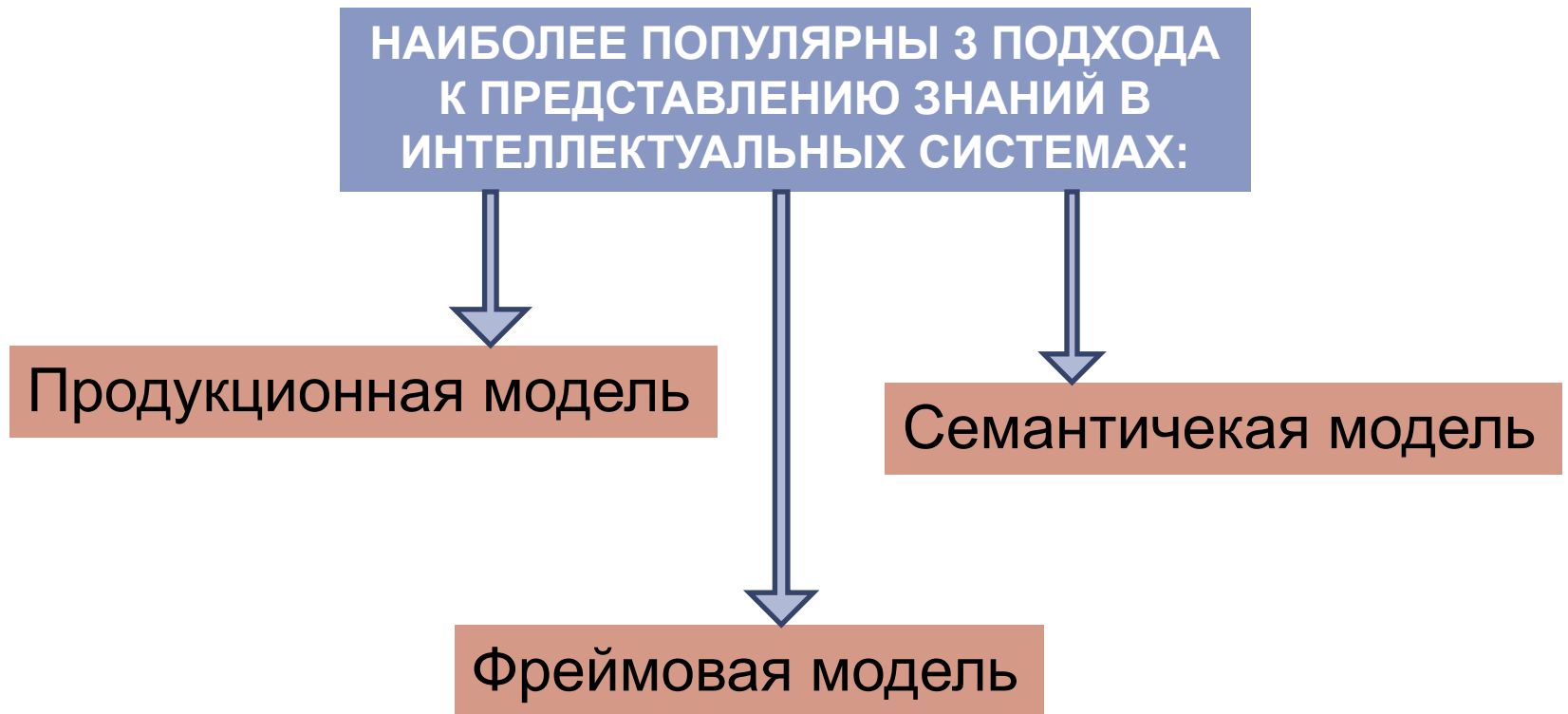


# Представление знаний в искусственном интеллекте

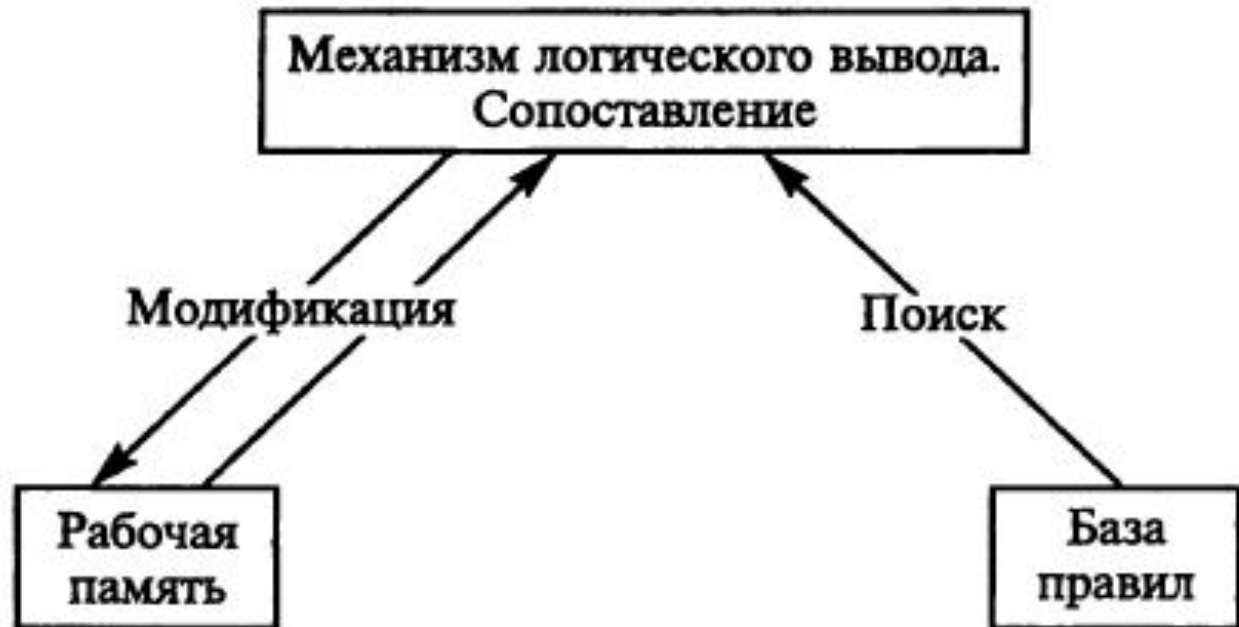


# Продукционные системы.

**Продукционная модель знания** — модель, основанная на правилах, позволяет представить знание в виде предложений типа «Если (условие), то (действие)»

Продукционные системы основаны на использовании базы правил и механизмов логического вывода.

Блок- схема продукционной системы



# Элементы продукционной системы

**Рабочая память** нужна для хранения исходных данных к задаче и выводов (новых фактов), полученных в ходе работы.

**База правил** состоит из фактов и продукционных правил.

**Механизмы логического вывода.** Различают два вида:

**прямой вывод (прямая цепочка рассуждений)**

**обратный вывод (обратная цепочка рассуждений)**

# Продукционная модель представления знаний

## БАЗА ЗНАНИЙ:

**Факт:** это некоторое утверждение ***X есть Y***, обозначают  $Y(X)$ .

### ● «Сократ есть человек, Платон тоже человек»:

- Человек (Сократ), Человек (Платон)

## Продукционное правило:

- ***ЕСЛИ А, ТО В ; Обозначают:  $B :- A$  (Если А истина, то В тоже истина)***

## Примеры:

- «Если «некто» – человек, то он смертен».  
(Или «Смертен «некто», если он человек»)

Пусть  $Y$  – «некто», тогда продукционное правило выглядит так:

- Смертен ( $Y$ ):-Человек ( $Y$ ).

# Пример запросов

## БАЗА ЗНАНИЙ

- Человек (Сократ).
- Человек (Платон).
- Смертен (Y):-Человек (Y).

### Запрос 1:

Человек (X)?

### Вывод продукционной системы:

X = Сократ

X = Платон

Yes (да) – выдаст система в конце.

