

# СЕМАНТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

В.Ш.Гумиров  
П.Ю.Матюков  
Д.Е.Пальчунов



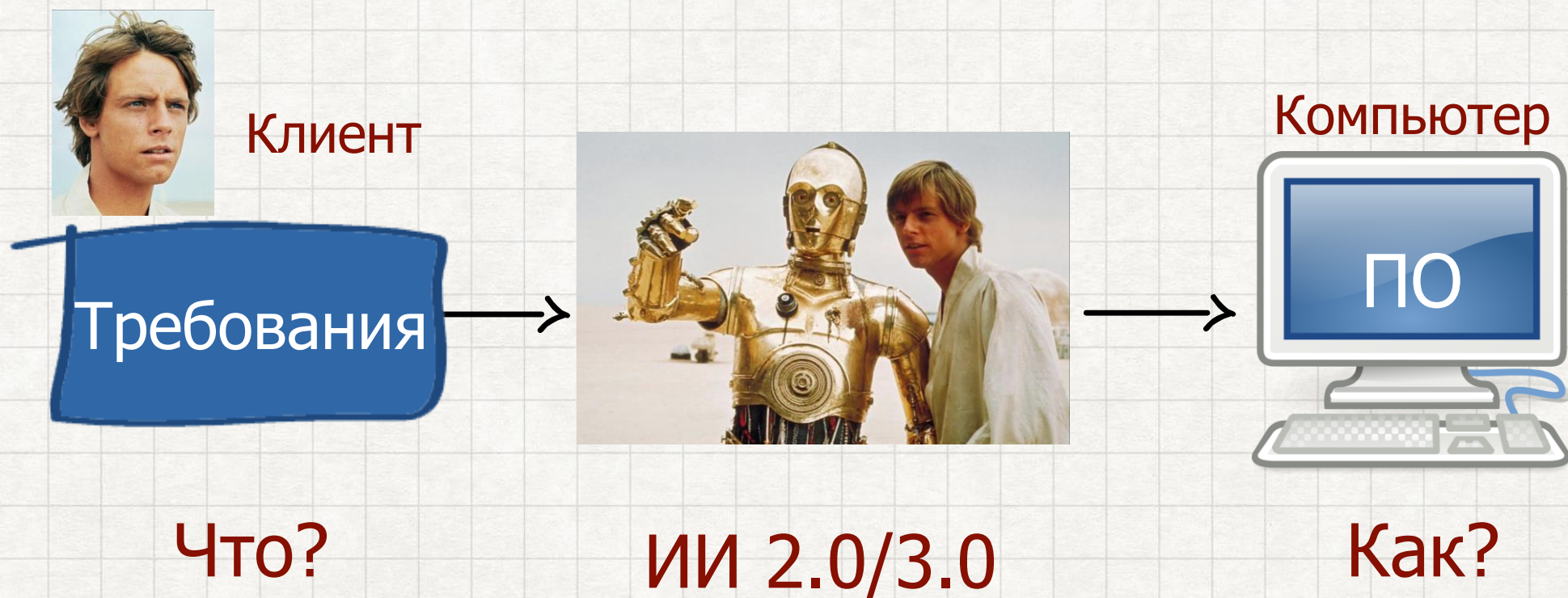
# ПРОБЛЕМА ПРОГРАММИСТА

ДЛЯ ОБЩЕНИЯ С КОМПЬЮТЕРОМ НУЖЕН ТОЛМАЧ



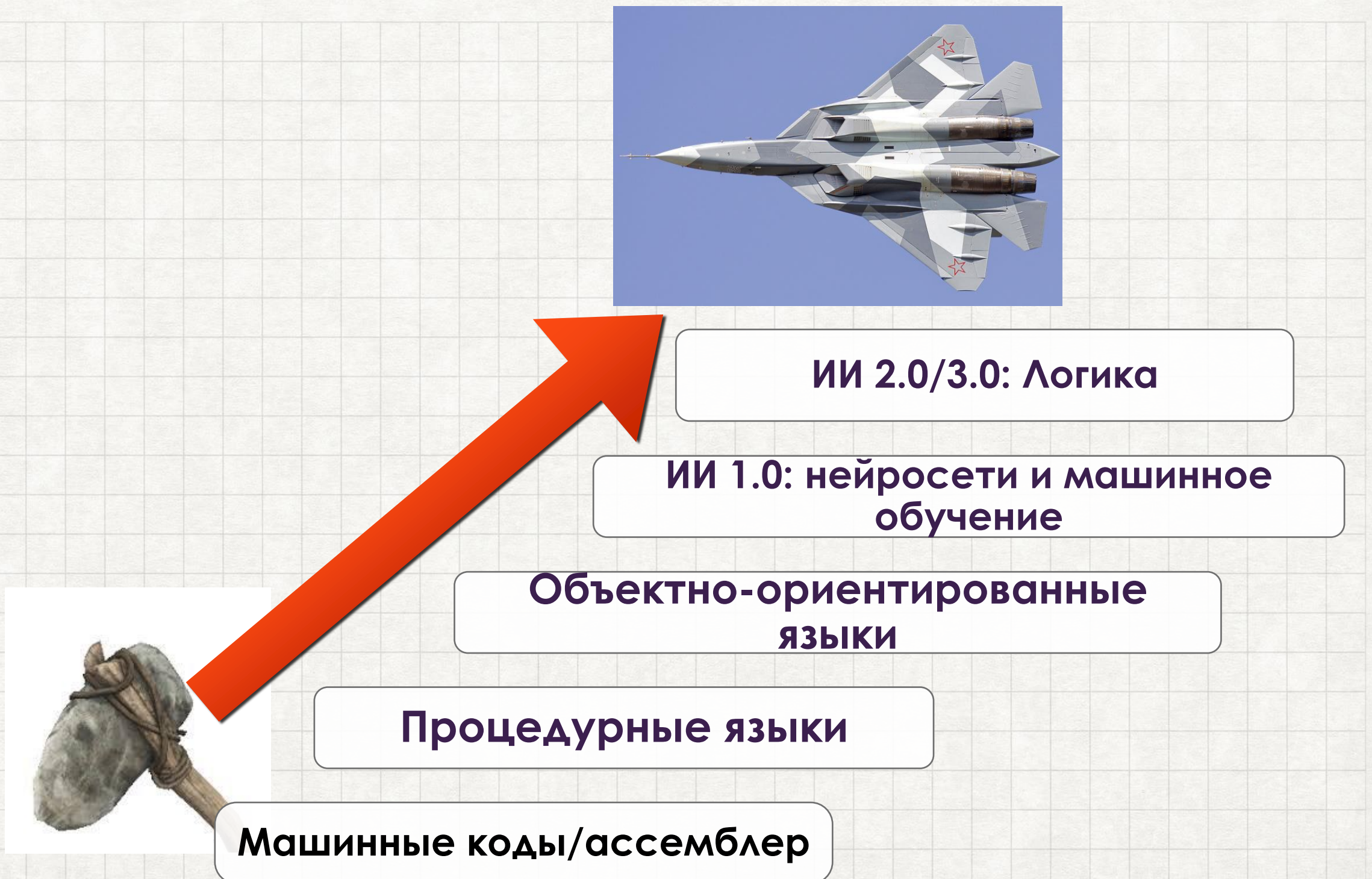


# ИСПОЛНИМЫЕ СПЕЦИФИКАЦИИ





# ЭВОЛЮЦИЯ ИТ





# СТАДИИ РАЗВИТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

## Теория

## Технология

**AI 1.0**

Решение задач на основе  
статистики

Машинное обучение и  
нейросети

**AI 2.0**

Решение задач на основе  
логики  
Исполнимые спецификации

Семантическое  
моделирование

**AI 3.0**

Логический анализ гипотез,  
машинное обучение на  
основе логики

Семантическое извлечение  
знаний  
Автоматическое  
доказательство, ризонеры

**AI 4.0**

Рефлексирование  
Понятийные рассуждения  
Намерения, желания,  
целеполагание

**Требует  
разработки**

# АИ 4.0

- Рассуждения, обоснования, рефлексия
- Детальная спецификация смысла понятий
  - Явное описание способа (алгоритма) ответа на вопрос: верно ли  $P(c_1, \dots, c_n)$
- Реализация эмоциональной оценки утверждений и ситуаций: соответствие целям, намерениям, желаниям, ценностям



# Уровни представления знаний

- Знания верхнего уровня общности:
  - Состояние здоровья, болезни, синдромы, ...
