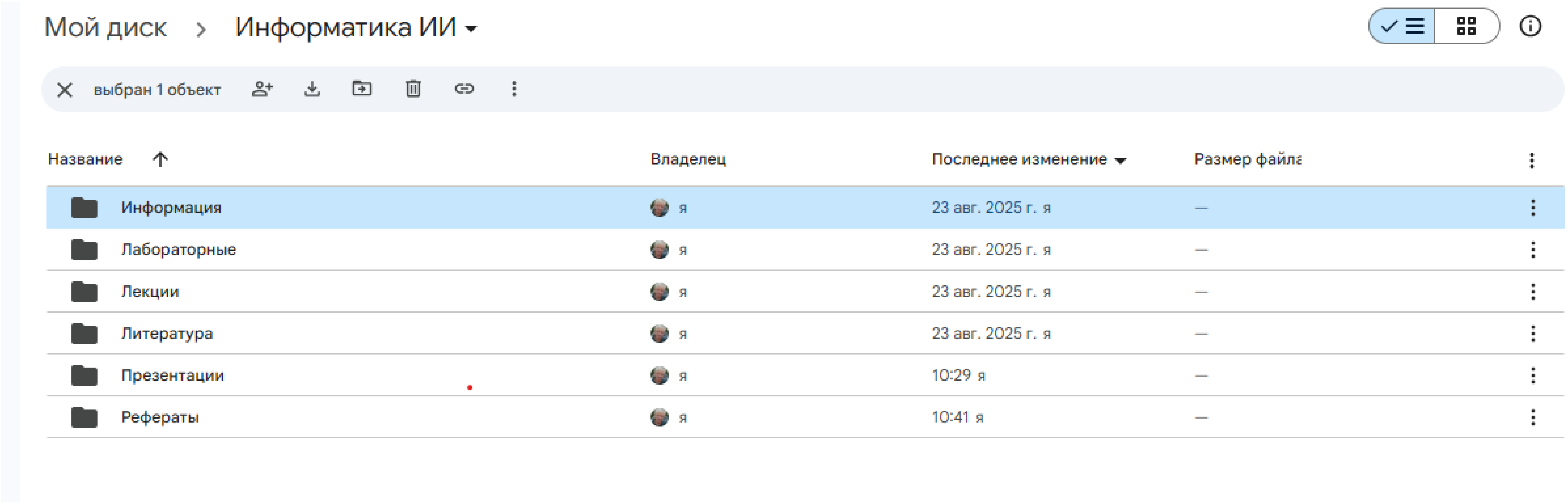
Комментарий к лекции 1

ВВЕДЕНИЕ

1. **О курсе**
2. **История ИИ + приложения**
3. **Методология**

https://drive.google.com/drive/folders/1VeKz0oNYI1SfEBH7sDGD2V8FyQDXXb7O?usp=sharing



Obraztsov@bsu.by

Название	Владелец	Последнее изменение	Размер файла
ИИС 1 группа - оценки за рефераты_тесты_и выполнение с...	я	08:47 я	197 КБ
ИИС 2 группа - оценки за рефераты_тесты_и выполнение с...	я	08:48 я	201 КБ
ИИС 3 группа - оценки за рефераты_тесты_и выполнение с...	я	08:48 я	210 КБ
ИИС 4 группа - оценки за рефераты_тесты_и выполнение с...	я	08:49 я	200 КБ

Название	Владелец	Последнее изменение	Размер файла
Методическая разработка к лабораторным по ИИ.pdf	я	22 янв. 2025 г. я	520 КБ

Название	Владелец	Последнее изменение	Размер файла
КИБЕРНЕТИЧЕСКИЙ ПОДХОД К ЭВОЛЮЦИИ.pdf	я	1 янв. 1980 г. я	2,1 МБ
Р.Пенроуз - Новый ум короля. О компьютерах, мышлении и...	я	1 янв. 1980 г. я	4,3 МБ
Book - Хокинс Дж.,Блейксли С. Об интеллекте.(2007).djvu	я	1 янв. 1980 г. я	1,7 МБ
book_ai_luger.djvu	я	1 янв. 1980 г. я	15,2 МБ
Hofshtadter D.R. (_Hofstadter_) Gyodel_, Esher, Bah.. E_ta bes...	я	1 янв. 1980 г. я	11,5 МБ
newmind.djvu	я	1 янв. 1980 г. я	2,6 МБ
penrose_large_the_small_and_the_human_mind.pdf	я	1 янв. 1980 г. я	5,5 МБ
Shapiro S.I. Ot algoritmov k suzhdeniyam (1973)(ru)(L)(T)(145...	я	1 янв. 1980 г. я	3,4 МБ

График и характер проведения лабораторных работ

Тематика	№ лабораторной	Смысл работы	№ занятия	Характер работы	Ссылка на описание работы
Логический вывод*	-	Ознакомление с задачей, постановка	1.	УСР	Стр.2
	1	Прямой вывод в БЗ продукционного типа	2.	Работа над заданием	Стр. 3-4
			3.		
			4.	Отчет	
	2	Табличный вывод в БЗ продукционного типа	5.	Работа над заданием	Стр. 5-7
			6.		
			7.	Отчет	
	3	Обратный вывод в БЗ продукционного типа	8.	Работа над заданием	Стр. 8
			9.		
			10.	Отчет	
Распознавание образов**	-	Ознакомление с задачей, постановка	11.	УСР	Стр. 9-11
	4	Решение задачи распознавания образов без обучения	12.	Работа над заданием	Стр. 12-13
			13.		
			14.	Отчет	
	5	Решение задачи распознавания образов с обучением	15.	Работа над заданием	Стр. 14-17
			16.		
			17.	Отчет	

Общие замечания по выбору задачи в лабораторных работах.

*– для лабораторных 1-3 необходимо самостоятельно построить базу правил продукционного типа. Требования к базе (задаче) см. стр. 2.
**– для лабораторных 4-5 необходимо выбрать датасет (ссылки на стр. 9-11) для задачи распознавания с обучением. Требования к задаче см. стр. 11.

4 курс, I семестр 2024/2025 учебного года **28.12.2024**

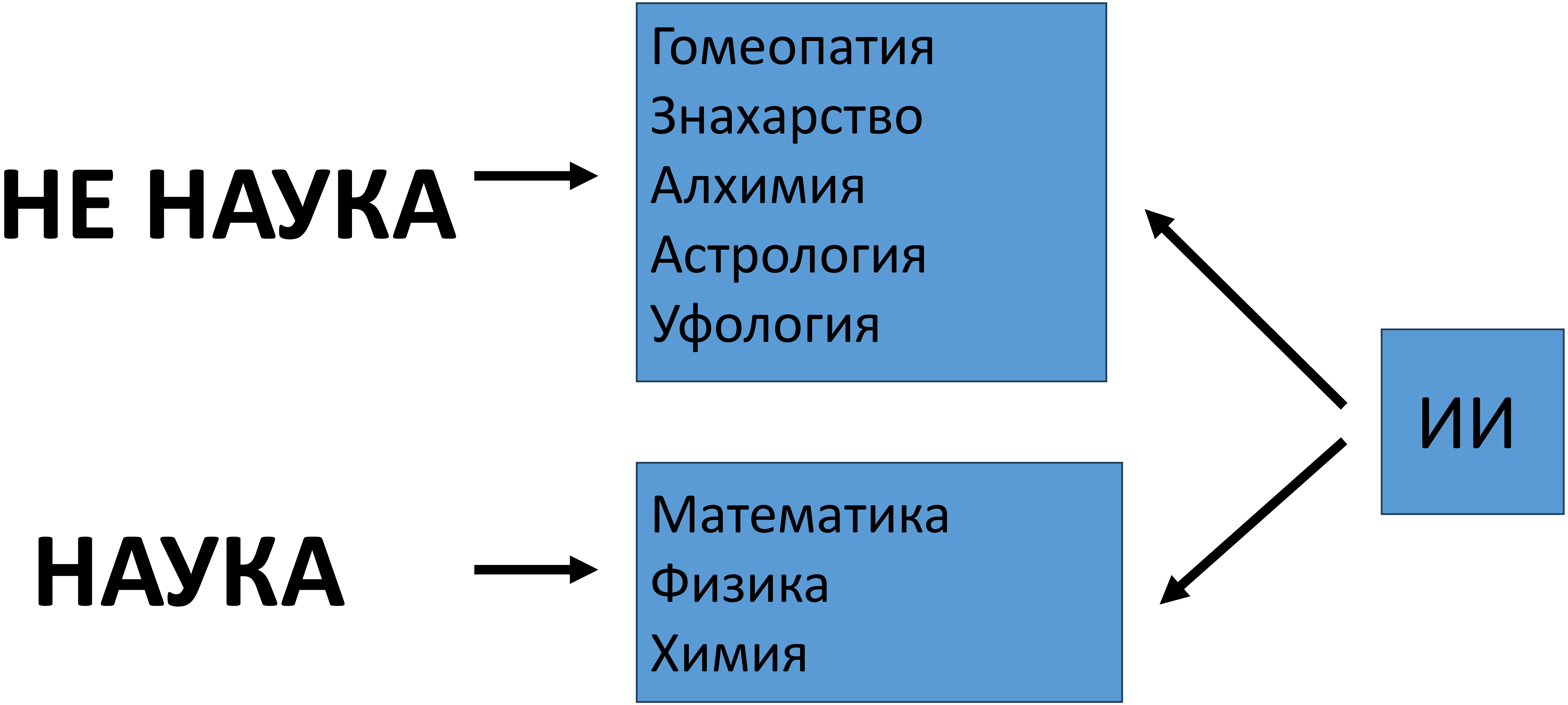
1 группа (МСС)

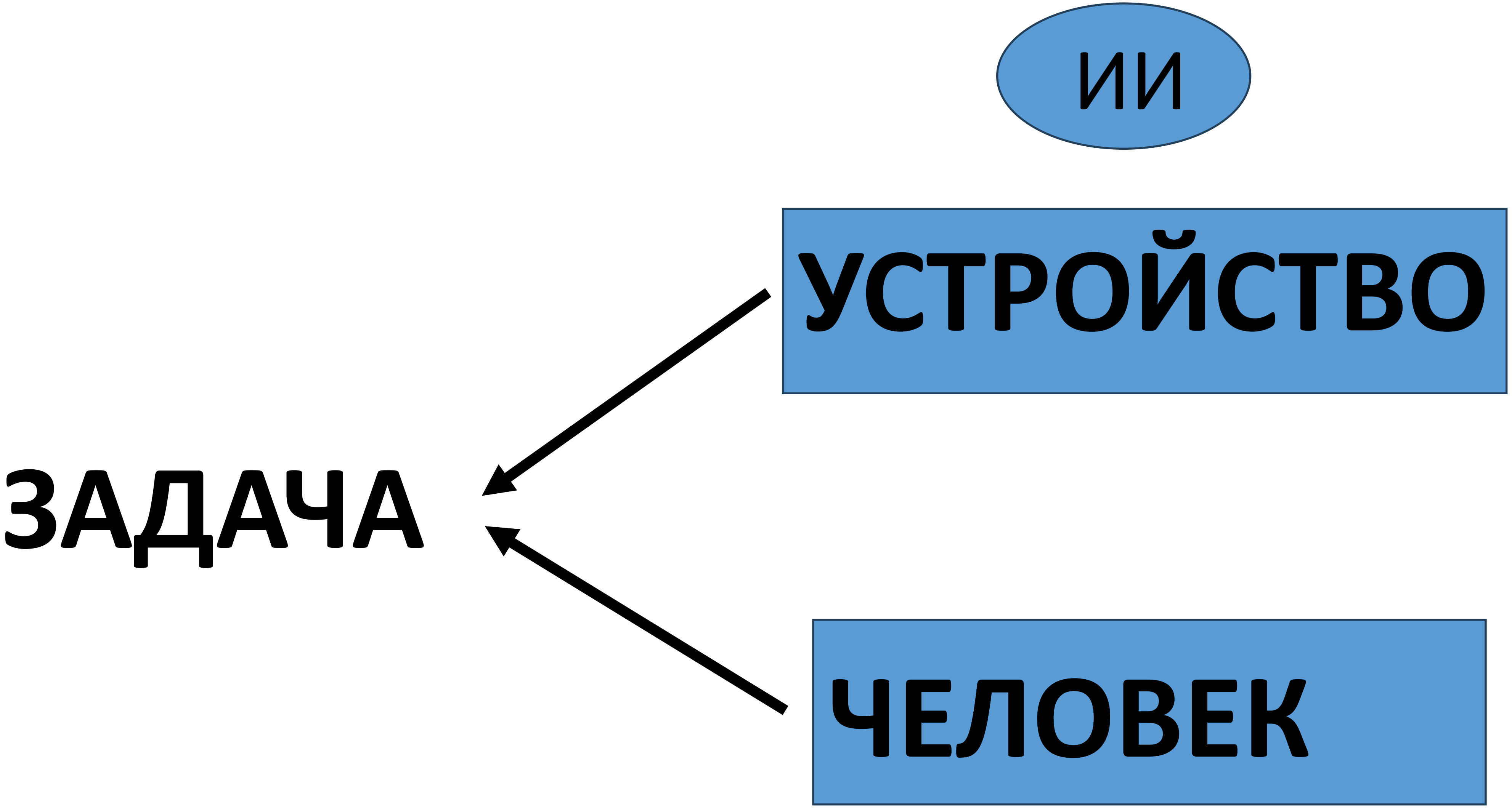
№	Фамилия, Имя, Отчество	Лабораторные	Тесты				Рефераты			пропуски	Сумма баллов
			1	2	3	4	25.10	22.11	20.12		
1.	Евсеев Матвей Олегович	3	1.5+5	5.5	6+11			10	15		57
2.	Кондратьев Евгений Иванович	-27		2.5+2							-22.5
3.	Коноплич Андрей Владимирович	-7						4			-3
4.	Рисник Павел Станиславович	10	2+6	4.5					6		28.5
5.	Рубин Дмитрий Валерьевич	-∞	2+0.5			2+3					-∞
6.	Рудович Дмитрий Игоревич	-28		1.5				3	10		-17.5
7.	Русаков Марк Андреевич	-33	1								-33
8.	Савков Евгений Анатольевич	-25						4			-25
9.	Саврицкий Игорь Александрович	-1	2+4		5	7		4			21
10.	Сивакова Кристина Денисовна	-20		1.5				6			-16.5
11.	Синькевич Анна Николаевна	-30									-30
12.	Стефанович Артём Витальевич	-12	1+3		9	6					7
13.	Такун Матвей Русланович	-9	1.5+5.5	+3.5		7					8.5
14.	Тилигузов Егор Александрович	-∞									-∞
15.	Титович Станислав Петрович	-33									-33
16.	Чикун Ксения Сергеевна	-2		3+1.5	7	5+12		4	12		42.5
17.	Шикунов Владислав Денисович	-8						5			-3
18.	Щербач Анастасия Константиновна	-15									-15
19.	Яцынович Максим Леонидович	-11						4			-7

Примечания:

1. Если сумма баллов по тестам не более 15, то необходимо на экзамене решать задачи;
2. Красным шрифтом в тексте выделены фамилии тех, кто не допущен к экзамену (**не сдал лабораторные**);

1. **МНОЖЕСТВО** (определение, способы задания)
2. **ОТНОШЕНИЯ** (определение, эквивалентность)
3. **ОТОБРАЖЕНИЕ** (определение, примеры)
4. **ФУНКЦИЯ** (способы задания, непрерывность, дискретность)
5. **МОДЕЛЬ** (определение, примеры)
6. **МЕТОД** (определение, зачем нужен)
7. **АЛГОРИТМ** (определение, примеры, сложность)
8. **ПРОГРАММА** (определение, языки программирования)
9. **ПРОГРАММНЫЕ ТЕХНОЛОГИИ** (примеры, программные системы)
10. **МЕТОДОЛОГИЯ** (определение, примеры)


НЕ НАУКА
Гомеопатия
Знахарство
Алхимия
Астрология
Уфология
ИИ
НАУКА
Математика
Физика
Химия


ИИ
УСТРОЙСТВО
ЗАДАЧА
ЧЕЛОВЕК

Искусственный интеллект

Можно применить в любой науке

Горячая тема

Широкое поле для исследований и открытий

Высокая оплата в настоящее время

Чего не будет в курсе?

- Подробного разбора технологий ИИ (ChatCPT,…..)
- Разбора кода
- Подробного разбора аспектов работы с данными
- Сложной математики
- Домашних заданий

ИСТОРИЯ

1. Наивный ($50^е$ - $70^е$)

2. Технологический ($80^е$ - $00^е$)

3. Нейросетевой ($00^е$ – наст. время)

РАЗВИТИЕ ИИ НА ПЕРВОМ ЭТАПЕ

1943

Сети бинарных нейронов могут выполнять логические операции

1947

Синаптическая пластичность. Правило Хебба (Hebb)

1955

Кибернетика (Wiener)

1958

Perceptron - вероятностная модель организации и хранения информации в мозге (Rosenblatt)

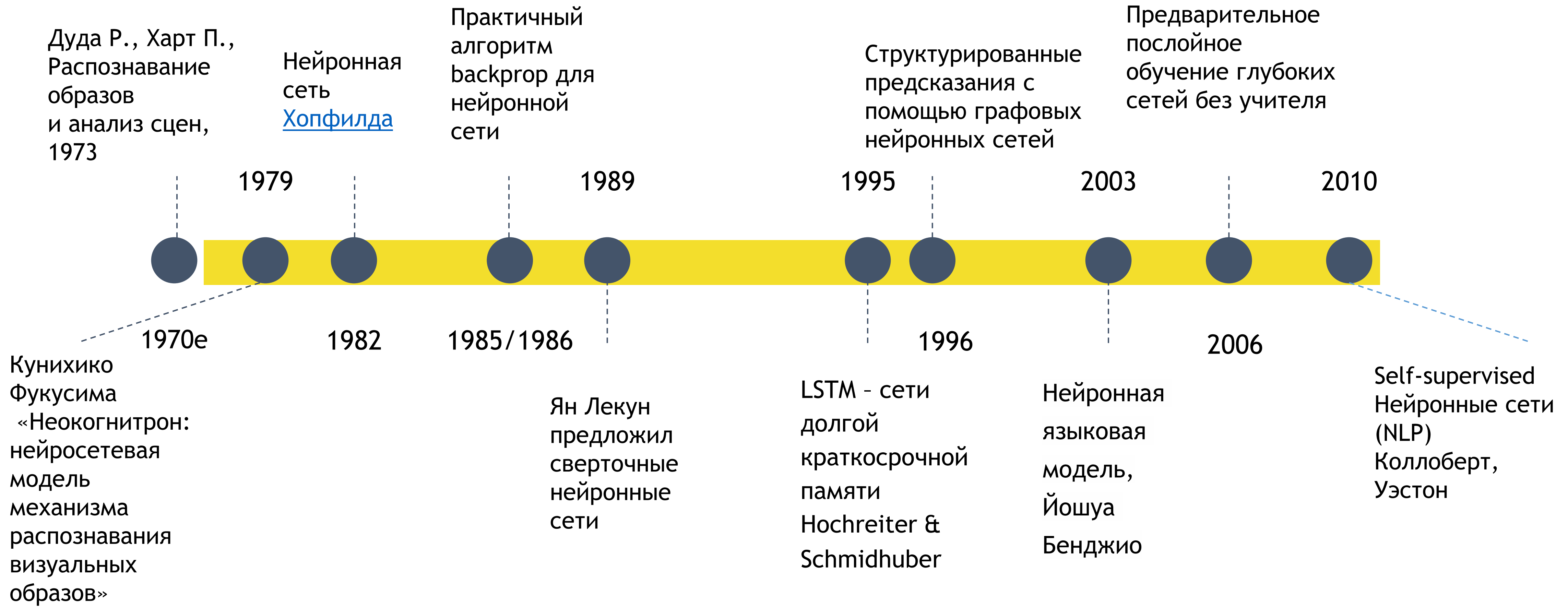
1962

Архитектура визуального кортекса (Hubel & Wiener)

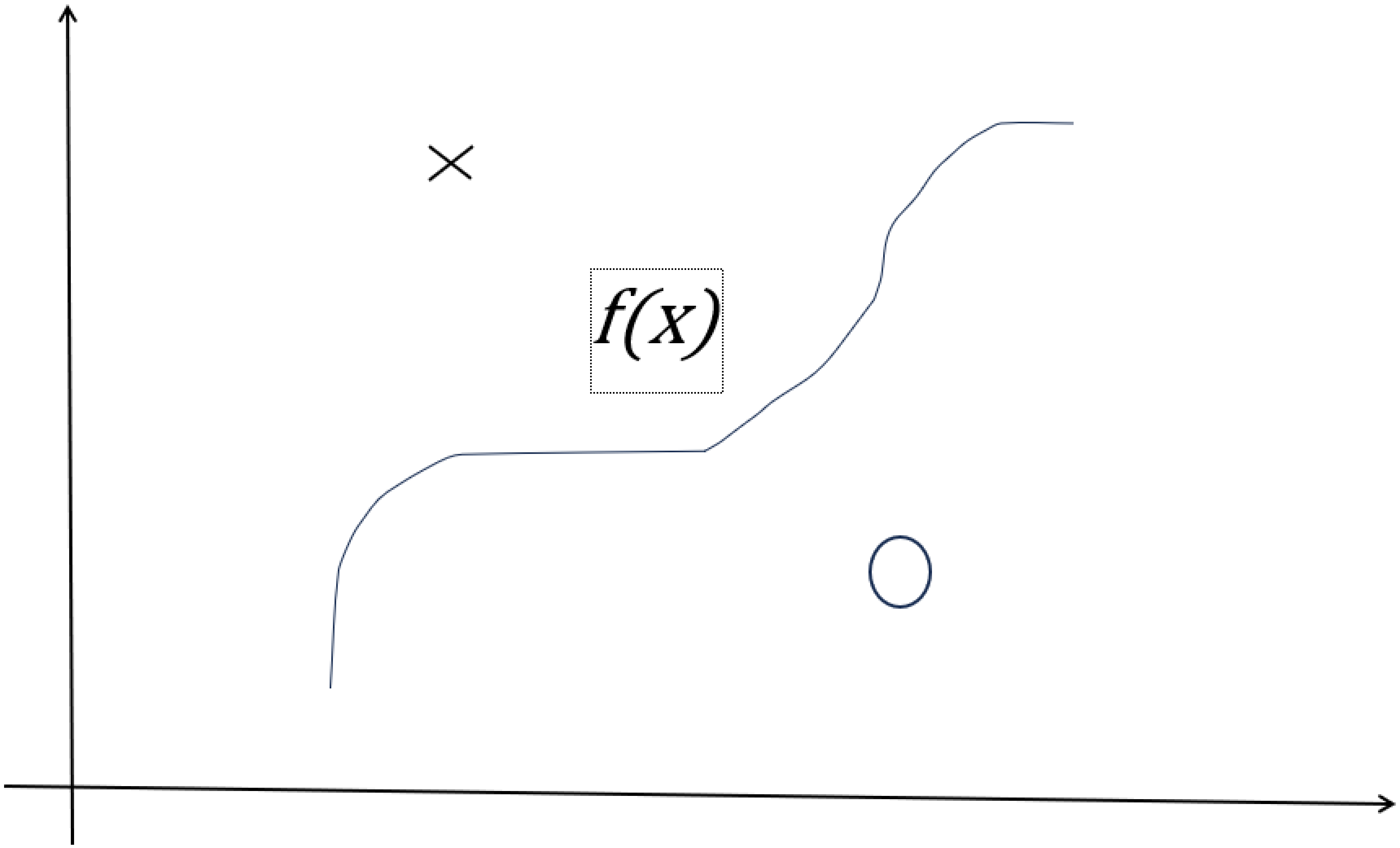
1969

Ограничения Персептрона (Minsky & Papert)

РАЗВИТИЕ ИИ НА ВТОРОМ ЭТАПЕ



ЗАДАЧА



$$\text{ЗАДАЧА} \longrightarrow \text{ИНФОРМАЦИЯ} \longrightarrow \begin{cases} f(x) > 0 \\ f(x) < 0 \end{cases}$$

Модель

Метод решения

Алгоритм

Программа

Результат

ИИ

Задача

Информация (Знания)

Модели ИИ (Логический вывод, Нейронные сети,…)

Программы

⋮

ТЕМА РЕФЕРАТА

Формально определите понятие **ЗАДАЧИ** в математике (и/или информатике)

Комментарий к лекции 2

ЗАДАЧИ + ИИ

1. **Задача в ИИ**
2. **Примеры задач**
3. **Общая схема решения задач**

ИНТЕЛЛЕКТ — это качество психики или общая когнитивная способность, состоящая из знаний, осознавать новое, обучаться и запоминать ситуации на основе опыта, понимать и применять абстрактные концепции, а также использовать знания для управления развитием мира. Он выбирает разные познавательные способности: восприятие, восприятие, память, мышление, воображение, внимание и революция. Интеллект проявляется **в способностях решать проблемы, адаптироваться к новым условиям и разумно мыслить** (Вики).

ИНТЕЛЛЕКТ

ЕСТЕСТВЕННЫЙ
(человек)

ИСКУССТВЕННЫЙ
(устройство)

1. **Ограниченный ИИ (narrow AI)** $\xrightarrow{\text{применение}}$

-доказательство теорем;
-игры;
-распознавание образов;
-принятие решений;
-обработка данных на естественном языке.

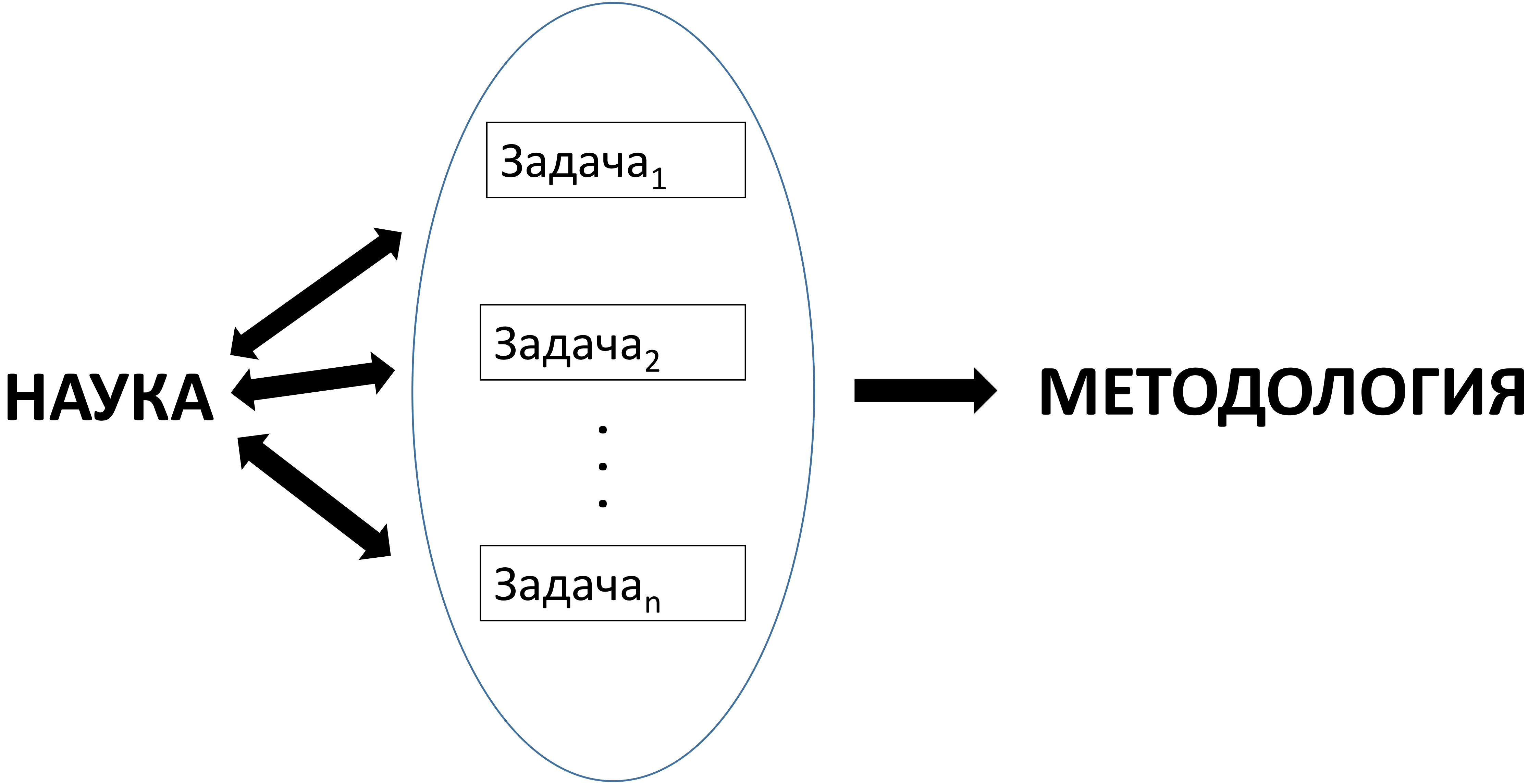
2. **Общий ИИ (general AI)** $\xrightarrow{\text{смысл}}$

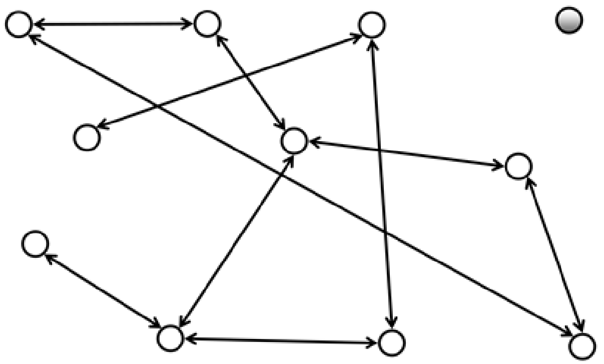
Объединение всех задач NAI

3. **Супер ИИ (super AI)** $\xrightarrow{\text{смысл}}$

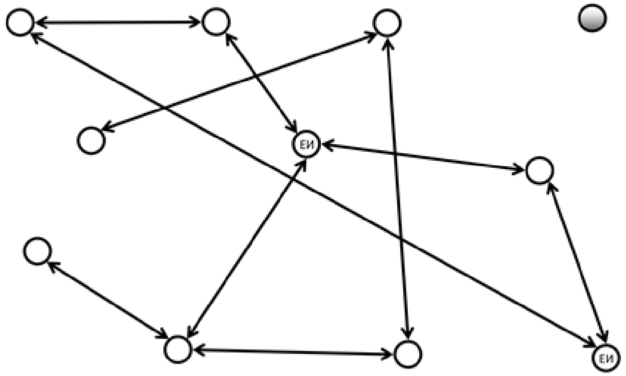
GAI + возможность развития (эволюции)

У Р О В Н И   С О З Н А Н И Я


НАУКА
Задача$_1$
Задача$_2$
Задача$_n$
МЕТОДОЛОГИЯ

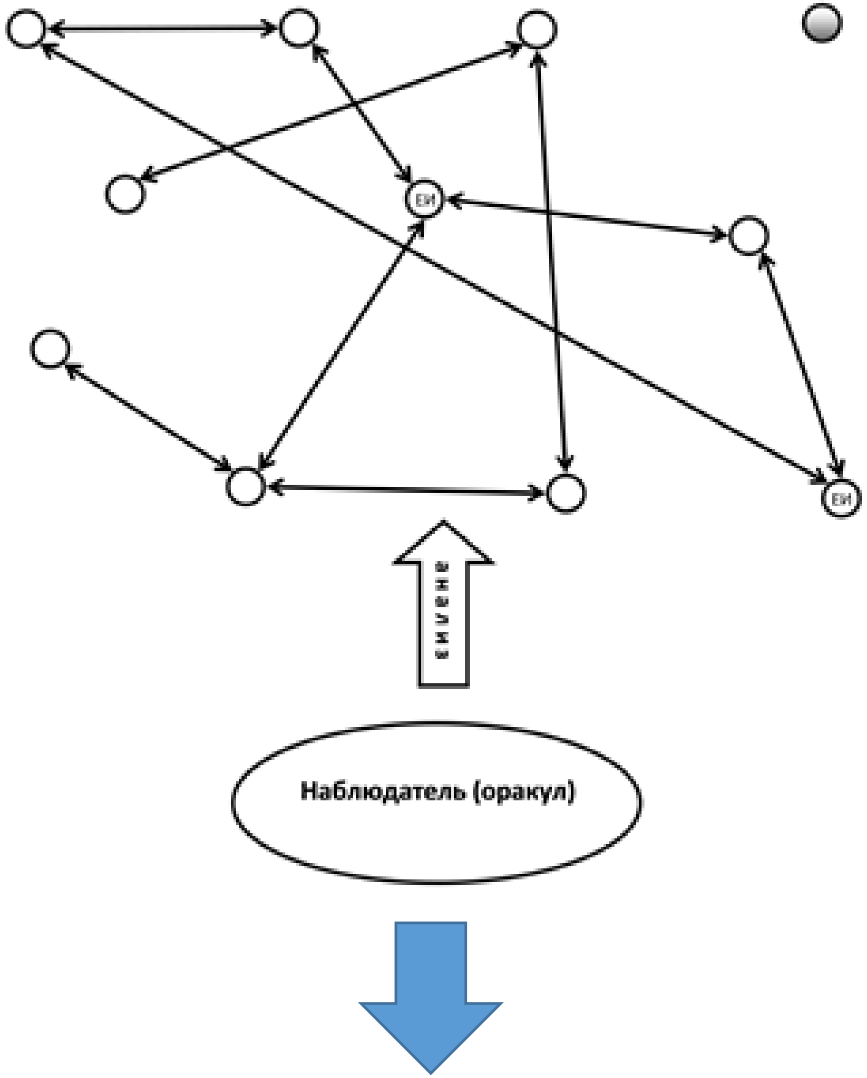


а)

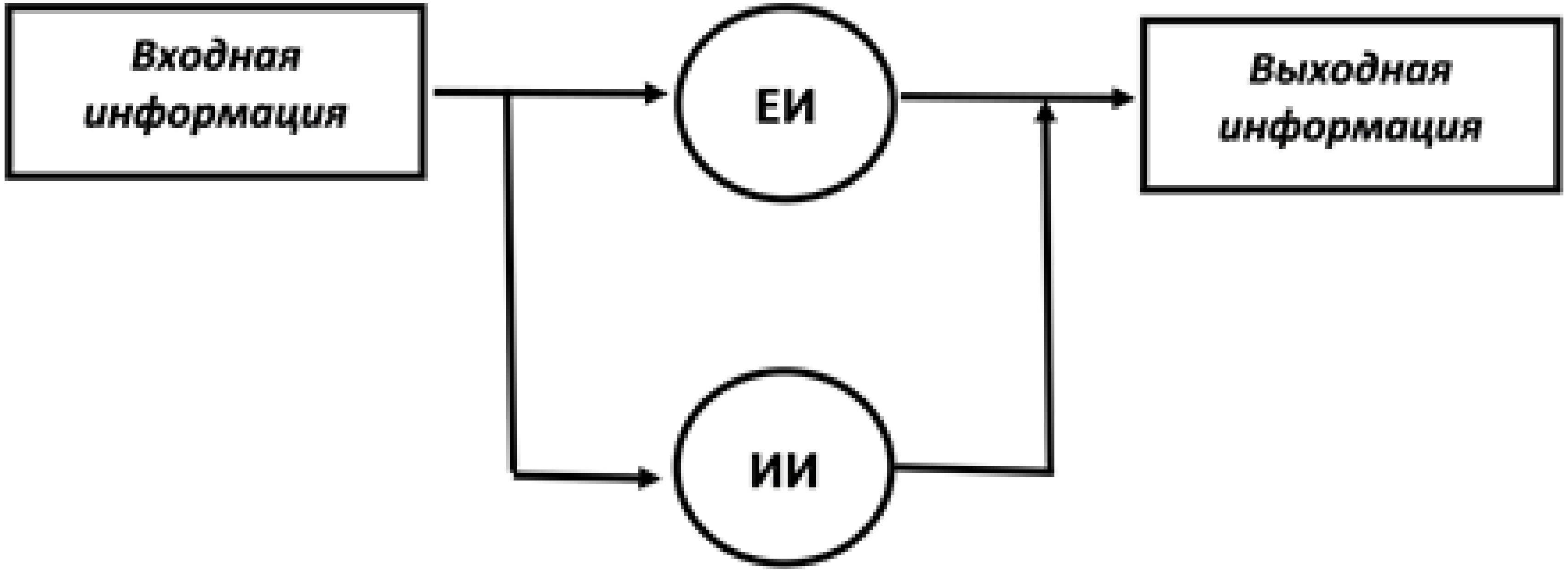


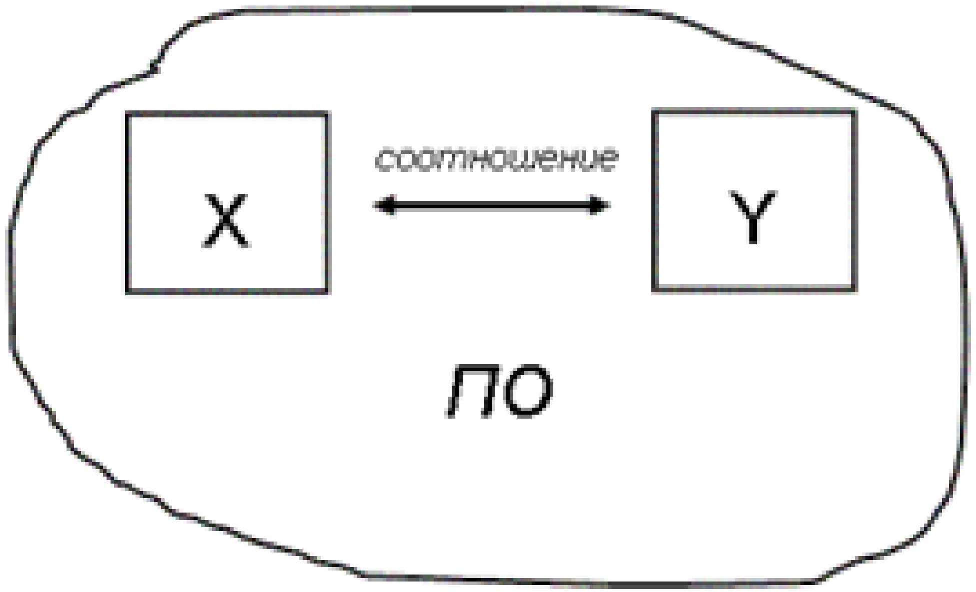
б)

Объектная схема реальной действительности



ТЕСТ ТЬЮРИНГА





X – входная информация

Y – выходная информация

$$X \times Y \qquad Z \subseteq X \times Y$$

Примеры:

1. задача в математике

Дано: некоторая предметная область (ПО). В ПО определено пространство исходной информации и объекты в нем. Целью существования этих объектов являются некоторые, возможно выделенные, объекты в пространстве решений (ответов, результатов).

Требуется: установить связь, соотношение между объектами в пространствах исходной информации и решений.