### Запрос 2:

Смертен (Сократ)?

Вывод: Yes (да).

### Запрос 3:

Смертен (кто)?

Результат:

кто = Сократ

кто = Платон

Yes (да).

# Пример БЗ

- Цифра (0).
- Цифра(1).
- Сочетание (X, Y):- Цифра (X), Цифра (Y).

Цель: Вывести сочетание нулей и единиц для двузначных чисел.

**Запрос:** Сочетание (A, B)?

**Результат:**

A = 0

B = 0

A = 0

B = 1

A = 1

B = 0

A = 1

B = 1

Yes (да).

# Пример

- **Правило 1 (П1).** Если необходимо лечение и высокая температура, то принять аспирин.
- **Правило 2 (П2).** Если ощущение озноб, то высокая температура.
- **Цель(запрос):** Необходимо лечение, ощущение озноб.
- **Рабочая память (РП):**
  - 1) необходимо – лечение; ощущение – озноб,
  - 2) необходимо – лечение; ощущение – озноб, **высокая температура**
  - 3) необходимо – лечение; ощущение – озноб, высокая температура, **принять аспирин.**
- **Заключение (вывод):**  
высокая температура, принять аспирин.

# ПРЯМОЙ ВЫВОД – ВЫВОД ОТ ДАННЫХ К ПОИСКУ ЦЕЛИ

**Суть:** формирование цепочки вывода заключается в многократном повторении элементарных шагов "сопоставить – выполнить".

- Механизм вывода начинает сопоставлять образцы из условных частей правил с образцами, хранимыми в рабочей памяти. Если образцы из условной части имеются в рабочей памяти, то условная часть считается **истинной**, в противном случае – **ложной**.

**Правило 1 (П1).** Если необходимо лечение и высокая температура, то принять аспирин.

**Правило 2 (П2).** Если ощущение озноб, то высокая температура.

**Рабочая память (РП):**

- 1) необходимо – лечение; ощущение – озноб,
- 2) необходимо – лечение; ощущение – озноб, *высокая температура*
- 3) необходимо – лечение; ощущение – озноб, высокая температура, *принять аспирин.*



В данном примере, при рассмотрении П1, **«необходимое лечение»** имеется в РП, но вторая часть условия – **«высокая температура»** в рабочей памяти отсутствует, поэтому условная часть П1 считается **ложной**. Далее рассматривается условная часть П2, она считается истинной, и в рабочую память система сама добавляет резолюцию **«высокая температура»**.

**Заключение (вывод):**  
**высокая температура, принять аспирин.**

Правила закончились, но при их применении РП была изменена, знания были изменены, из этого следует, что **правила начинают проверяться второй раз по циклу**. Теперь срабатывает П2. В рабочую память добавляется запись **«принять аспирин»**. Таким образом система делает заключение: **«высокая температура, принять аспирин»**.

## ОБРАТНЫЙ ВЫВОД

От цели, для ее подтверждения, к данным

**Суть:** "Что нужно, чтобы правая часть данного правила была справедлива, и есть ли необходимые суждения в рабочей памяти?"

Добавляем цель *«принять аспирин»*. Для того, чтобы достичь этой цели необходимо, чтобы была *«высокая температура»*, значит это условие становится для системы новой целью.

Далее рассматривается П2. Мы видим, что для достижения цели *«высокая температура»*, необходимо *«ощущение озноб»*, оно попадает в память.



# Разрешение конфликтов

Правило 3 (П3): Введем в систему 3-е правило (П3):

Если необходимо лечение, то вызвать врача.

**!!! Введение этого правила приводит к возникновению проблемы выбора – какое правило принять при схожих условиях П1 или П3**

Правило 1 (П1). Если необходимо лечение и высокая температура, то принять аспирин.