  - Экономические прогнозы, ожидание стабильности или нестабильности, валютных кризисов и т.д.
  
- Знания нижнего уровня общности:
  - Симптомы, результаты анализов, ...
  - Конкретные данные о курсах валют, ценах на нефть, ...

# Цены на нефть

## Нефть и драгметаллы

Нефть марки Brent	55.8500	-0.8400	02.10
Aluminium	2102.0000	-4.0000	29.09
Copper	6481.0000	-20.0000	29.09

Gold	1279.7200	0.00	01.10
Nickel	10500.0000	-20.0000	29.09
Palladium	934.5000	+0.4000	29.09

Platinum	909.5000	-0.5000	29.09
Silver	16.6200	0.00	01.10
Zinc	3162.0000	+1.0000	29.09

## НЕФТЬ МАРКИ BRENT



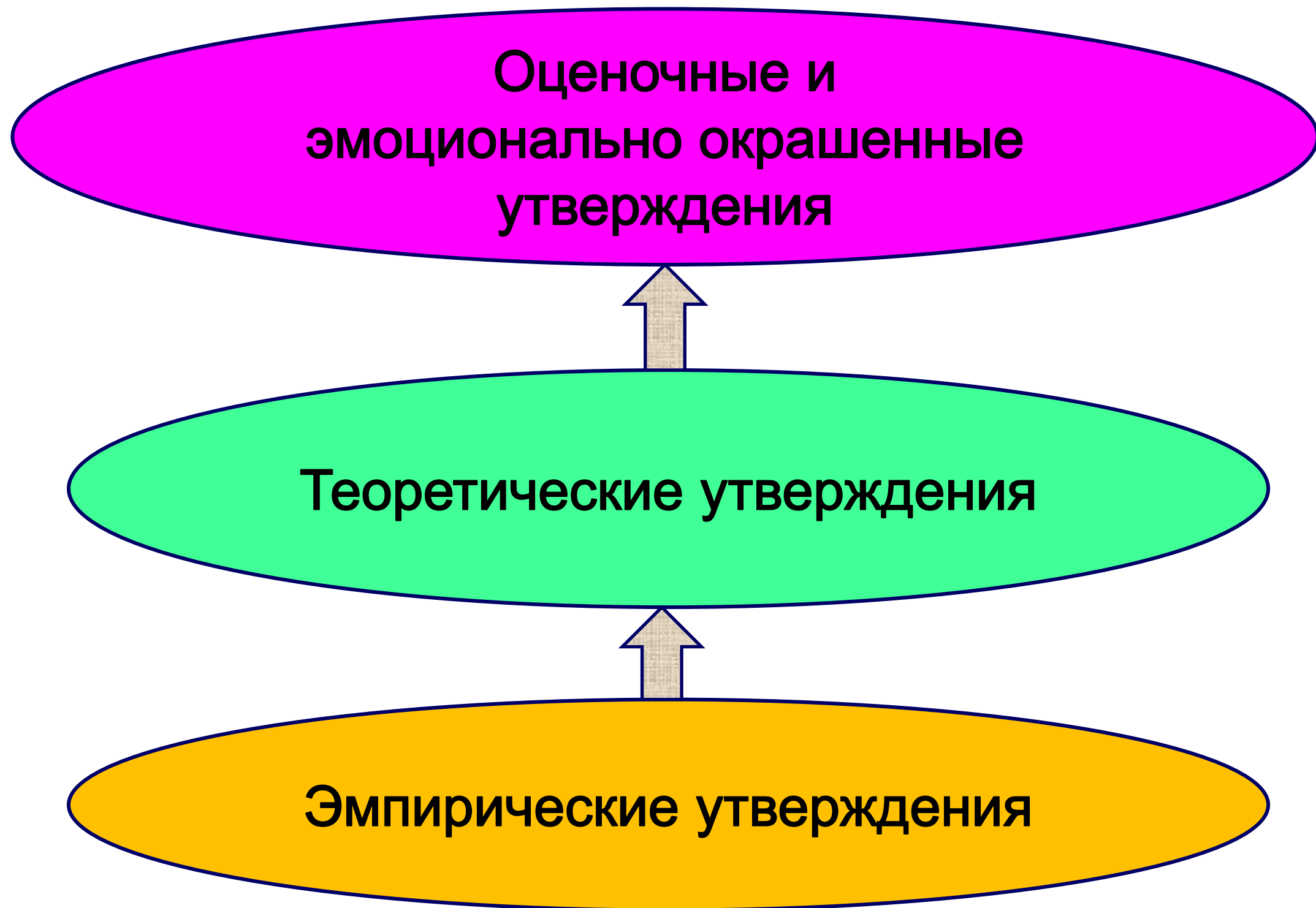




# Уровни представления знаний

- Представление и извлечение знаний различной степени общности
- Использование понятий различной степени общности:
  - ☐ Эмпирические понятия
  - ☐ Теоретические понятия
- Пополнение сигнатуры «верхнеуровневыми» понятиями

# Уровни представления знаний





# Эмпирические и теоретические термины

- Р.Карнап: «Философские основания физики»
- Эмпирические термины – понятия, которые являются непосредственно наблюдаемыми: цвет, температура, давление, объем
- Смысл теоретических терминов является абстрактным и гипотетическим: энтропия, кинетическая энергия, масса электрона