2. Задача распознавания свойства четности

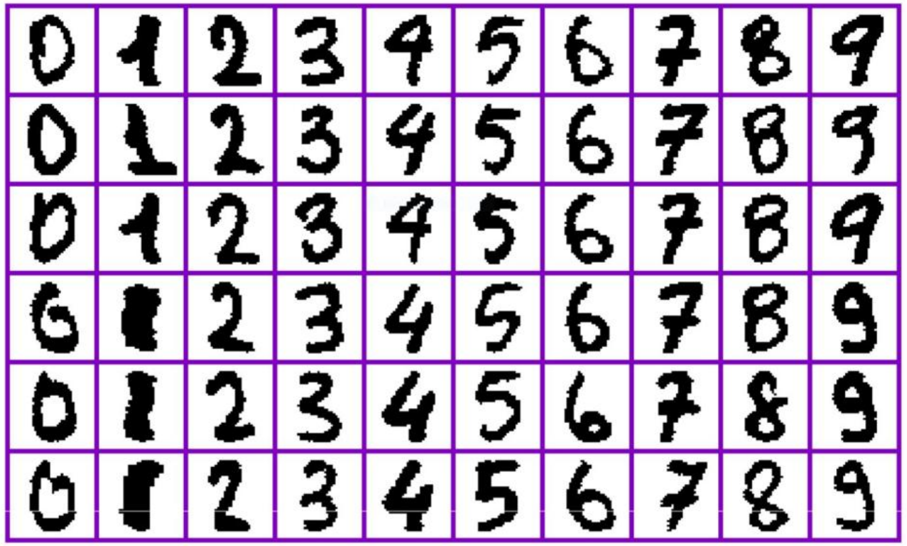
$$\begin{tikzcd}
 & \mathbb{N}_{\text{ч}} \\
\mathbb{N} \arrow[ur] \arrow[dr] & \\
 & \mathbb{N}_{\text{нч}}
\end{tikzcd}$$

$$\begin{cases} \mathbb{N}_{\text{ч}} \cap \mathbb{N}_{\text{нч}} = \varnothing \\ \mathbb{N}_{\text{ч}} \cup \mathbb{N}_{\text{нч}} = \mathbb{N} \end{cases}$$

$$I(\mathbb{N}_{\text{ч}}, \mathbb{N}_{\text{нч}})$$

$$\forall n \in \mathbb{N} \quad A: n \times I(\mathbb{N}_{\text{ч}}, \mathbb{N}_{\text{нч}}) \to \textit{результат}$$

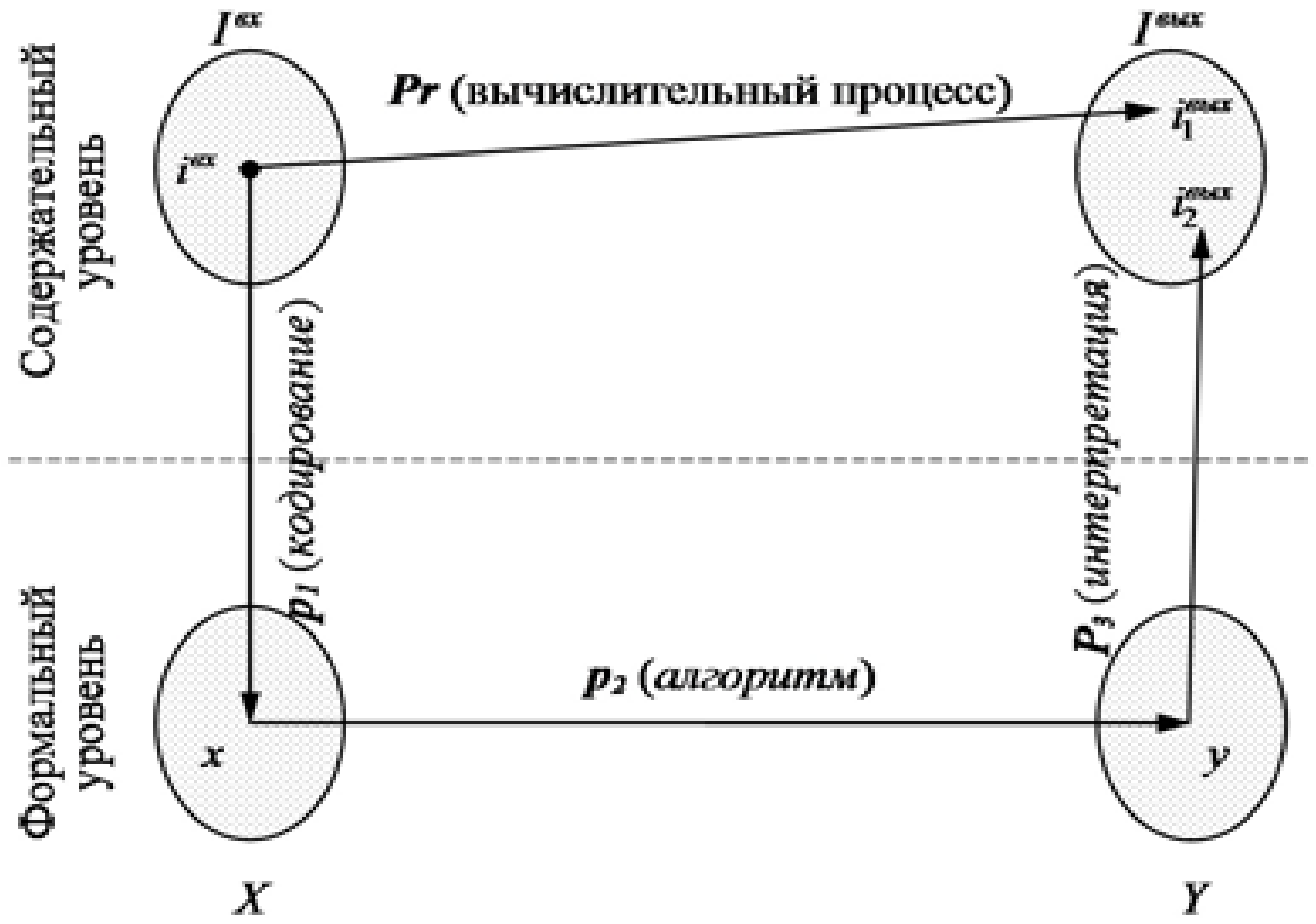
3. Задача распознавания арабских цифр



$$\begin{cases} X_0, \ldots, X_9 \\ X_0 \cup \ldots \cup X_9 \subseteq X \end{cases}$$

$$X^0 \quad I(X^0, X_0, \ldots, X_9)$$

$$\forall x \in X \quad A : x \times I(X^0, X_0, \ldots, X_9) \to p$$



ОБЩАЯ СХЕМА РЕШЕНИЯ ЗАДАЧ

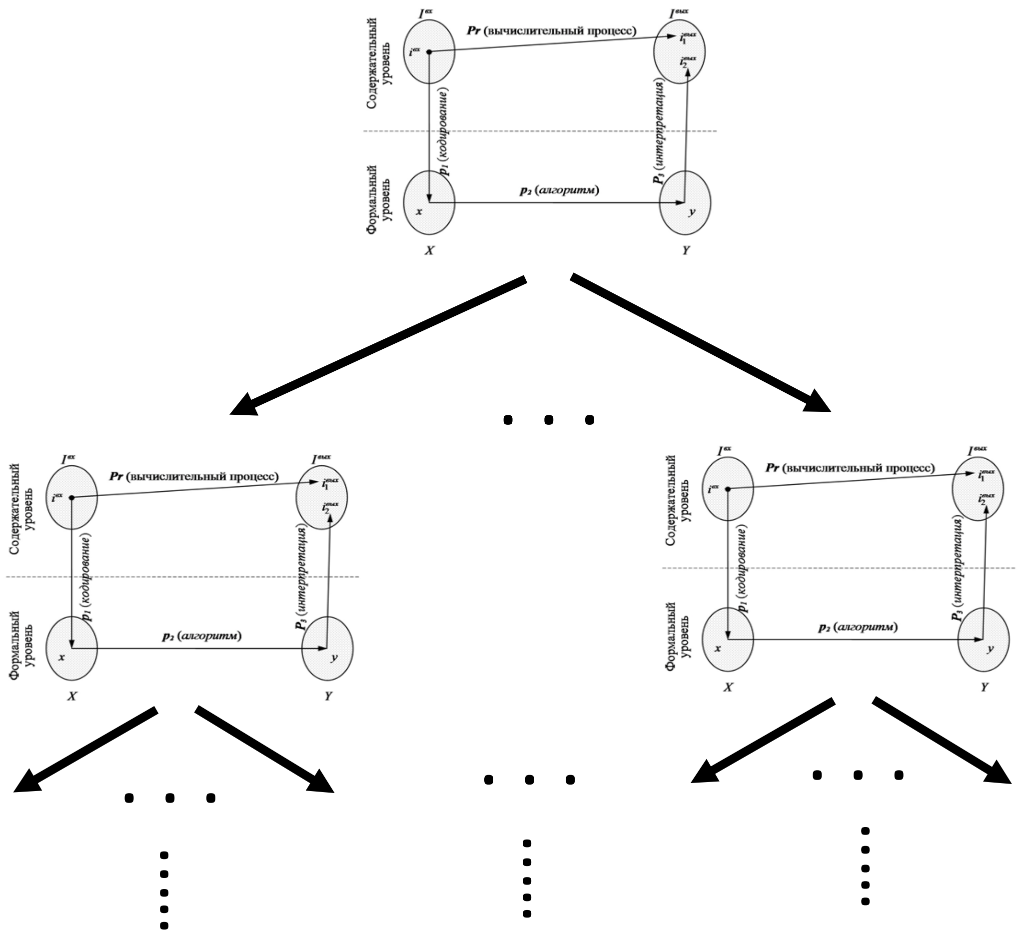
Задача на уровне природы → Объектная схема реальности

1 уровень

2 уровень

⋮

k уровень



Такая схема используется для построения теорий

Процесс решения задачи

<u>ИНФОРМАЦИЯ К РАЗМЫШЛЕНИЮ</u>

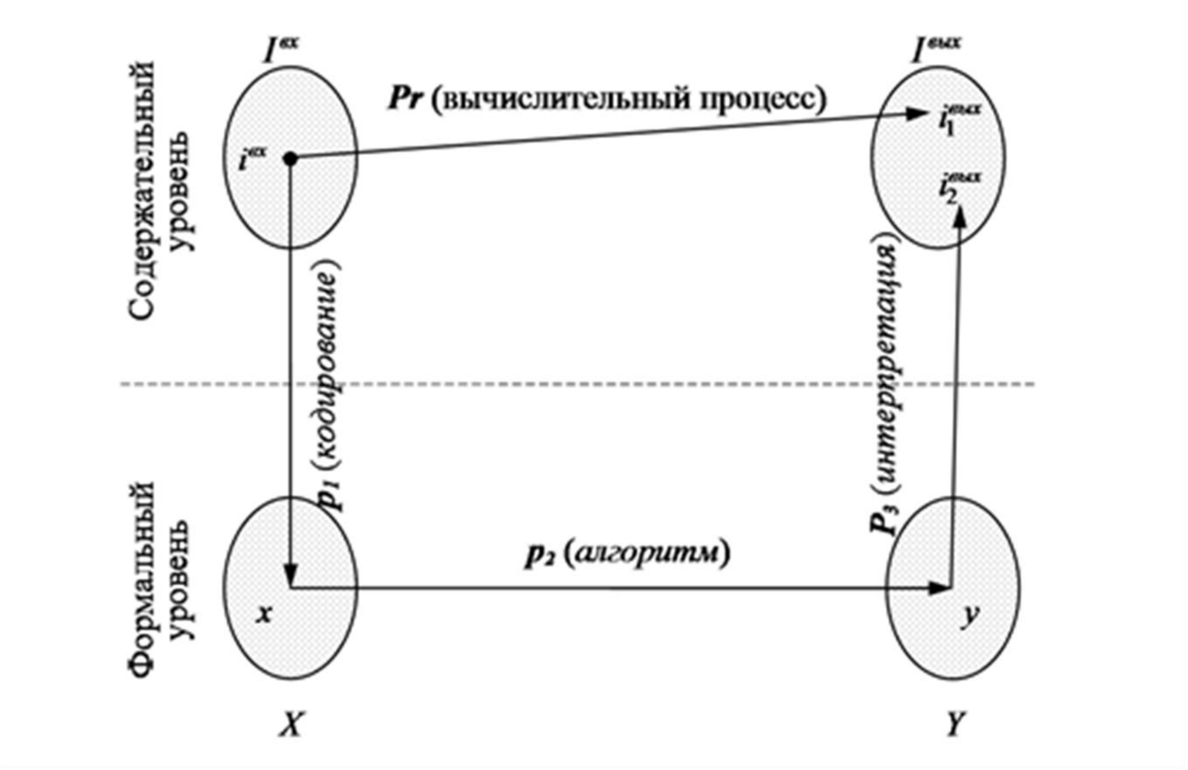
1.Приведите пример **ЗАДАЧИ** и покажите как задача решается с учетом общей схемы.

2. Можете привести пример **ЗАДАЧИ**, которая не “вписывается” в общую схему. Опишите как она решается.

Комментарий к лекции 3

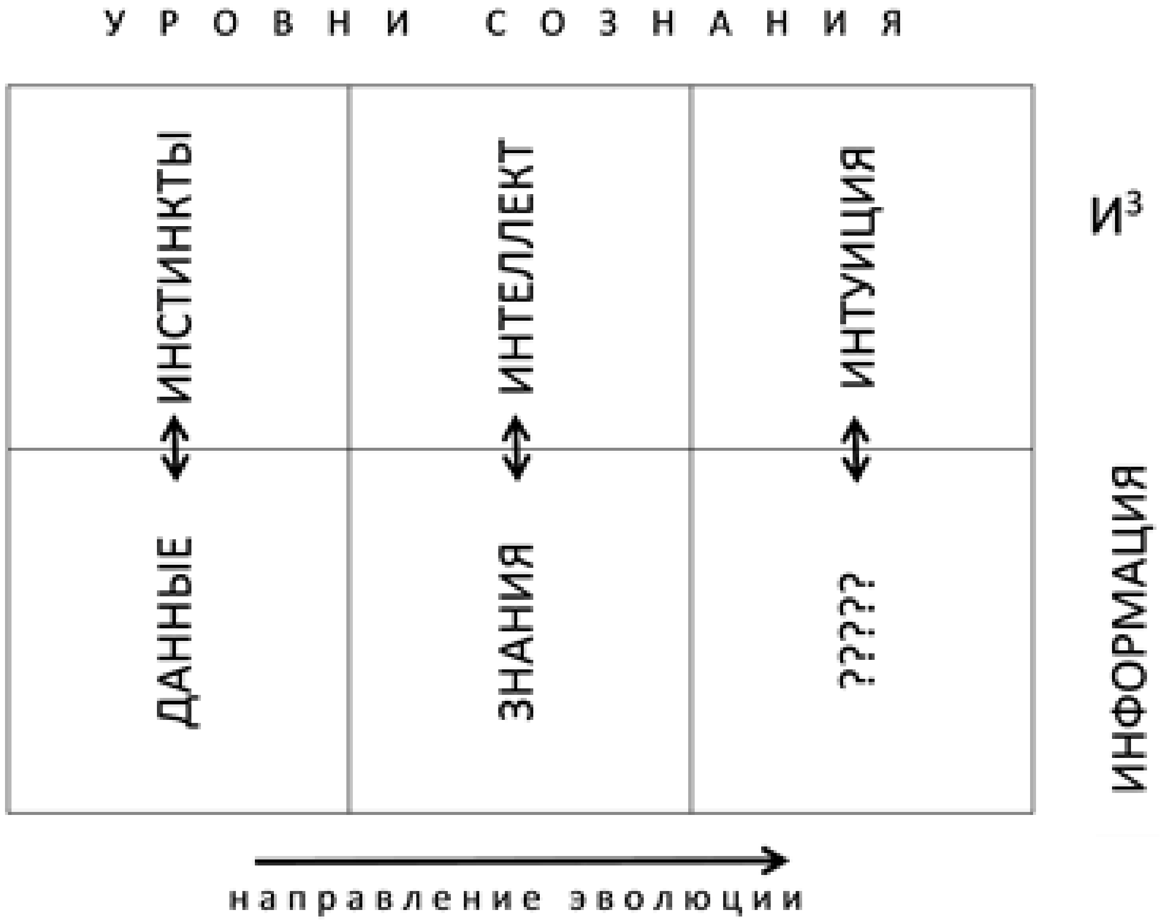
ОПРЕДЕЛЕНИЕ ЗАДАЧ ИИ

1. **Задача и информация**
2. **Дедуктивные и индуктивные задачи**
3. **Свойства задач ИИ + методология решения**



У Р О В Н И   С О З Н А Н И Я

ИНФОРМАЦИЯ

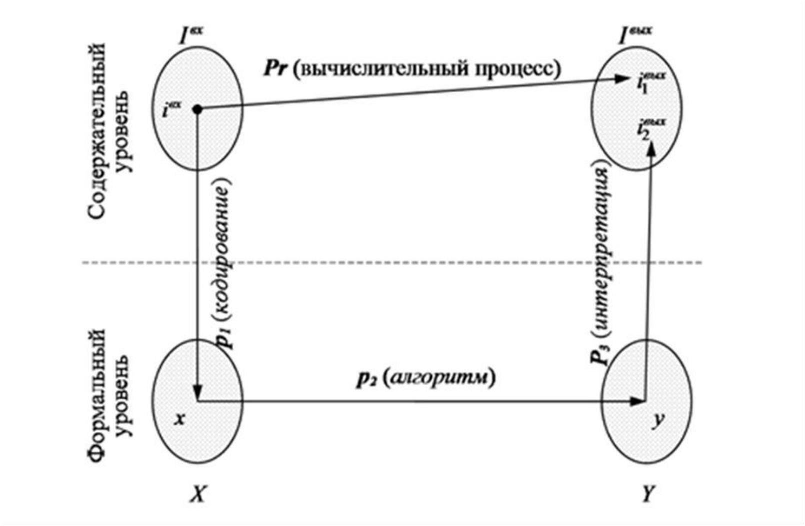


$$Z_{cod} \subseteq I^{вх} \times I^{вых}$$

$$Z_{\text{форм}} \subseteq X \times Y$$

$$информация = данные \cup знания \cup ???$$

$$информация \Leftrightarrow энергия$$



$$\Pr(i^{\text{вх}}) = i_1^{\text{вых}} \in I^{\text{вых}},$$

$$p_3 \circ p_2 \circ p_1(i^{\text{вх}}) = i_2^{\text{вых}} \in I^{\text{вых}}.$$

$$i_1^{\text{вых}} \overset{?}{\Leftrightarrow} i_2^{\text{вых}}$$

$$\forall i^{\text{вх}} \in I^{\text{вх}} \; ((\Pr(i^{\text{вх}}) = p_3 \circ p_2 \circ p_1(i^{\text{вх}})) \Leftrightarrow (i_1^{\text{вых}} = i_2^{\text{вых}}))$$

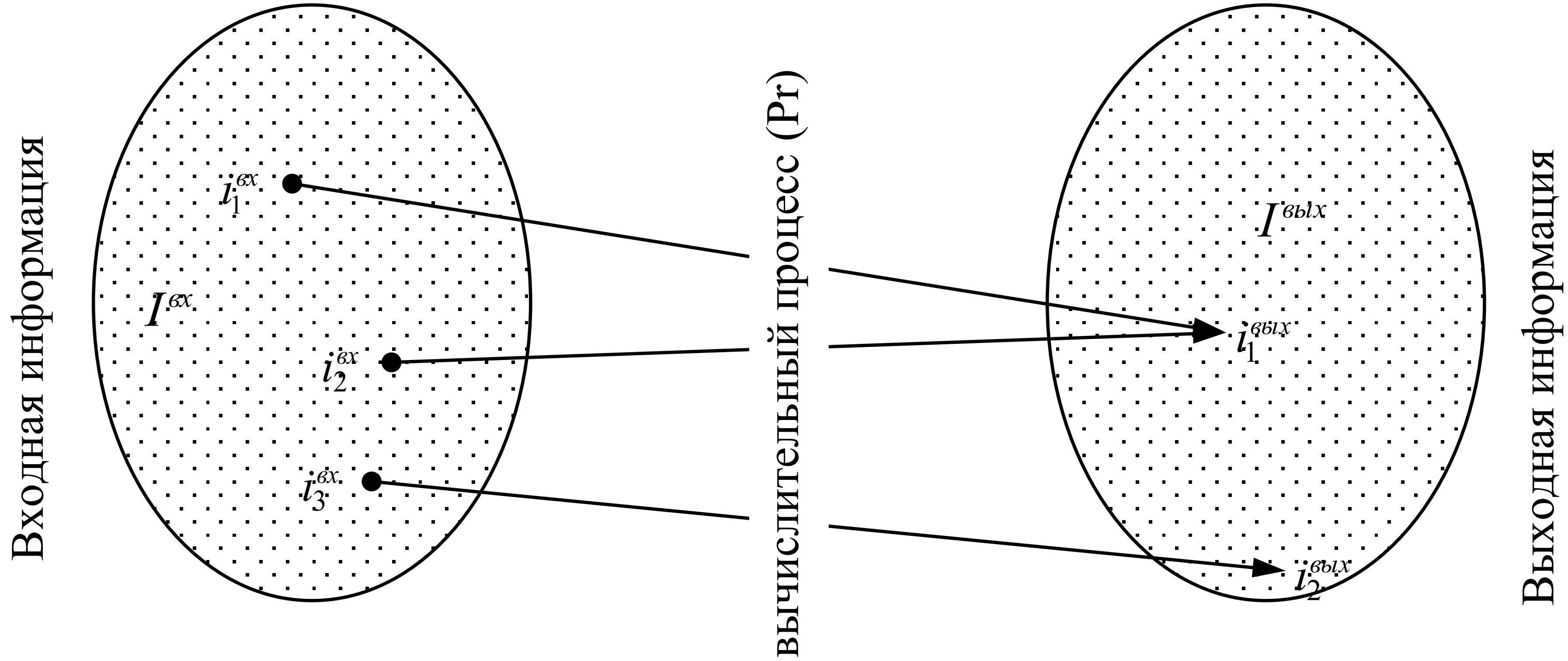
$$\psi : I^{вых} \times I^{вых} \to R^{+}$$

ψ- *метрика*

$$\varphi : R^{+} \to [0,1]$$

$$\varphi(r) = \begin{cases} 1, r = 0, \\ 0, r = \infty \end{cases}$$

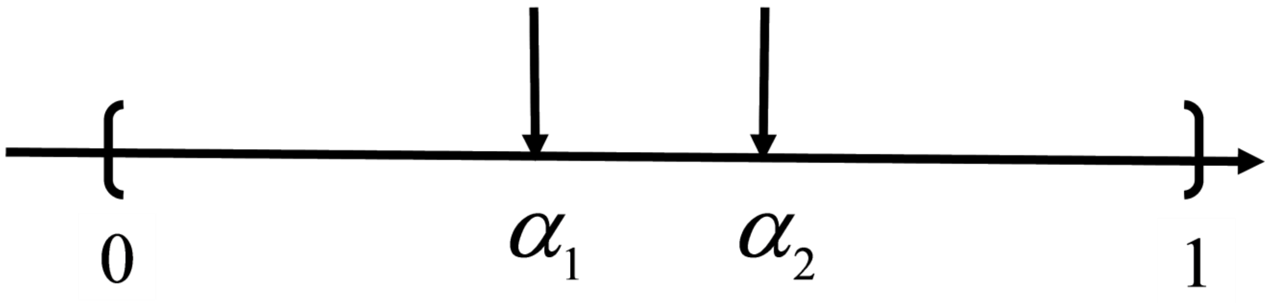
$$\Phi(I^{вх}) = \sum_{i^{in} \in I^{input}} \varphi(\psi(\Pr(i^{вх}), p_3 \circ p_2 \circ p_1(i^{вх}))) \cdot | I^{вх} |^{-1}$$


Входная информация
$I^{вх}$
$i_1^{вх}$
$i_2^{вх}$
$i_3^{вх}$
вычислительный процесс (Pr)
$I^{вых}$
$i_1^{вых}$
$i_2^{вых}$
Выходная информация

$$\Phi(I^{вх}) \Rightarrow \begin{cases} \Phi(I^{вх}) = 1 \\ \Phi(I^{вх}) < 1 \end{cases}$$

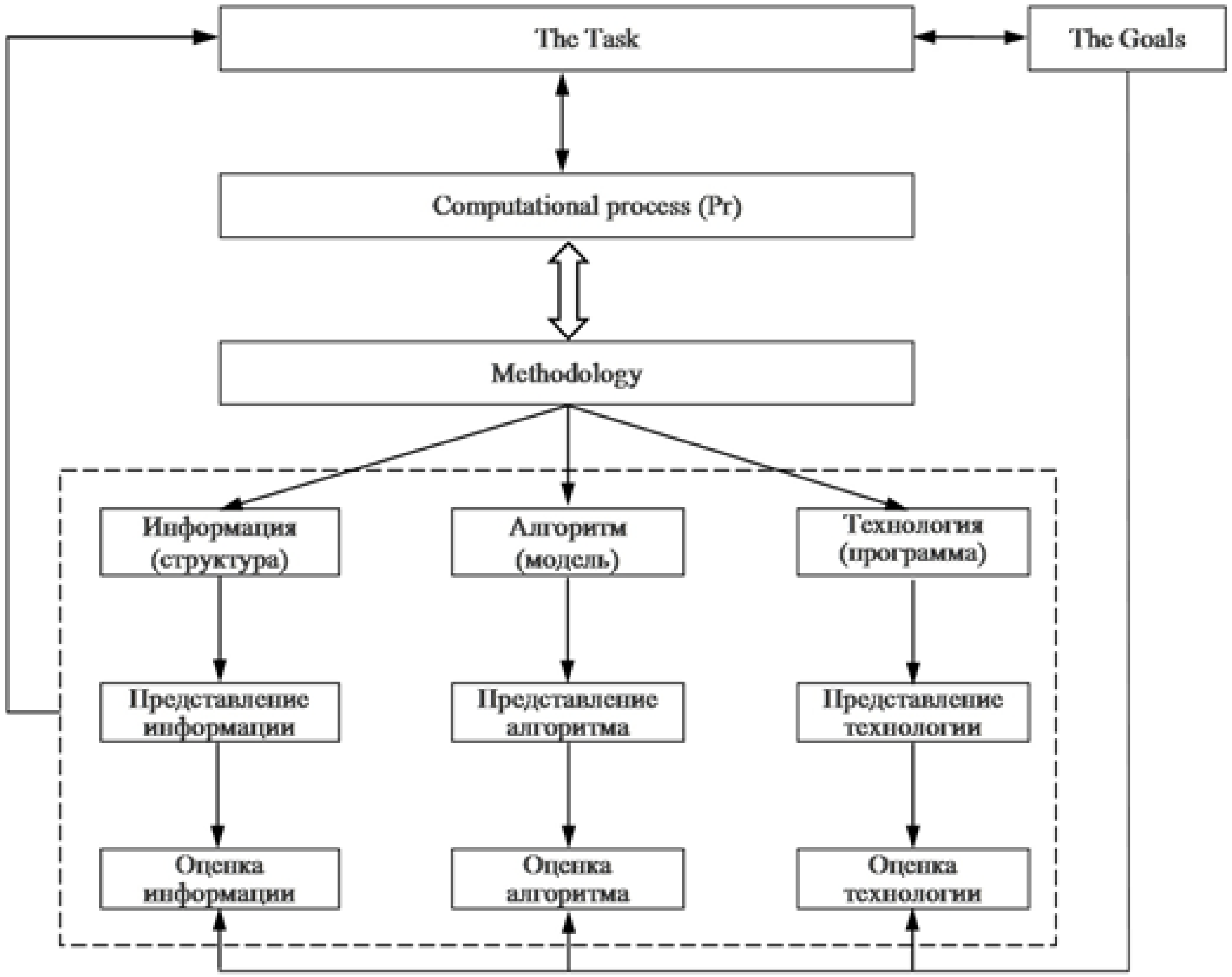
Дедуктивность

Индуктивность



Эволюция

1. ИНДУКТИВНОСТЬ
2. ЗНАНИЯ
3. ЭВОЛЮЦИЯ



Методология решения задач ИИ

ИНФОРМАЦИЯ К РАЗМЫШЛЕНИЮ

Предположим, что информация в ЗРСЧ задана не по **принципу свертки**, а с помощью **прецедентности**. Например: Задана обучающая выборка:

$$\mathbb{N}^0 = \{1397, 275,145,51,50,18\}$$

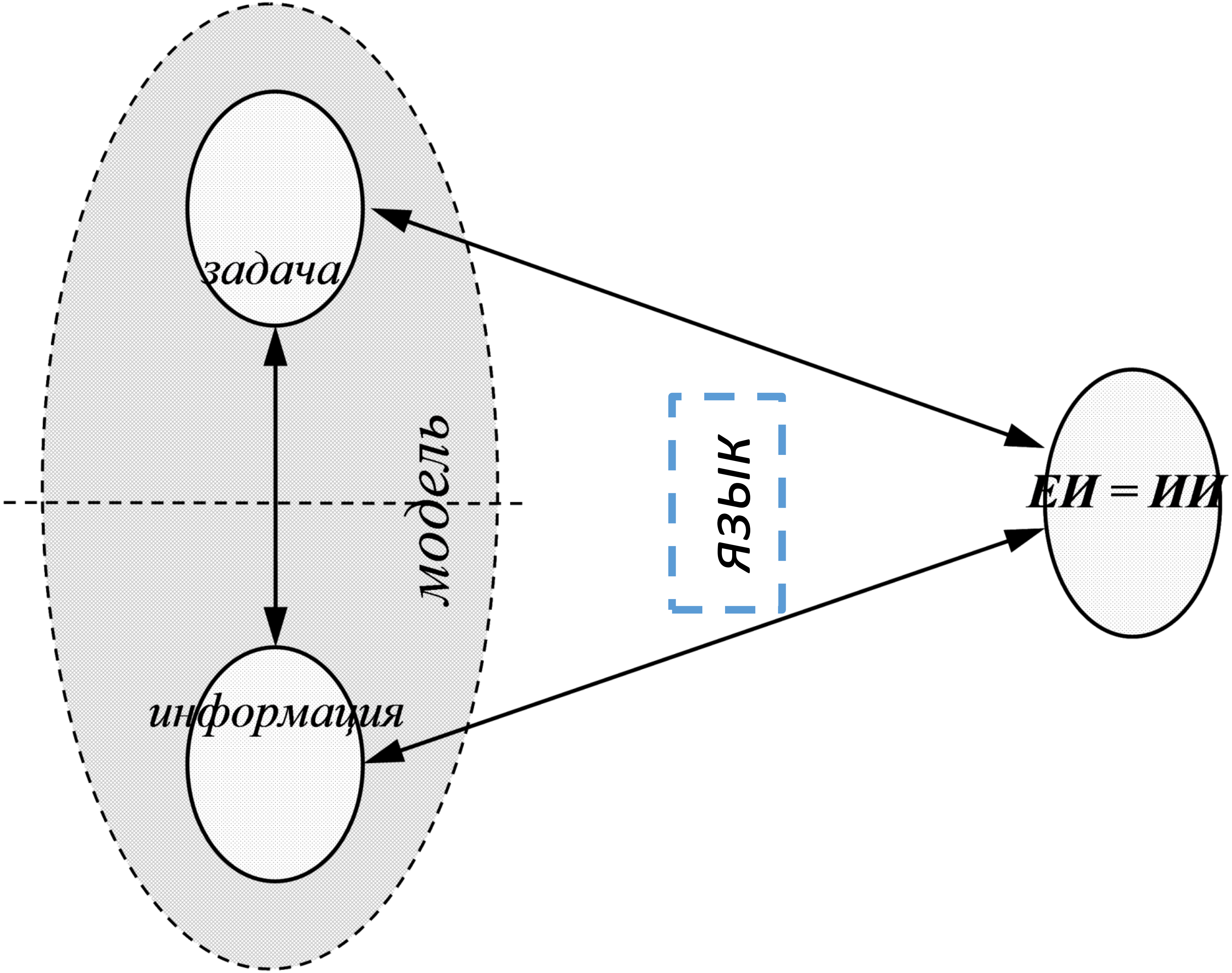
$$\mathbb{N}^0_{\text{ч}} = \{50,18\} \qquad \mathbb{N}^0_{\text{нч}} = \{1397, 275,145,51\}$$

Может ли ЗРСЧ быть решена в такой постановке. Если **может**, то укажите **алгоритм**.

Комментарий к лекции 4

МОДЕЛИ ИИ

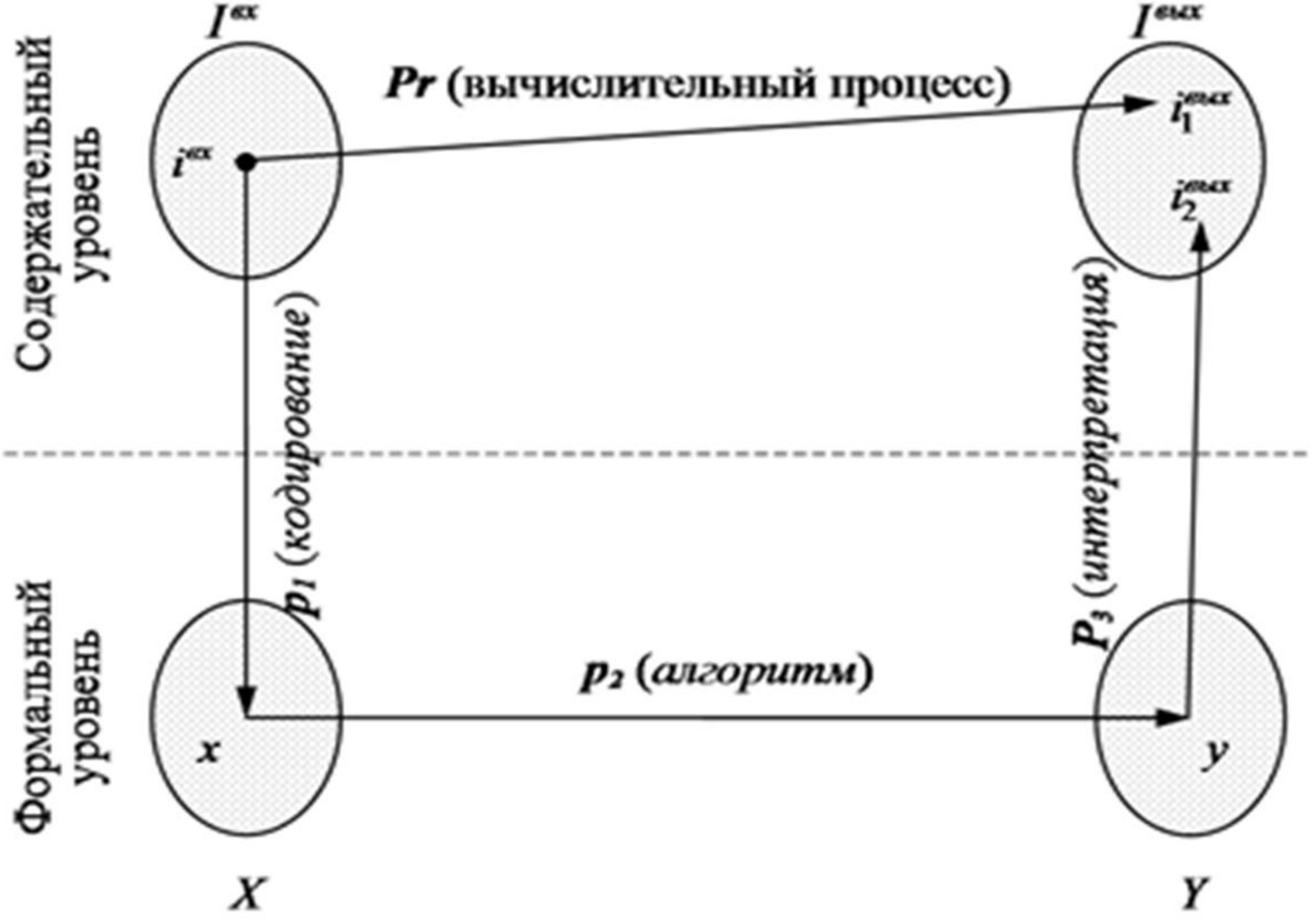
1. **Представление о моделировании**
2. **Классификация моделей**
3. **Общая структура логических моделей**



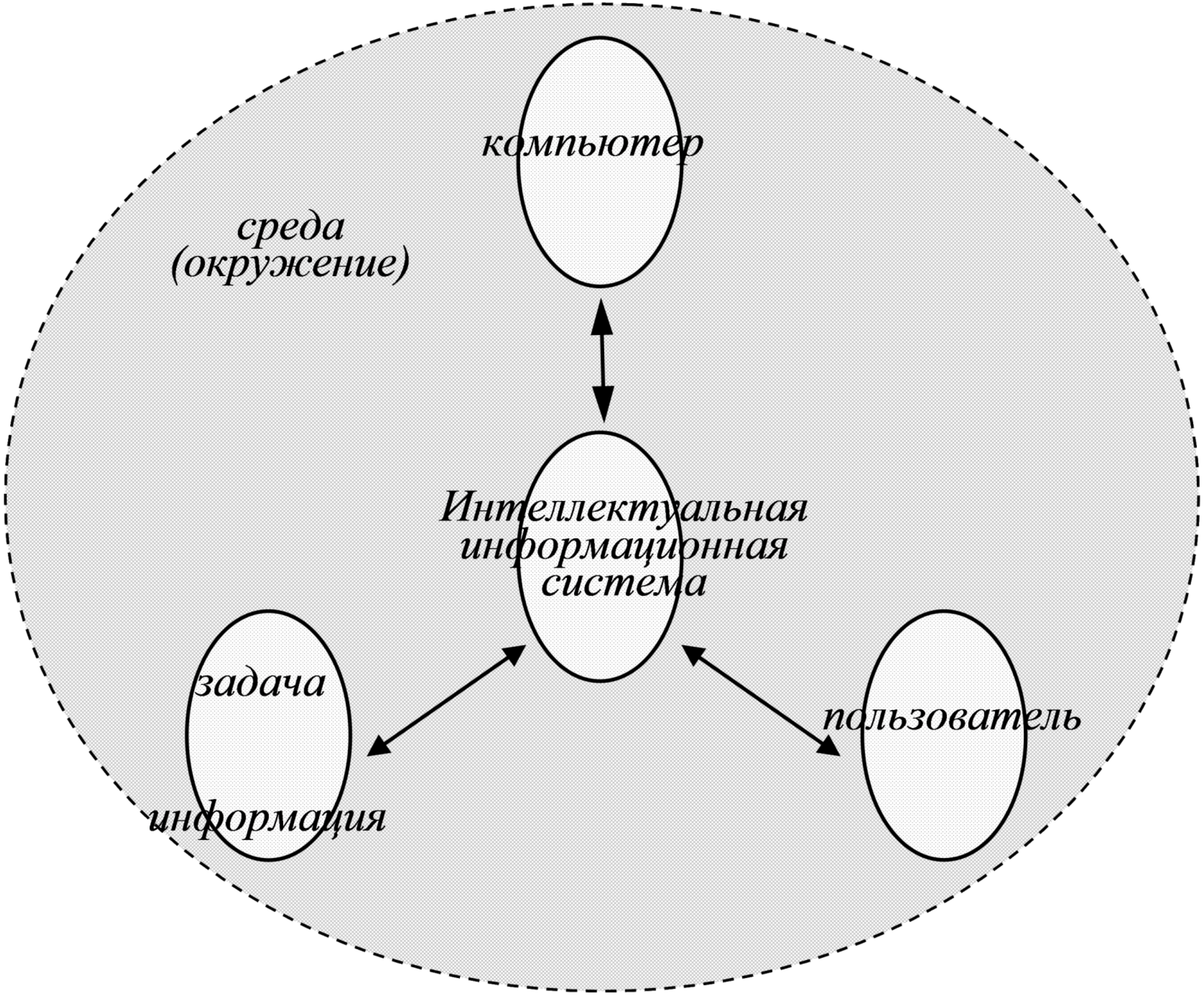
задача
информация
модель
ЯЗЫК
ЕИ = ИИ

- **Модель газа** как системы бильярдных шаров — классическая модель изображения, поясняющая поведение газа через взаимодействие частиц.
- **Модель атома Бора** — одна из первых квантовых моделей атома, описывающая энергетические уровни электрона.
- **Модель двойной спирали ДНК** — биологическая модель структуры ДНК, поясняющая механизм наследственности.
- **Модель разрешения поля Фарадея** — концептуальная модель для описания взаимодействия зарядов.
- **Модель эволюционного развития животных мира** — теоретическая модель биологической цивилизации.
- **Математические модели в экологии** — для анализа и прогнозирования экосистемных процессов.
- **Имитационные модели** — например, воспроизводят реальные процессы, движение молекул в газе или поведение колоний микробов.