Соответствующая ситуация, когда на каждом этапе логического вывода существует множество применимых правил, ситуация **называется ситуацией (множеством) конфликтного набора.**

**!!! Проблема разрешения выбора является крайне важной, поскольку именно от нее зависит эффективность этого вывода.**

Один из механизмов такого вывода – у каждого выводимого решения присутствует определенный коэффициент – точность или оптимальность (например, в диапазоне от 0 до 1).

Такая же ситуация может быть и с обратным выводом.

**Правило 4 (П4): Добавим 4-е правило (П4).**


Если ощущение – жар, то высокая температура.

Если на основании этого условия подтверждается цель «принять аспирин», то для достижения первоначальной цели нужно применить лишь правило 1.

**Правило 1 (П1).** Если необходимо лечение и высокая температура, то принять аспирин.

Однако, чтобы подтвердить новую цель *«высокая температура»*, открывается возможность применения **правила 1**, нужно использовать либо правило 2, либо правило 4. Если сначала применить правило 2, то это будет самый удачный выбор, поскольку сразу же можно применить и правило 1.

**Правило 2 (П2).** Если ощущение озноб, то высокая температура.



В обратных выводах применяют **метод исключения из рассмотрения правил**, не имеющих прямого отношения к заданной цели.

**Это позволяет повысить эффективность вывода.**

## Продукционная модель в языке Prolog

- Программа на Прологе состоит из предложений (утверждений). Каждое предложение заканчивается точкой.
- Предложения бывают трех видов:
  - факты,
  - правила,
  - вопросы.

# Факт

- **Факт** – это предложение, у которого нет тела. В терминах логики предикатов, **факт** – это и есть предикат, который является истинным. Он фиксирует (определяет) некоторое отношение между объектами.
- Факт представляет собой безусловно истинное утверждение.
- Например, факт, что Наташа является мамой Даши, может быть записан в виде (в SWI-Prolog строки-константы записываются в одинарных кавычках):  
mata('Наташа', 'Даша').

# Правило

- **Правило** – предложение, истинность заголовка которого в виде предиката зависит от истинности одной или нескольких формул, указанных в теле. Обычно правило содержит несколько хвостовых целей, которые должны быть истинными для того, чтобы само правило было истинным.



# Правила

- В правилах «запятая» – « $,$ » означает операцию И (конъюнкцию), а « $;$ » - ИЛИ (дизъюнкция), например можно так:

grandmama(X,Y):-

    mama(X, Z), mama(Z, Y).

grandmama(X,Y):-

    mama(X, Z), papa(Z, Y).

А можно и так:

grandmama(X,Y):-

    mama(X, Z), mama(Z, Y);

    mama(X, Z), papa(Z, Y).

- Или так:

grandmama(X,Y):- mama(X, Z), (mama(Z, Y); papa(Z, Y)).

# Вопрос (запрос, цель)

- **Вопрос** (запрос, цель) – предложение, состоящее только из тела
- Вопросы используют для выяснения выполнимости некоторого отношения между описанными в программе объектами. Автоматическая система логического вывода Пролога рассматривает вопрос как цель, к которой надо стремиться. Ответ на вопрос может оказаться положительным (true) или отрицательным (false), в зависимости от того, может ли быть достигнута соответствующая цель.

# Внутренняя и внешняя цель

- Программа может содержать вопрос в теле (**внутренняя цель**). Если программа содержит внутреннюю цель, то после запуска программы на выполнение система сразу проверяет достижимость заданной цели. Если внутренней цели в программе нет, то после запуска программы система выдает приглашение вводить вопросы в диалоговом режиме (**внешняя цель**). Программа, компилируемая в исполняемый файл, обязательно должна иметь внутреннюю цель.
- Если цель достигнута, система отвечает «yes» («true»), в противном случае «no» («false»). Следует отметить, что ответ «no» на вопрос не всегда означает, что он отрицательный. Система может дать такой ответ и в том случае, когда у нее просто недостаточно информации, позволяющей положительно ответить на вопрос. Т.е. Пролог основан на т.н. «**модели закрытого мира**», в которой все, что можно получить на основе описания модели является истиной, а остальное – ложью.