# Эмпирические и теоретические термины

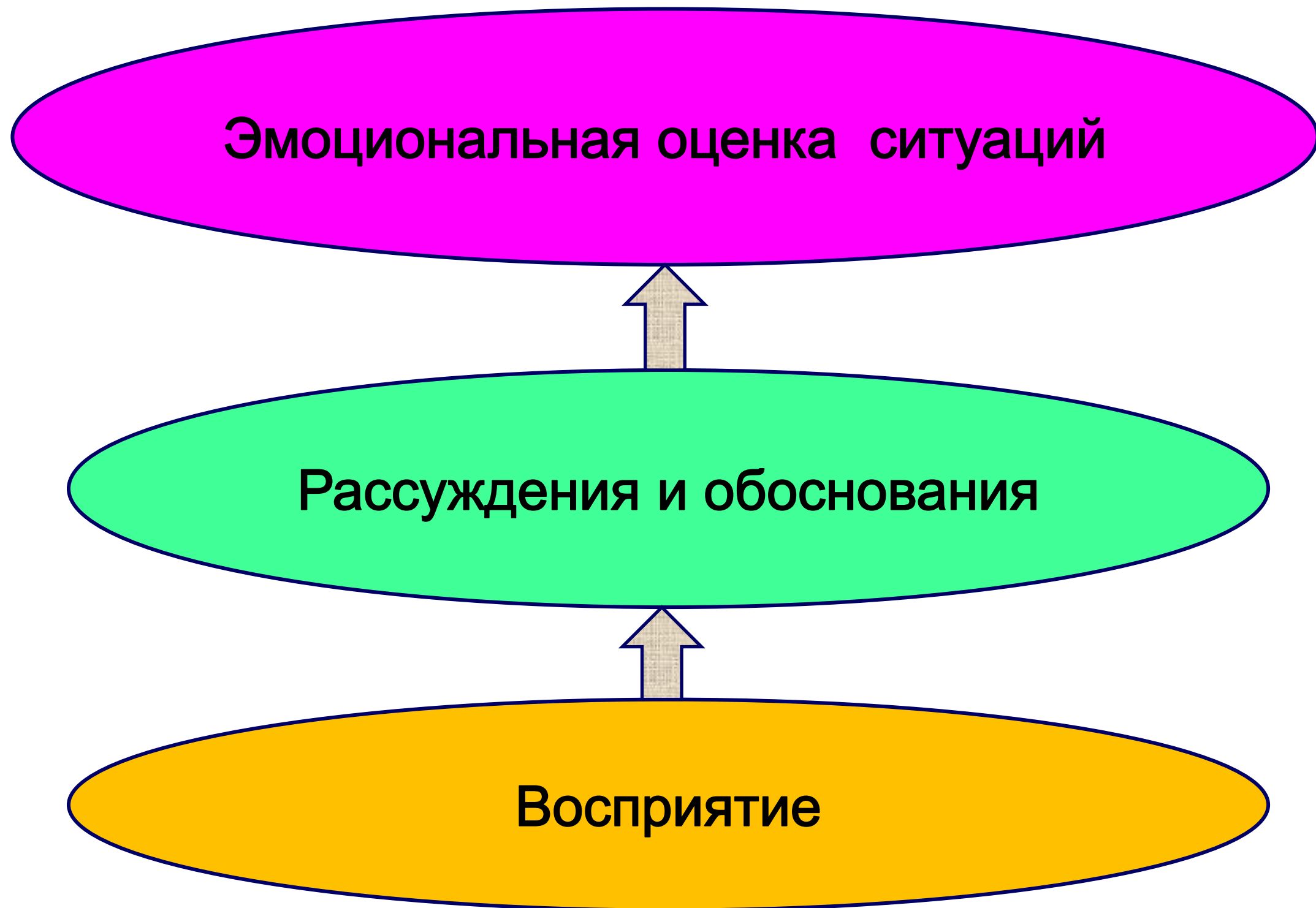
- Свойство быть эмпирическим или теоретическим является относительным: один набор терминов является эмпирическим относительно другого; в этом случае второй набор терминов будет теоретическим относительно первого – понятия и метапонятия
- Теоретические термины нужны для того, чтобы с их помощью формулировать теоретические законы, из которых, в свою очередь, можно получать эмпирические следствия



# Взаимосвязь уровней общности понятий

- Определение понятий одного уровня через понятия другого уровня
- Принцип редукции **vs.** принцип несводимости понятий верхнего уровня к понятиям нижних уровней:
  - несводимость биологического уровня к физическому
  - несводимость сознания к реакциям нейронов

# Естественный интеллект





# Реализация интеллектуальной деятельности

Функция эмоциональной оценки предложений и ситуаций



Фрагменты атомарных диаграмм,  $\Delta_0$ -предложения



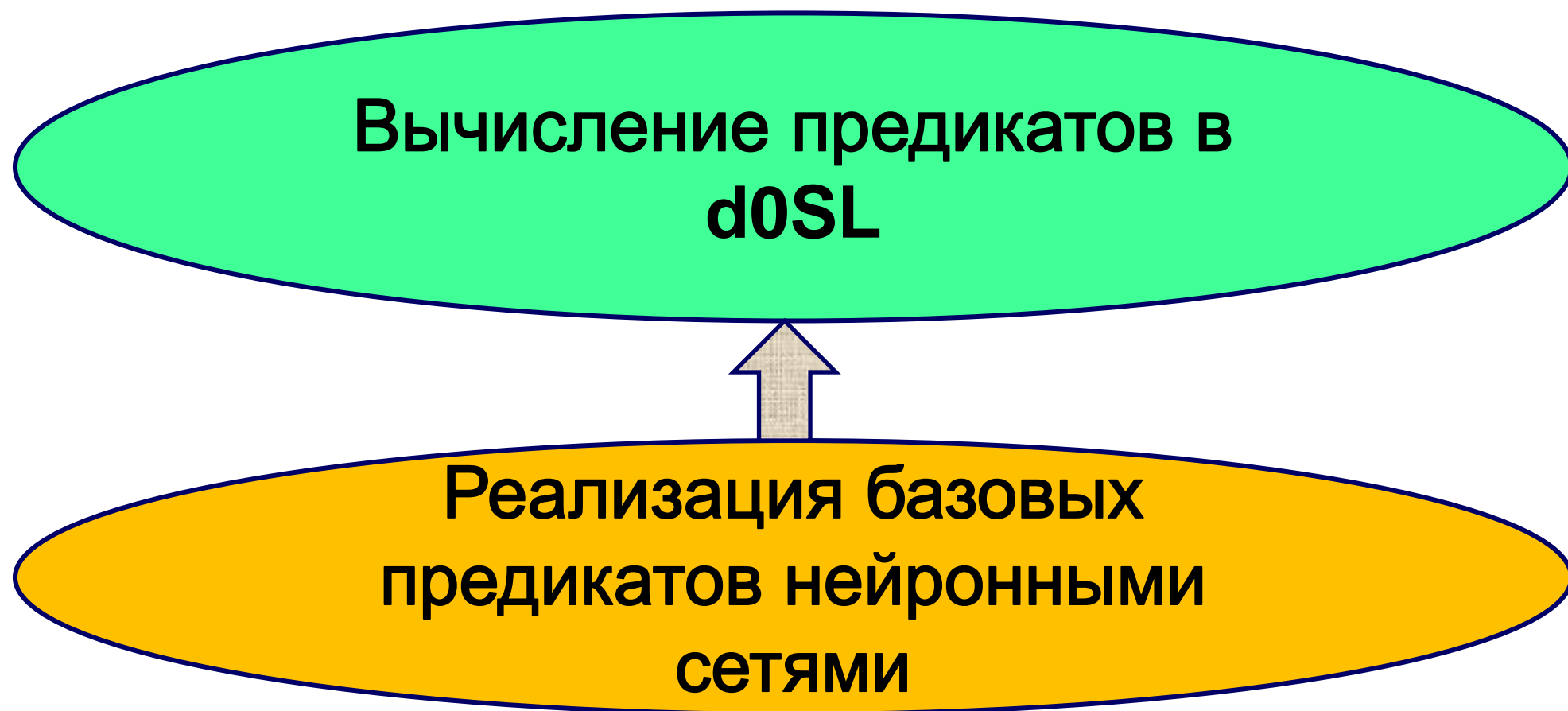
Машинное обучение:  
нейронные сети, ...