- Уравнение Пуассона — применяется в механике и электротехнике для определения потенциалов.
- Уравнение теплопроводности — моделирует изменения температуры и атмосферного потока.
- Волновое уравнение — описывает распространение волн, в том числе звуковых, световых.
- Модель гармонического осциллятора — описывает колебание без затухания, например колебание пружины или маятника.
- Модель с затуханием — наблюдает уменьшение размеров колебаний со временем.
- Модель движения тела с постоянной скоростью — простая линейная модель.
- Модель равноускоренного движения — использует квадратичную функцию для описания ускоренного движения, включая падение падения.

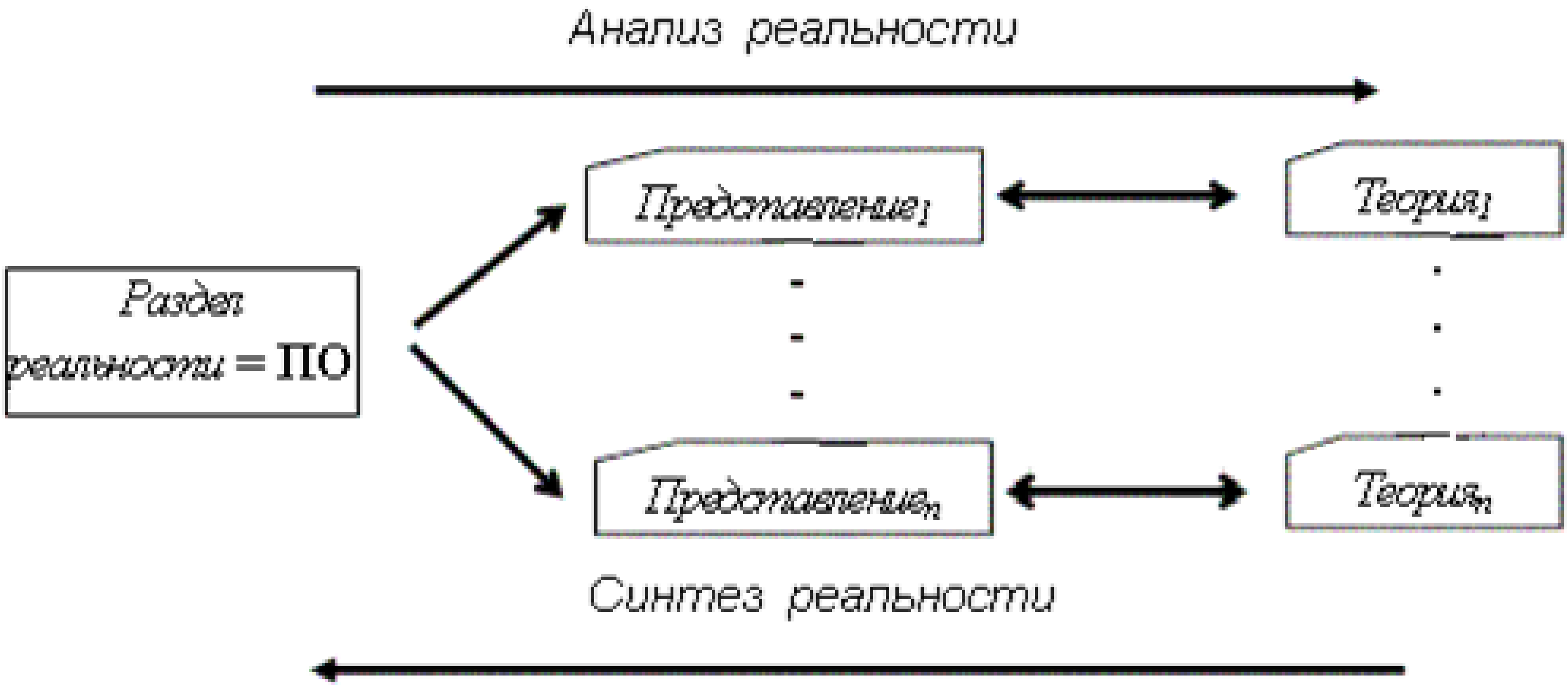


$I^{вх}$
$I^{вых}$
Pr (вычислительный процесс)
$i^{вх}$
$i_1^{вых}$
$i_2^{вых}$
Содержательный уровень
p_1 (*кодирование*)
p_3 (*интерпретация*)
Формальный уровень
p_2 (*алгоритм*)
x
y
X
Y


компьютер
среда
(окружение)
Интеллектуальная
информационная
система
задача
информация
пользователь

Реальность $\Rightarrow$ (представления о реальности $\Leftrightarrow$ теория, формализация)

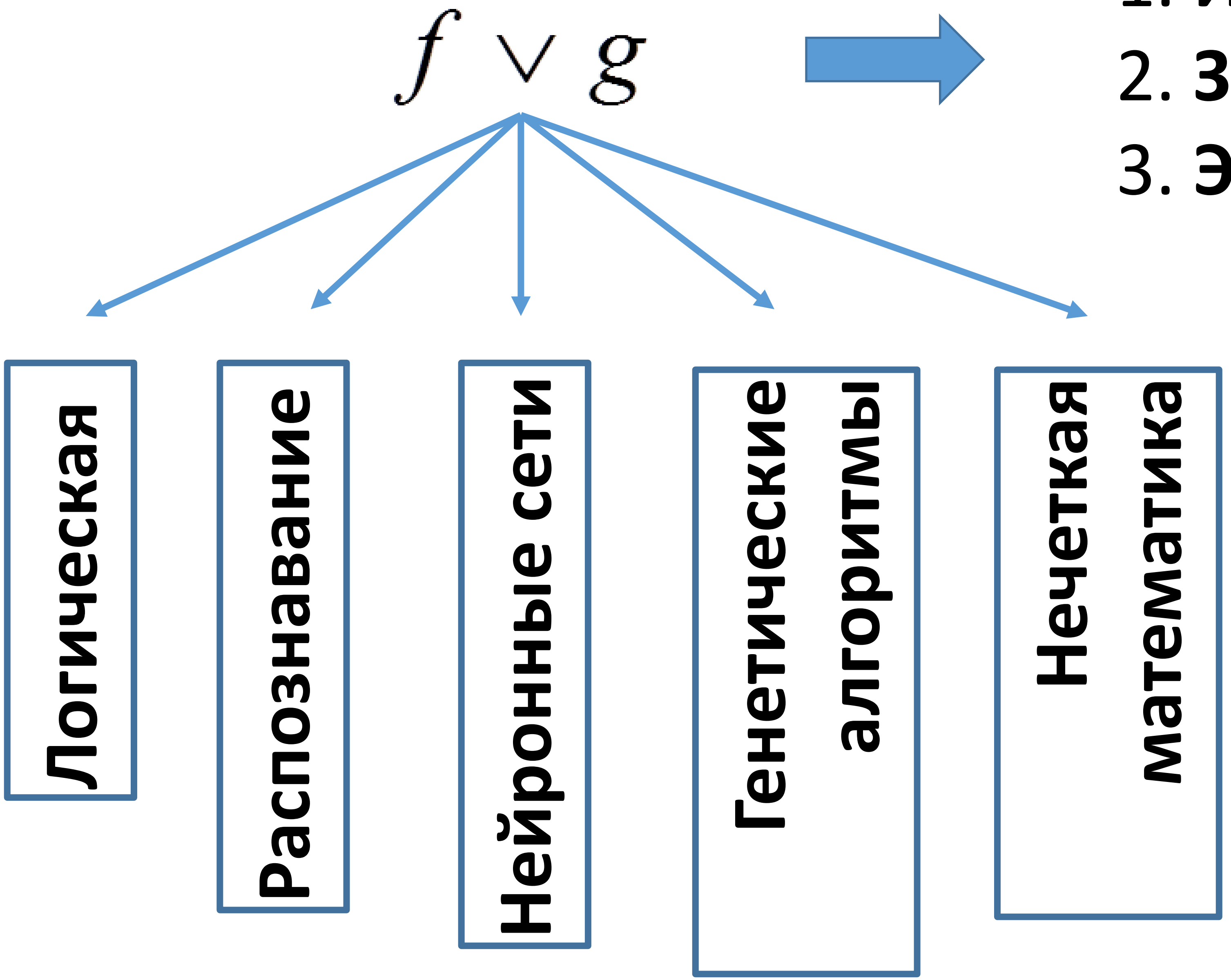
- **представления элементов реальности;**
- **манипулирования представленными объектами с целью получения информации о существующих (синтезируемых) элементах.**


Анализ реальности
Представление$_1$
Теория$_1$
Раздел реальности = ПО
Представление$_n$
Теория$_n$
Синтез реальности

$$Z_{сод} \subseteq I^{вх} \times I^{вых} \quad \text{или} \quad \mathrm{Z}_{\text{форм}} \subseteq \mathrm{X} \times \mathrm{Y}$$

$$g : I^{вх} \to I^{вых} \qquad f : \mathrm{X} \to \mathrm{Y}$$

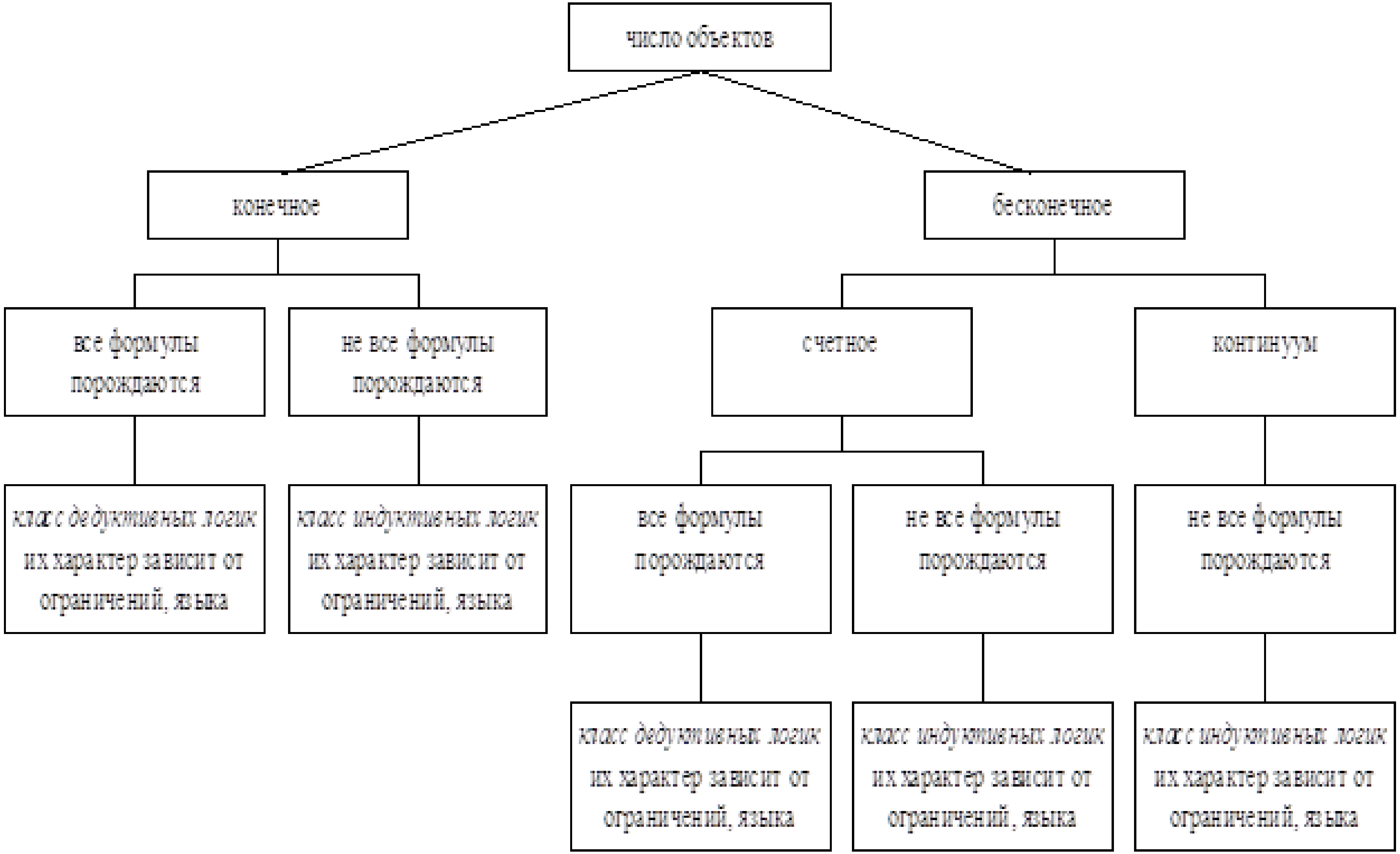
$$g^{-1} : I^{вых} \to I^{вх} \qquad f^{-1} : \mathrm{Y} \to \mathrm{X}$$



1. **ИНДУКТИВНОСТЬ**
2. **ЗНАНИЯ**
3. **ЭВОЛЮЦИЯ**

$$f : \mathrm{X} \to \mathrm{Y} \quad \Leftrightarrow \quad P_f(x, y) = \begin{cases} 1, f(x) = y \\ 0, \textit{иначе} \end{cases}$$

P_f - предикат, представляющий функцию f


число объектов
конечное
бесконечное
все формулы порождаются
не все формулы порождаются
счетное
континуум
класс дедуктивных логик их характер зависит от ограничений, языка
класс индуктивных логик их характер зависит от ограничений, языка
все формулы порождаются
не все формулы порождаются
не все формулы порождаются
класс дедуктивных логик их характер зависит от ограничений, языка
класс индуктивных логик их характер зависит от ограничений, языка
класс индуктивных логик их характер зависит от ограничений, языка

Логическая модель считается определенной, если:

- **задан конечный алфавит** (множество символов);
- **определена процедура построения формул** или слов формальной системы;
- **выделено множество формул, называемых аксиомами**;
- задано **конечное множество правил вывода**, которые позволяют получать из аксиом другие формулы.

- **невозможность вывода отрицания** уже доказанного утверждения, т.е.

$$E \vdash A \Rightarrow E \wedge \neg A \vdash A$$

- логическая модель должна быть **минимальной**, т.е. не содержащей бесполезных аксиом и правил вывода.

- логическая модель должна быть адекватной, т.е. любая теорема должна быть **общезначимой** и **полной**

$$\vdash A \Rightarrow \models A \qquad\qquad \models A \Rightarrow \vdash A$$

<u>логическая модель JP:</u>

- словарь: {a, b, =}.
- формулы-любая комбинация символов из множества {a, b, =}.
- аксиома: a = a.
- правило вывода: $c_1 = c_2 \Rightarrow bc_1 = bc_2$

Очевидно, что в данной модели могут быть получены следующие теоремы:

$$a = a$$

$$ba = ba$$

$$bba = bba$$

Между тем, bab = a – является формулой, но не выводимой.

логическая модель DH

- словарь: {M, I, U}.
- формулы - любая последовательность символов из множества {M, I, U}.
- аксиома: MI.
- правила вывода: mI → mIU

  Mm → Mmm

  mIII → mU

  mUU → m

В данной модели может быть следующий вывод:

MI → (2) MII → (2) MIIII → (1) MIIIIU → (3) MIUU → (4) MI

Существует различие между концепциями доказательства и истинности, т.к. эти два понятия относятся к различным областям. Всего имеется 4 варианта взаимоотношения между доказательством и значением истинности. Обозначим теоремы через **Т**, не теоремы – **NT**. Их интерпретация может быть истинной (**И**), либо ложной (**Л**). Соотношение между этими вариантами можно представить следующим образом:

Т, И	**Т, Л**
NT, И	**NT, Л**

ПРИМЕРЫ

- Условимся, что символ a означает 0, a b – последующее целое число за символом, стоящим справа от него. Т.е. bba $\rightarrow$ 2, bbba $\rightarrow$ 3 и т.д. Символом же = обозначим знак равенства. Тогда аксиома интерпретируется как 0 = 0. Теоремы же – 1 =1, 2 = 2 и т.д. Очевидно, что и аксиома, и теоремы являются истинными при такой интерпретации. В противоположность этому формула 1 = 2, которая не интерпретирует никакую из теорем, является ложной. Заметим также, что в этой интерпретации множества (Т, Л) и (NT, И) пусты.
- Пусть теперь a означает высказывание “Сократ смертен”, b – символ отрицания, a = - идентичность двух высказываний. Тогда аксиома формулируется: “Сократ смертен = Сократ смертен”. Формула “Сократ бессмертен = Сократ бессмертен”, которая является теоремой – также истинна. При такой интерпретации формула “$\neg\neg$Сократ смертен = Сократ смертен”, которая является не теоремой, также истинна. В этой интерпретации непустыми будут классы (NT, И) и (NT,Л)

ИНФОРМАЦИЯ К РАЗМЫШЛЕНИЮ

Является ли MU теоремой системы DH? Приведите доказательство своего утверждения.

Комментарий к лекции 5

МОДЕЛЬ ИСЧИСЛЕНИЯ ВЫСКАЗЫВАНИЙ

1. **Построение модели и основные ее свойства**
2. **Проблема доказуемости формул**
3. **Алгоритмы доказательства разрешимости**
4. **Метод резолюции**

1. **Алфавит:**

a.буквы p, q, r, s, t,….

b.логические операторы $\neg$, $\supset$

c.скобки ()

2. **Построение формул:**

a.если m – формула, то (m) – также формула

b.если m_1, m_2, m_3 – формулы, то $\neg m_1$, $m_2 \supset m_3$ – также формулы

c.всякое высказывание (буква) есть формула

d.формулы однозначно получаются с помощью правил a-b.

3. **Аксиомы:** (m_1, m_2, m_3 – формулы или буквы)

(A1) $(m_1 \supset (m_2 \supset m_1))$

(A2) $((m_1 \supset (m_2 \supset m_3)) \supset ((m_1 \supset m_2) \supset (m_1 \supset m_3)))$

(A3) $((\neg m_1 \supset \neg m_2) \supset ((\neg m_1 \supset m_2) \supset m_1))$

4. Правила вывода

если m_1 и $(m_1 \supset m_2)$ – теоремы, то m_2 также теорема

Дополнительные логические связки: $\vee$ (дизъюнкция), $\wedge$ (конъюнкция), $\equiv$ (эквивалентность).

$(m_1 \equiv m_2)$ равносильно $(m_1 \supset m_2) \wedge (m_2 \supset m_1)$
$(\neg m_1 \vee m_2)$ равносильно $(m_1 \supset m_2)$
$\neg(m_1 \wedge m_2)$ равносильно $(\neg m_1 \vee \neg m_2)$

Формула исчисления высказываний может быть интерпретирована, т.е. ей может быть приписано одно из двух значений – И или Л.

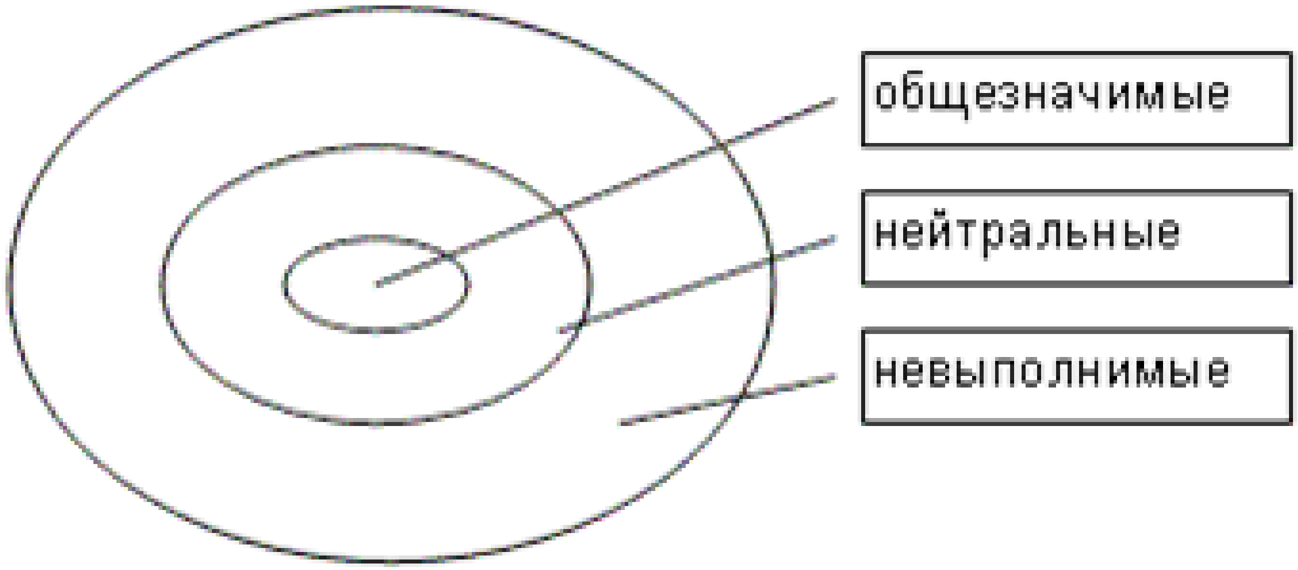
p	q	$p\wedge q$	$p\vee q$	$p\supset q$	$p\equiv q$
И	И	И	И	И	И
И	Л	Л	И	Л	Л
Л	И	Л	И	И	Л
Л	Л	Л	Л	И	И

p	$\neg p$
И	Л
Л	И

Формула **выполнима** (нейтральна), если при некоторых значениях входящих в нее символов (высказываний) она допускает интерпретацию со значением И.

Формула **общезначима** (тавтология), если она истинна, независимо от значений входящих в нее символов (высказываний).

Формула **невыполнима**, если при всех значениях входящих в нее символов (высказываний) она интерпретируется со значением Л.



Теорема Поста: Формула A доказуема в ИВ тогда и только тогда, когда она является тавтологией.

ИВ характеризуется:

1. **непротиворечивостью**, т.е. A и $\neg$ A не могут быть одновременно выводимыми;
2. **полнотой**, т.е. теоремы в точности соответствуют тавтологиям;
3. **разрешимостью**, т.е. существует процедура решения проблемы доказуемости.

Для других формул применяется редуцирование проблем, смысл которого основан на следующем **утверждении**: А является логическим следствием из В тогда и только тогда, когда импликация (В $\supset$ А) есть тавтология, т.е.

$$B \models A \Leftrightarrow \models (B \supset A)$$

Это утверждение имеет и следующую, более общую, формулировку:

$$(H_1,\ldots, H_n) \models A \Leftrightarrow \models ((H_1,\ldots, H_n) \supset A)$$

H_i называют **гипотезами**. В последнем виде способ доказательства называют **принципом дедукции**. Другая форма принципа дедукции

$$(H_{\mathit{1}} \wedge \ldots \wedge H_{n}) \models A \Leftrightarrow (H_{\mathit{1}} \wedge \ldots \wedge H_{n} \wedge \neg A) \models Л$$

Два принципа **математической индукции**:

- *принцип полной математической индукции*

$$(P(1) \wedge P(n) \Rightarrow P(n+1)) \Rightarrow \forall n \; P(n)$$

- *принцип трансфинитной индукции*

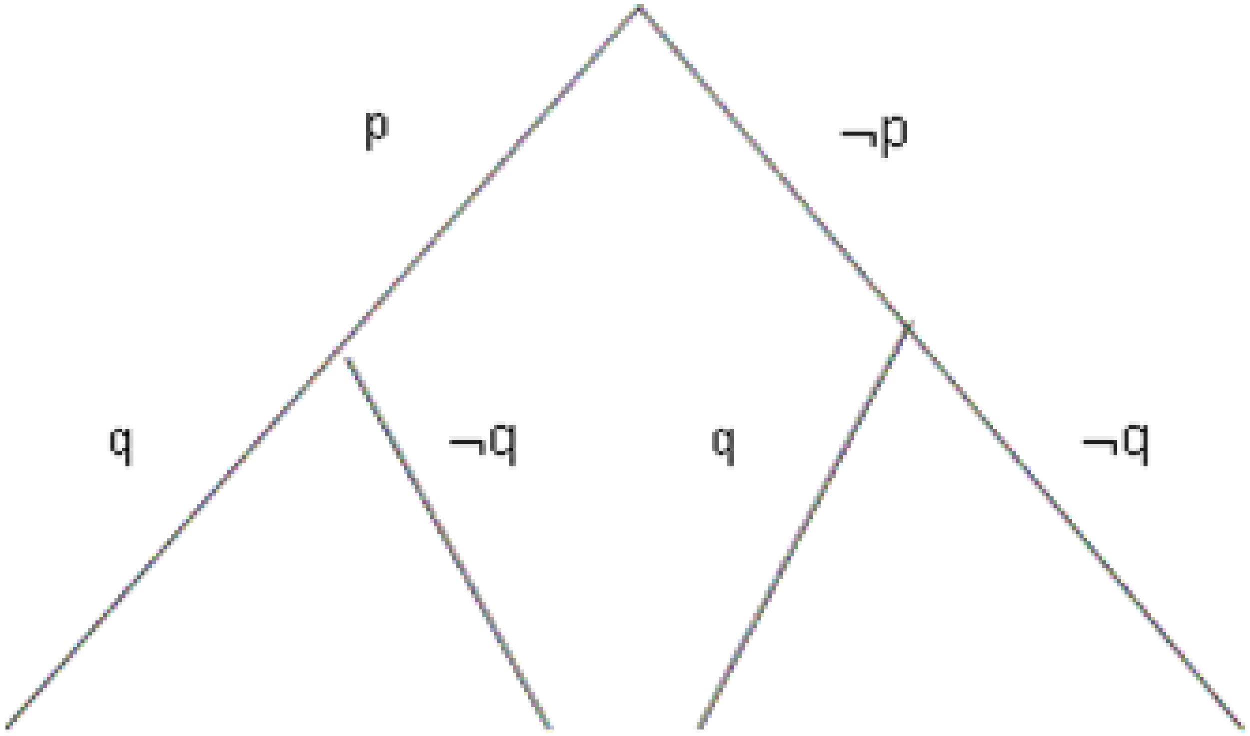
$$(P(1) \wedge \forall m < n \; P(m) \Rightarrow P(n)) \Rightarrow \forall n \; P(n)$$

где *P* – некоторое свойство (предикат), *m*, *n* – натуральные числа.

СЕМАНТИЧЕСКОЕ ДЕРЕВО

- каждая дуга помечена позитивной или негативной литерой, взятой из P;
- литеры, которыми помечены две дуги, выходящие из одного узла, противоположны;
- никакая ветвь не содержит более одного вхождения каждой литеры;
- никакая ветвь не содержит пару противоположных литер.

$P = \{p, q\}$



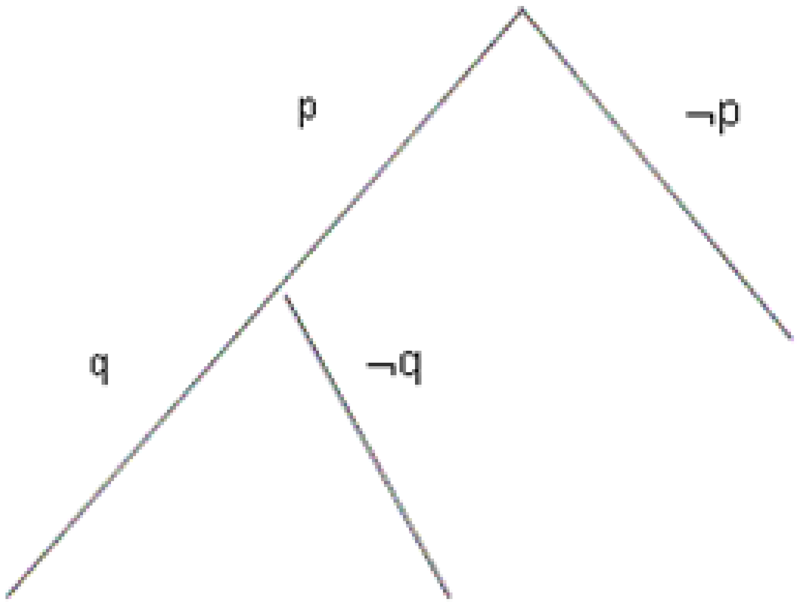
Тривиальный алгоритм.

смысл: просмотр полного семантического дерева

Алгоритм Куайна.

смысл: если при всех возможных расширениях некоторой частичной интерпретации, исследуемая формула принимает одно и то же истинностное значение из {И, Л}, то бесполезно строить поддерево, исходящее из узла, соответствующего этой частичной интерпретации

Пример: $(((p \wedge q) \supset r) \wedge (p \supset q)) \supset (p \supset r)$



Алгоритм редукции.

смысл: доказывается общезначимость формул путем приведения к абсурду (от противного)

Пример: $((p \wedge q) \supset r) \supset (p \supset (q \supset r))$

Принцип резолюций.

Дизъюнктом называется дизъюнкция конечного числа литер, т.е. формула вида:

$$(l_1 \vee \ldots \vee l_n)$$

Конъюнктивной нормальной формой (КНФ) называется конъюнкция конечного числа дизъюнктов.

Утверждение: Любая формула имеет логически эквивалентную ей КНФ.

Алгоритм приведения к КНФ

1. Исключаем операции $\equiv, \supset$. Для этого заменяем $(m_1 \equiv m_2)$ на

$$(m_1 \supset m_2) \wedge (m_2 \supset m_1)$$

а затем $(m_1 \supset m_2)$ на $(\neg m_1 \vee m_2)$.

2. Добиваемся того, чтобы операция $\neg$ оставалась только непосредственно перед литерами. Для этого необходимое число раз применяем преобразования:

$$\neg(m_1 \wedge m_2) \to (\neg m_1 \vee \neg m_2),$$

$$\neg(m_1 \vee m_2) \to (\neg m_1 \wedge \neg m_2),$$

$$\neg\neg m_1 \to m_1$$

3. Необходимое число раз применяются преобразования, основанные на дистрибутивности операций:

$$m_1 \vee (m_2 \wedge m_3) \rightarrow (m_1 \vee m_2) \wedge (m_1 \vee m_3),$$
$$m_1 \wedge (m_2 \vee m_3) \rightarrow (m_1 \wedge m_2) \vee (m_1 \wedge m_3).$$

4. Дизъюнкты, содержащие литеру (высказывание) и ее отрицание

$$(m \vee \neg m) = И$$

могут быть опущены, т.к. они общезначимы. Опускаются также повторения литер (высказываний) в пределах одного дизъюнкта.

Примеры. a. $(p \supset (q \supset r))$ b. $((p \wedge s) \supset r)$

- множество дизъюнктов невыполнимо в том и только том случае, когда Л является логическим следствием из него. Таким образом, невыполнимость множества формул *S* можно проверить порождая логические следствия из *S* до тех пор, пока не получим пустой дизъюнкт (Л).
- для порождения логических следствий может использоваться простая схема рассуждений. Пусть m_1, m_2, m_3 - формулы. Предположим, что две формулы $(m_1 \vee m_3)$ и $(m_2 \vee \neg m_3)$ истинны. Если формула m_3 истинна, то можно сделать заключение, что m_2 истинна. В противном случае, формула m_1 истинна. Во всех случаях формула $(m_1 \vee m_2)$ является истинной. Получается следующее правило:

$$\{m_1 \vee m_3, m_2 \vee \neg m_3\} \models m_1 \vee m_2$$

когда m_3 - высказывание, а m_1, m_2 - дизъюнкты, это правило называется **правилом резолюций**. Его общезначимость выражается в следующем **утверждении**:

Пусть S_1, S_2 - дизъюнкты некоторой формулы S, l – литера. Если $l \in S_1$ и $\neg l \in S_2$, то дизъюнкт

$$r = (S_1 \setminus l) \vee (S_2 \setminus \neg l)$$

является логическим следствием формулы S.

Замечания.

1. Нетрудно показать, что формулы S и $S \wedge r$ эквивалентны.
2. Дизъюнкт r называется резольвентой дизъюнктов S_1, S_2.

Алгоритм (метод резолюций).

1. при условии, что $Л \notin S$ переходим к шагу 2. В противном случае – к шагу 5.
2. выбираем l, s_1, s_2 такие, что $l \in s_1$ и $\neg l \in s_2$.
3. вычисляем резольвенту r.
4. заменяем S на $S \wedge r$ и переходим к шагу 1.
5. конец работы алгоритма.

Пример. $S = \{p \vee q, p \vee r, \neg q \vee \neg r, \neg p\}$

Свойства:

- если S не содержит ни одной пары дизъюнктов, допускающих резольвенту, то S выполнимо;
- Если S невыполнимо и содержит резольвенты своих дизъюнктов, то оно обязательно содержит и пустой дизъюнкт;
- Если S невыполнимо, то этот факт всегда можно установить алгоритмом резолюций.

Пример. $\{h, \neg h \vee p \vee q, \neg p \vee c, \neg q \vee c\} \supset c$

Комментарий к лекции 6

МОДЕЛЬ ИСЧИСЛЕНИЯ ПРЕДИКАТОВ

1. **Построение модели**
2. **Доказательство выполнимости в ИП**
3. **Метод резолюции**

СИЛЛОГИЗМ АРИСТОТЕЛЯ

Все люди смертны, Сократ – человек. Следовательно, Сократ смертен

Обозначим высказывания, образующие это рассуждение через p, q и r.

$$(p \wedge q) \supset r$$

Силлогизм можно переписать следующим образом:

Для всех x, если x является человеком, то x смертен. Сократ является человеком, следовательно, Сократ смертен

Введем предикатные константы $H(x), M(x)$, которые будут означать свойства “являться человеком” и “быть смертным”. Тогда силлогизм может быть приведен к виду:

$$((\forall x (H(x) \supset M(x)) \wedge H(\textit{Сократ})) \supset M(\textit{Сократ}))$$

1. **Алфавит:**

a. константы p, q, r, s, t,….

b. индивидуальные переменные x, y, z, …

c. предикаты A, B, C, …

d. логические операторы $\neg, \supset$

e. скобки ()

f. квантор всеобщности $\forall$

2. **Построение формул:**

Формулы ИП образуются аналогично формулам ИВ. Кроме того:

a. каждому предикату приписывается вес k. Выражение $A(x_1,\ldots,x_k)$ является формулой, если и только если вес A равен k.

b. Выражение $(\forall x_1\, A(x_1,\ldots,x_k))$ представляет собой формулу, в которой x_1 есть связанная переменная , а все остальные $x_i\ (i \geq 2)$ - свободные.

3. **Аксиомы**:

Имеются все три аксиомы ИВ (А1-А3), а также:

$$(A4) \quad (\forall t)A(t) \supset A(И)$$

$$(A5) \quad (\forall t)(m_1 \supset m_2) \supset (m_1 \supset (\forall t)m_2)$$

где m_1, m_2 - формулы, t – не является свободной переменной в m_1 .

4. **Правила вывода**:

$(m_1 \wedge (m_1 \supset m_2) \supset m_2$ modus ponens

$m_1 \supset (\forall t) m_1$ обобщение

где t – свободная переменная в m_1.

Доказательство выполнимости в ИП

Алгоритм приведения к КНФ состоит из четырех основных этапов.

1. Приведение к *предваренной (префиксной)* форме.

Предваренной называется формула, которая имеет вид

$$Q_1 x_1 \ldots Q_n x_n M,$$

где Q_i обозначает один из кванторов $\forall, \exists$, M – формула без кванторов.

Утверждение: *для любой формулы ИП существует логически эквивалентная ей предваренная формула*.

Алгоритм приведения к предваренной форме:

a. исключить операции эквивалентности и импликации;
b. если необходимо, переименовать связанные переменные таким образом, чтобы никакая переменная не имела бы одновременно свободных и связных вхождений;
c. удалить кванторы, которые не распространяются на квантифицированную переменную;
d. преобразовать формулу в соответствии со следующими правилами (снятие отрицаний или уменьшение радиуса действия операции $\neg$):

$$\neg\forall xA \to \exists x\neg A,\ \neg\exists xA \to \forall x\neg A,$$

$$\neg(A \wedge B) \to (\neg A \vee \neg B),\ \neg(A \vee B) \to (\neg A \wedge \neg B),$$

$$\neg\neg A \to A$$

e. переместить все кванторы в начало формулы. Для этого используются следующие правила:

$$(\forall xA \wedge \forall xB) \to \forall x(A \wedge B),\ (\forall xA \vee \forall xB) \to \forall x(A \vee B)$$

$$(\forall xA \wedge B) \to \forall x(A \wedge B),\ (\forall xA \vee B) \to \forall x(A \vee B)$$ (A не содержит x)

$$(A \wedge \forall xB) \to \forall x(A \wedge B),\ (A \vee \forall xB) \to \forall x(A \vee B)$$ (B не содержит x)

$$(\exists xA \wedge B) \to \exists x(A \wedge B),\ (\exists xA \vee B) \to \exists x(A \vee B)$$ (B не содержит x)

$$(A \wedge \exists xB) \to \exists x(A \wedge B),\ (A \vee \exists xB) \to \exists x(A \vee B)$$ (A не содержит x)

Пример.

$$\forall x(P(x) \wedge \forall y \exists x(\neg Q(x, y) \supset \forall z R(a, x, y)))$$

$$\left\{\begin{array}{ll}
\forall x(P(x) \wedge \forall y \exists x(\neg\neg Q(x, y) \vee \forall z R(a, x, y))) & \text{(п. a)} \\
\forall x(P(x) \wedge \forall y \exists u(\neg\neg Q(u, y) \vee \forall z R(a, u, y))) & \text{(п. b)} \\
\forall x(P(x) \wedge \forall y \exists u(\neg\neg Q(u, y) \vee R(a, u, y))) & \text{(п. c)} \\
\forall x(P(x) \wedge \forall y \exists u(Q(u, y) \vee R(a, u, y))) & \text{(п. d)} \\
\forall x \forall y \exists u(P(x) \wedge (Q(u, y) \vee R(a, u, y))) & \text{(п. e)}
\end{array}\right.$$

2. Приведение к сколемовской нормальной форме (удаление $\exists$).

Каждой формуле A сопоставляется формула S_A , которые выполнимы либо нет одновременно. Доказано, что формула $S_A \supset A$ тавтология.

Алгоритм:

a. сопоставить каждой $\exists$- кв. переменной список $\forall$- кв. переменных, предшествующих ей, а также функцию f, число переменных у которой равно мощности составленного списка;
b. заменить каждое вхождение $\exists$- кв. переменной на терм в виде функции f, полученной на предыдущем шаге;
c. удалить $\exists$- квантификацию.

Примеры.

$$\forall x \forall y \exists u\, (P(x) \wedge (Q(u, y) \vee R(a, u, y)))$$

$u \to f(x, y)$ подстановка

$\forall x \forall y\, (P(x) \wedge (Q(f(x, y), y) \vee R(a, f(x, y), y)))$ СНФ

$$\exists u \forall v \exists w \forall x \forall y \exists z\, M(u, v, w, x, y, z)$$

$u \to a \qquad w \to f(v) \qquad z \to g(v, x, y)$ подстановки

$\forall v \forall x \forall y\, M(a, v, f(v), x, y, g(v, x, y))$ СНФ

3. Удаление кванторов всеобщности.

Пусть задана произвольная формула А, которая зависит от некоторых свободных переменных $x_1,\ldots,x_n$. Обозначим U_A - $\forall$ - замыкание формулы А (т.е. $\forall x_1\ldots\forall x_n A$), и E_A - $\exists$ - замыкание А (т.е. $\exists x_1\ldots\exists x_n A$). Тогда, можно доказать, что:

$$\models A \Leftrightarrow \models U_A, \quad \models \neg A \Leftrightarrow \models \neg E_A$$

4. Приведение к КНФ.

Для этого можно использовать те же правила, которые использовались для приведения к КНФ формулы ИВ.

Метод резолюций.

Пусть А – произвольная формула, а Q_A- КНФ формулы А. В общем случае доказано, что:

$$\models A \Leftrightarrow \models Q_A, \quad \models \neg A \Leftrightarrow \models \neg Q_A$$

Число интерпретаций формулы Q_A, невыполнимость которой требуется показать, бесконечно и несчетно. Эрбран показал, что для каждой формулы существует ограниченная область H_A такая, что:

> Q_A является невыполнимой тогда и только тогда, когда она принимает значение Л при всех интерпретациях именно на области H_A .

Эрбранова область формулы Q_A (приведенной к КНФ) – это минимальное множество H_A, удовлетворяющее следующим условиям:

a. для любой константы *a*, имеющей вхождение в А, область H_A содержит соответствующую Эрбранову константу;

b. если А не содержит константы, то тем не менее H_A содержит Эрбранову константу, обозначаемую *c*;

c. для каждой *n* – местной функции *f*, имеющей вхождение в А, вводится функциональная константа *F*. Принадлежность $h_1,\ldots,h_n$ к H_A влечет принадлежность к H_A также $F(h_1,\ldots,h_n)$.

