Слизерин

ТО ?x не успеваает в семестре

# Демонстрация работы ЭС на правилах

## **ФАКТЫ:**

- 0) Миллисент живет в подземелье Слизерин
- 1) Миллисент амбициозна
- 2) Симус живет в башне Гриффиндор
- 3) Симус целуется с Миллисент

**ГИПОТЕЗА:** Миллисент дружится с Гермионой