# Вычисление предикатов

- Формальное описание знаний – на языке логики предикатов: язык  $\Delta_0$ -формул
- Предикаты  $P(x_1, \dots, x_n)$  делятся на
  - базовые и
  - формульно определяемые через уже заданные
- Значения истинности базовых предикатов задаются извне модели, как оракулы
- Формульно определяемые предикаты описываются средствами  $d_0SL$



# Вычисление предикатов



# Вычисление предикатов

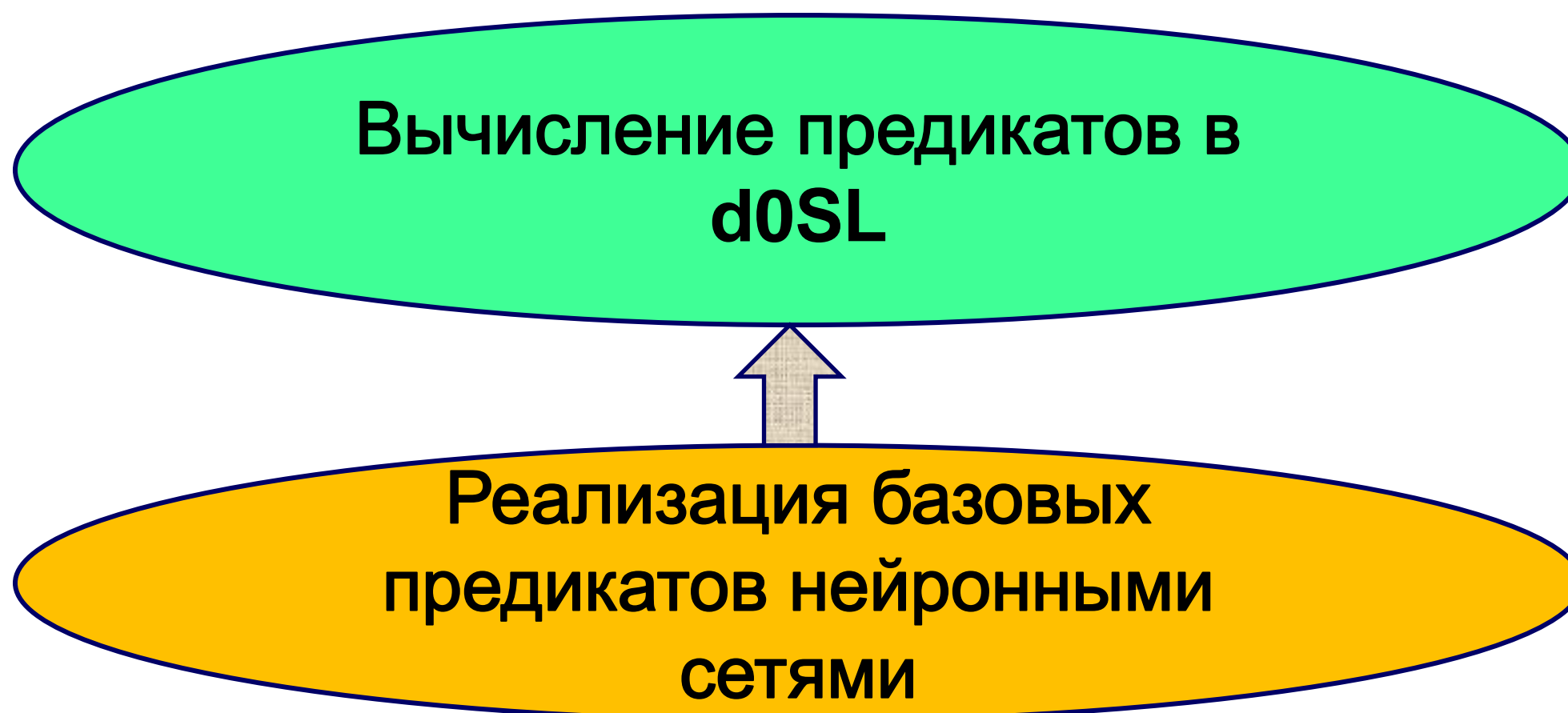
- Формальное описание знаний – на языке логики предикатов: язык  $\Delta_0$ -формул
- Предикаты  $P(x_1, \dots, x_n)$  делятся на
  - базовые и
  - формульно определяемые через уже заданные
- Значения истинности базовых предикатов задаются извне модели, как оракулы
- Формульно определяемые предикаты описываются средствами  $d_0SL$