Пример.

Доказать, что *C* является логическим следствием гипотез H_1, H_2:

$$H_1 : \forall x (P(x) \supset Q(x))$$

$$H_2 : \forall x (Q(x) \supset R(x))$$

$$C : \ \forall x (P(x) \supset R(x))$$

На основании принципа дедукции это будет доказано, если

$$(H_1 \wedge H_2 \wedge \neg C)$$

невыполнимо.

Приведем формулу к КНФ.

$$\forall x((P(x) \supset Q(x)) \wedge (Q(x) \supset R(x)) \wedge \exists x \neg (P(x) \supset R(x)))$$

$$\forall x((\neg P(x) \vee Q(x)) \wedge (\neg Q(x) \vee R(x)) \wedge \exists x \neg (\neg P(x) \vee R(x)))$$

$$\forall x((\neg P(x) \vee Q(x)) \wedge (\neg Q(x) \vee R(x)) \wedge \exists x(\neg \neg P(x) \wedge \neg R(x)))$$

$$\forall x((\neg P(x) \vee Q(x)) \wedge (\neg Q(x) \vee R(x)) \wedge \exists y(\neg \neg P(y) \wedge \neg R(y)))$$

$$\exists y \forall x((\neg P(x) \vee Q(x)) \wedge (\neg Q(x) \vee R(x)) \wedge (P(y) \wedge \neg R(y)))$$

$$((\neg P(x) \vee Q(x)) \wedge (\neg Q(x) \vee R(x)) \wedge P(c) \wedge \neg R(c))$$

$$\Downarrow$$

$$((\neg P(c) \vee Q(c)) \wedge (\neg Q(c) \vee R(c)) \wedge P(c) \wedge \neg R(c))$$

Применим алгоритм резолюций и покажем невыполнимость полученной формулы

1. $\neg P(c) \vee Q(c)$
2. $\neg Q(c) \vee R(c)$
3. $P(c)$
4. $(\neg R(c)$
5. $Q(c)$ (1, 3)
6 $\neg Q(c)$ (2, 4)
7. Л (5, 6)

Более эффективный способ получается, если работать прямо с дизъюнктами без явного рассмотрения интерпретаций. Для этого используется операция **унификации**.

Комментарий к лекции 7

ДЕДУКТИВНЫЕ И ИНДУКТИВНЫЕ ЛОГИКИ

1. **Смысл дедуктивности логических систем**
2. **Примеры дедуктивных логик**
3. **Примеры индуктивных логик**

стандартная схема определения математической логики

множества $\Rightarrow$ *характеристическая функция* $\Rightarrow$ *абстракция (логическая переменная)* $\Rightarrow$ *(операции над множествами* $\Rightarrow$ *операциям над логическим переменным)* $\Rightarrow$ *построение логических систем и требования к ним*

$$X \Leftrightarrow \quad \mu_{\mathrm{X}}(x) = \begin{cases} 1, & x \in \mathrm{X} \\ 0, & otherwise \end{cases}$$

$$z \in \overline{\mathrm{X}} \Leftrightarrow \mu_{\overline{\mathrm{X}}}(z) = 1 - \mu_{\mathrm{X}}(z) = 1 \Leftrightarrow z \notin \mathrm{X}$$

$$z \in \mathrm{X} \cap \mathrm{Y} \Leftrightarrow \mu_{\mathrm{X}\cap\mathrm{Y}}(z) = \mu_{\mathrm{X}}(x) \wedge \mu_{\mathrm{Y}}(y) = 1 \Leftrightarrow x, y \in \mathrm{X} \cap \mathrm{Y}$$

$$z \in \mathrm{X} \cup \mathrm{Y} \Leftrightarrow \mu_{\mathrm{X}\cup\mathrm{Y}}(z) = \mu_{\mathrm{X}}(x) \vee \mu_{\mathrm{Y}}(y) = 1 \Leftrightarrow x, y \in \mathrm{X} \cup \mathrm{Y}$$

Построение логических систем

1. **Стандартная аксиоматическая система** $\Rightarrow$ доказательство “в смысле Гильберта”

аксиомы:

$$(A1) \qquad (X \supset (Y \supset X))$$

$$(A2) \qquad ((X \supset (Y \supset Z)) \supset ((X \supset Y) \supset (X \supset Z)))$$

$$(A3) \qquad ((\neg X \supset \neg Y) \supset ((\neg X \supset Y) \supset X))$$

правило вывода:

$$\textit{Если } X \textit{ и } (X \supset Y) - \textit{теоремы, то } Y - \textit{есть теорема}$$

2. **Система натурального вывода** $\Rightarrow$ доказательство “в смысле Генцена”

Форма записи правил:

$$\frac{S_1,\dots,S_n}{S}$$

Базисные правила (формализуют принцип монотонности рассуждений, вывода)

$$\frac{}{E, A \Rightarrow A} \quad и \quad \frac{E \Rightarrow A}{E, B \Rightarrow A}$$

Правила введения связок

(∧) $$\frac{E \Rightarrow A \quad E \Rightarrow B}{E \Rightarrow A \wedge B}$$

(∨) $$\frac{E \Rightarrow A}{E \Rightarrow A \vee B} \qquad \frac{E \Rightarrow B}{E \Rightarrow A \vee B}$$

(⊃) $$\frac{E, A \Rightarrow B}{E \Rightarrow A \supset B}$$

(¬) $$\frac{E, A \Rightarrow \text{Л}}{E \Rightarrow \neg A}$$

(≡) $$\frac{E, A \Rightarrow B \quad E, B \Rightarrow A}{E \Rightarrow A \equiv B}$$

Правила удаления связок

(∧) $$\frac{E \Rightarrow A \wedge B}{E \Rightarrow A} \qquad \frac{E \Rightarrow A \wedge B}{E \Rightarrow B}$$

(∨) $$\frac{E \Rightarrow A \vee B \quad E, A \Rightarrow C \quad E, B \Rightarrow C}{E \Rightarrow C}$$

(⊃) $$\frac{E \Rightarrow A \supset B}{E, A \Rightarrow B}$$

(¬) $$\frac{E \Rightarrow A \quad E \Rightarrow \neg A}{E \Rightarrow \text{Л}} \qquad \frac{E \Rightarrow \neg\neg A}{E \Rightarrow A}$$

(≡) $$\frac{E \Rightarrow A \equiv B}{E, A \Rightarrow B} \qquad \frac{E \Rightarrow A \equiv B}{E, B \Rightarrow A}$$

3. **Стандартная аксиоматическая система для квантификации** $\Rightarrow$ доказательство “в смысле Гильберта”

аксиомы:

$$(A1) \qquad (P \supset (Q \supset P))$$

$$(A2) \qquad ((P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R)))$$

$$(A3) \qquad ((\neg P \supset \neg Q) \supset ((\neg P \supset Q) \supset P))$$

$$(A4) \qquad \forall x (P \supset Q) \supset (P \supset \forall x Q)$$

правило вывода:

$$\textit{Если } X \textit{ и } (X \supset Y) - \textit{теоремы, то } Y - \textit{есть теорема}$$

правило обобщения:

Если x не связана в теореме P, то $\forall x P$ - теорема

4. **Система натурального вывода для квантификации** $\Rightarrow$ доказательство "в смысле Генцена"

к системе натурального вывода (пример 2) добавляются:

Правила введения кванторов

$$(\forall) \qquad \frac{E \Rightarrow A(x)}{E \Rightarrow \forall x\ A(x)}$$

$$(\exists) \qquad \frac{E \Rightarrow A(t)}{E \Rightarrow \exists x\ A(x)}$$

Правила удаления кванторов

$$(\forall) \qquad \frac{E \Rightarrow \forall x\ A(x)}{E \Rightarrow A(t)}$$

$$(\exists) \qquad \frac{E \Rightarrow \exists x\ A(x)}{E \Rightarrow A(c)}$$

Дедуктивность логических систем

Направления исследований логических систем:

- *выразительная сила языка*, используемого для построения логики;
- *сравнение логических систем* (в этом смысле логики в примерах 1, 2 и 3, 4 примерно одинаковы);
- *конструктивность системы* (существование и реализуемость, численные характеристики соответствующих алгоритмов вывода).

Общим для всех логических систем является возможность использования принципа (метатеоремы) дедукции:

Необходимым и достаточным условием выводимости $\boldsymbol{A}$ из гипотез (теорем) $\boldsymbol{E}, \boldsymbol{B}$ является выводимость ($\boldsymbol{B} \supset \boldsymbol{A}$) из $\boldsymbol{E}$

Формально: $E, B \vDash A \quad \Leftrightarrow \quad E \vDash (B \supset A)$

Принцип дедукции:

- является верным в полной системе типа 1, 2 и может оставаться верным в неполной системе;
- не применим к системам с квантификацией (типа 3, 4), если в выводе формулы A используется правило обобщения по переменной , имеющей свободное вхождение в B. (например – $P \vDash \forall x P$ является законным выводом, в отличие от $\vDash P \supset \forall x P$).

Следствием принципа дедукции является универсальный *механизм вывода* – метод резолюций, а также *разрешимость* формальных систем.

Дедуктивная логика и реальный мир: возможность обоснования правильности рассуждений и выводов

ЕИ задача

1. некоторые объекты и связи между ними. Число объектов и связей может быть произвольным
2. семантика функционирования этих объектов и взаимодействие по установленным связям
3. определенные цели функционирования задачи
4. методология анализа и/или синтеза может быть известна (если задача имеет решение) либо нет (в противном случае)

Модель формальной логики (ИИ)

1. язык логического символизма и правила построения формул (высказываний)
2. некоторый набор тавтологий
3. правила манипулирования формулами, по которым получаются новые формулы и только формулы
4. алгоритм, позволяющий осуществлять классификацию любой исследуемой формулы. В худшем случае этот алгоритм (резолюция) не сможет закончить работу
5. ограниченная интерпретация формул в пространстве {**истина, ложь**}

Формально соотношение между реальностью и логическими формализмами вряд ли может быть установлено. Наиболее правдоподобным является предположение:

как бы ни была построена дедуктивная логика (по числу аксиом, правил вывода и др. компонентов), она никогда не будет достаточной для моделирования ЕИ.

Возможные направления модификации дедуктивных логик

принципы построения логических систем

- для описания конкретной логики должны быть заданы пространства – исходное (где определены переменные) и результирующее (где определены значения операций над переменными). В нашем случае это пространство B^2;
- определены наборы исходных аксиом и правил вывода.

свойства логических систем

рефлексивность - $\{p_1,\dots, p_n, q\} \vDash q$

монотонность - $\{p_1,\dots, p_n\} \vDash q \Rightarrow \{p_1,\dots, p_n, r\} \vDash q$

транзитивность - $\{p_1,\dots, p_n\} \vDash r \wedge \{p_1,\dots, p_n, r\} \vDash q \Rightarrow \{p_1,\dots, p_n\} \vDash q$

направления возможной модификации логик:

- изменением B^2;
- манипуляцией над множеством аксиом либо правил вывода (их исключением или добавлением новых);
- описанием структур (через свойства).

Примеры:

- k – значная логика;
- интуиционистская логика;
- модальная логика.

$A \Rightarrow B$

Примеры индуктивных логик

1. Логика подтверждений Д.Пойа

	доказательная схема	***эвристические схемы***		
название	modus ponens	затушеванная доказательная	затушеванная индуктивная	индуктивная
посылки	$A \Rightarrow B$ *A* истинно	$A \Rightarrow B$ *A* более правдоподобно	$A \Rightarrow B$ *A* менее правдоподобно	$A \Rightarrow B$ *A* ложно
заключение	*B* истинно	*B* более правдоподобно	*B* несколько менее правдоподобно	*B* менее правдоподобно

2. Вероятностная логика

	доказательная схема	***эвристические схемы***		
название	modus ponens	затушеванная доказательная	затушеванная индуктивная	индуктивная
посылки	$A \Rightarrow B$ P(A) = 1	$A \Rightarrow B$ $\uparrow$ P(A)	$A \Rightarrow B$ $\downarrow$ P(A)	$A \Rightarrow B$ P(A) = 0
заключение	P(B) = 1	$\uparrow$ P(B)	$\downarrow$ P(B)	$\downarrow$ P(B)

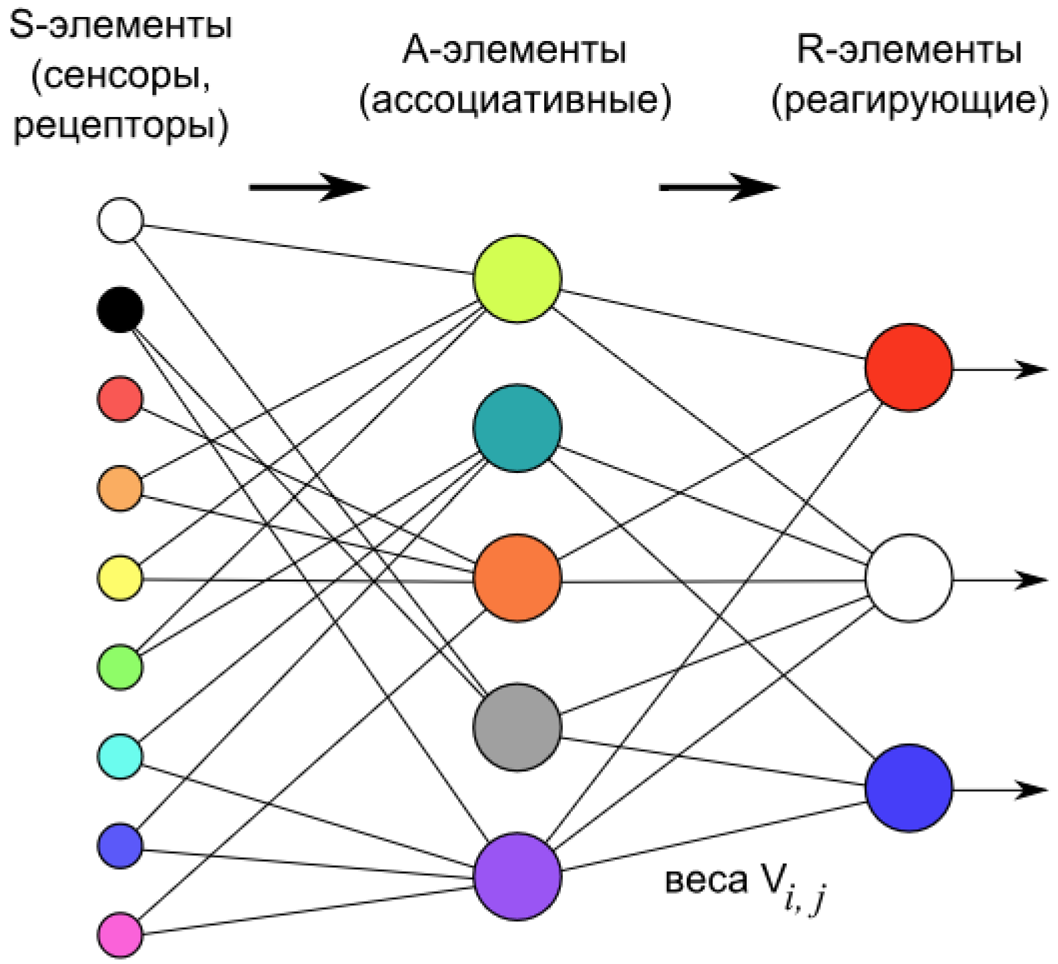
3. Аксиоматическое построение вероятностной логики

- Логика подтверждений Р.Карнапа;
- Логика подтверждений Х.Джефриса;
- Логика релевантности.

МОДЕЛИ РАСПОЗНАВАНИЯ ОБРАЗОВ

1. **История проблемы распознавания**
2. **Эволюция понимания задачи**
3. **Примеры**
4. **Общая схема решения задач**
5. **Формальная постановка задач распознавания**
6. **Связь между задачами распознавания с обучением и без**
7. **Что можно сказать о качестве**
8. **Простейшая алгоритмизация**

ПЕРСЕПТРОН РОЗЕНБЛАТТА



ИСТОРИЯ РАСПОЗНАВАНИЯ

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B = события

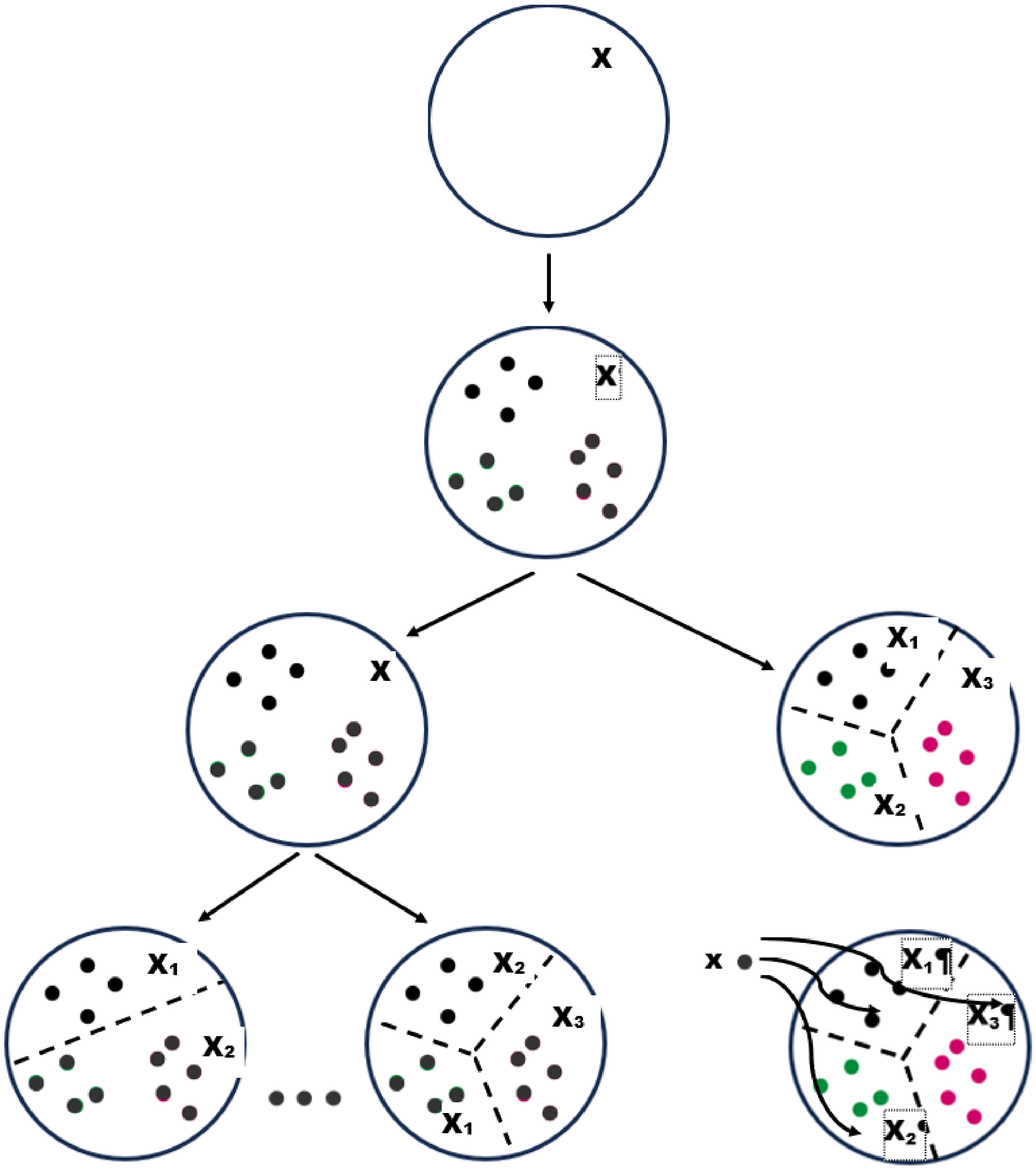
$P(A|B)$ = вероятность гипотезы А при наступлении события В

$P(B|A)$ = вероятность события В при наступлении события А

$P(A), P(B)$ = независимые вероятности А и Б

1. Алгоритмический этап
2. Модельный этап
3. Нейронные сети
4. Машинное обучение

ЭВОЛЮЦИЯ ПОНИМАНИЯ ЗАДАЧИ
РАСПОЗНАВАНИЯ ОБРАЗОВ



МНОЖЕСТВО

ПРЕДСТАВЛЕНО
ПРИМЕРАМИ

ДВА ВАРИАНТА
ЗАДАЧ

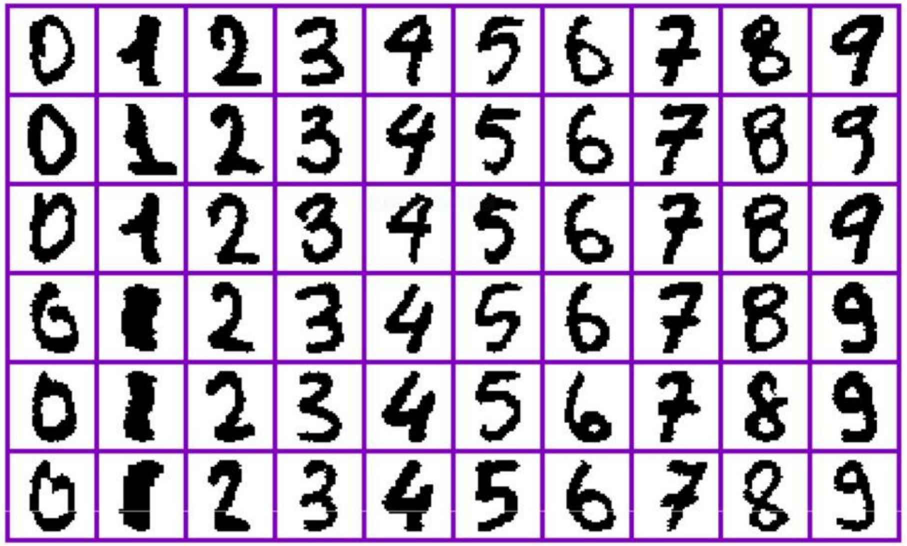
2. Задача распознавания свойства четности

$$
\begin{tikzcd}[row sep=small]
 & \mathbb{N}_{\text{ч}} \\
\mathbb{N} \arrow[ur] \arrow[dr] & \\
 & \mathbb{N}_{\text{нч}}
\end{tikzcd}
\qquad
\begin{cases}
\mathbb{N}_{\text{ч}} \cap \mathbb{N}_{\text{нч}} = \varnothing \\
\mathbb{N}_{\text{ч}} \cup \mathbb{N}_{\text{нч}} = \mathbb{N}
\end{cases}
$$

$$I(\mathbb{N}_{\text{ч}}, \mathbb{N}_{\text{нч}})$$

$$\forall n \in \mathbb{N} \quad A\colon n \times I(\mathbb{N}_{\text{ч}}, \mathbb{N}_{\text{нч}}) \to \textit{результат}$$

3. Задача распознавания арабских цифр

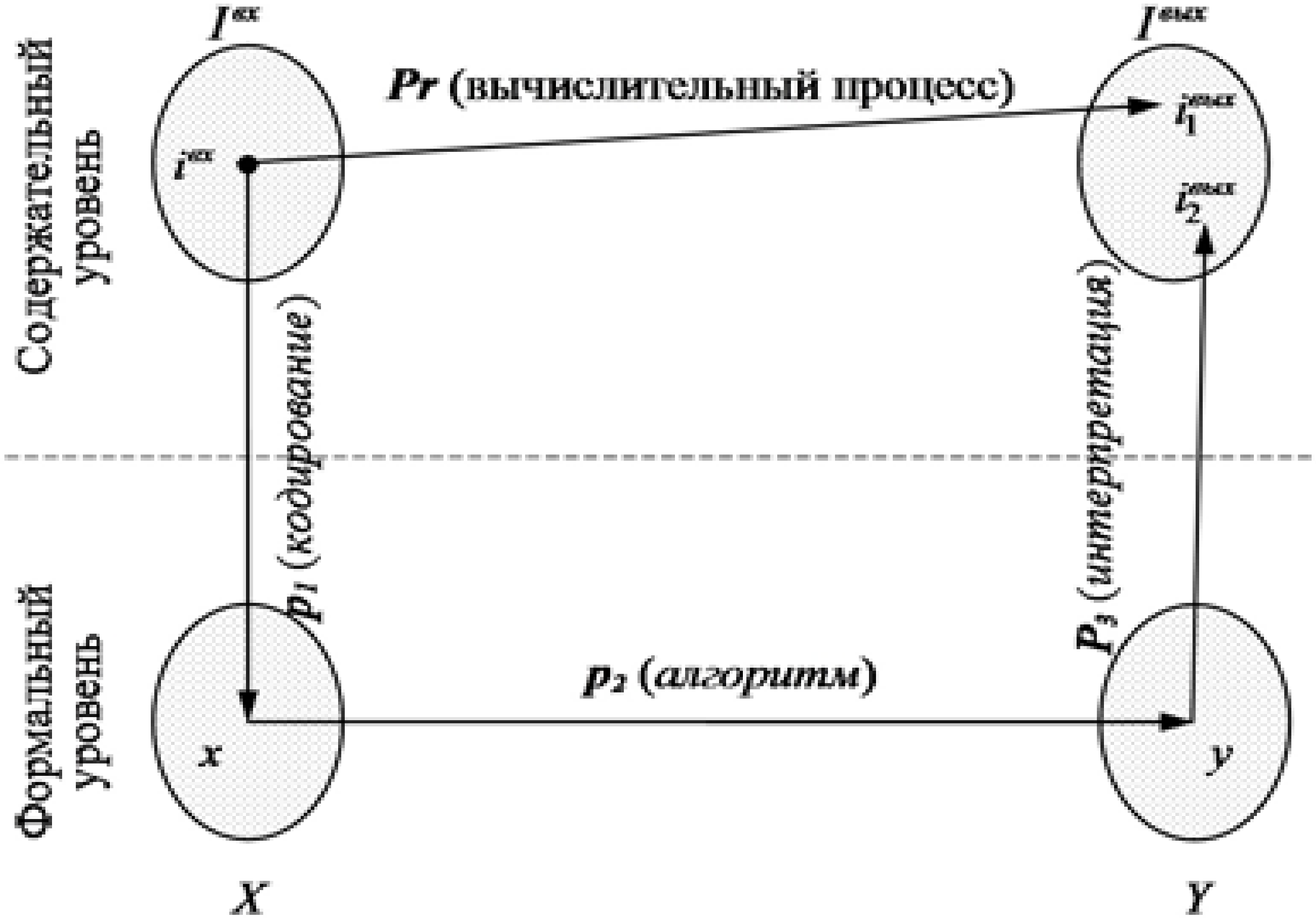


$$\left[\begin{array}{l} X_0,\ldots,X_9 \\ X_0 \cup \ldots \cup X_9 \subseteq X \end{array}\right.$$

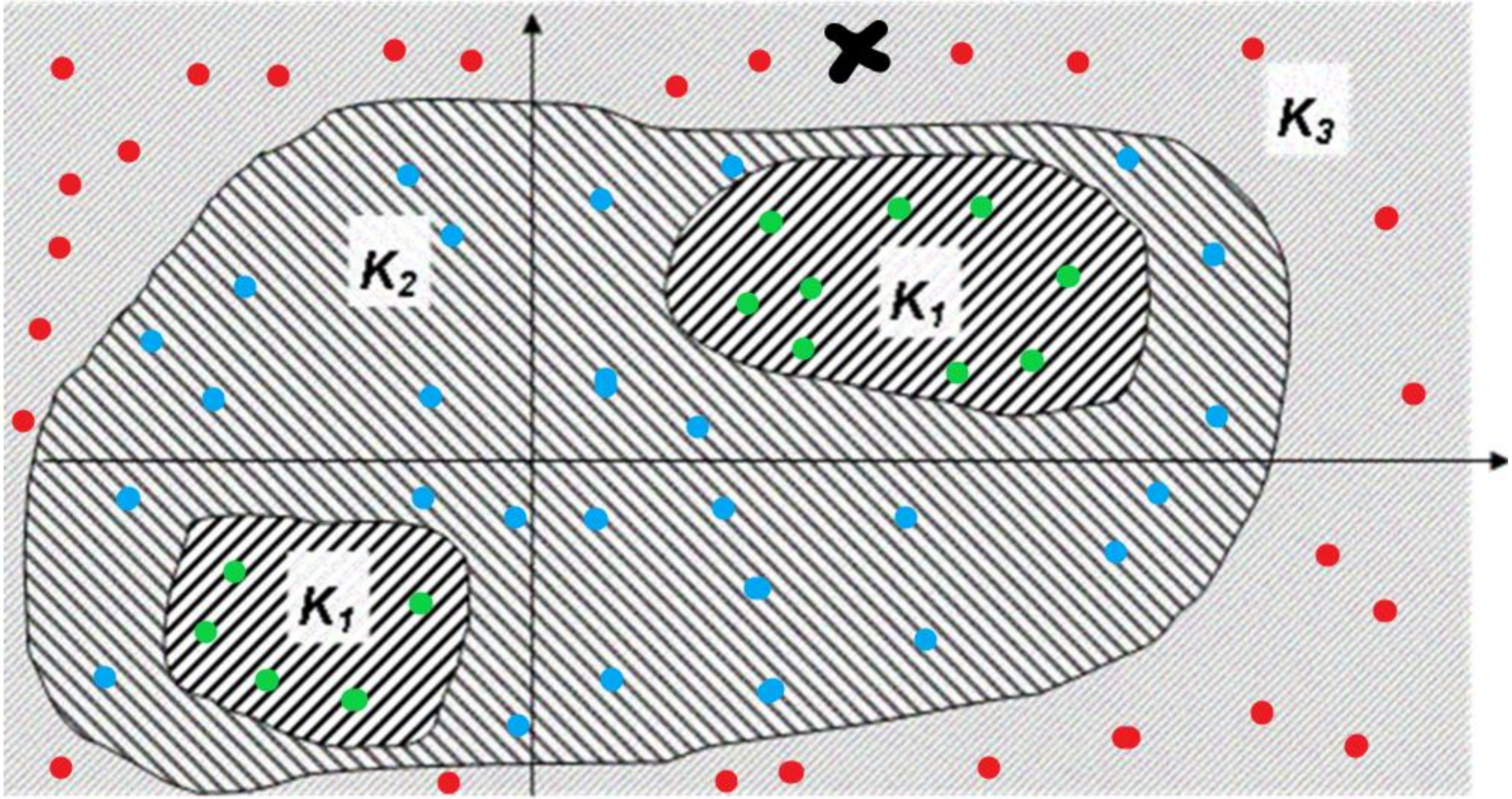
$$X^0 \quad I(X^0, X_0,\ldots,X_9)$$

$$\forall x \in X \quad A : x \times I(X^0, X_0,\ldots,X_9) \to p$$

ОБЩАЯ СХЕМА РЕШЕНИЯ ЗАДАЧ



В перечисленных задачах может быть выделено две задачи: **распознавания образов** и **классификации**.



Задача распознавания с обучением:

Дано: на множестве объектов X произвольной природы задано некоторое (возможно и бесконечное) число подмножеств (классов) $\boldsymbol{X}_1, \dots, \boldsymbol{X}_l$. Задана также некоторая информация о классах $\boldsymbol{I_0}(\boldsymbol{X}_0, \boldsymbol{X}_1, \dots, \boldsymbol{X}_l)$, где $\boldsymbol{X}_0$-конечная выборка объектов из X с известной принадлежностью к классам $\boldsymbol{X}_1, \dots, \boldsymbol{X}_l$. Информация $\boldsymbol{I_0}$ конечна ($|\boldsymbol{I_0}| < \infty$).

Требуется: Требуется, используя только информацию $\boldsymbol{I_0}$, указать алгоритм $\boldsymbol{A}$ (возможно наилучший в некотором смысле), который определен на всем множестве X и результат работы которого для каждого $\boldsymbol{x} \in \boldsymbol{X}$ можно интерпретировать в терминах принадлежности классам $\boldsymbol{X_i}$ $(\boldsymbol{i} \in \mathbb{N})$.

Замечания:

1. Для уточнения постановки задачи распознавания с обучением обычно используют следующие дополнительные обозначения. Через $(P_1, \dots, P_l)$ и $(P_1^A, \dots, P_l^A)$ обозначают системы предикатов вида: $P_i: X \to \{0,1\}$ и $P_i^A: X \to \{0,1\}$, которые определяются условиями:

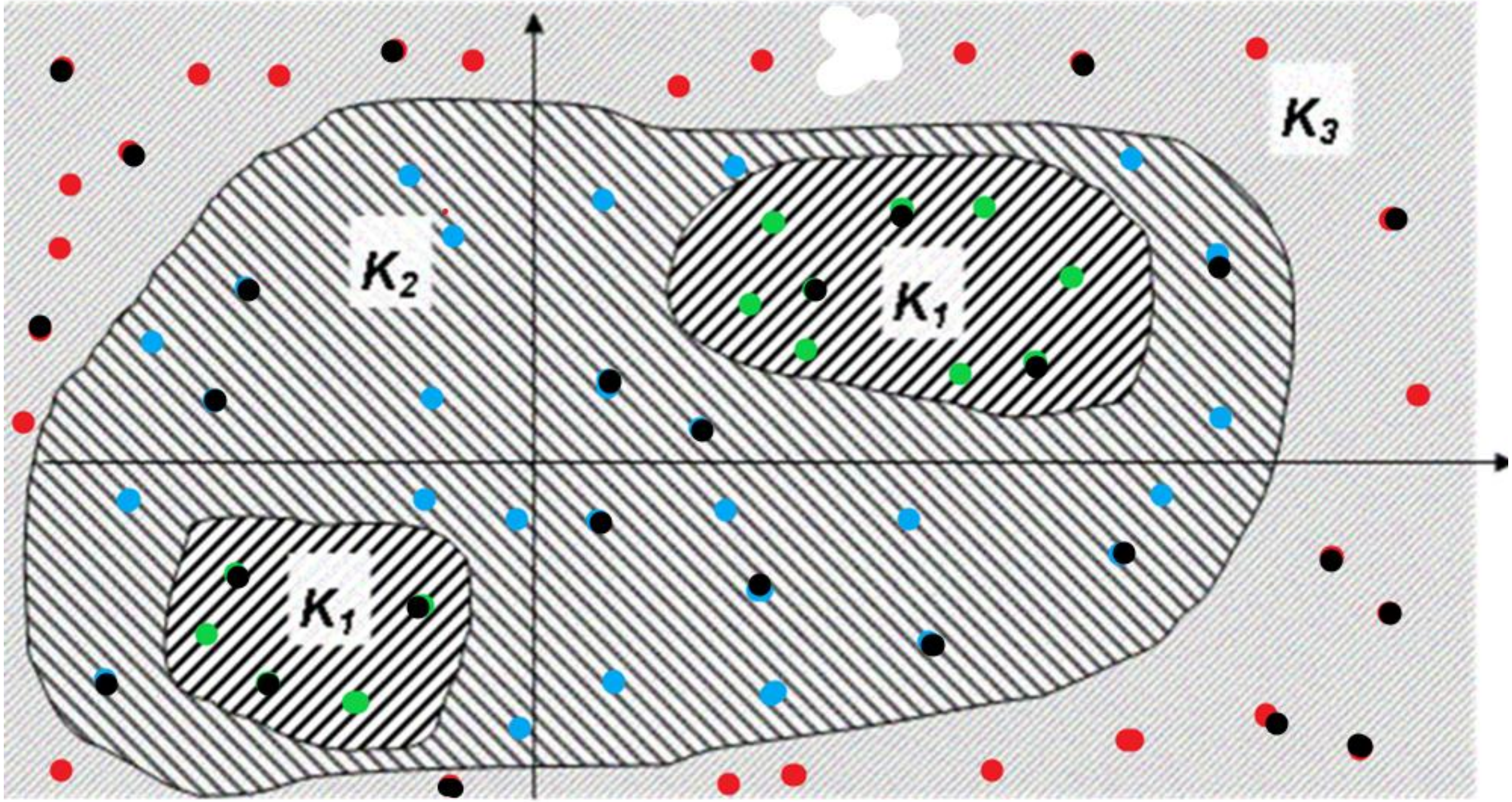
$$P_i(x) = \begin{cases} 1, \text{если } x \in X_i \\ 0, \text{иначе.} \end{cases}$$

$$P_i^A(x) = \begin{cases} 1, \text{если } x \text{ заносится алгоритмом } A \text{ в класс } X_i \\ 0, \text{иначе.} \end{cases}$$

2. В постановке задачи от алгоритма $\boldsymbol{A}$ требуется, чтобы он был “наилучшим в некотором смысле”. Очевидно, что это требование можно трактовать по-разному. Одна из таких трактовок может быть следующей: $P_i(x) = P_i^A(x) \quad \forall x \in X$. Очевидно, что левая часть этого соотношения известна только на заданном подмножестве $\boldsymbol{X}_0$. Если при выборе наилучшего алгоритма ограничиваться только $\boldsymbol{X}_0$, то искомым будет переборный алгоритм. Чтобы его исключить, выборку $\boldsymbol{X}_0$ разбивают на две подвыборки - $X_0^{\text{обуч}}$ и $X_0^{\text{контр}}$.

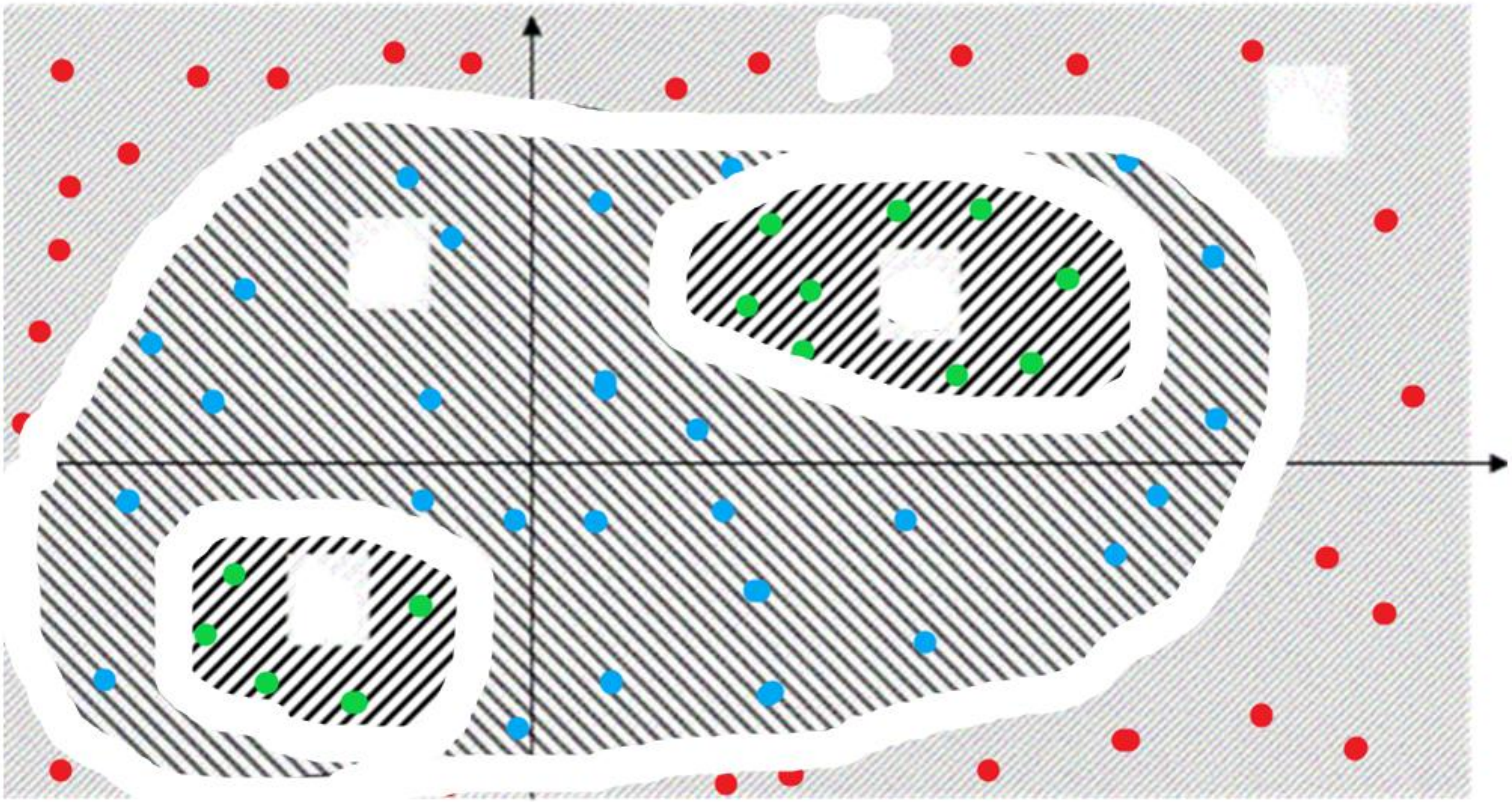
Задача распознавания с обучением: **(продолжение)**

Ограничения: выборку $\boldsymbol{X}_0$ разбивают на две непересекающиеся части - $X_0^{\text{обуч}}$ и $X_0^{\text{контр}}$. При построении алгоритма используют только ту часть $\boldsymbol{I_0}$, которая имеет отношение к $X_0^{\text{обуч}}$. При выборе “наилучшего в некотором смысле” алгоритма ***A*** исходят из оптимизации соотношения $P_i(x) = P_i^A(x)\ \forall x \in X_0^{\text{контр}}$.