# Переменные

- Во всех предложениях можно использовать **переменные**. Считается, что переменные в теле одного правила неявно связаны квантором всеобщности. Имя переменной в Прологе может состоять из букв латинского алфавита, цифр, знаков подчеркивания и должно начинаться с прописной буквы или знака подчеркивания. При этом переменные в теле правила неявно связаны квантором всеобщности и эквивалентны объектам предметной области. Переменные могут быть свободными или связанными.
- **Свободная переменная** – переменная, которая еще не получила значения. Она не равняется ни нулю, ни пробелу; у нее вообще нет никакого значения. Такие переменные еще называют **неконкретизированными**.
- Переменная, которая получила какое-то значение, называется **связанной**. Такой переменной не может быть присвоено новое значение, т. е., по сути, переменная становится константой.

# Анонимная переменная

- Областью действия переменной в Прологе является одно предложение. В разных предложениях может использоваться одно и то же имя переменной для обозначения разных объектов. Исключением из правила определения области действия является **анонимная** переменная, которая обозначается символом подчеркивания «\_». Анонимная переменная предписывает интерпретатору (компилятору) проигнорировать значение аргумента (терма). Если в правиле несколько анонимных переменных, то все они отличаются друг от друга, несмотря на то, что записаны с использованием одного и того же символа («\_»). Анонимные переменные могут записываться только в качестве терма предиката. Использовать их в выражениях (например, арифметических) нельзя. Пример использования анонимной переменной в вопросе «Есть ли у Даши мама?»:

```
?- мама(_, 'Даша').  
true.
```

# Суть логического вывода в Прологе

Таким образом, программа на Прологе состоит из фактов и правил, выражающих некоторые знания о предметной области. Вопрос - это также некоторый предикат, истинность которого нас интересует. Если вопрос не содержит переменных, то вычисление его значения дает ответ «true» при его истинности, либо ответ «false» при его ложности. Если же в предикате вопроса есть переменные, то ищутся их значения (интерпретация), при которых этот предикат и все предикаты программы становятся истинными. В этом и состоит суть вычислений (логического вывода) программы на Прологе.

# Примеры. База знаний

- `мама('Наташа', 'Даша').`  
`мама('Даша', 'Маша').`  
`мама('Наташа', 'Вася').`

`рапа('Вася', 'Маша').`

`grandмама(X, Y):-`

`мама(X, Z), (мама(Z, Y); рапа(Z, Y)).`

`grandрапа(X, Y):-`

`рапа(X, Z), (мама(Z, Y); рапа(Z, Y)).`

# Запросы

**Вопрос 1** – является ли Наташа мамой Даши:

```
?- мама('Наташа', 'Даша').  
true.
```

**Вопрос 2** – кто является мамой Даши:

```
?- мама(X, 'Даша').  
X = 'Наташа';  
false.
```

Первая строка сообщения означает, что ответ найдет и мамой Даши является Наташа; вторая – что в базе знаний для оставшихся предложений не обнаружены другие мамы Даши.



# Запросы

**Вопрос 3** – есть ли у Даши мама:

?- мама( \_, 'Даша').

True.

**Вопрос 4** – для кого Наташа является бабушкой:

?- grandmama('Наташа', X).

X = 'Маша' ;

X = 'Маша'.

В данном случае выдается два одинаковых ответа «Маша», т.к. правило grandmama в одном случае сработало по цепочке «Наташа – Даша – Маша», а в другом - «Наташа – Вася – Маша». Очевидно, что в приведенном примере базы знаний либо Наташи - это две различные женщины, либо Маши.

# Запросы

**Вопрос 5** – найти всех мам и детей:

?- мама(X,Y).

X = 'Наташа',

Y = 'Даша' ;

X = 'Даша',

Y = 'Маша' ;

X = 'Наташа',

Y = 'Вася'.

**false.**

В данном случае после третьего ответа не выдается «false», т.к. в базе знаний были перебраны все предложения и они все истинны.