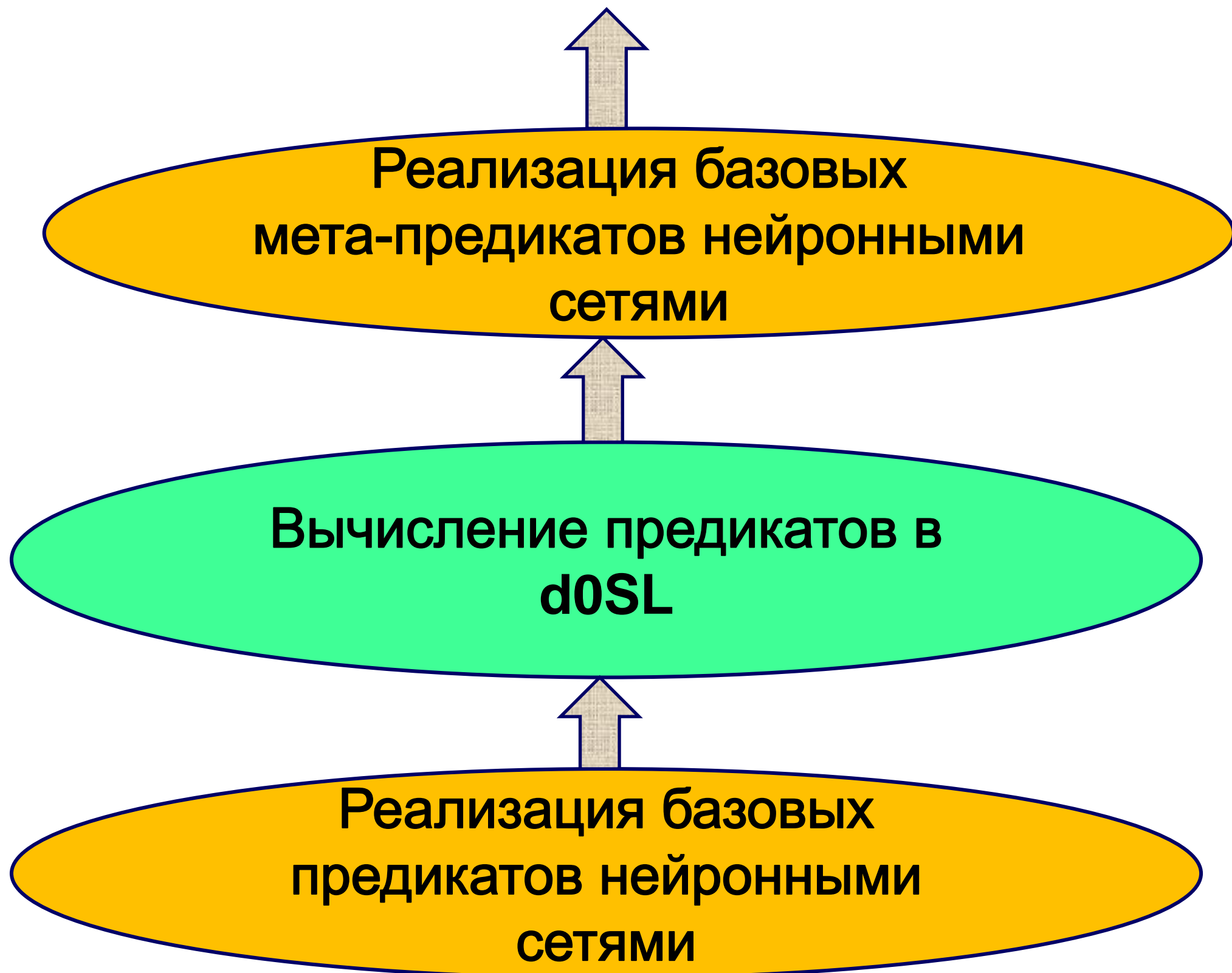
# «Слоёный пирог»

- Формальное описание знаний – на языке логики предикатов: язык  $\Delta_0$ -формул
- Предикаты  $P(x_1, \dots, x_n)$  делятся на
  - базовые и
  - формульно определяемые через уже заданные
- Значения истинности базовых предикатов задаются извне модели, как оракулы
- Формульно определяемые предикаты описываются средствами d0SL
- Далее могут вводиться предикаты мета-уровня, и их значения снова вычисляются

# «Слоёный пирог»



# «Слоёный пирог»





# Распределённый реестр и семантические контракты

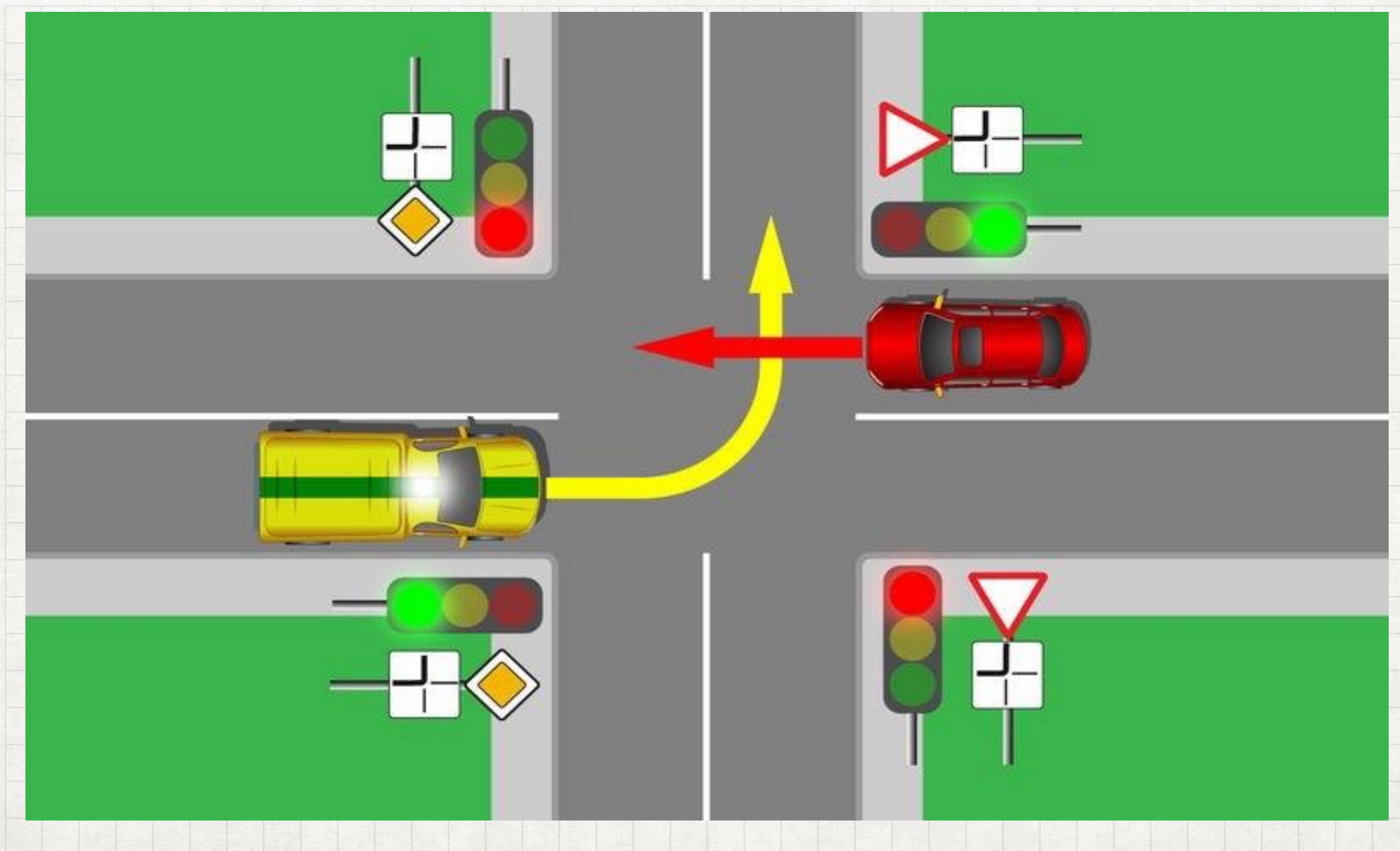
Выполнение семантических  
контрактов с оценкой успешности  
действий