K_3
K_2
K_1
K_1

Задача распознавания без обучения

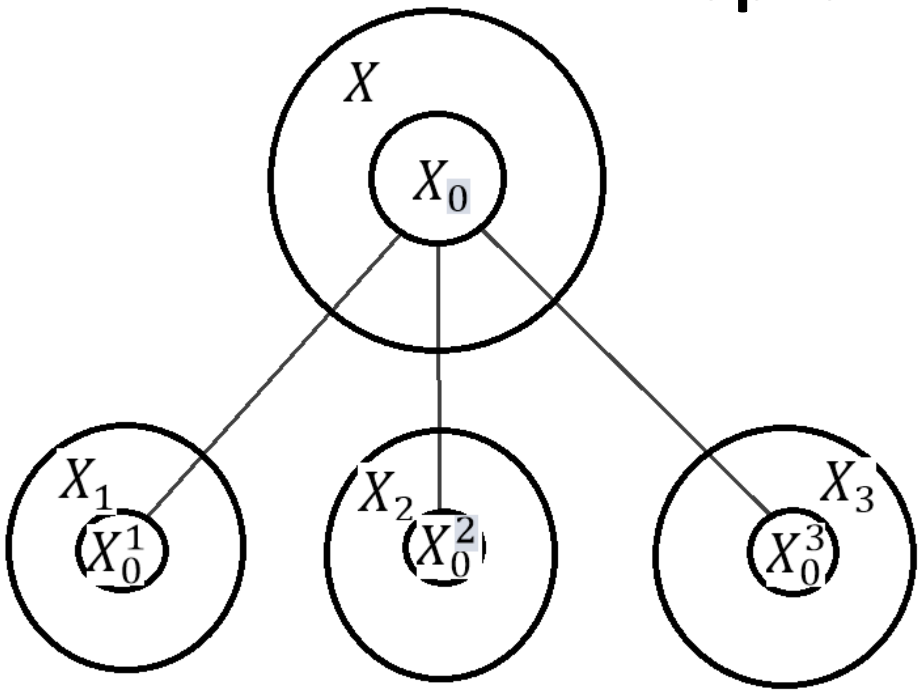


Задача распознавания без обучения:

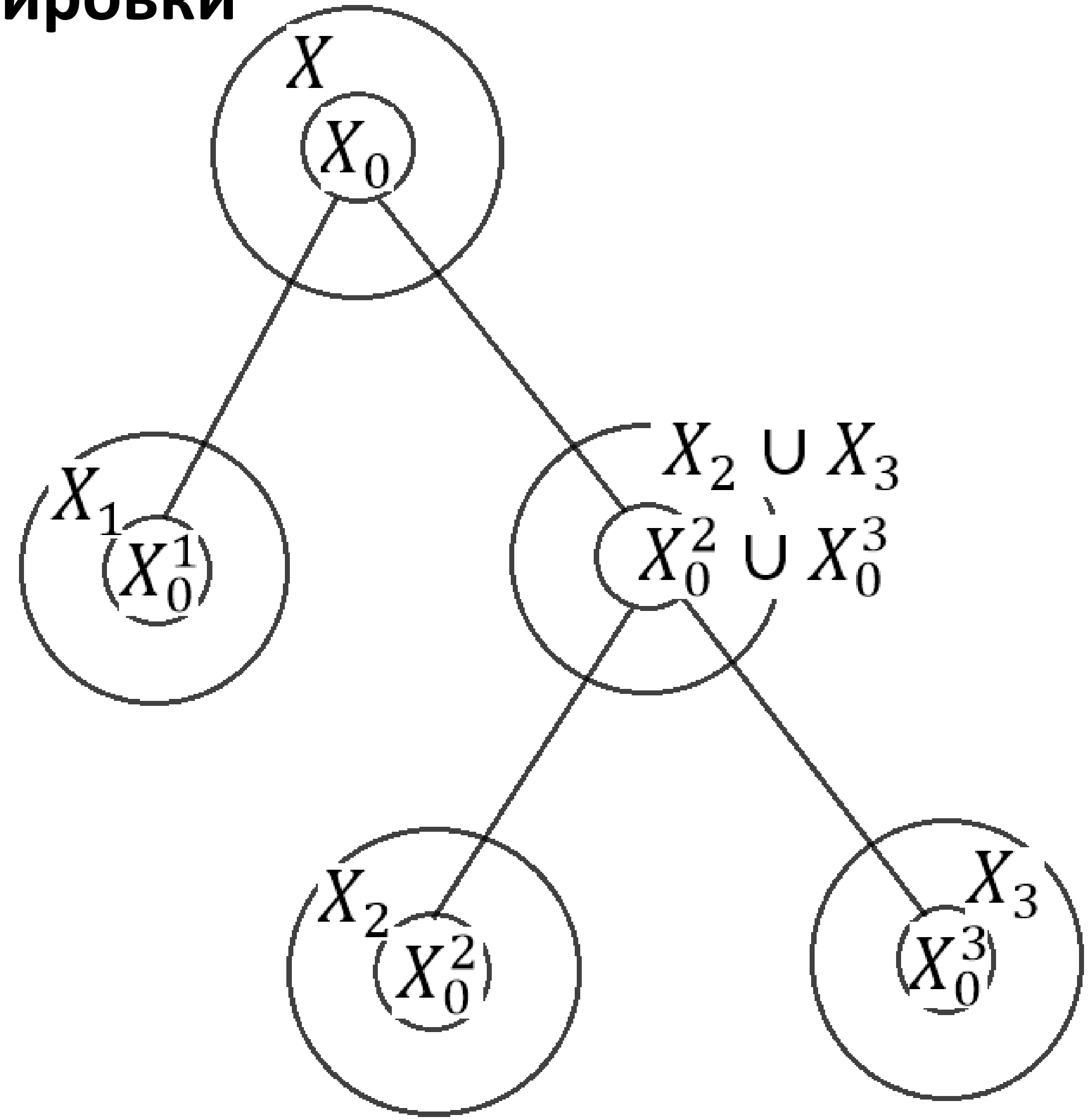
Дано: множество объектов X произвольной природы. Задана также некоторая информация о множестве $\boldsymbol{I_0}(\boldsymbol{X}_0)$. Информация $\boldsymbol{I_0}$ конечна ($|\boldsymbol{I_0}| < \infty$). Известно, что множество X может быть разбито на некоторое (конечное) число подмножеств (классов) $\boldsymbol{X}_1, \dots, \boldsymbol{X}_l$.

Требуется: Требуется, используя только информацию $\boldsymbol{I_0}$, указать алгоритм ***A*** (возможно наилучший в некотором смысле), который определен на всем множестве X и результат работы которого можно интерпретировать в терминах разбиения X на подмножества (классы) $\boldsymbol{X}_1, \dots, \boldsymbol{X}_l$.

Варианты группировки

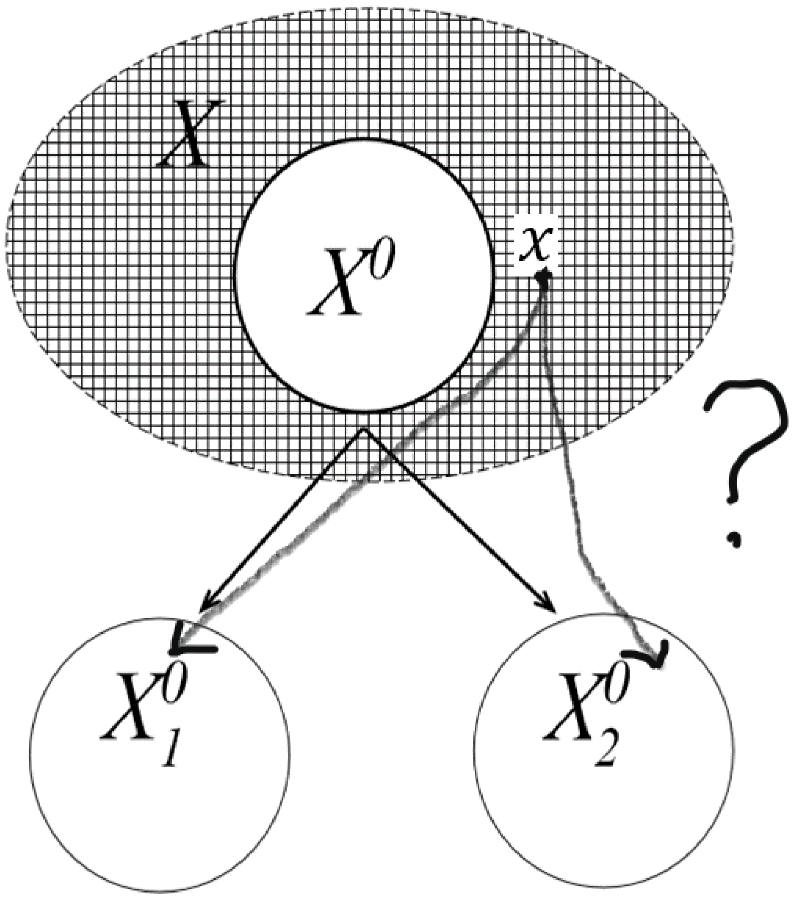


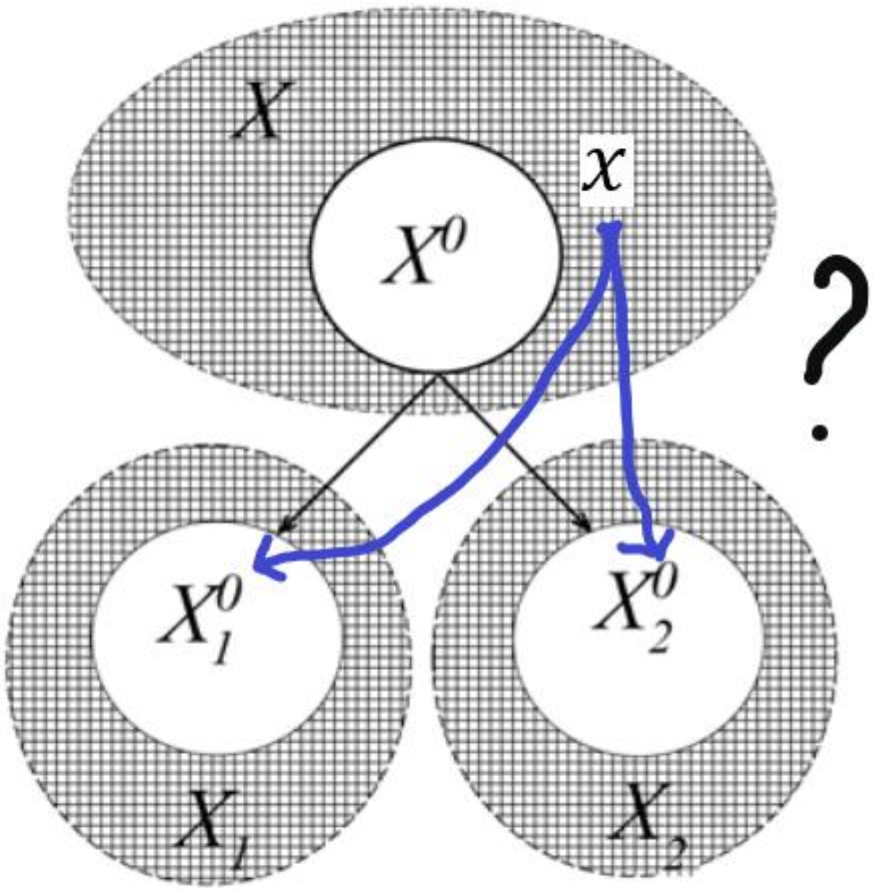
1 вариант



2 вариант

Связь между задачами/алгоритмами распознавания





Качество решения задачи распознавания с обучением

$$\mathbf{X}_i^{\mathbf{A}} := \left\{ x \in \mathbf{X} \;\middle|\; \mathbf{P}_i^{\mathbf{A}}(x) = 1 \right\}.$$

$$\mu : \mathbf{X}_i \times \mathbf{X}_i^{\mathbf{A}} \to [0,1], \qquad \psi : [0,1]^l \to [0,1]$$

$$\mu(\mathrm{X}_i, \mathrm{X}_i^{\mathrm{A}}) = \begin{cases} 1, & if \quad \mathrm{X}_i = \mathrm{X}_i^{\mathrm{A}}, \\ 0, & if \quad \mathrm{X}_i \cap \mathrm{X}_i^{\mathrm{A}} = \varnothing, \end{cases} \qquad \psi(a_1,\ldots,a_l) = \begin{cases} 1, & if \quad a_1 = \ldots = a_l = 1, \\ 0, & if \quad a_1 = \ldots = a_l = 0. \end{cases}$$

$$\boldsymbol{\Phi}_{\mathbf{A}}(\mathbf{X}) := \psi(\mu(\mathbf{X}_1, \mathbf{X}_1^{\mathbf{A}}), \ldots, \mu(\mathbf{X}_l, \mathbf{X}_l^{\mathbf{A}})) \quad \Longleftrightarrow \quad \Phi^{\mathrm{A}}(\boldsymbol{X}_0)$$

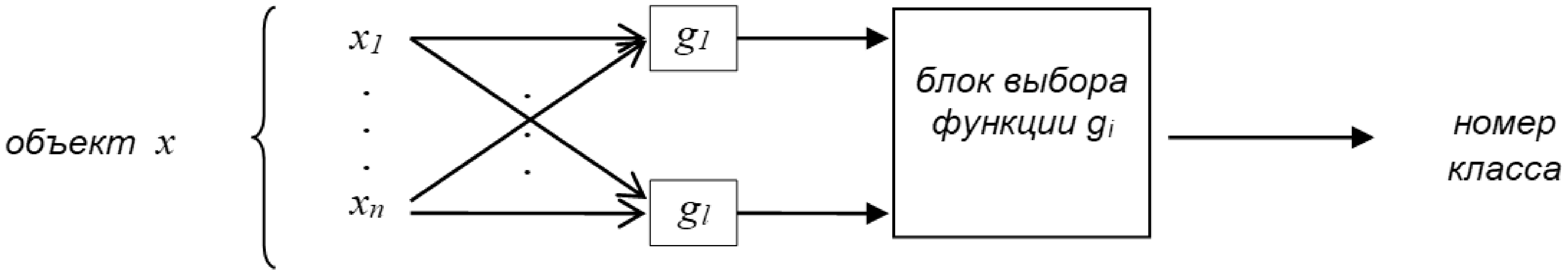
$$\forall\, \mathbf{A} \quad (\boldsymbol{\Phi}_{\mathbf{A}}(\mathbf{X}) < 1 \;\Leftrightarrow\; \exists\, \mathbf{X}^0 \;\; \boldsymbol{\Phi}_{\mathbf{A}}(\mathbf{X}^0) < 1)$$

Confusion matrix (матрица ошибок)

	Истинная классификация	
Предсказанный класс	класс 1	класс 2
класс 1	True Positive (*tp*)	False Positive (*fp*)
класс 2	False Negative (*fn*)	True Negative (*tn*)
	tp+ *fn* = общее число объектов I класса	*fp*+ *tn* = общее число объектов II класса

№	Обозначение функции	Формула
1	False positive rate (FPR), ложноположительный коэффициент	$\frac{fp}{fp + tn}$
2	False negative rate (FNR), ложноотрицательный коэффициент	$\frac{fn}{fn + tp}$
3	Sensitivity и Recall, чувствительность	$\frac{tp}{tp + fn}$
4	Specificity, специфичность	$\frac{tn}{tn + fp}$
5	Precision, точность	$\frac{tp}{tp + fp}$
6	Accuracy, правильность	$\frac{tp + tn}{n}$
7	индекс Жаккара (Jaccard)	$\frac{tp}{tp + fn + fp}$

Алгоритмы A строятся на базе следующей гипотезы:



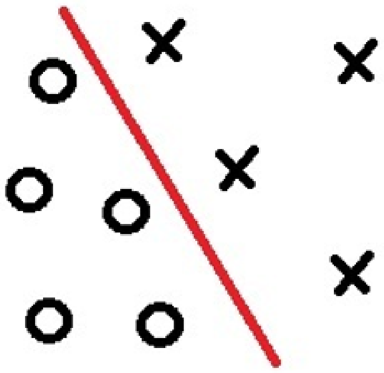
Выбор функции g_i может быть сделан следующим образом:

$$\forall x \in R^n \quad (A(x)=i \Leftrightarrow g_i(x)=\max_{j\in\{1,\dots,l\}}\{g_j(x)\}).$$

$\{x \mid x\in R^n,\ g_i(x)>g_j(x),\ j\neq i\}$ - область решения для класса K_i

$\{x \mid x\in R^n,\ g_i(x)=g_j(x)\}.$ - разделяющая гиперплоскость

ДЕТЕРМИНИСТСКИЕ АЛГОРИТМЫ



$$\exists g_1, g_2 \ (g_i : X \to \{0,1,2\}),$$

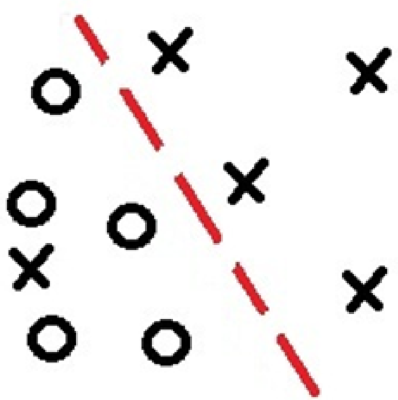
$$g_1(x) = \begin{cases} 1, x \in K_+ \\ 0, \textit{иначе} \end{cases} \quad g_2(x) = \begin{cases} 2, x \in K_\circ \\ 0, \textit{иначе} \end{cases}$$

$$\exists \ g_1, g_2 \ (g_i : X \to R),$$

$$g_1(x) - g_2(x) = 0,$$

$$A(x) = \begin{cases} 1, g_1(x) - g_2(x) > 0 \\ 2, g_1(x) - g_2(x) < 0 \\ 0, \textit{иначе} \end{cases}$$

Контрпример:



$$X \mapsto X' = (X, Y),\ Y = \{1, -1\}$$

Линейные алгоритмы распознавания

$$K_1, \dots, K_l$$

$$\exists g_1, \dots, g_l \qquad g_j = \sum_{i=1}^{n} a_i x_i + a_{n+1}, \qquad j = 1, \dots, l \qquad g_j = \max_t \{g_t(x)\}$$

Исходя из вида функций $g_1,\dots,g_l$, уравнение границы между классами K_j и K_t имеет вид

$$g_{jt}(x)=g_j(x)-g_t(x)=\sum_{i=1}^{n} b_i\, x\ +b_{n+1}=0$$

$$\begin{cases} b_1 x_1^1+\dots\dots\dots\dots\dots\dots+b_n x_n^1+b_{n+1}>0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ b_1 x_1^m+\dots\dots\dots\dots\dots\dots+b_n x_n^m+b_{n+1}>0 \\ b_1(-x_1^{m+1})+\dots\dots+b_n(-x_n^{m+1})-b_{n+1}>0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ b_1(-x_1^k)+\dots\dots\dots+b_n(-x_n^k)-b_{n+1}>0 \end{cases}$$

СТАТИСТИЧЕСКИЕ АЛГОРИТМЫ РАСПОЗНАВАНИЯ

Это возможно в случае, когда все элементы задачи наделяются вероятностными характеристиками.

Байесовский классификатор:

Предположим, что:

1. известна или может быть оценена априорная вероятность класса K_j: $p(K_j), j=1,...,l$ (т.е. вероятность наблюдения объекта из класса K_j);
2. функция плотности вероятности $p(x|K_j)$ известна или может быть оценена;
3. апостериорная вероятность $p(K_j|x)$ неизвестна.

Математическое ожидание потерь, связанных с решением $x \in K_j$:

$$r_j(x) = \sum_{i=1}^{l} a_{ij}\, p(K_i \mid x)$$

Алгоритм:

шаг 1. для объекта x вычисляем значения условного риска $r_1(x), \ldots, r_l(x)$.
шаг 2. объект x заносим в класс K_j , при условии

$$r_j(x) = \min_i \{r_i(x)\}$$

Называется этот алгоритм обычно ***байесовским классификатором***.

Величины $p(K_j \mid x)$ вычисляются с использованием формулы Байеса

$$p(K_j \mid x) = (p(K_j) \cdot p(x \mid K_j)) \cdot (p(x))^{-1}$$

Тогда

$$r_j(x) = (p(x))^{-1} \cdot \sum_{i=1}^{l} a_{ij} \cdot (p(K_i) \cdot p(x \mid K_i))$$

ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ДЕТЕРМИНИСТСКИХ АЛГОРИТМОВ

1. 0сновные идеи алгоритмизации
2. Типы алгоритмов
3. Общая схема алгоритмов, основанных на похожести
4. Виды похожести
5. Реализация функций похожести

ОСНОВНЫЕ ИДЕИ АЛГОРИТМИЗАЦИИ

1. **Задача должна быть разрешимой**

$$P(x) = \big(P_1(x), \dots, P_l(x)\big) \qquad\qquad P_i(x) = \begin{cases} 1, \text{если } x \in X_i \\ 0, \text{иначе.} \end{cases}$$

Задача узнавания объектов называется **разрешимой,** если:

$$\exists g\colon X \to \mathbb{B}_2^l \quad \forall x \in X \ (P_i(x) = 1 \Longrightarrow g(x) = y \wedge (y_j = 1 \Leftrightarrow j = i)).$$

Гипотеза:

Задача распознавания образов с обучением является **разрешимой** в том и только том случае, если

$$\exists g: X \to Y \quad \forall x_1, x_2 \in X \quad \big(P(x_1) \neq P(x_2) \Rightarrow g(x_1) \neq g(x_2)\big).$$

Как описывается множество X и функции g

Языки = математические формализмы

Детерминистские

Вероятностные

Нечеткие

2. **Вид функции $\boldsymbol{g}$: $\boldsymbol{X} \to \boldsymbol{Y}$**

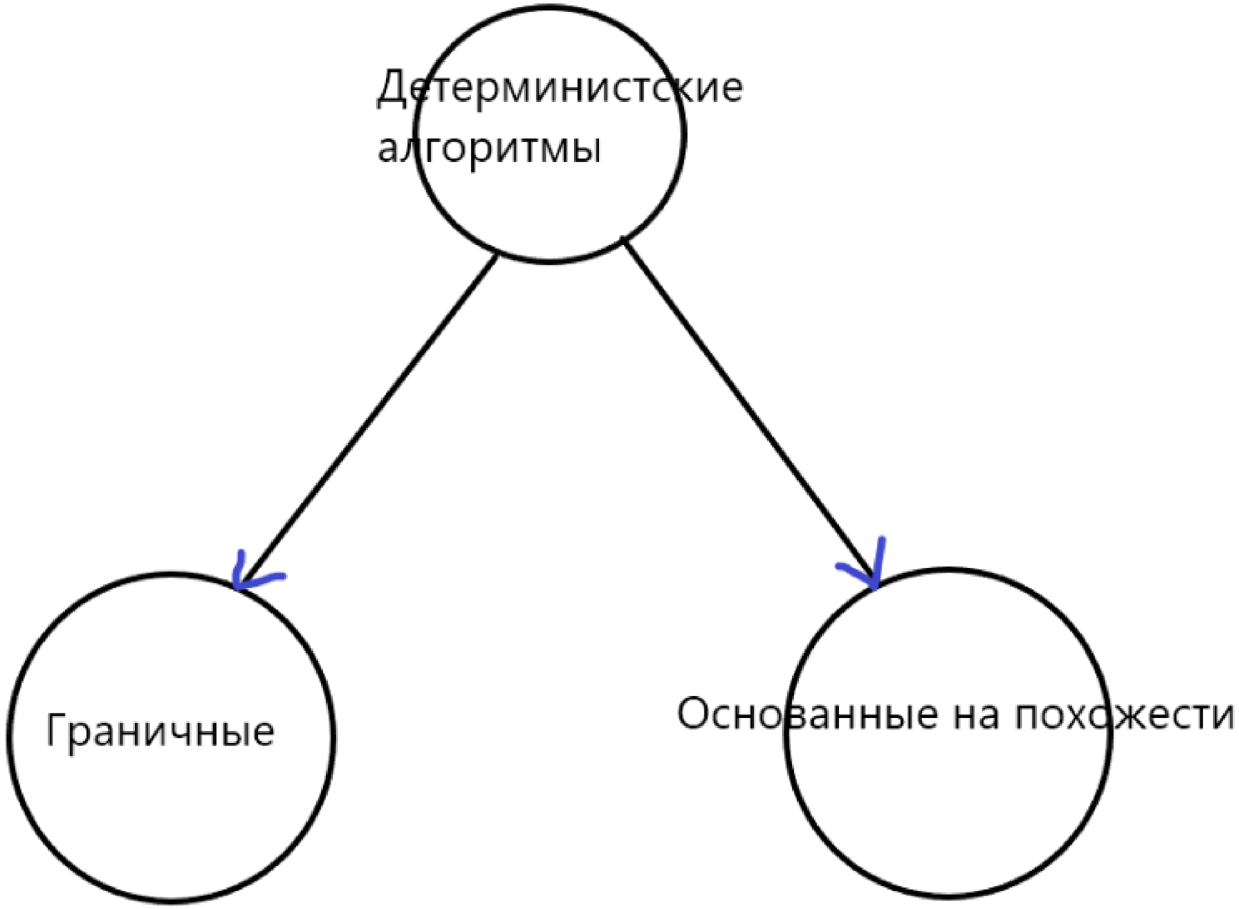
Удобнее всего функцию g строить (искать) в виде:

$$g = (g_1, \dots, g_l)$$

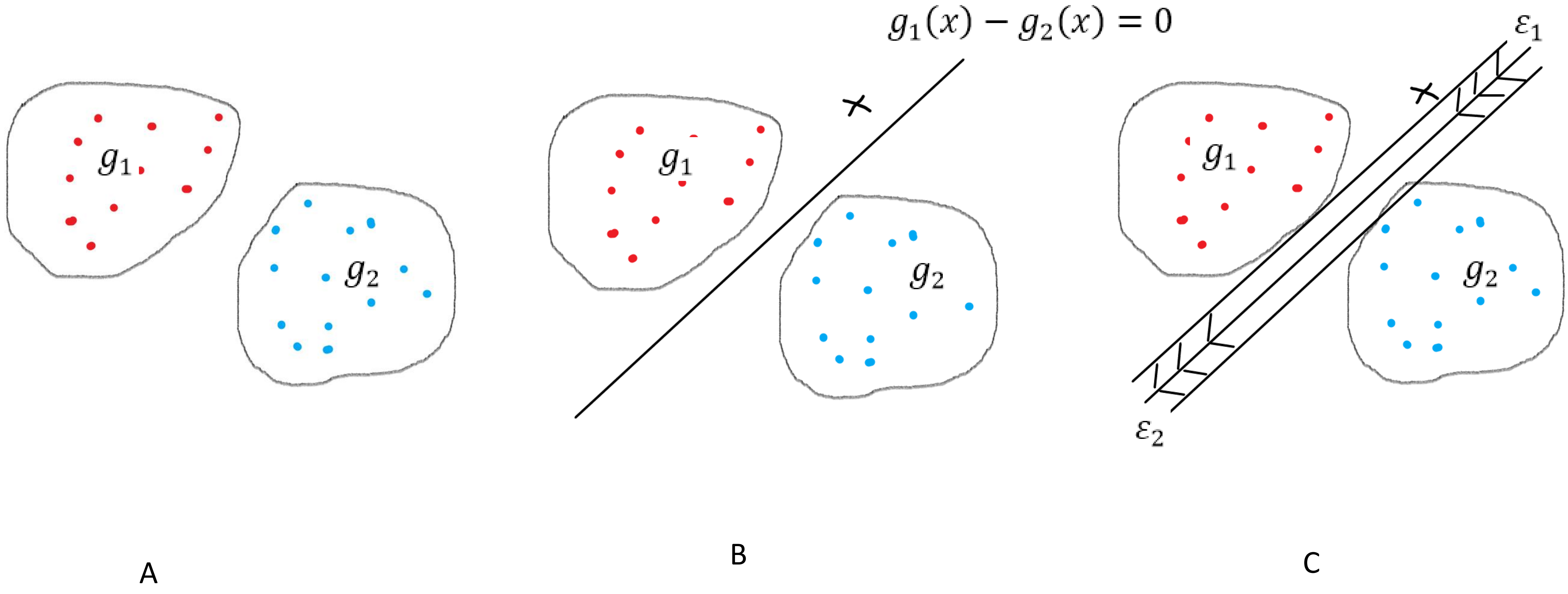
3. **Функция $\boldsymbol{g}$ непрерывна?**

“Схожие” объекты в пространстве X должны при использовании функции $g = (g_1, \dots, g_l)$ приводить к “схожим” результатам в пространстве Y

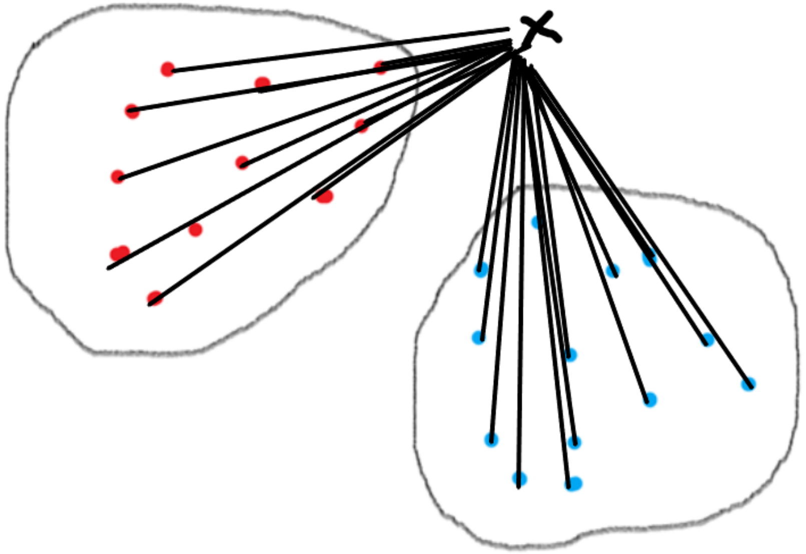
ТИПЫ ДЕТЕРМИНИСТСКИХ АЛГОРИТМОВ



Пример граничного алгоритма (линейного)



Основная идея алгоритма похожести



Для обучающей выборки:

$$\begin{bmatrix} (X_1, X_1) & \cdots & (X_1, X_l) \\ \vdots & \ddots & \vdots \\ (X_l, X_1) & \cdots & (X_l, X_l) \end{bmatrix}$$

Общая схема алгоритмов похожести

Вход: объект x

↓

Сравнение объекта x с объектами из выборки $X_0^{\text{обуч}}$

↓

Определение похожести объекта x с объектами из класса $\boldsymbol{X}_i$

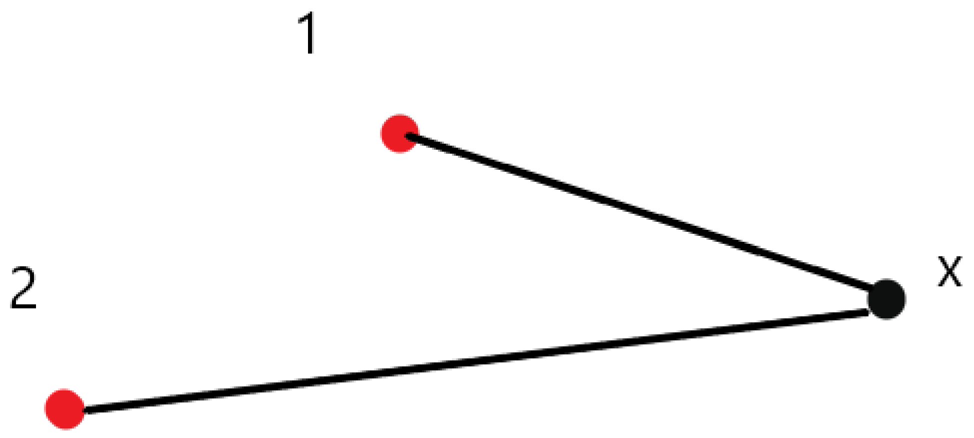
↓

Вычисление принадлежности объекта x к классу $\boldsymbol{X}_i$

Метрическая близость

$$s: X \times X \to \mathbb{R}$$

$$\forall x_1, x_2, x_3 \in X \begin{cases} s(x_1, x_2) = 0 \ \Leftrightarrow x_1 = x_2; \\ s(x_1, x_2) = s(x_2, x_1); \\ s(x_1, x_2) \leq s(x_1, x_3) + s(x_3, x_2). \end{cases}$$



– метрика Евклида ($X \subseteq \mathbb{R}^n$)

$$s(x_1, x_2) = (\sum_{i=1}^{n} (x_{1i} - x_{2i})^2)^{1/2}$$

– метрика Хэмминга ($X \subseteq \mathbb{B}^n$)

$$s(x_1, x_2) = \sum_{i=1}^{n} |x_{1i} - x_{2i}|$$

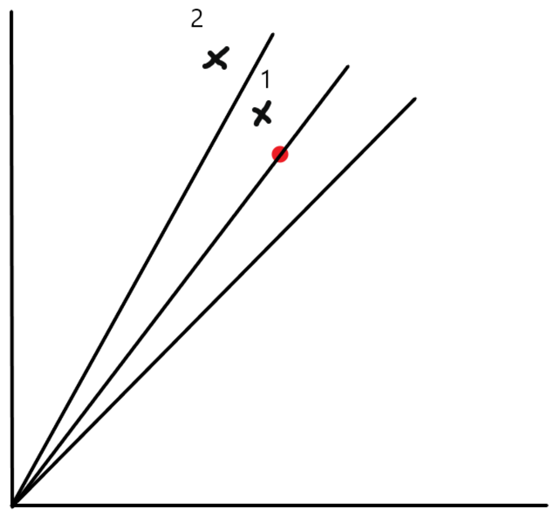
– метрика Минковского ($X \subseteq \mathbb{R}^n, p \in \mathbb{N}$)

$$s(x_1, x_2) = (\sum_{i=1}^{n} (x_{1i} - x_{2i})^p)^{1/p}$$

Метрическое подобие

$s: X \times X \to \mathbb{R}$

$$\forall x_1, x_2, x_3 \in X \begin{cases} s(x_1, x_2) = 1 \iff x_1 = x_2; \\ s(x_1, x_2) = \left(s(x_2, x_1)\right)^{-1}; \\ s(x_1, x_2) = s(x_1, x_3) \times s(x_3, x_2). \end{cases}$$



Пример: (для случая_ $x_1, x_2 \in \mathbb{R}_+^n$, $\mathbb{R}_+ \stackrel{\text{def}}{=} \{x | x \in \mathbb{R}, x > 0\}$) Если условие не выполняется, то описанный ниже метод не применим.