Семантические контракты на  
d0SL

Реализация базовых  
предикатов нейронными  
сетями

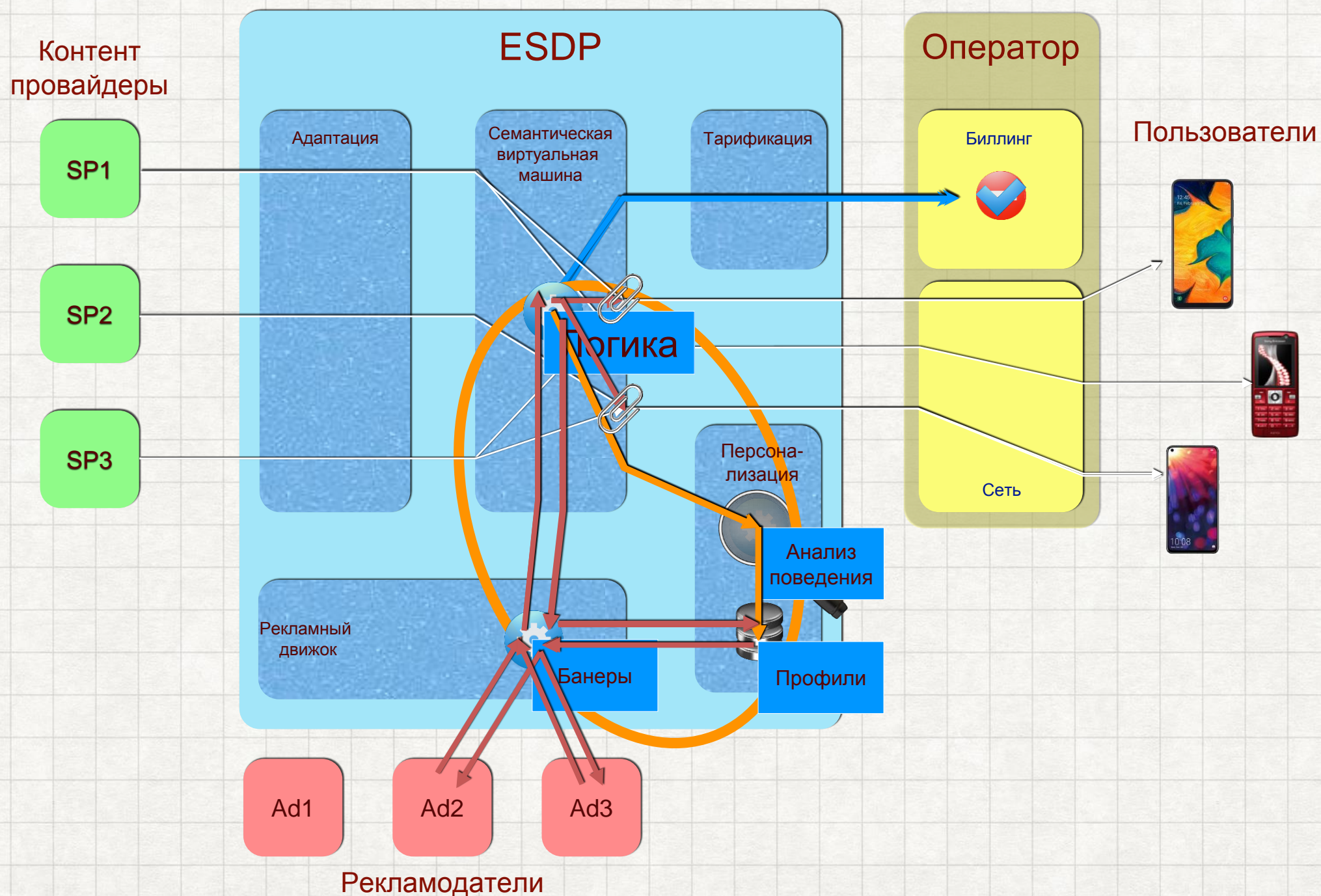
# Логика (ИИ 2.0) vs алгоритмы

ПРАВИЛА ДОРОЖНОГО ДВИЖЕНИЯ



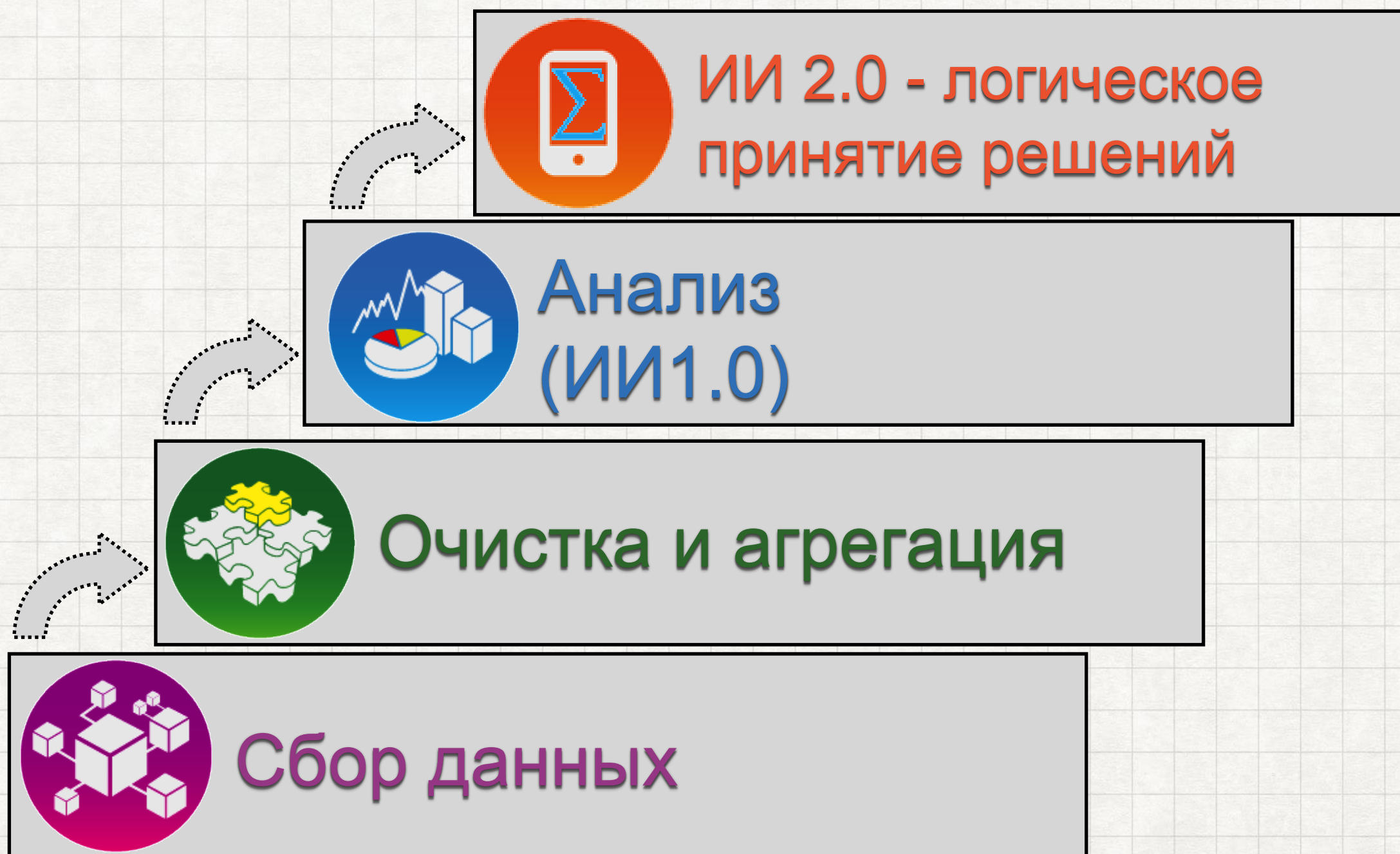


# ИИ 2.0: предбиллинг VAS у мобильного оператора





# ИИ и автономное принятие решений





# Индустрия 4.0: ВЫЗОВЫ

## Кадры



## Безопасность



## Управляемость





# Кадры: кто все это будет программировать?



18 МЛРД IoT  
УСТРОЙСТВ К 2022  
ГОДУ



## Количество программистов в России

350 000

Имеющееся количество программистов

1 000 000

Требуемое количество  
программистов (по  
оценке Минкомсвязи РФ)



Каждый 400-й в России — программист



© ООО «Доктор Веб», 2003 — 2016



# Безопасность



BOEING 737 MAX 8

- **Зависимость** человечества от ИТ
- Как обеспечивать **качество** и **надежность** инфраструктуры?
- **Нейронные сети и алгоритмы:** черный ящик и проблема переобучения