Инициализация: Полагаем ниже по определению $n + 1 \stackrel{\text{def}}{=} 1$ и вычисляем

$$s(x_1, x_2) = (\prod_{k=1}^{n} \frac{\max\{x_{1k}, x_{1k+1}\} \cdot (\min\{x_{1k}, x_{1k+1}\})^{-1}}{\max\{x_{2k}, x_{2k+1}\} \cdot (\min\{x_{2k}, x_{2k+1}\})^{-1}})^{1/n}$$

Метрическая прецедентность

$s: X \times X \to \mathbb{R}$

$$\forall x_1, x_2 \in X \begin{cases} s(x_1, x_2) = 1 \Leftrightarrow x_1 = x_2; \\ s(x_1, x_2) = s(x_2, x_1); \\ s(x_1, x_2) + s(x_1, \bar{x}_2) = 0. \end{cases}$$

$\bar{x}$ - отрицание объекта x (+)

В силу условия (+) метрическая прецедентность определена только в пространстве $\mathbb{B}^2$ ($\mathbb{B} = \{0,1\}$)

Пример. Для случая $x_1, x_2 \in \mathbb{B}_2^n, \mathbb{B}_2 = \{0,1\}$,
функция s может быть определена следующим образом:

$$s(x_1, x_2) = \left(\sum_{j=1}^{n} a_{ij}\right)^{-1} \times (\sum_{j=1}^{n} (-1)^u \times a_{ij}) \Bigg)$$

$$u = \begin{cases} 1, \text{если } x_{1j} \neq x_{2j} \\ 2, \text{иначе} \end{cases}$$

$$(a_1, \ldots, a_n) \in \mathbb{R}_+^n$$

Упражнения:

Показать, что μ – метрика:

$$\mu(x_1, x_2) = 1 - \min\{s(x_1, x_2), s(x_2, x_1)\}$$ (s - подобие)

$$\mu(x_1, x_2) = 1 - s(x_1, x_2)$$ (s - прецедентность)

Комментарий к лекции 9

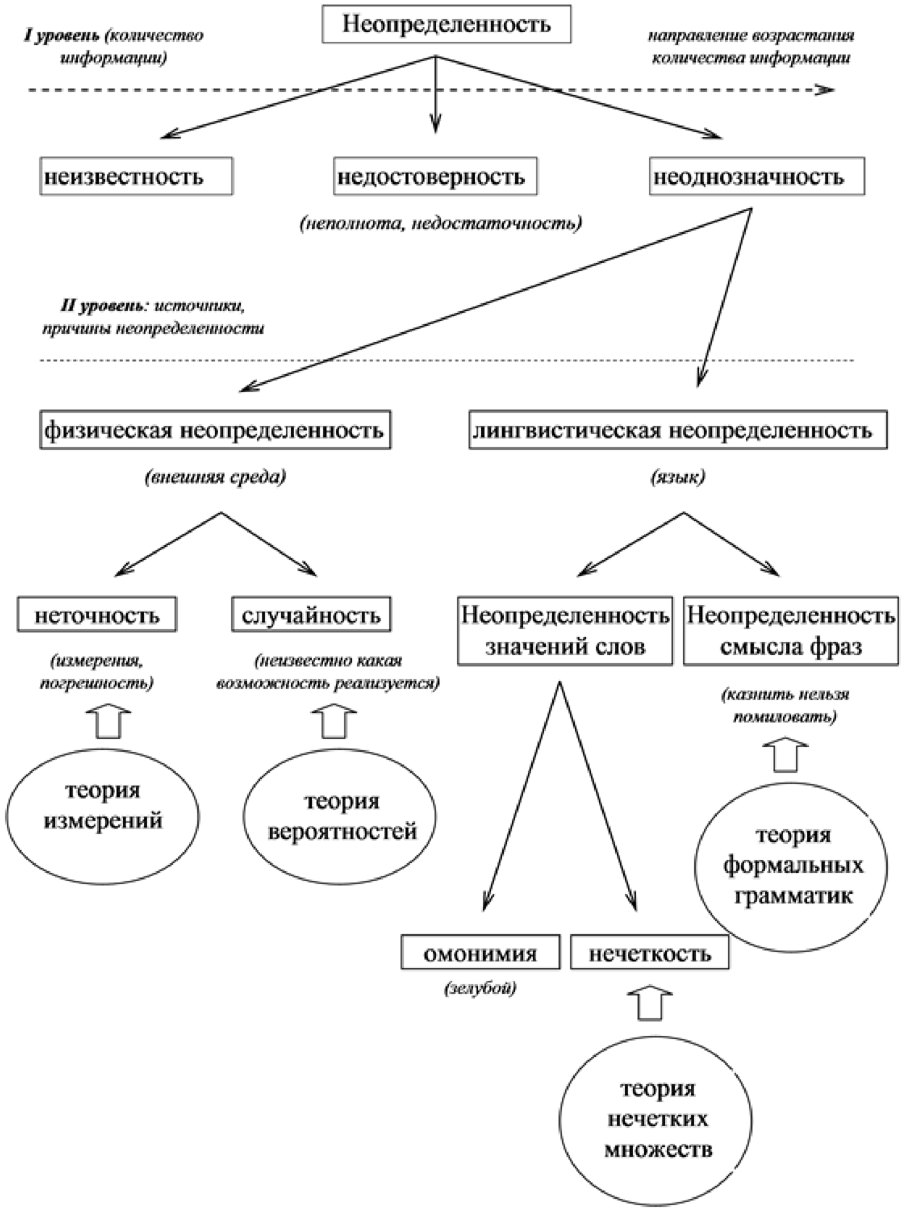
НЕЧЕТКАЯ МАТЕМАТИКА

1. **Необходимость нечеткой математики**
2. **Формализация языка**
   - **основные нечеткие конструкции**
   - **принципы нечеткой математики**
3. **Нечеткая математика в ИИ**
4. **Примеры использования**

Необходимость нечеткой математики – **неопределенность** среды существования ЕИ

1. Логика подтверждений Д.Пойа

	доказательная схема	***эвристические схемы***		
название	modus ponens	затушеванная доказательная	затушеванная индуктивная	индуктивная
посылки	$A \Rightarrow B$ *A* истинно	$A \Rightarrow B$ *A* более правдоподобно	$A \Rightarrow B$ *A* менее правдоподобно	$A \Rightarrow B$ *A* ложно
заключение	*B* истинно	*B* более правдоподобно	*B* несколько менее правдоподобно	*B* менее правдоподобно



ФОРМАЛИЗАЦИЯ

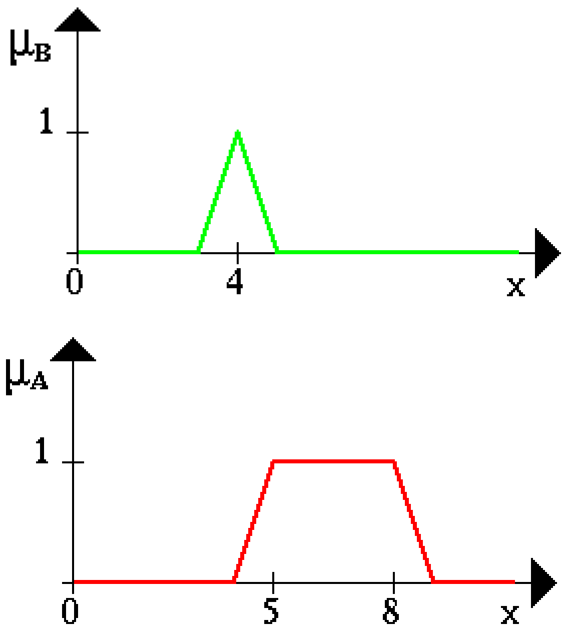
$$\mu_X(x) = \begin{cases} 1, если\ x \in X, \\ 0, если\ x \notin X. \end{cases}$$

$$X = \{(x_1, \mu_X(x_1)), \ldots, (x_n, \mu_X(x_n))\}, n \in N, \mu_X(x_n) = 1$$

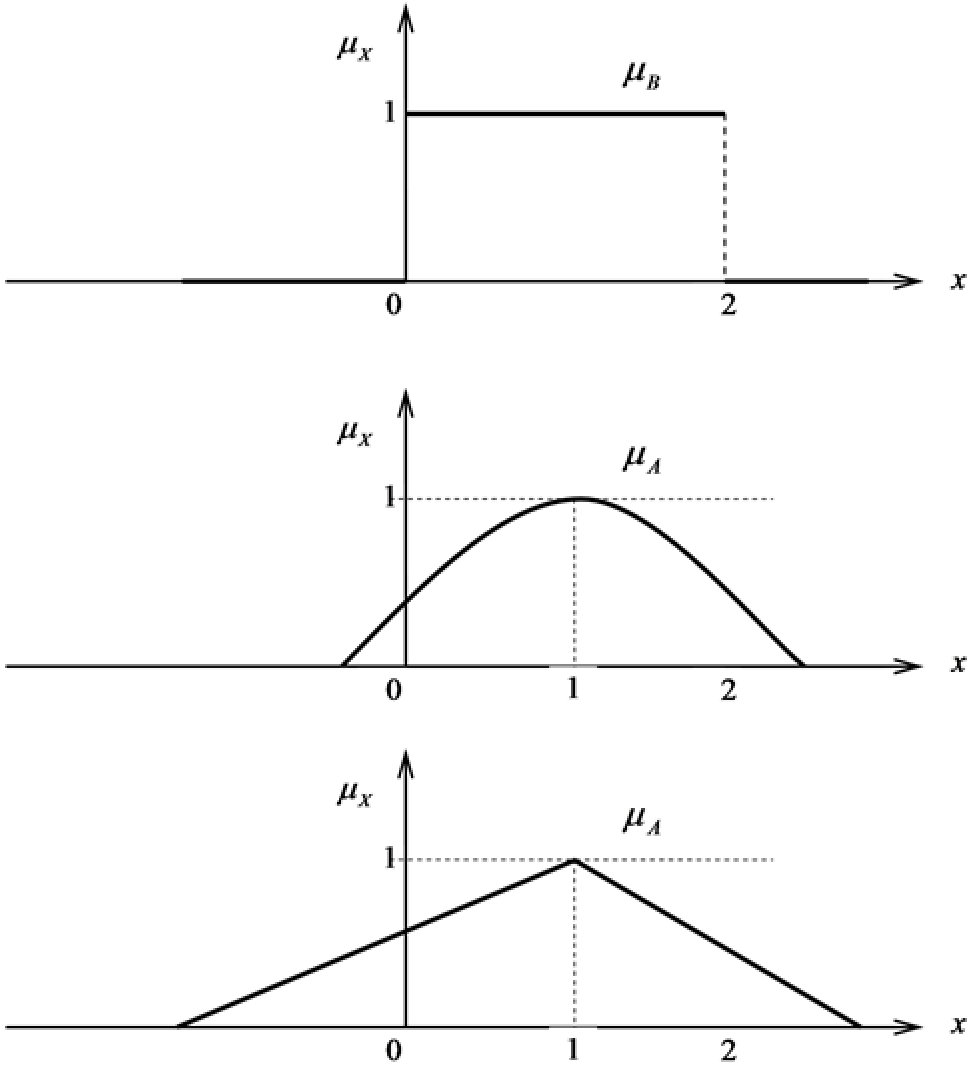
$$X = \{x_i | \mu_X(x_i) = 1\}$$

$$\varphi_X : X \to [0,1] \longrightarrow A = \{(x_1, \varphi_A(x_1)), \ldots, (x_n, \varphi_A(x_n))\}, x_i \in X$$

Пример: $A = \{x \mid \text{"значение } x \text{ близко к } 1\text{"}\}$



П
А
Т
О
Л
О
Г
И
Я



$$supp\ A = \{x \mid \mu_A(x) > 0, x \in X\}$$

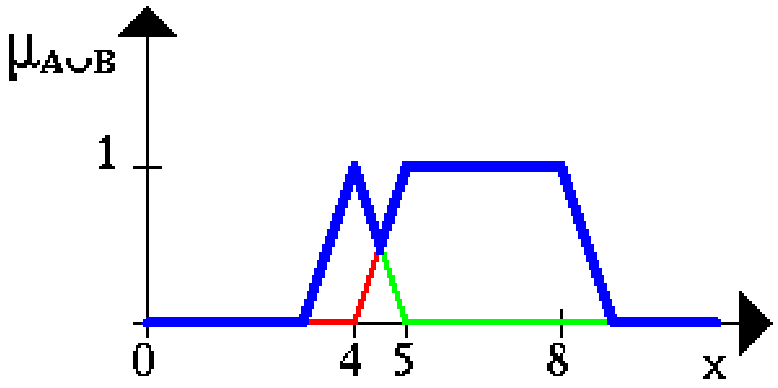
Операции над нечеткими множествами

Заданы: A, B φ_A, φ_B

Объединение $C = \{x, \mu_C(x)\}$ $(x \in X) \wedge (x \in A \vee x \in B)$

$$(a) \quad \varphi_C(x) = \max\{\varphi_A(x),\ \varphi_B(x)\},$$

$$(b) \quad \varphi_C(x) = \begin{cases} 1, \text{если } \varphi_A(x) + \varphi_B(x) \geq 1 \\ \varphi_A(x) + \varphi_B(x), \text{иначе} \end{cases}$$

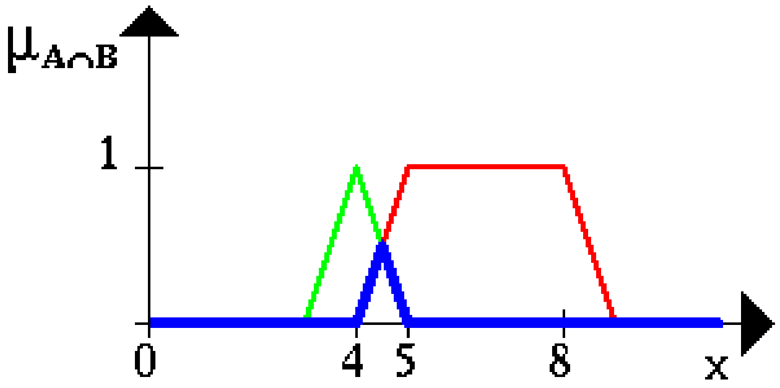


$$(c) \quad \varphi_C(x) = \varphi_A(x) + \varphi_B(x) - \varphi_A(x) \cdot \varphi_B(x)$$

Пересечение $C = \{x, \mu_C(x)\}$ $(x \in X) \wedge (x \in A \wedge x \in B)$

$$(a) \qquad \varphi_C(x) = \min\ \{\varphi_A(x),\ \varphi_B(x)\},$$

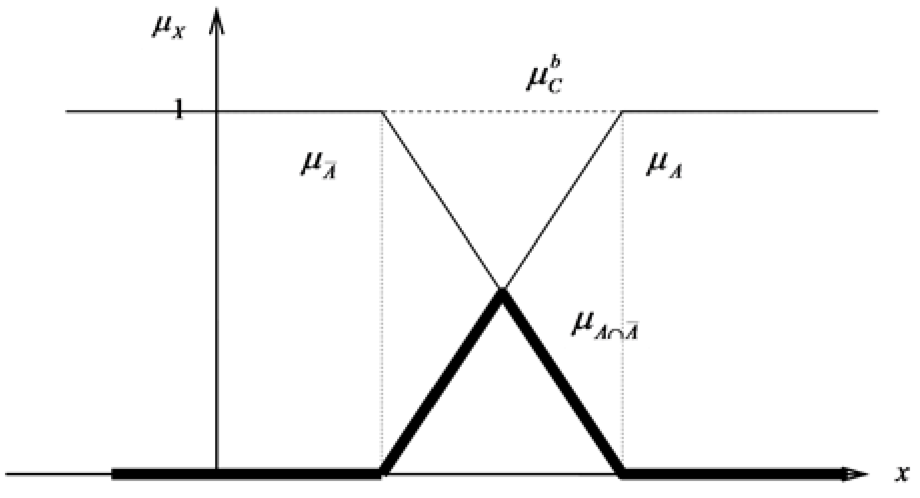
$$(b) \qquad \varphi_C(x) = \varphi_A(x) \cdot \varphi_B(x)$$

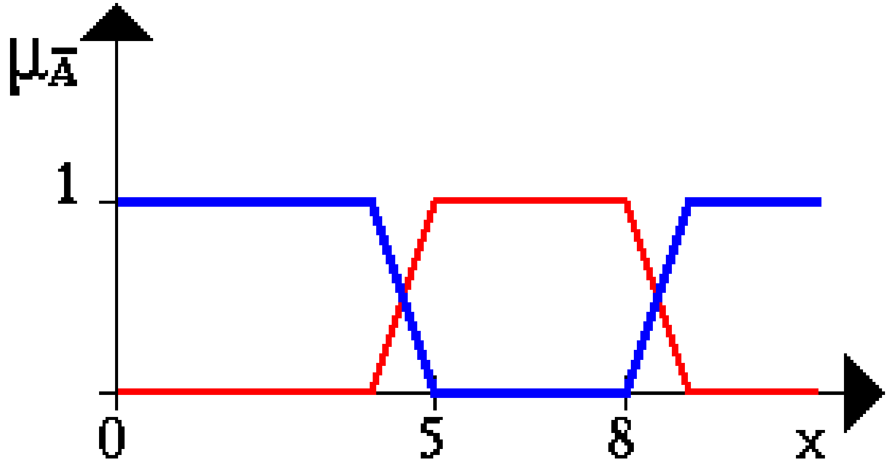


Дополнение $\overline{A} = \{x, \mu_{\overline{A}}(x)\}$ $\forall x \in X\ \ (\mu_{\overline{A}}(x) = 1 - \mu_A(x))$

Свойство нечетких множеств: $A \cap \overline{A} \neq \varnothing$

A= {множество чисел, гораздо больших 0}



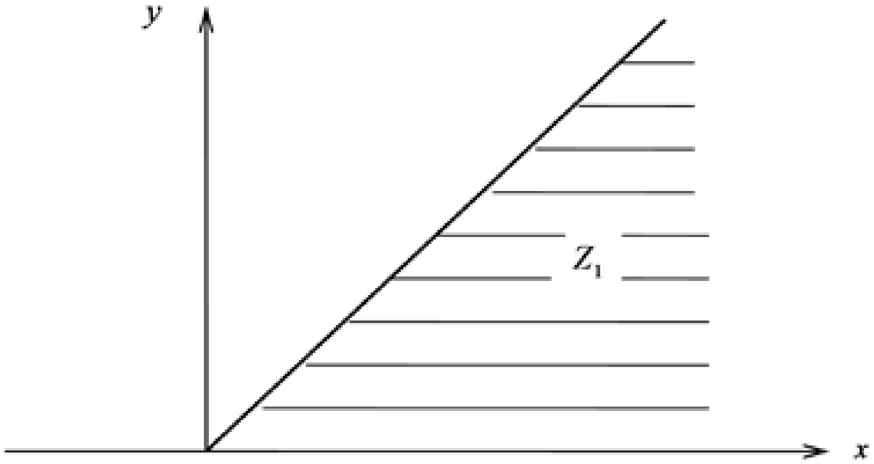


Операции концентрирования и растяжения

A $\longrightarrow$ $A^{\alpha} \quad \alpha \in R^{+} \qquad \mu_{A^{\alpha}}(x) = \mu_{A}^{\alpha}(x)$

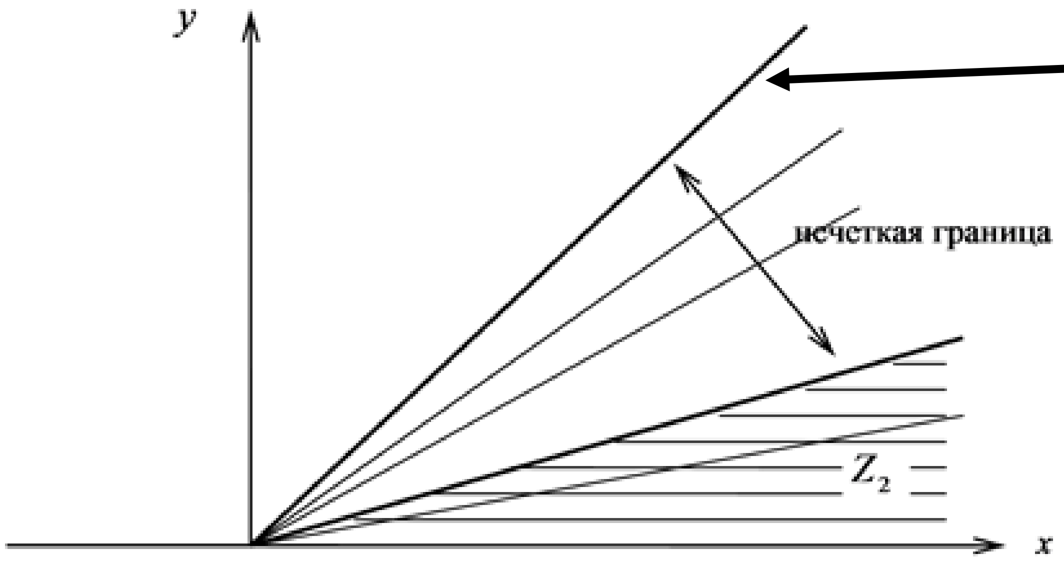
НЕЧЕТКИЕ ОТНОШЕНИЯ

$R \longrightarrow \quad Z_1 = \{(x, y) | (x, y) \in R \times R, x \geq y\}$



$X \qquad R \subseteq X \times X \qquad \mu_R : X \times X \rightarrow [0,1]$

Пример: $Z_2 = (R^2, >>)$ - “много больше”



$Z_1 = (R^2, \geq)$

линейный порядок

Операции над нечеткими отношениями

A, B - нечеткие отношения в X×X с функциями принадлежности μ_A μ_B

Объединение $C = A \cup B$ $C = \{(x_1, x_2), \mu_C(x_1, x_2)\}$

$$C : ((x_1, x_2) \in X \times X) \wedge ((x_1, x_2) \in A \vee (x_1, x_2) \in B)$$

$$\forall (x_1, x_2) \in X \times X \quad \left[\begin{array}{l} \mu_C(x_1, x_2) = \max\{\mu_A(x_1, x_2), \mu_B(x_1, x_2)\} \\ \mu_C(x) = \begin{cases} 1, \text{если } \mu_A(x_1, x_2) + \mu_B(x_1, x_2) \geq 1, \\ \mu_A(x_1, x_2) + \mu_B(x_1, x_2), \text{ в противном случае} \end{cases} \\ \mu_C(x_1, x_2) = \mu_A(x_1, x_2) + \mu_B(x_1, x_2) - \mu_A(x_1, x_2) \cdot \mu_B(x_1, x_2) \end{array} \right.$$

Максминное произведение:

C=A∘B $$C:((x_1,x_2)=(x_1,x_i)\circ(x_i,x_2),(x_1,x_2)\in X\times X)\wedge$$
$$\wedge((x_1,x_2)\in A\vee(x_1,x_2)\in B)$$
$$\forall(x_1,x_i),(x_i,x_2)\in X\times X$$
$$(\mu_C(x_1,x_2)=\sup_{x_i\in X}\min\{\mu_A(x_1,x_i),\mu_A(x_i,x_2)\})$$

Свойства нечетких отношений

Рефлексивность: $\mu_A(x,x)=1 \quad A=\{(x_1,x_2)\in R^2 \,|\, "x_1 \text{ и } x_2 \text{ примерно равны}"\}$

Симметричность: $\mu_A(x_1,x_2)=\mu_A(x_2,x_1)$

$$A=\{(x_1,x_2)\in R^2 \,|\, "x_1 \text{ и } x_2 \text{ сильно отличаются}"\}$$

Пример:

Пусть $X = \{x_1, x_2\}$ и отношения A и B заданы следующими матрицами:

$$\mu_A(x,y) = \begin{array}{c|c} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{Vmatrix} 0.2 & 0.6 \\ 0.5 & 0.8 \end{Vmatrix} \end{array} \qquad \mu_B(x,y) = \begin{array}{c|c} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{Vmatrix} 0.5 & 0.7 \\ 0.3 & 0.1 \end{Vmatrix} \end{array} \qquad \mu_{A \circ B}(x,y) = \begin{array}{c|c} & \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{Vmatrix} 0.3 & 0.6 \\ 0.5 & 0.8 \end{Vmatrix} \end{array}$$

$$\mu_{A \circ B}(x_1, x_1) = \begin{array}{lll} \mu_A(x_1, x_1) & \mu_B(x_1, x_1) & \xrightarrow{\min} 0.2 \\ \mu_A(x_1, x_2) & \mu_B(x_2, x_1) & \xrightarrow{\min} 0.3 \\ & & \quad \max \end{array}$$

$$\mu_{A \circ B}(x_1, x_2) = \begin{array}{lll} \mu_A(x_1, x_1) & \mu_B(x_1, x_2) & \xrightarrow{\min} 0.2 \\ \mu_A(x_1, x_2) & \mu_B(x_2, x_2) & \xrightarrow{\min} 0.6 \\ & & \quad \max \end{array}$$

Транзитивность: $A \circ A \subseteq A$

$$\mu_A(x_1, x_i) \cdot \mu_A(x_i, x_3) \le \min(\mu_A(x_1, x_i), \mu_A(x_i, x_3)) \le$$
$$\le \max(\mu_A(x_1, x_i), \mu_A(x_i, x_3))$$

$$A = \{(x_1, x_2) \in R^2 \,|\, "x_1 \textit{ намного больше } x_2"\}$$

Структуры

Эквивалентность – рефлексивно, симметрично и транзитивно

Частичные и линейные порядки - рефлексивно, антисимметрично и транзитивно

Нечеткие функции

$$f : X \to Y$$

$$f : X \to Y \;\Leftrightarrow\; Z_f \subseteq X \times Y$$

$$(y, \mu_B(y)) \overset{df}{=} (f(x), \mu_A(x)),\;\; x \in X$$

$$\mu_B(y) \overset{df}{=} \sup_{x \in f^{-1}(y)} \mu_A(x), \qquad f^{-1}(y) = \{x \in X \mid f(x) = y\}$$

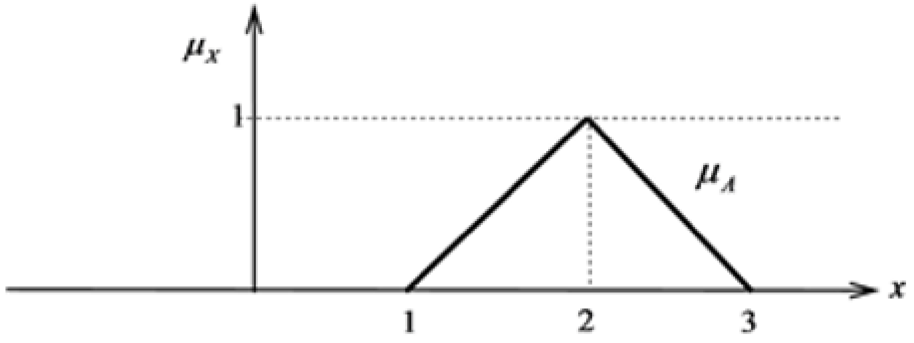
нечеткость

Нечеткие операции

$f : X \times Y \to Z$ - бинарные

Если X=Y=Z и все они числовые, то приходим к арифметике

НЕЧЕТКАЯ АРИФМЕТИКА

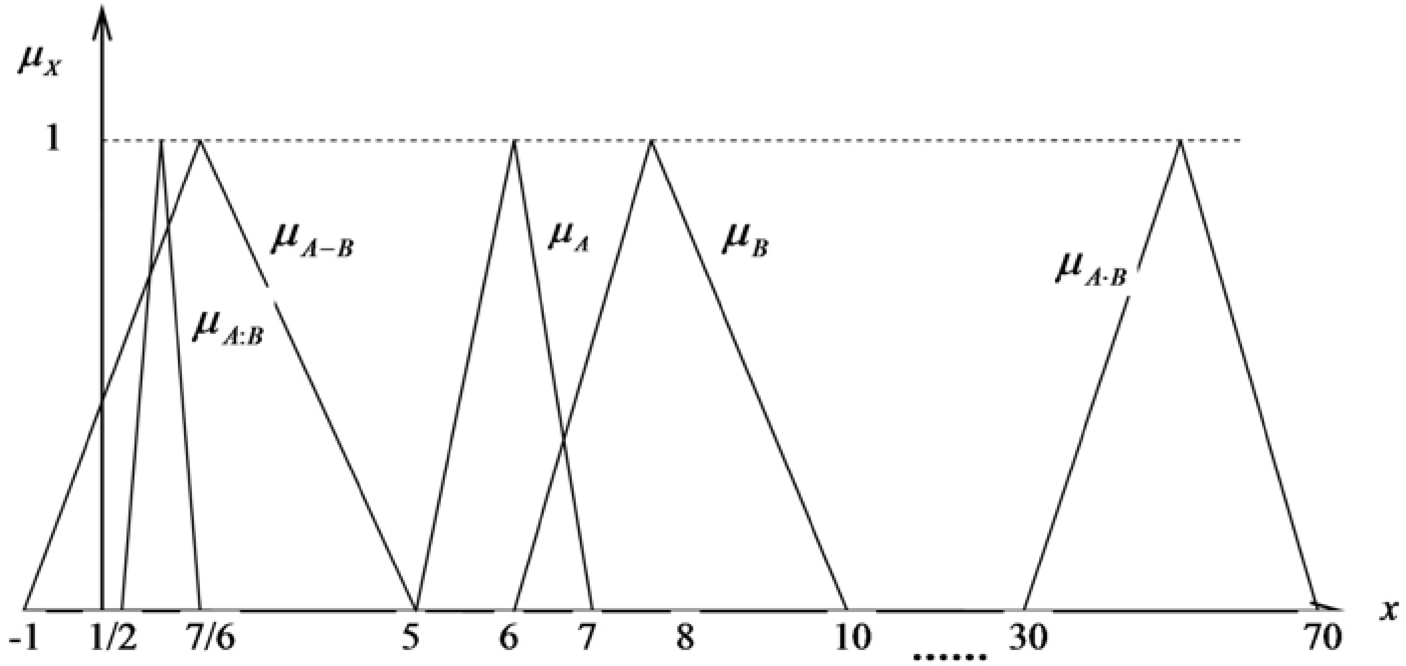


$$g : R \times R \to R \qquad D = g(A,B)$$

$$\mu_D(x) = \sup_{\substack{x=g(a,b)\\ a\in S_A,\, b\in S_B}} \min(\mu_A(a), \mu_B(b))$$

$$S_D = \{x \mid x = g(a,b), a \in S_A, b \in S_B\}$$

$$\begin{aligned} S_D = \{&\min(g(a_1,b_1), g(a_1,b_3), g(a_3,b_1), g(a_3,b_3)),\\ &\max(g(a_1,b_1), g(a_1,b_3), g(a_3,b_1), g(a_3,b_3))\} \end{aligned}$$



$$\forall A, B \quad (A + B \neq 0),$$

$$\forall A, B \quad (A : B \neq 1),$$

$$A \cdot X + B = C$$

НЕЧЕТКАЯ ЛОГИКА

$P^F : X^n \to [0,1]$ -- предикат

$\mu_{P^F}(x)$

Нечеткие формулы представляют собой логическую комбинацию нечетких предикатов, которая строится с помощью операций $\wedge, \vee, \neg$

Пусть теперь P^{F_1} и P^{F_2} – различные нечеткие предикаты.

$$\mu_{P^{F_1} \vee P^{F_2}}(x) = \max\{\mu_{P^{F_1}}(x), \mu_{P^{F_2}}(x)\}$$

$$\mu_{P^{F_1} \wedge P^{F_2}}(x) = \min\{\mu_{P^{F_1}}(x), \mu_{P^{F_2}}(x)\}$$

$$\mu_{\neg P^F}(x) = 1 - \mu_{P^F}(x)$$

СТЕПЕНИ НЕОПРЕДЕЛЕННОСТИ

$m : 2^X \to [0,1]$ - распределение уверенности $\quad A \subseteq X$

$$\begin{cases} m(\varnothing) = 0, \\ \sum_{A \subseteq X} m(A) = 1, \end{cases}$$

Функция уверенности - $b : 2^X \to [0,1]$

$$b(A) = \sum_{B \subseteq A} m(B)$$

Пример: $X = \{x_1, x_2, x_3\}$

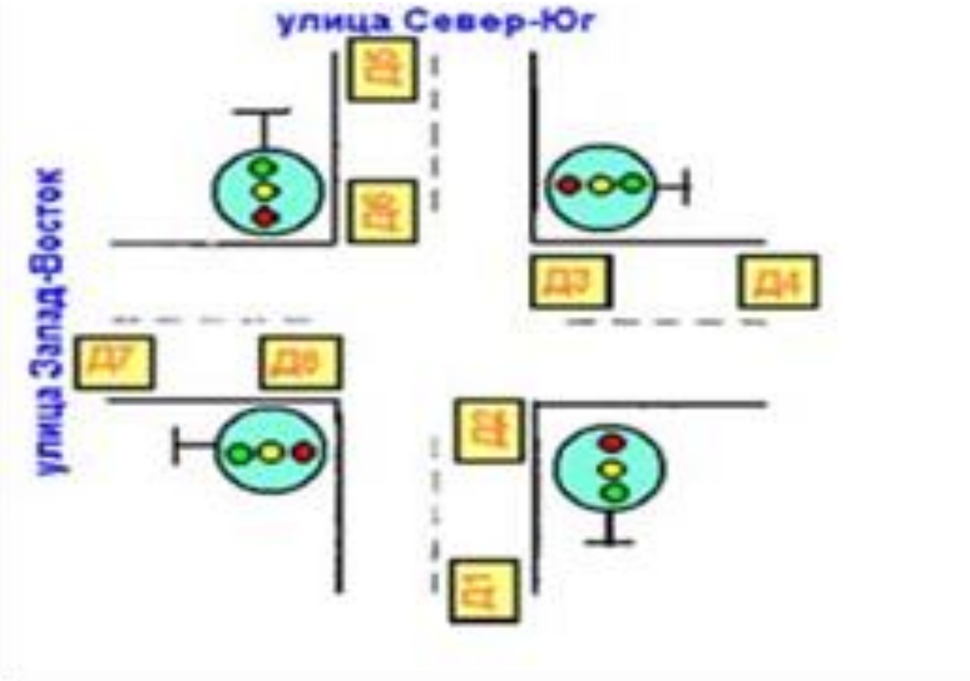
A	$\varnothing$	x_1	x_2	x_3	x_1, x_2	x_2, x_3	x_1, x_3	x_1, x_2, x_3
$m(A)$	0	0.2	0.3	0.1	0.3	0.1	0	0
$b(A)$	0	0.2	0.3	0.1	0.8	0.5	0.3	1
$p(A)$	0	0.5	0.7	0.2	0.9	0.8	0.7	1

$$p(A) = 1 - b(\neg A) = \sum_{\substack{B \subseteq X \\ B \cap A = \emptyset}} m(B)$$

p –степень правдоподобия подмножества *A*

$$b(A) \le истинная\,вероятность(A) \le p(A)$$

Моделирование работы светофора с нечеткой логикой



Приложения нечеткой математики

- Автоматическое управление воротами плотины на **гидроэлектростанциях** *(Tokyo Electric Pow.)*
- Упрощенное управление **роботами** *(Hirota, Fuji Electric, Toshiba, Omron)*
- **Наведение телекамер** при трансляции спортивных событий *(Omron)*
- Замена экспертов при анализе работы **биржи** *(Yamaichi, Hitachi)*
- Предотвращение нежелательных температурных флуктуаций в **системах кондиционирования воздуха** *(Mitsubishi, Sharp)*
- Эффективное и стабильное управление **автомобильными двигателями** *(Nissan)*
- Управление экономичной скоростью **автомобилей** *(Nissan, Subaru)*
- Улучшение эффективности и оптимизация **промышленных систем управления** *(Aptronix, Omron, Meiden, Sha, Micom, Mitsubishi, Nisshin-Denki, Oku-Electronics)*
- Позиционирование приводов в **производстве полупроводников.** wafer-steppers *(Canon)*
- Оптимизированное планирование **автобусных расписаний** *(Toshiba, Nippon-System, Keihan-Express)*
- Системы архивации **документов** *(Mitsubishi Elec.)*
- **Системы прогнозирования** землетрясений *(Inst. of Seismology Bureau of Metrology, Japan)*
- **Медицина:** диагностика рака *(Kawasaki Medical School)*

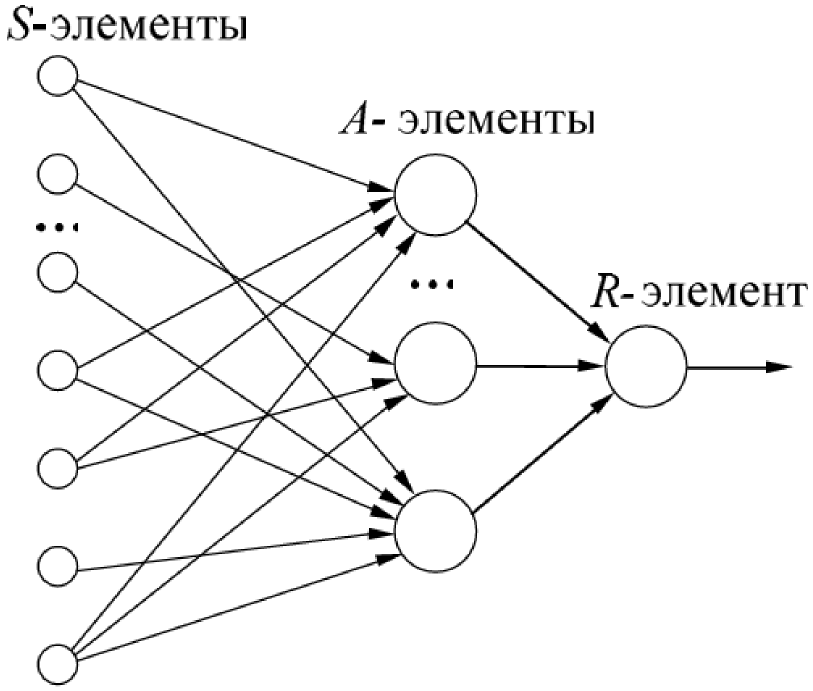
- Сочетание методов нечеткой логики и **нейронных сетей** *(Matsushita)*
- Распознавание рукописных символов в **карманных компьютерах (записных книжках)** *(Sony)*
- Распознавание движения изображения в **видеокамерах** *(Canon, Minolta)*
- Автоматическое управление двигателем **пылесосов** с автоматическим определением типа поверхности и степени засоренности *(Matsushita)*
- Управление освещенностью в **камкодерах** *(Sanyo)*
- Компенсация вибраций в **камкодерах** *(Matsushita)*
- Однокнопочное управление **стиральными машинами** *(Matsushita, Hitachi)*
- **Распознавание** рукописных текстов, объектов, голоса *(CSK, Hitachi, Hosai Univ., Ricoh)*
- Вспомогательные средства полета **вертолетов** *(Sugeno)*
- Моделирование **судебных процессов** *(Meihi Gakuin Univ, Nagoy Univ.)*
- **САПР** производственных процессов *(Aptronix, Harima, Ishikawajima-OC Engeneering)*
- Управление скоростью линий и температурой при **производстве стали** *(Kawasaki Steel, New-Nippon Steel, NKK)*
- Управление **метрополитенами** для повышения удобства вождения, точности остановки и экономии энергии *(Hitachi)*
- Оптимизация потребления бензина в **автомобилях** *(NOK, Nippon Denki Tools)*
- Повышение чувствительности и эффективности **управления лифтами** *(Fujitec, Hitachi, Toshiba)*
- Повышение безопасности **ядерных реакторов** *(Hitachi, Bernard, Nuclear Fuel div.)*

Комментарий к лекции 10

НЕЙРОННЫЕ СЕТИ

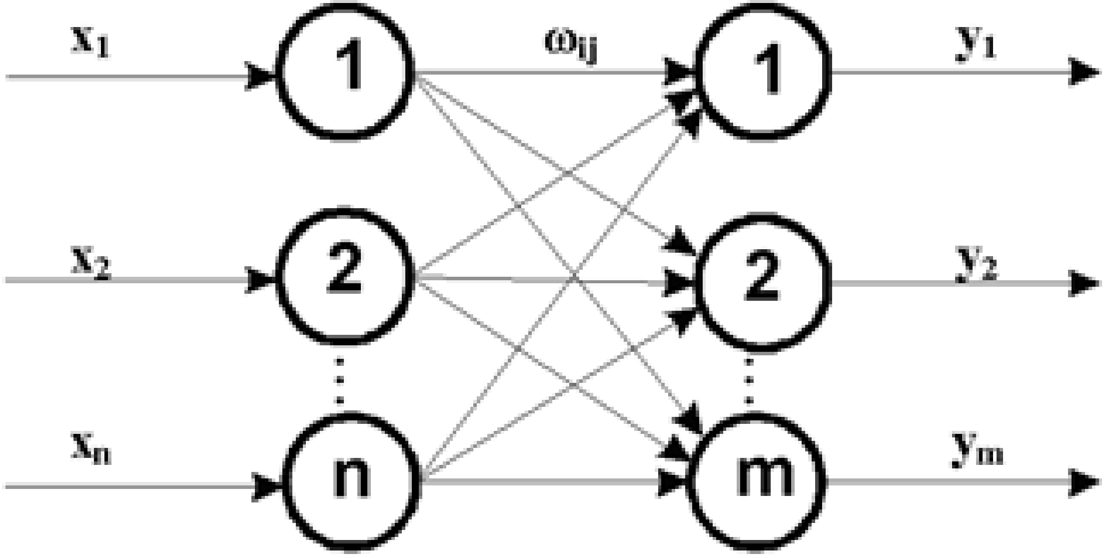
1. **Основы моделирования НС**
2. **Варианты построения нейронной сети**
3. **Типы НС**
4. **Большие языковые модели LLM**
5. **Применение НС**

ПЕРСЕПТРОН РОЗЕНБЛАТТА



Шаг 0.	Значения $W(t=0)$ полагаются случайными
Шаг 1.	$x^{\alpha} \Rightarrow \tilde{y}^{\alpha} \neq y^{\alpha}$
Шаг 2.	$\delta^{\alpha} = g(\tilde{y}^{\alpha} - y^{\alpha})$
Шаг 3.	$W(t+\Delta t) = W(t) + \eta \cdot x^{\alpha}(\delta^{\alpha})^{T}$ $0 < \eta < 1$
Шаг 4.	Шаги 1-3 повторяются для всех обучающих векторов x^{α}