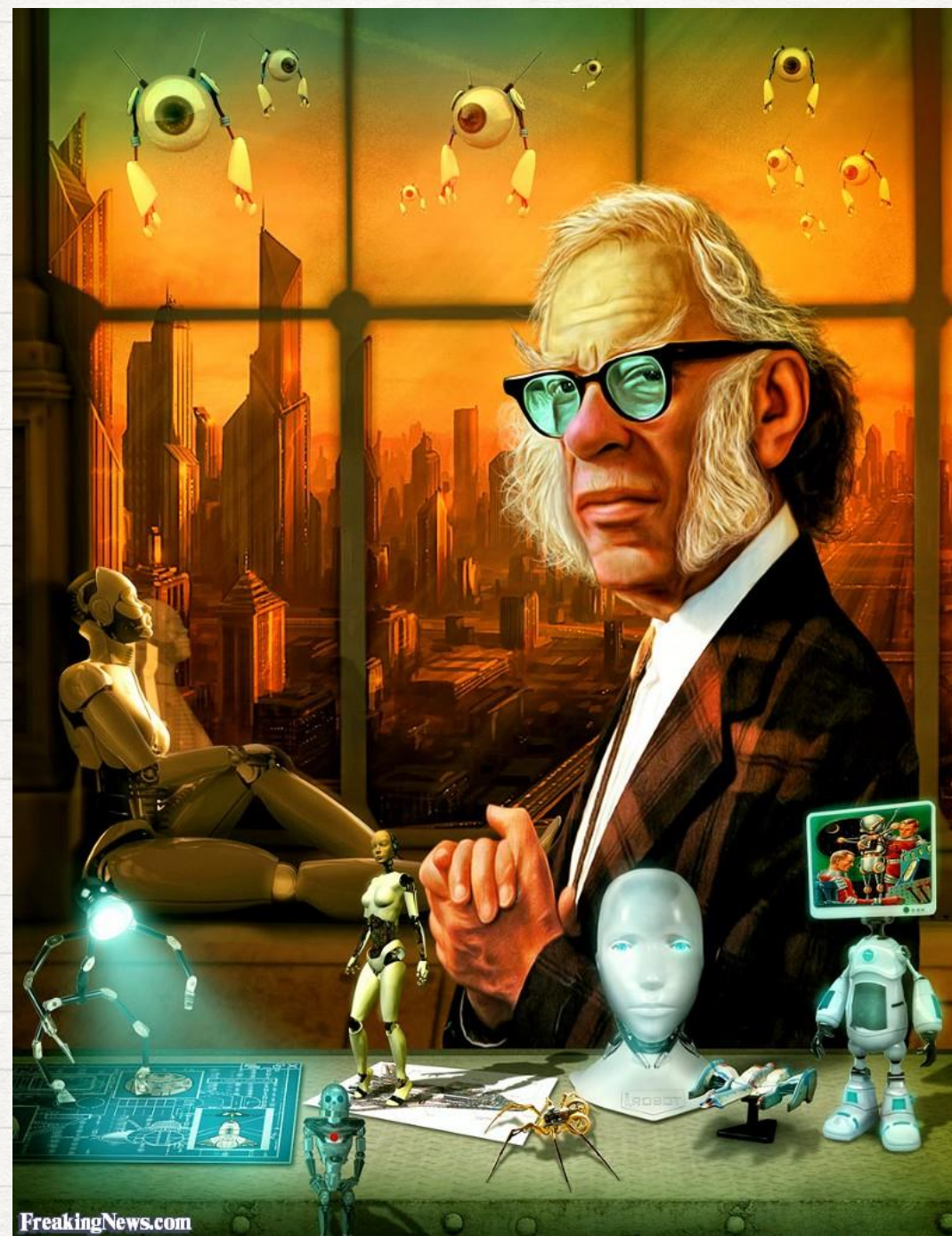


# УПРАВЛЯЕМОСТЬ: ответственный ИИ

- **Законы  
Робототехники**

ПРОЕКТ ДИРЕКТИВЫ ЕС

<https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai>





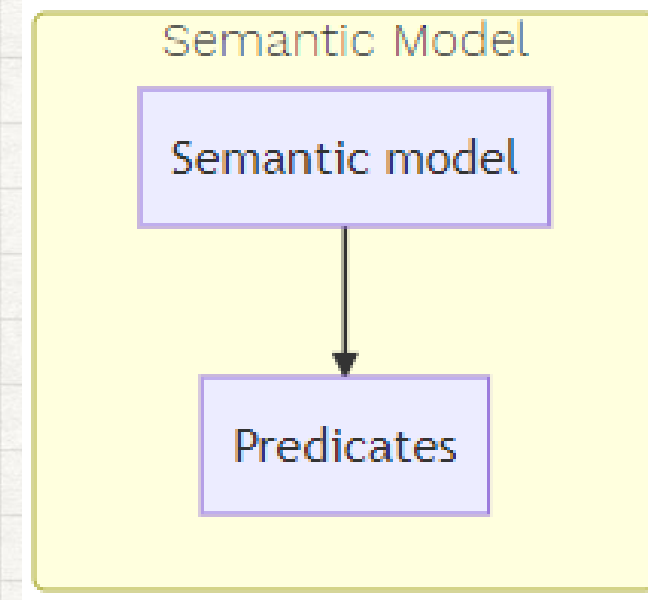
# d0s1 : СЕМАНТИЧЕСКАЯ И ДОМЕННАЯ МОДЕЛЬ

```
model ChessAII def
  use ChessDSL from org.d0s1.examples.chess.ChessDSL

  def start() means
    check all
      ChessDSL.start()
    end
  end def

  def check board(board : ChessBoard) means
    check all
      var queens = ChessDSL.get queens(board)
      for all q1, q2 in queens
        not ChessDSL.on one line(q1, q2) and not ChessDSL.on one diagonal(q1, q2)
      end
    end def
  end def

end ChessAII
```

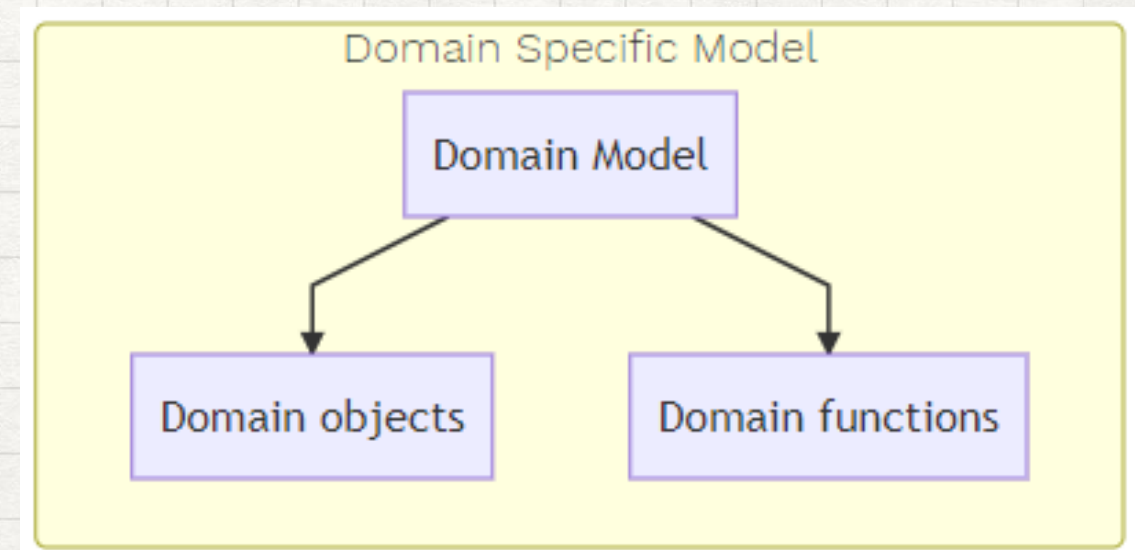


```
domain specific model Math def
  # Math library

  # Square root
  fun sqrt(value : numeric) returns numeric

  # Sinus and Cosinus
  fun sin(value : numeric) returns numeric
  fun cos(value : numeric) returns numeric

  # Power function
  fun pow(value : numeric, power : numeric) returns numeric
end Math
```





# Элементы языка d0s1

ДОСТАТОЧНО ДЛЯ ШИРОКОГО КРУГА ЗАДАЧ

Определяемые предикаты	Логические функции
Доменные функции	Функции доменной модели
Логические операции	<b>and</b> , <b>or</b> , <b>not</b> , <b>check all</b> (n-арная <b>and</b> )
Импликация <b>⇒</b>	<b>If then</b>
Квантор всеобщности <b>∀</b>	<b>for all</b> по ограниченным спискам
Переменные	<b>var</b> не изменяемы после инициализации
Операторы сравнения	<b>&lt;</b> , <b>&lt;=</b> , <b>==</b> , <b>&gt;=</b> , <b>&gt;</b>