Архитектура однослойной нейронной сети

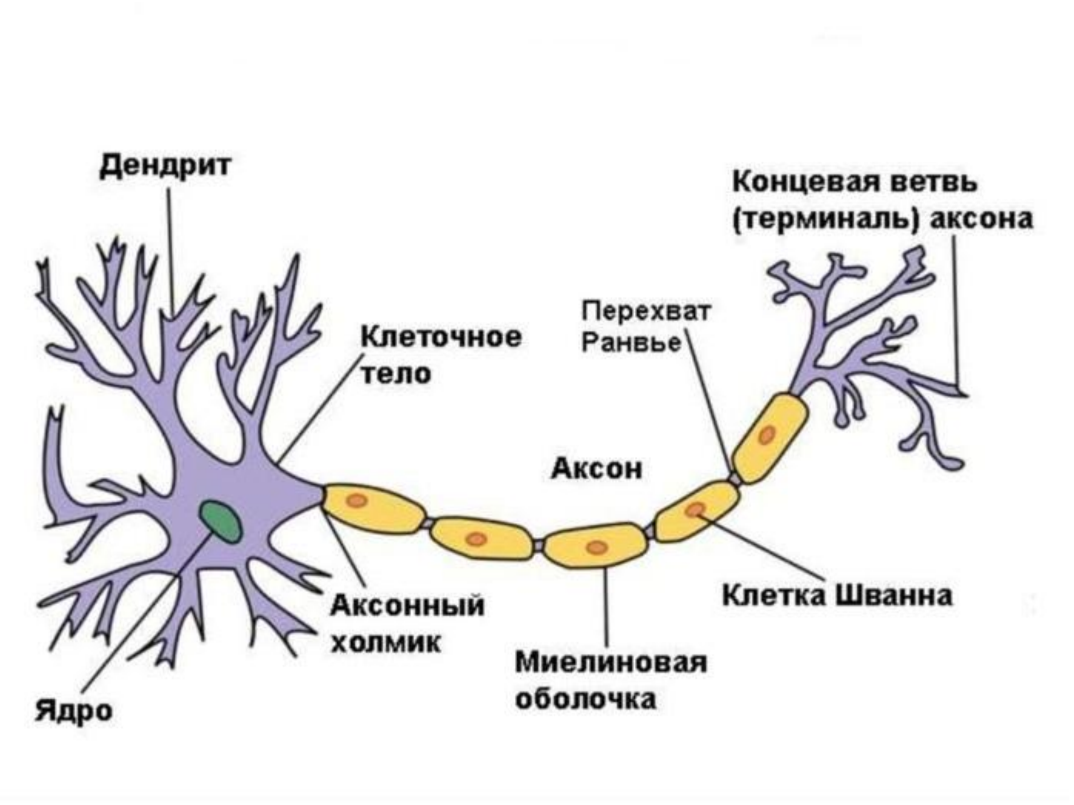


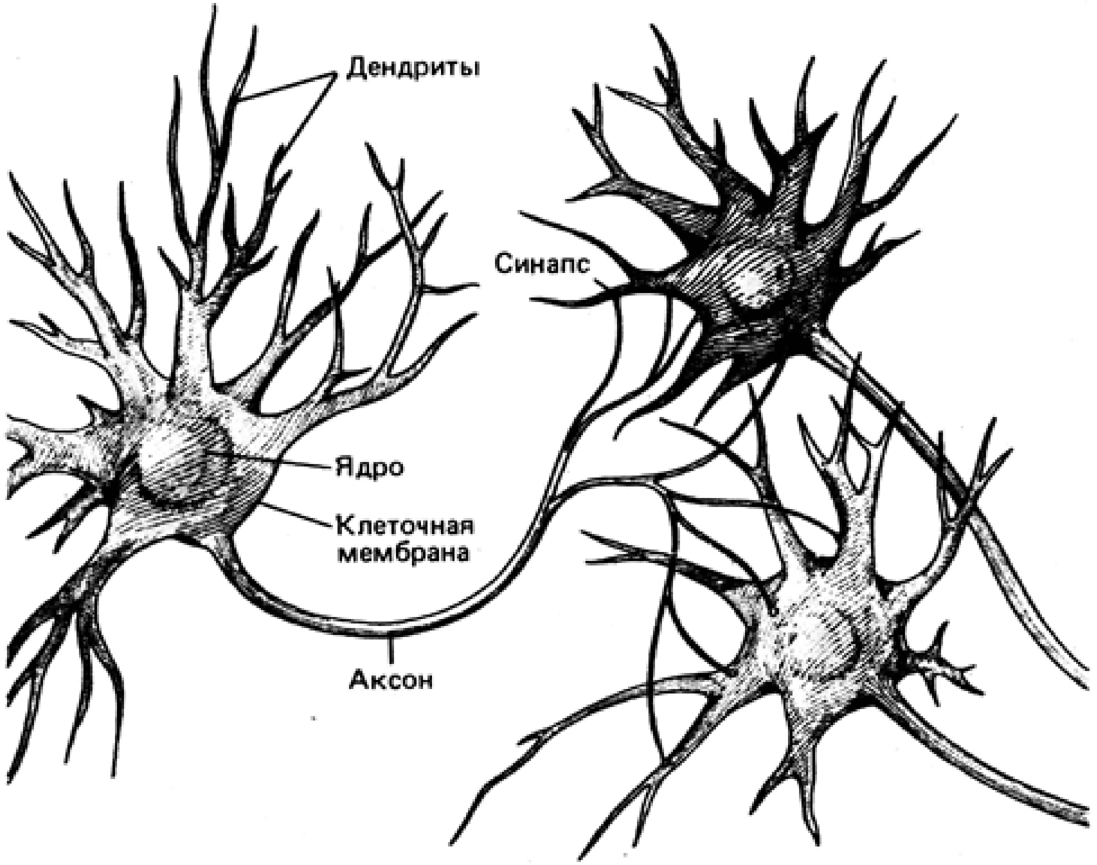
$$y_j = F(S_j) = F(\sum_{i=1}^{n} \omega_{ij} x_i - T_j),$$

$$W = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2m} \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nm} \end{bmatrix}$$

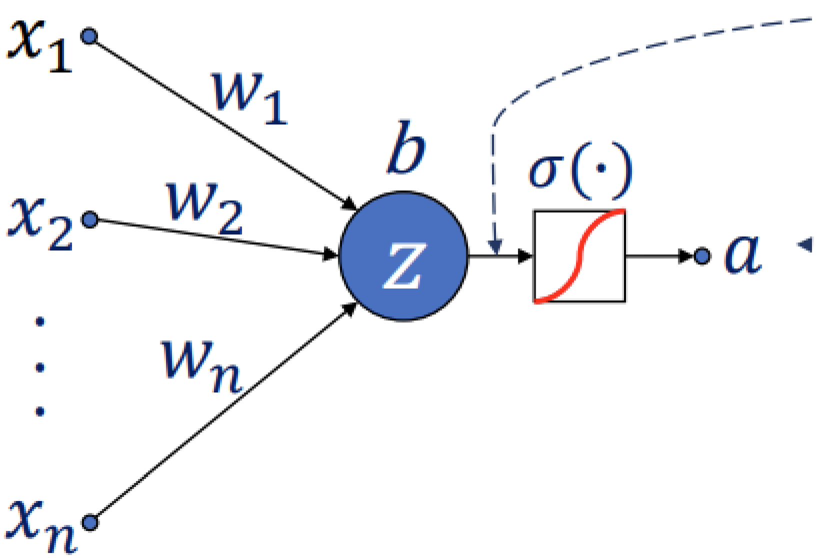
$$S = W^T X - T,$$

Биологический нейрон





Искусственный (формальный) нейрон



Индуцированное локальное поле:

$$z = b + \sum_{i=1}^{n} w_i x_i$$

$$\boxed{a = \sigma(z)}$$

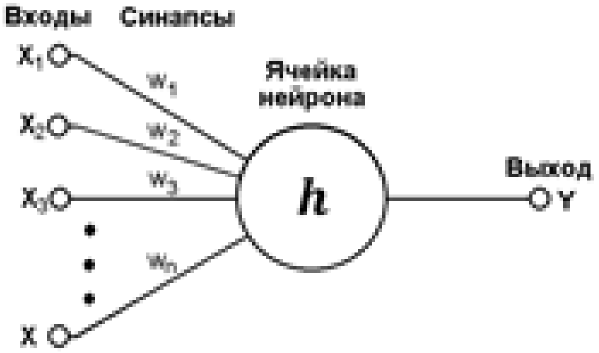
$\vec{x} = (x_1, x_2, \ldots, x_n)$ – входные сигналы: $x_i \in \mathbb{R}$

$\vec{w} = (w_1, w_2, \ldots, w_n)$ – синапсические веса: $w_i \in \mathbb{R}$

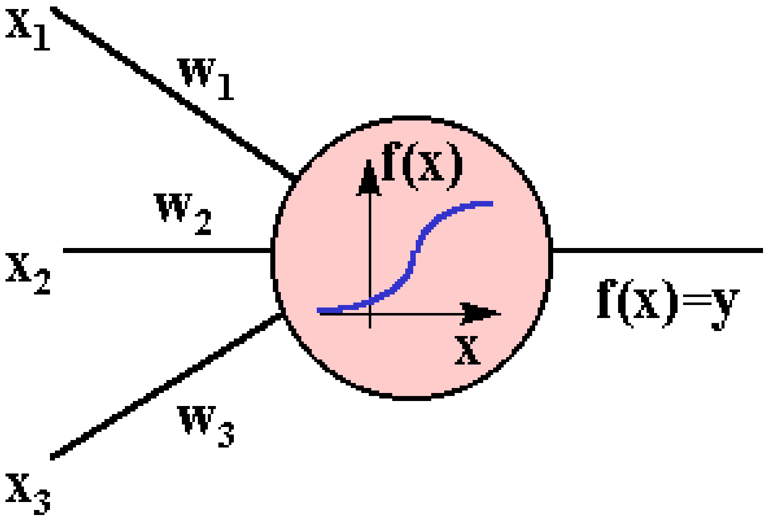
b – смещение: $b \in \mathbb{R}$

a – выходной сигнал: $0 < a < 1$

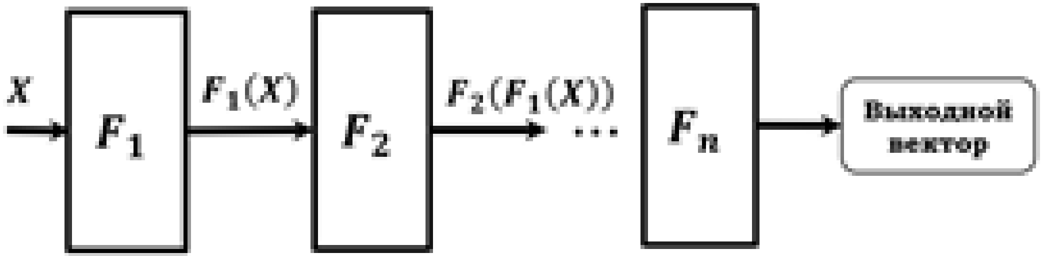
Формальный нейрон



$$h = \sum_i x_i * w_i \qquad y = f(h)$$



Нейронная сеть



$$\Delta = \boxed{\text{Выходной вектор}} - \boxed{\text{Эталонный выход}}$$

$$\frac{\partial \Delta}{\partial W_n} = \frac{\partial \Delta (F_n)}{\partial F_n} * \frac{\partial F_n (F_{n-1}, W_n)}{\partial W_n}$$

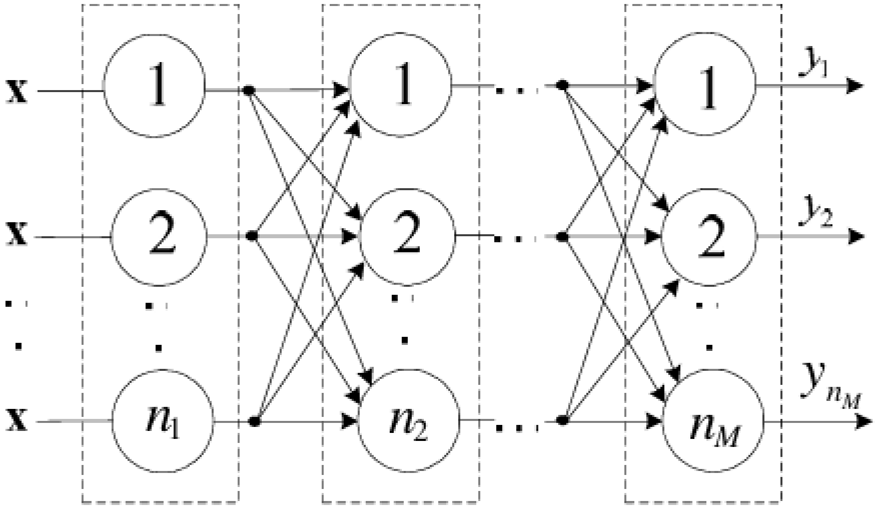
Функция f в модели нейрона называется активационной.

Способы задания f:

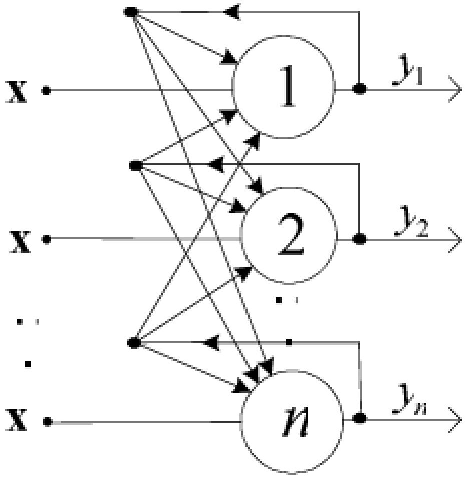
Функции активации нейронов

Название	Формула	Область значений
Линейная	$f(s)=k\,s$	$(-\infty, \infty)$
Полулинейная	$f(s) = \begin{cases} k\,s, & s > 0, \\ 0, & s \le 0 \end{cases}$	$(0, \infty)$
Логистическая (сигмоидальная)	$f(s) = \dfrac{1}{1+e^{-as}}$	$(0, 1)$
Гиперболический тангенс (сигмоидальная)	$f(s) = \dfrac{e^{as} - e^{-as}}{e^{as} + e^{-as}}$	$(-1, 1)$
Экспоненциальная	$f(s) = e^{-as}$	$(0, \infty)$
Синусоидальная	$f(s) = \sin(s)$	$(-1, 1)$
Сигмоидальная (рациональная)	$f(s) = \dfrac{s}{a+\lvert s\rvert}$	$(-1, 1)$
Шаговая (линейная с насыщением)	$f(s) = \begin{cases} -1, & s \le -1, \\ s & -1 < s < 1, \\ 1 & s \ge 1 \end{cases}$	$(-1, 1)$
Пороговая	$f(s) = \begin{cases} 0, & s < 0 \\ 1, & s \ge 0 \end{cases}$	$(0, 1)$
Модульная	$f(s) = \lvert s\rvert$	$(0, \infty)$
Знаковая (сигнатурная)	$f(s) = \begin{cases} 1, & s > 0, \\ -1, & s \le 0 \end{cases}$	$(-1, 1)$
Квадратичная	$f(s) = s^2$	$(0, \infty)$

Базовые архитектуры нейронных сетей

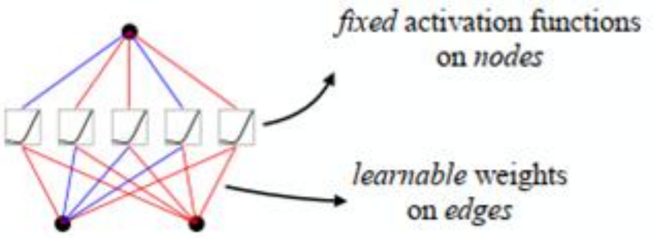
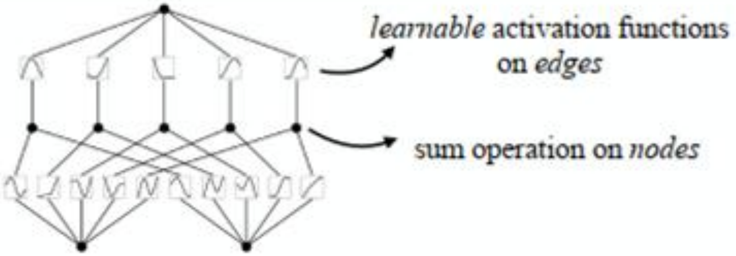
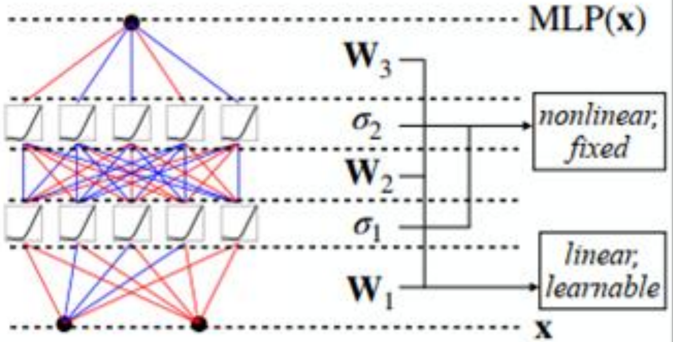
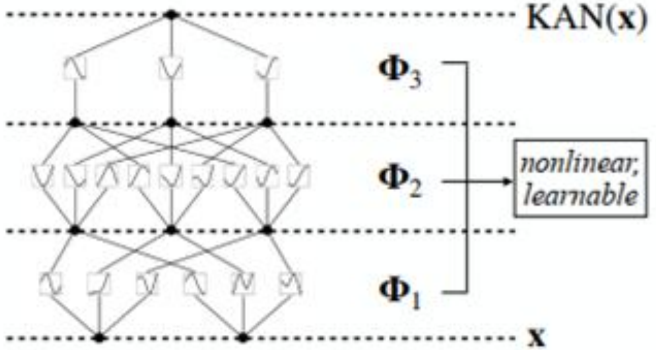


Многослойная НС



Полносвязная

Варианты построения нейронной сети

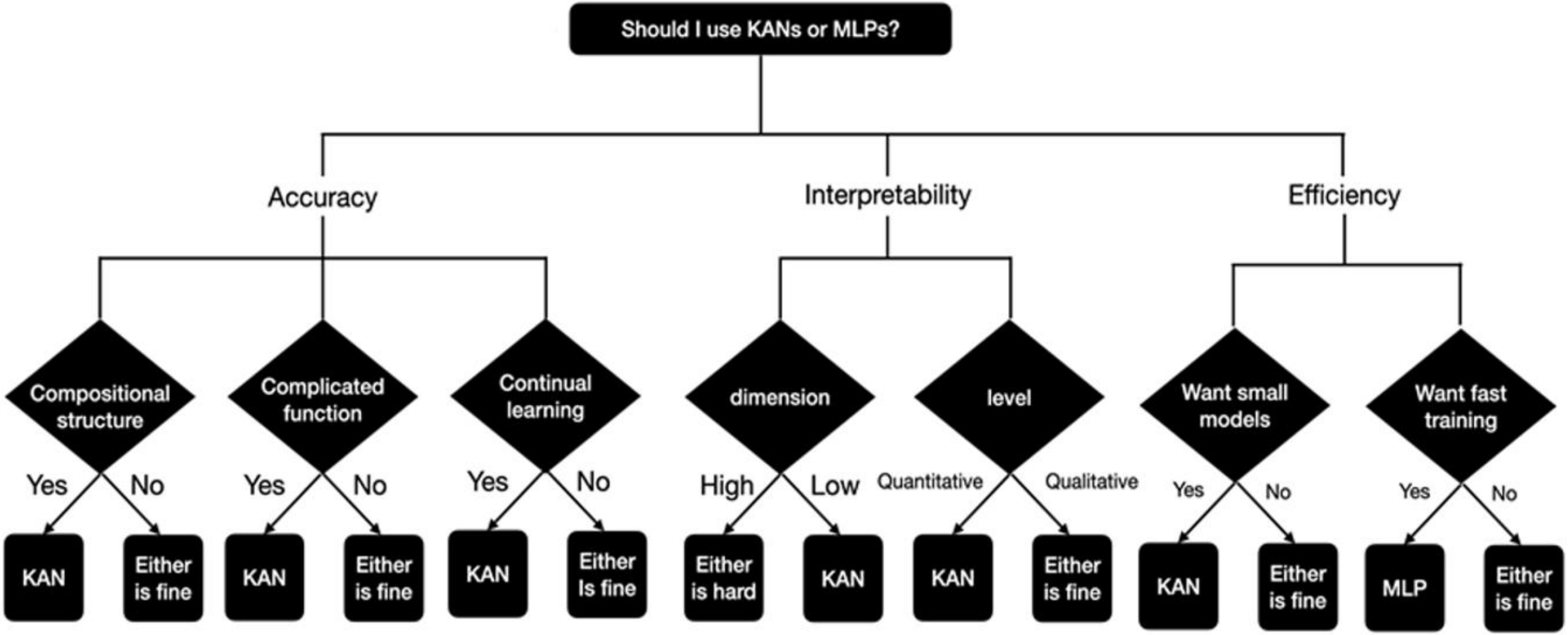
Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i\sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q\left(\sum_{p=1}^{n} \phi_{q,p}(x_p)\right)$
Model (Shallow)	(a) 	(b) 
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)	(c) 	(d) 

Универсальная теорема аппроксимации:
искусственная нейронная сеть с одним скрытым слоем и достаточно большим числом нейронов $N(\boldsymbol{\varepsilon})$ может аппроксимировать с любой степенью точности $\boldsymbol{\varepsilon}>0$ любую непрерывную функцию $f(x_1, \ldots, x_n)$.

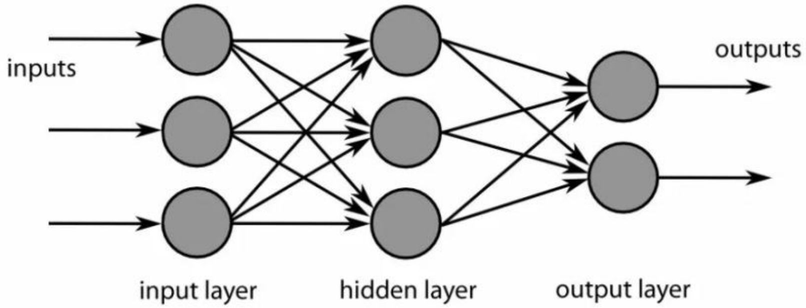
Теорема КАН:
Любую функцию $f\colon [0,1]^n \to \mathbb{R}$ можно представить в виде

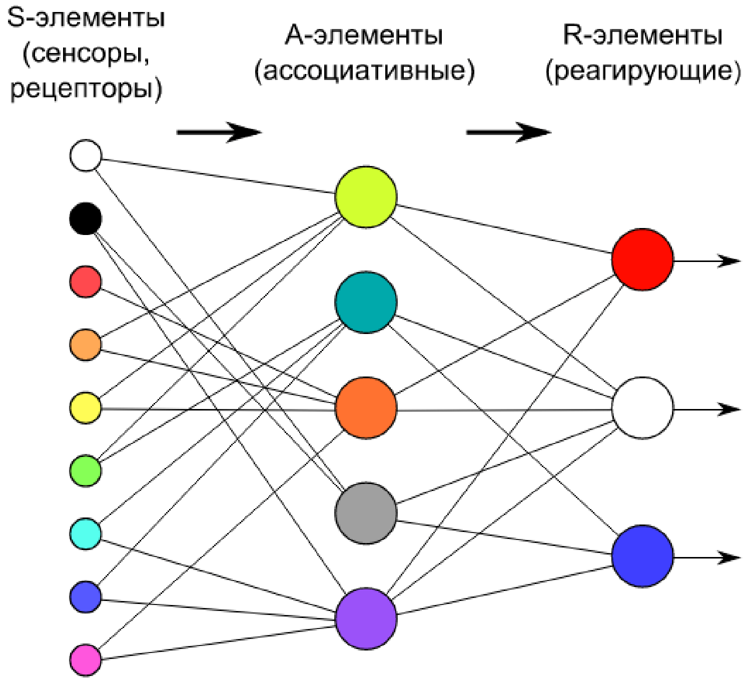
$$f(x) = f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$

$\phi_{q,p}\colon [0,1] \to \mathbb{R}$, $\Phi_q\colon \mathbb{R} \to \mathbb{R}$


Should I use KANs or MLPs?
Accuracy
Interpretability
Efficiency
Compositional structure
Yes
No
KAN
Either is fine
Complicated function
Yes
No
KAN
Either is fine
Continual learning
Yes
No
KAN
Either Is fine
dimension
High
Low
Either is hard
KAN
level
Quantitative
Qualitative
KAN
Either is fine
Want small models
Yes
No
KAN
Either is fine
Want fast training
Yes
No
MLP
Either is fine

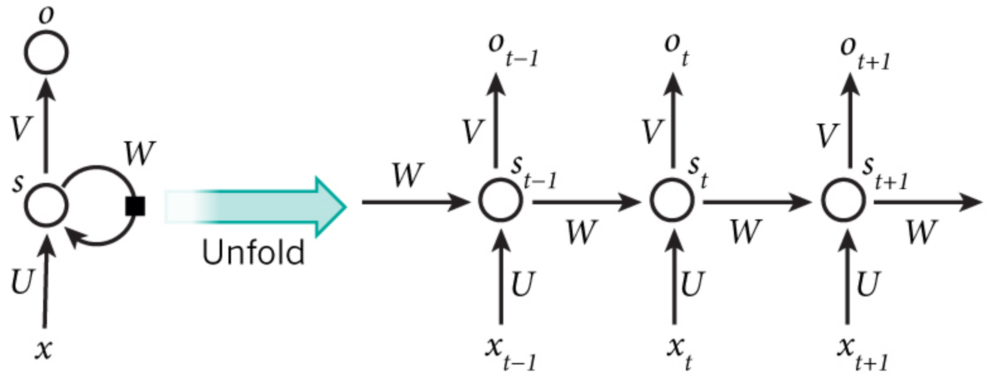
Сеть прямого распространения (FFNN)



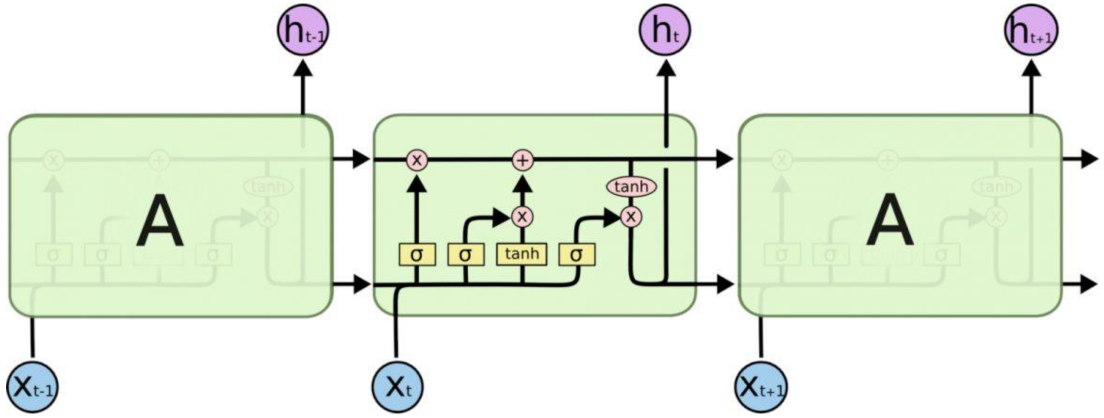


- Не могут работать с последовательной информацией.
- Учитывают только последний ввод.
- Не запоминают предыдущие данные.

Рекуррентные нейронные сети (RNN)

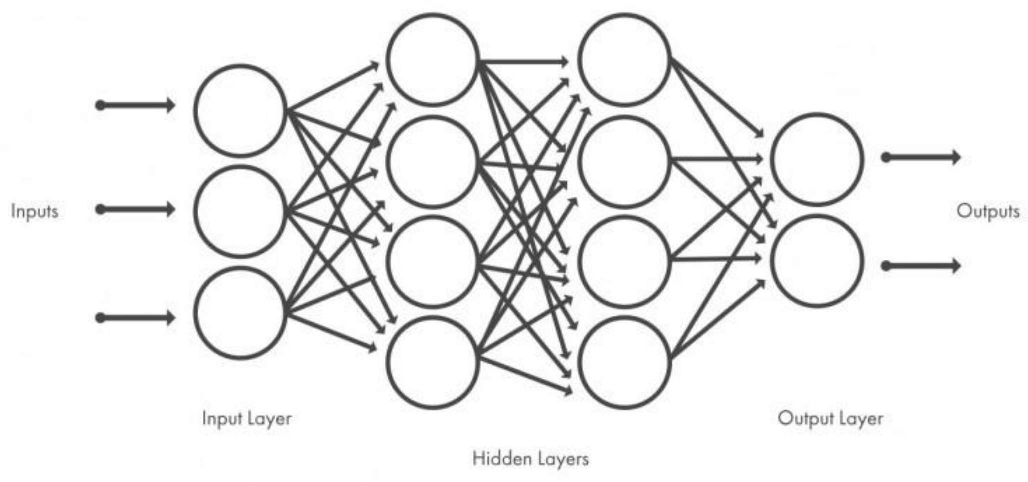


Долгая краткосрочная память (Long short-term memory, LSTM)



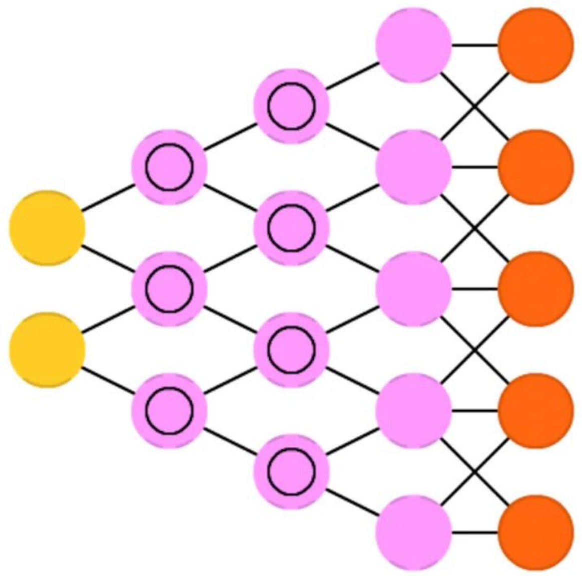
- OpenAI в 2018 создал бота для Dota 2.
- DeepMind в 2019 разработал бота для Starcraft II.

Сверточные нейронные сети (CNN)



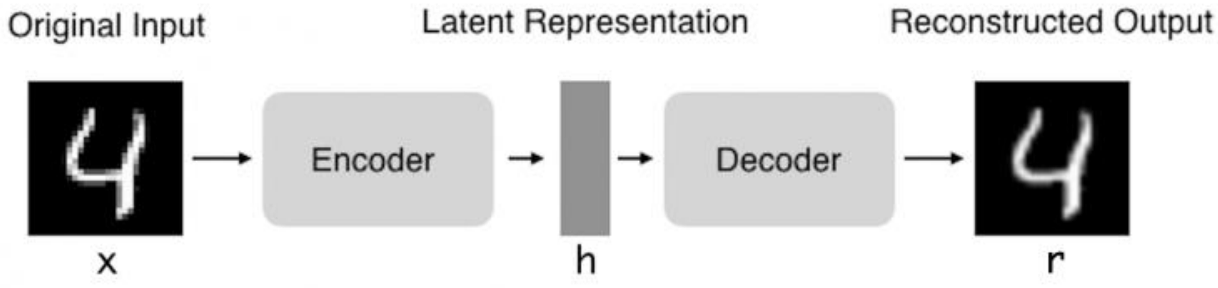
- Распознавание объектов с сеточной структурой признаков (изображения, видео).
- Два слоя – **сверточный** и пулинг.
- Автоматическое извлечение признаков.
- Возможность переобучения для выполнения новых задач без создания другой системы.

Деконволюционные сети (DNN)



- Применяются в задачах восстановления и генерации изображений, в случаях когда необходимо увеличить разрешение для получения более детализированного результата.
- Деконволюционные нейросети используют операцию обратную свертке (**конволюцию**).
- Используются для извлечения структуры признаков из массива информации.

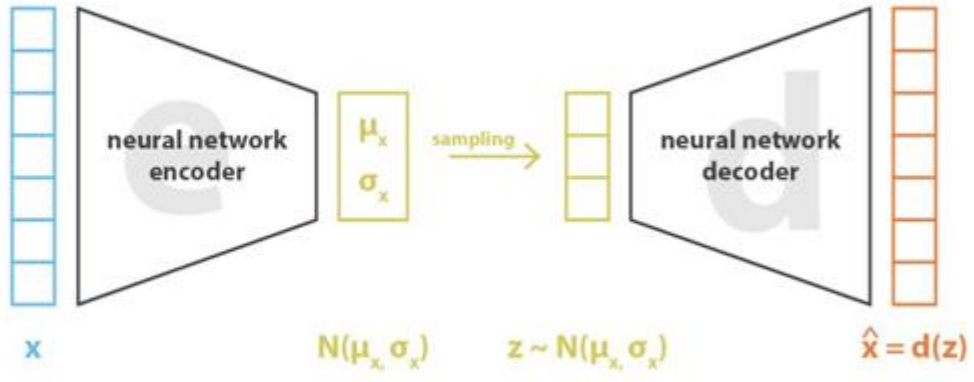
Автоэнкодер



Autoencoder — нейронная сеть, которая вначале сжимает данные, уменьшая их размер, а затем максимально точно восстанавливает исходные данные, минимизируя потери.

Цель — получить на выходном слое отклик, близкий к полученному.

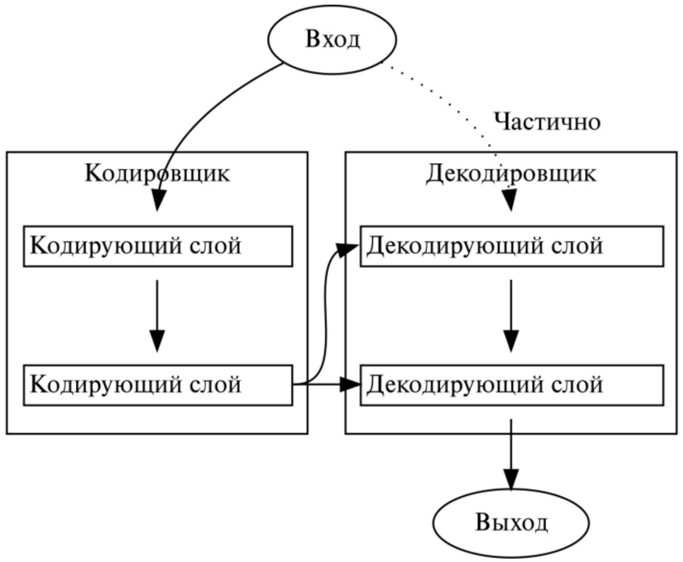
Автоэнкодер имеет одинаковое количество нейронов на входе и выходе.

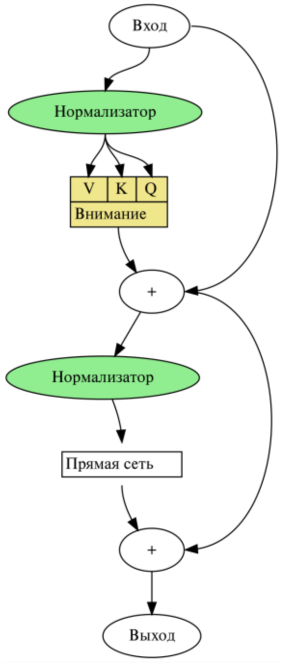


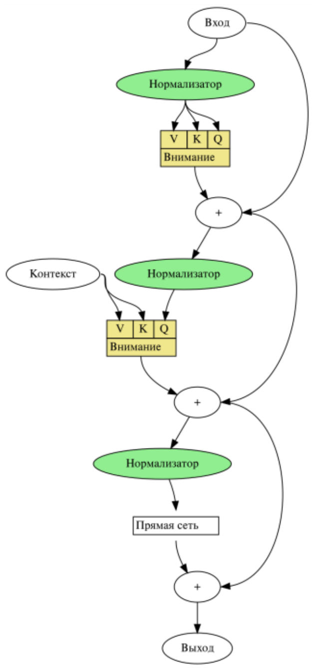
Вариационный автоэнкодер

Используется для **генерации** на выходе случайных данных, похожих на тренировочный набор

Трансформеры







- **Двунаправленные трансформеры**
  Двунаправленные кодировщики на моделях трансформеров (BERT) обрабатывают слова в предложении взаимосвязанно, а не по отдельности.
- **Предварительно обученные генеративные трансформеры (GPT)**
  Способны предсказывать следующее значение в последовательности на основе предыдущих.
- **Трансформеры для мультимодальных задач**
  Предназначены для обработки преимущественно текста и изображений.
- **Трансформеры машинного зрения**
  Предназначены для классификации изображений.

Токены. Интерпретируются как вектора, цифровое значение которых кодирует значение каждого токена. Близкие токены ⇨ близкие вектора.

Генеративные трансформеры — это семейство нейронных сетей,, которые способны создать текст после предварительного обучения в огромных объемах текстовых данных. Основная идея заключается в том, что модель сначала обучается предсказать следующее слово в тексте, а затем может быть дообучена для решения различных задач естественной обработки языка (НЛП), таких как генерация текста, перевод, классификация и другие.

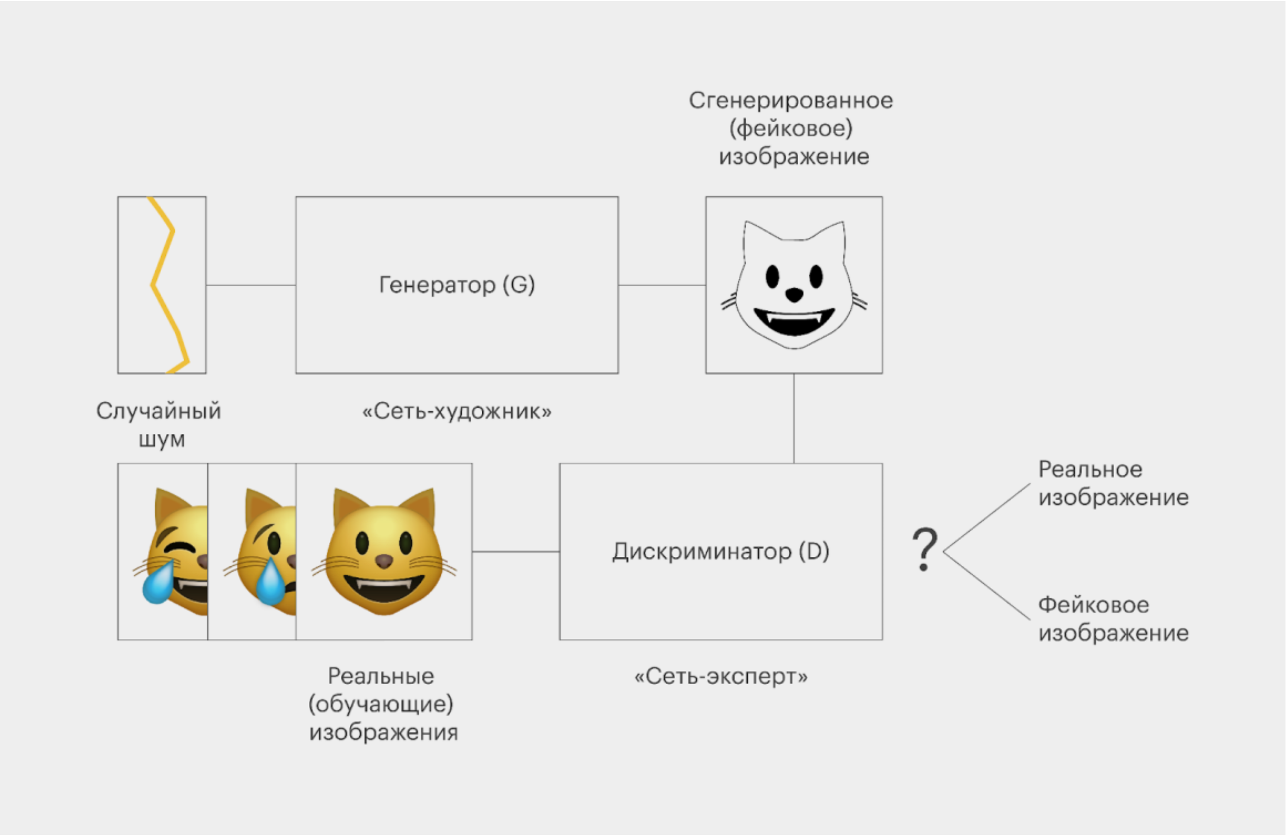
Архитектура и принцип работы

- Генеративные трансформеры используют механизм внимания (механизм внимания), который позволяет модели одновременно анализировать все слова в предложениях и определять, какие из них наиболее актуальны для текущего контекста.
- В отличие от рекуррентных сетей, трансформеры обрабатывают весь текст параллельно, что делает их значительно быстрее и эффективнее при обучении на больших объемах данных.
- GPT-модели используют только декодировщик трансформера, так как их основная задача — генерация текста на основе контекста.

Применение

- Генеративные трансформеры применяются для создания текстов, чат-ботов, ручной трансляции, анализа и генерации речи, а также в других областях, где требуется обработка и генерация естественного языка.