типы	<b>Строки</b> , <b>числа</b> , <b>логические константы</b> ( <b>true</b> , <b>false</b> , <b>none</b> ) , <b>списки</b> , <b>типы доменной модели</b>
использование других семантических и доменных моделей	<b>use</b>



# Примеры d0s1

## Predicate example

```
def check board(board : ChessBoard) means
  check all
    var queens = ChessDSL.get queens(board)
    for all q1, q2 in queens
      not ChessDSL.on one line(q1, q2) and not ChessDSL.on one diagonal(q1, q2)
    end
  end def
```

```
if AutodromeDSL.road sign(car) then
  AutodromeDSL.road sign allows move(car)
end
```

```
var queens = ChessDSL.get queens(board)
for all q1, q2 in queens
  not ChessDSL.on one line(q1, q2) and not ChessDSL.on one diagonal(q1, q2)
```

```
var queens = ChessDSL.get queens(board)
for all q in queens
  not for all qq in queens
    not ChessDSL.on near line(q, qq)
  end
for all q1, q2 in queens
  not ChessDSL.on one line(q1, q2) and not Ch
```

## Variable example

```
# Testing for sin & cos
def test() means
  check all
    var angle = 35
    var cosinus = Math.cos(angle)
    var sinus = Math.sin(angle)
    var sum of squares = Math.pow(cosinus, 2) + Math.pow(sinus, 2)

    # considering the features of the library java.lang.Math
    # and inaccurate calculations when converting degrees to radians
    sum of squares <= 1
    sum of squares >= 0.999999
  end
end def
```



# СПАСИБО!

В.Ш. Гумиров, [vit@eyeline.mobi](mailto:vit@eyeline.mobi)

П.Ю. Матюков, [peter@eyeline.mobi](mailto:peter@eyeline.mobi)

Д.Е. Пальчунов, [palch@math.nsc.ru](mailto:palch@math.nsc.ru)