Генеративно-состязательные сети (GAN)

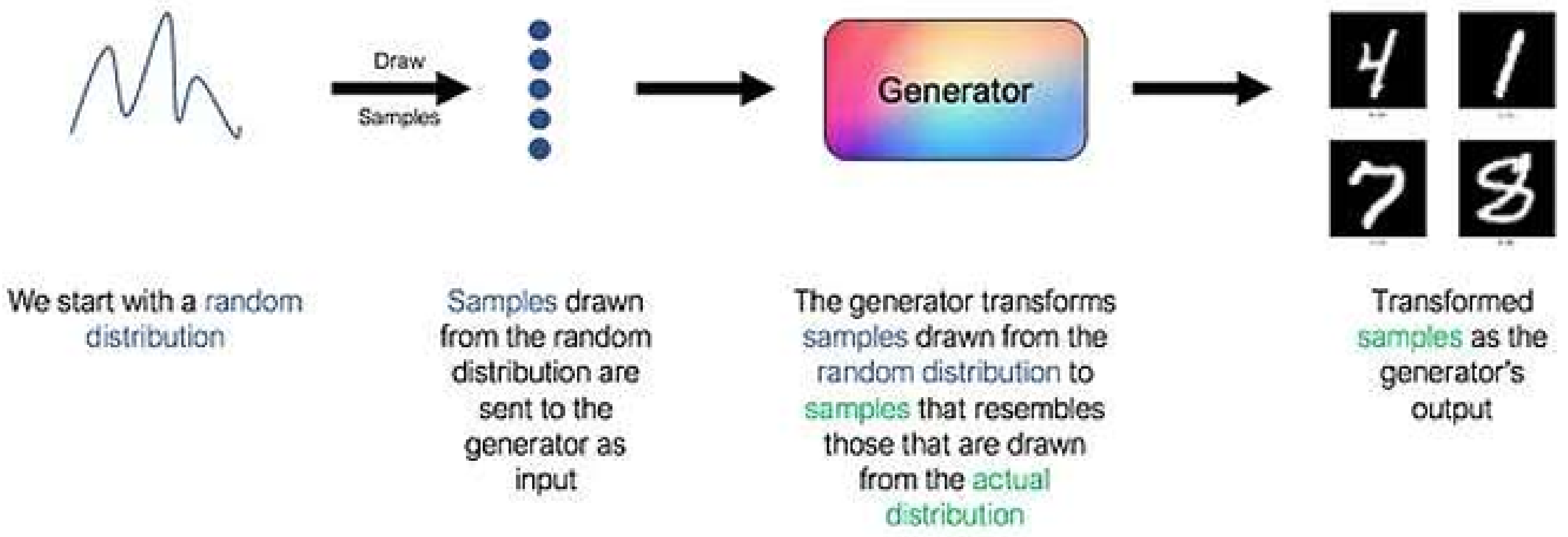


- прогнозирование и принятие решений;
- анализ сложных данных;
- оптимизация процессов;
- распознавание изображений и голосовых записей;
- генерация контента.

- Midjorney;
- Chat GPT;
- Bard;
- Kandinsky;
- Giga chat.

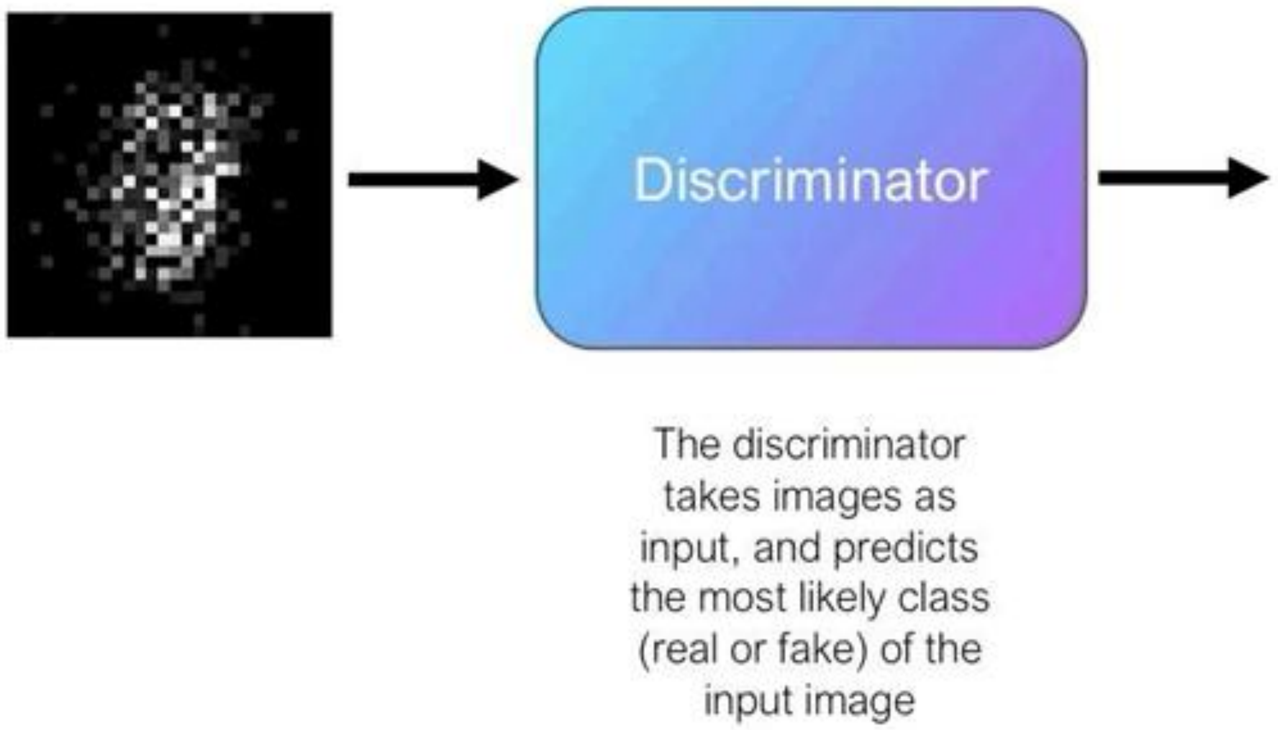
Генератор — модель обучения распределению

В GAN генератор — это нейронная сеть, которая изучает базовое распределение данных. Чтобы быть более конкретным, генератор принимает в качестве входных данных случайное распределение (также известное как «шум» в литературе по GAN) и изучает функцию отображения, которая отображает входные данные в желаемый результат, который является фактическим базовым распределением данных.



Дискриминатор — противник Генератора

Роль дискриминатора состоит в оценке качества выходных изображений генератора. Технически дискриминатор представляет собой двоичный классификатор. Он принимает изображения в качестве входных данных и выводит вероятность того, что изображение является реальным (т. е. фактическим тренировочным изображением) или фальшивым (т. е. полученным от генератора).

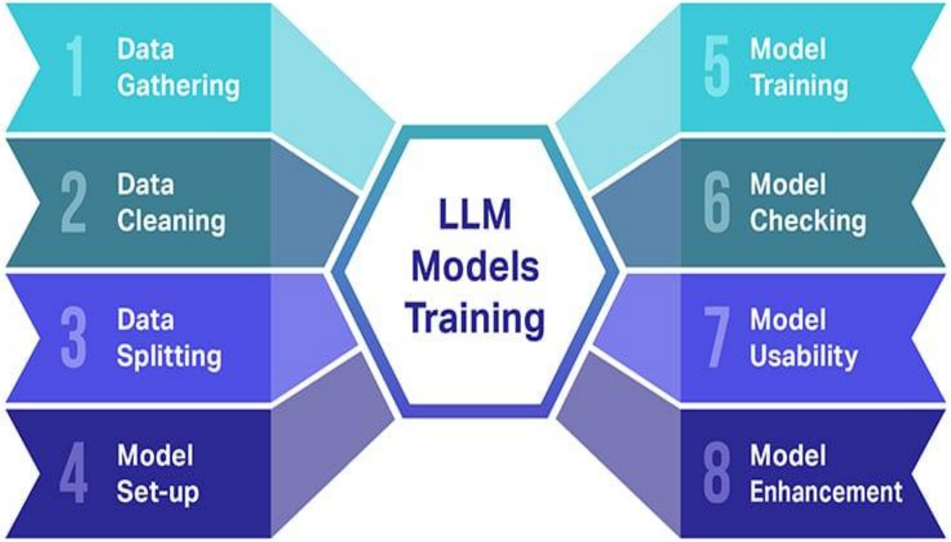


Большие языковые модели LLM

Большие языковые модели (LLM) — это системы искусственного интеллекта (ИИ), предназначенные для обработки, понимания и создания текста.

Варианты использования LLM :

ЭТАПЫ ОБУЧЕНИЯ LLM



1. **Сбор текстовых данных.**
2. **Очистка данных.**
3. **Разделение данных.**
4. **Настройка модели.**
5. **Обучение модели.**
6. **Проверка модели.**
7. **Использование модели.**
8. **Улучшение модели.**

Принцип работы больших языковых моделей

- Модель получает на вход «промпт» (запрос от пользователя) и далее подбирает наиболее подходящее следующее слово.
- После этого полученная строка вновь подается на вход модели и она подбирает еще одно слово.
- И так далее.

Как LLM генерирует связный текст

Допустим, модель начала свой ответ с фразы «**AI сегодня преобразует**». Как она строит разумное предложение дальше?

Embedding (встраивание)

+ **смысл**

AI сегодня преобразует +

общество	4,7%
инновации	3,2%
промышленность	4,3%
медицину	3,5%
все	5,1%

Best Large Language Models in 2025

Модель	Developer	Основные характеристики	Контекстное окно	Особенности и применение	Цена/доступность
GPT-5	Open AI	Унифицированная модель с балансом между скоростью и глубиной рассуждений	До 400,000 токенов	Мультимодальность, высокая точность, подходит для стандартных задач и простых рассуждений.	Бесплатно (ограничено), $20/мес Pro
Claude 4.5 Sonnet	Anthropic	Отлично справляется со сложными и многошаговыми задачами	До 200,000 токенов	Оптимизация для частных задач, длительная автономная работа	Бесплатно (ограничено), $20/мес Pro
Gemini 2.5 Pro	Google	Лидер по мультимодальной обработке (текст, изображение, аудио, видео)	1 000 000 токенов (расширяемое до 2 млн)	Глубокое логическое мышление («Deep Think»), интеграция с экосистемой Google	Коммерческая, не публично указано
Grok 3	X / Илон Маск	Реальное время, доступ к новейшей информации и тенденциям, интеграция с X	Стандартное контекстное окно	Адаптация к задаче с необходимостью самой свежей информации	Коммерческая
Meta Llama 2	Meta Superintelligence Labs	8 млрд и 70 млрд параметров, открытый доступ	Стандартное контекстное окно	Лучшее соотношение цены/качества, наладка под конкретную задачу	Открытый доступ
Mistral	Mistral AI	Быстрая и энергоэффективная модель	Средний размер	Хорошо подходит для предприятий в системах с ограниченными задачами	Коммерческая

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ

Нейросеть	Описание
Convolutional Neural Network (CNN)	Используется для обработки изображений и распознавания паттернов в них. Применяется в компьютерном зрении.
Recurrent Neural Network (RNN)	Подходит для работы с последовательными данными, такими как текст и аудио. Применяется в языковых моделях.
Generative Adversarial Network (GAN)	Используется для генерации новых данных, таких как изображения и звуки. Применяется в синтезе контента.
Transformer	Базовая архитектура для моделей обработки естественного языка, таких как BERT и GPT.
Deep Q-Network (DQN)	Применяется в обучении с подкреплением для игр и задач управления.
U-Net	Широко используется в сегментации изображений, включая медицинское изображение.
YOLO (You Only Look Once)	Одна из популярных архитектур для реального времени обнаружения объектов на изображении.
ResNet	Известная архитектура с нейронными блоками остаточных связей, обычно используется для классификации.
ChatGPT	Разработанный OpenAI, является мощной языковой моделью, основанной на архитектуре GPT (Generative Pre-trained Transformer). ChatGPT способен генерировать тексты и поддерживать разговоры с пользователем, делая его популярным инструментом для задач генерации текста и чат-ботов.

Нейронные сети с прямой связью (FFNN)	Простой тип нейронной сети, в которой информация перемещается только одним путем, от входного слоя к выходному слою через один или несколько скрытых слоев.
Сети с долговременной кратковременной памятью (LSTM)	Форма RNN, созданная для решения проблемы исчезновения градиентов в стандартных RNN. Долгосрочные зависимости в последовательных данных лучше фиксируются с помощью LSTM.
Автоэнкодеры	Неконтролируемая обучающаяся нейронная сеть, в которой сеть обучается воспроизводить свои входные данные на своем выходном слое. Сжатие данных, обнаружение аномалий и шумоподавление изображения могут выполняться с помощью автокодировщиков.
Генеративно-состязательные сети (GAN)	Генеративная нейронная сеть — это форма нейронной сети, которая обучена производить новые данные, сопоставимые с обучающим набором данных. GAN состоят из двух сетей: сети генератора, которая создает свежие данные, и сети дискриминатора, которая оценивает качество созданных данных.

Комментарий к лекции 11

ИНФОРМАЦИЯ В ЗАДАЧАХ ИИ

1. **Что такое информация**
2. **Формы информации**
3. **Формализация понятия данных и знаний, связь между ними**
4. **Свойства знаний.**

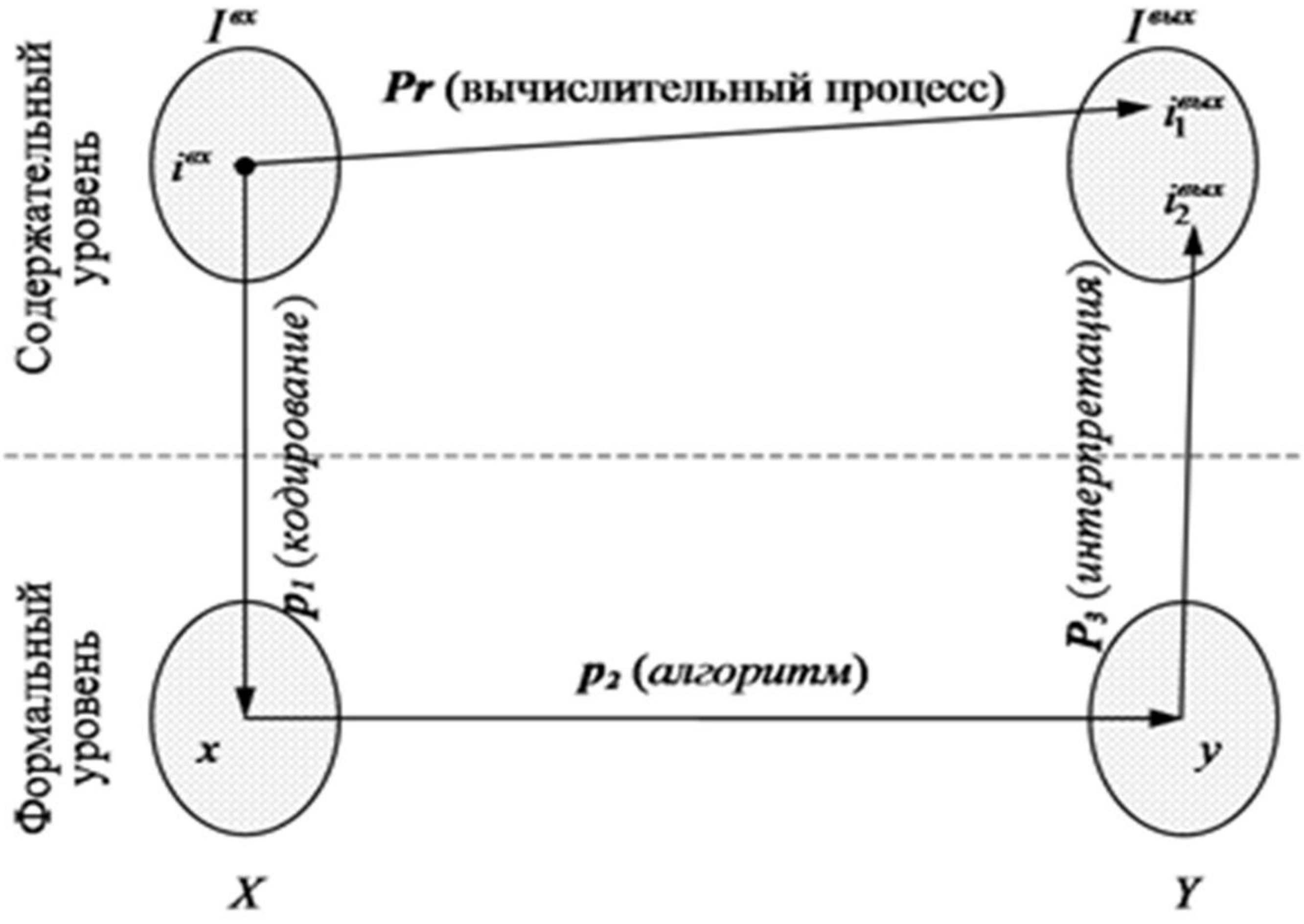
ХАРАКТЕРИСТИКИ МОДЕЛЕЙ ИИ

СВОЙСТВА

ЯЗЫК НЕЧЕТКОЙ МАТЕМАТИКИ

	Индуктивность	Знания	Эволюция	Достоинства	Недостатки
ЛОГИЧЕСКИЕ МОДЕЛИ	+	+	?		
ГЕНЕТИЧЕСКИЕ МОДЕЛИ	+	+	+		
НЕЙРОННЫЕ СЕТИ	+	+	+		
МОДЕЛИ РАСПОЗНАВАНИЯ	+	+	+		

Понятие **информации** связано с **процессом решения** задачи

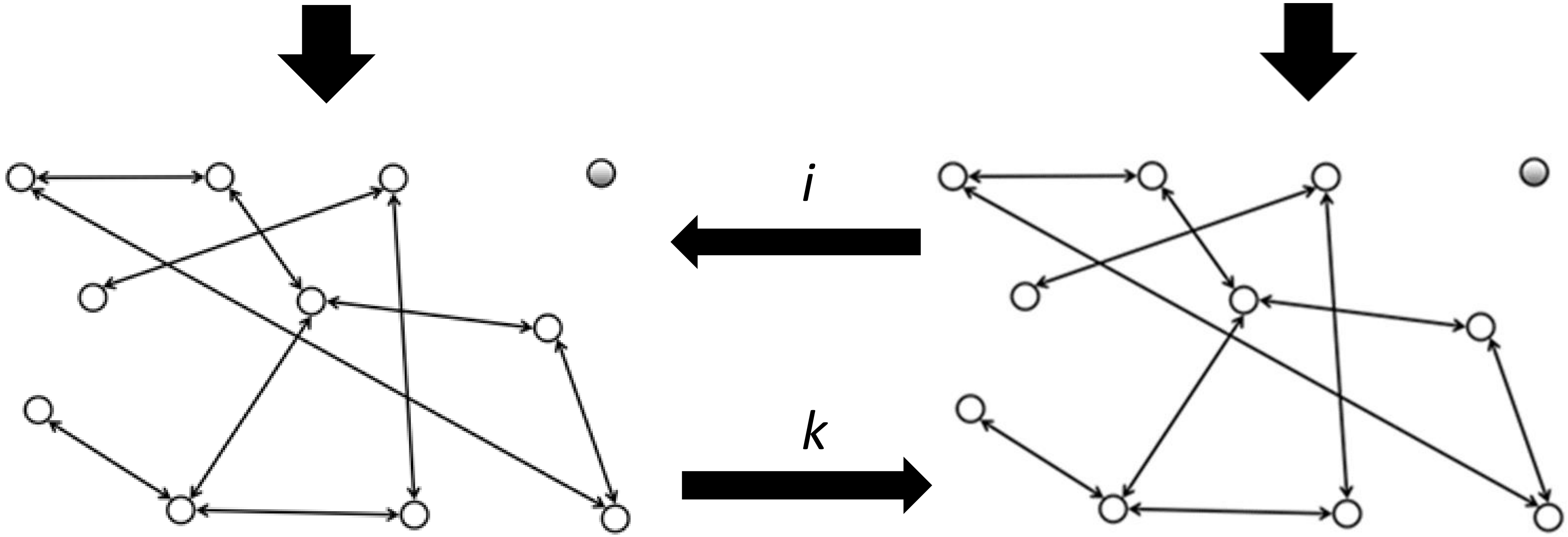


$$Z_{сод} \subseteq I^{вх} \times I^{вых}$$

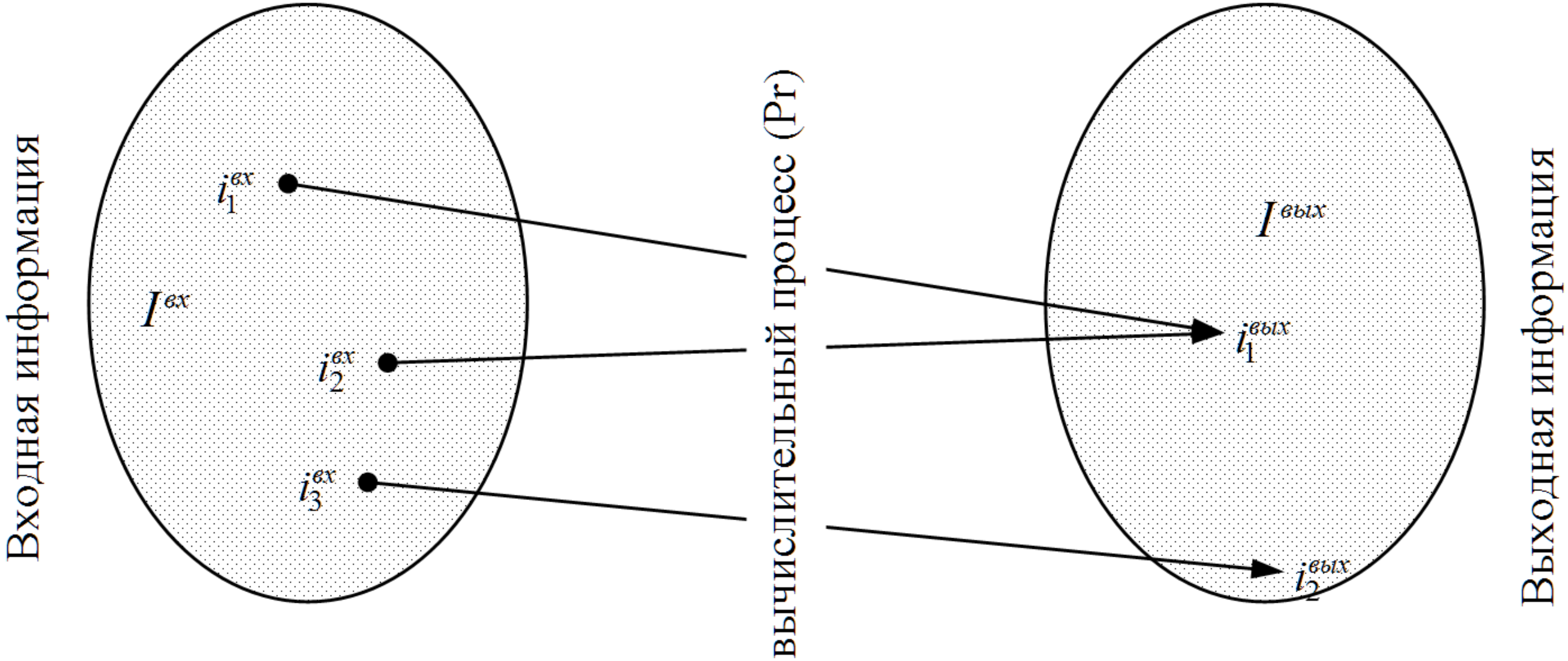
$$Z_{\text{форм}} \subseteq X \times Y$$

Процесс решения задачи можно описать в терминах взаимодействия объектов

$I^{вх} \times I^{вых}$ $\qquad$ $X \times Y$



Содержательный уровень $\qquad$ Формальный уровень


Входная информация
$I^{вх}$
$i_1^{вх}$
$i_2^{вх}$
$i_3^{вх}$
вычислительный процесс (Pr)
$I^{вых}$
$i_1^{вых}$
$i_2^{вых}$
Выходная информация

Информация— *сведения*, воспринимаемые человеком или устройством, как отражение фактов материального мира в процессе коммуникации.

Информация— *знания* относительно фактов, событий, вещей, идей и понятий, которые в определённом контексте имеют конкретный смысл.

Информация должна обрести некоторую *форму представления* (то есть превратиться в *данные*), чтобы ей можно было обмениваться.

Термин «**информация**» —общенаучное понятие, включающее обмен *сведениями* между людьми, устройствами, человеком и устройством; обмен *сигналами* в животном и растительном мире (кибернетика)…

Информацию можно разделить на виды:

по ***объектам*** *информационного* ***взаимодействия***:

Человек — человек.

Человек — устройство.

Устройство— устройство.

Обмен сигналами в животном и растительном мире.

по ***способу восприятия***:

Визуальная — воспринимаемая органами зрения.

Звуковая — воспринимаемая органами слуха.

Тактильная — воспринимаемая тактильными рецепторами.

Обонятельная — воспринимаемая обонятельными рецепторами.

Вкусовая — воспринимаемая вкусовыми рецепторами.

По форме представления:

Текстовая — передаваемая в виде символов.

Числовая — в виде цифр и знаков.

Графическая — в виде изображений, предметов, графиков.

Звуковая —в виде записи и передачи лексем языка аудиальным путём.

Видеоинформация — передаваемая в виде видеозаписи.

По назначению:

Массовая — содержит сведения, понятные большей части социума.

Специальная — содержит сведения, не понятные части социума.

Секретная — передаваемая узкому кругу лиц.

Личная (приватная) — набор сведений о какой-либо личности.

По значению:

актуальная — информация, ценная в данный момент времени.
достоверная — информация, полученная без искажений.
понятная — информация, понятная тому, кому она предназначена.
полная — информация, достаточная для принятия правильного решения или понимания.
полезная — полезность информации определяется субъектом, получившим информацию в зависимости от объёма возможностей её использования.

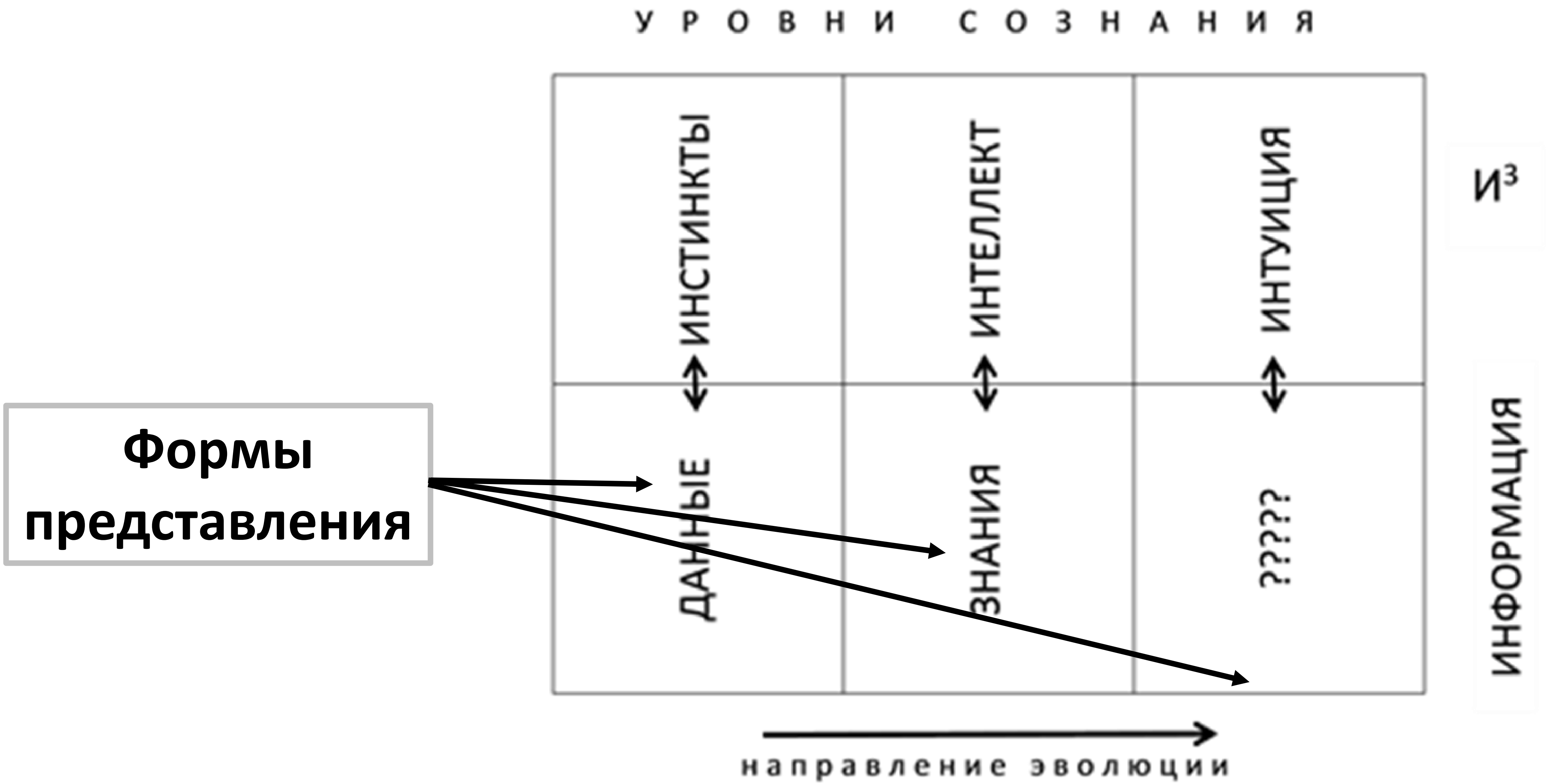
По истинности:

истинная
ложная

Информация – сведения о взаимодействии объектов в процессе решения задачи, представленные в любой **доступной** человеку или устройству **форме**.

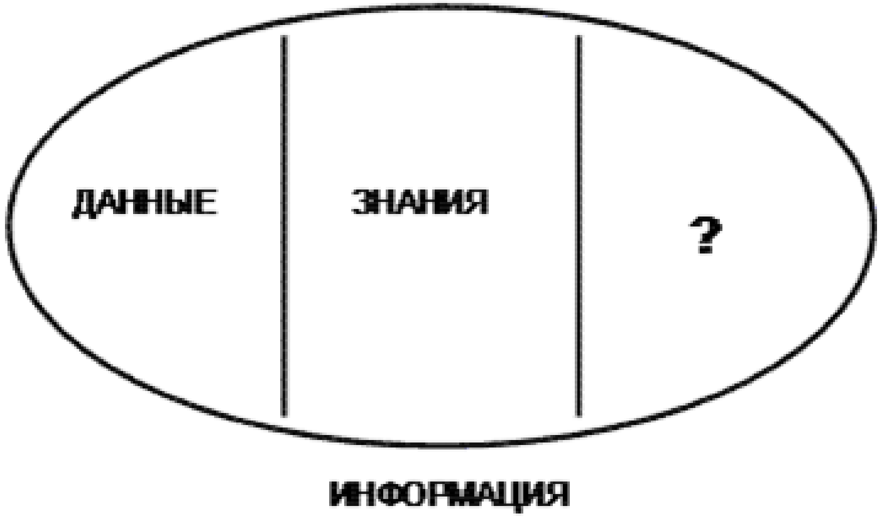
Доступность означает, что они могут быть закодированы и проинтерпретированы.

Формы представления ???



У Р О В Н И С О З Н А Н И Я
ИНСТИНКТЫ
ИНТЕЛЛЕКТ
ИНТУИЦИЯ
И³
Формы представления
ДАННЫЕ
ЗНАНИЯ
?????
ИНФОРМАЦИЯ
направление эволюции

$$информация = данные \cup знания \cup ???$$

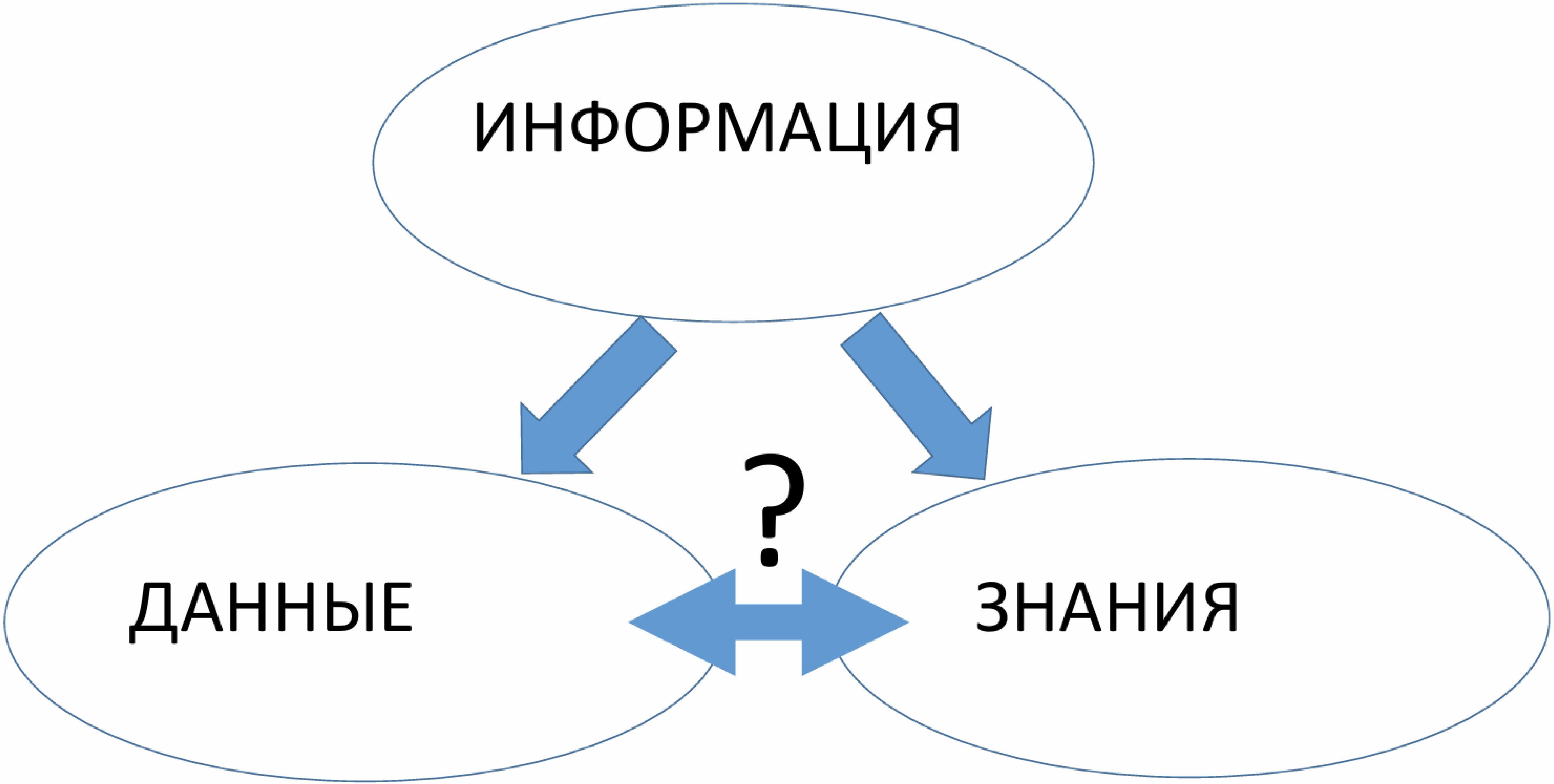


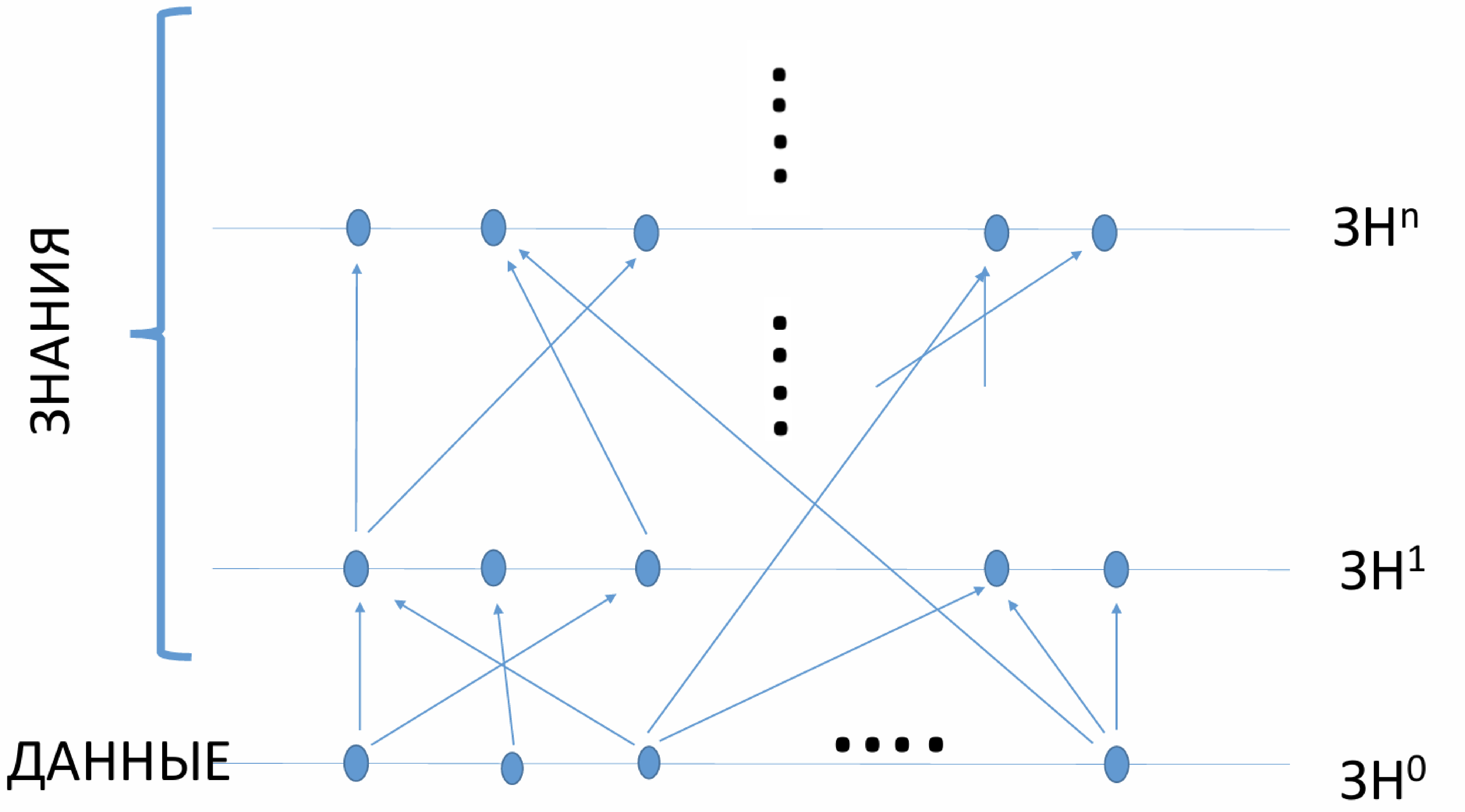
PS. $информация \Leftrightarrow энергия$

PS. $\text{\textit{информация}} \Leftrightarrow \text{\textit{энергия}}$

Энергия — скалярная физическая величина, являющаяся единой мерой различных форм движения и взаимодействия материи, мерой перехода движения материи из одних форм в другие.

Информация — скалярная величина, являющаяся единой мерой ……?? (*для различных способов измерения энергии*)


ИНФОРМАЦИЯ
?
ДАННЫЕ
ЗНАНИЯ


ЗНАНИЯ
ДАННЫЕ
ЗН^n
ЗН^1
ЗН^0

Формализация понятий данных и знаний

$$объект: \pi_1 \times \ldots \times \pi_n \to D_1 \times \ldots \times D_n$$

π_i - наименование признака

D_i - множество его возможных значений

$n \in N$ - арность объекта

$$объект_j \leftrightarrow объект_k;\ j,k \in N$$

В настоящее время наиболее распространенными и логически завершенными являются следующие концепции знаний:

- первая основывается на постулате: ***знания есть интерпретированные данные***;
- вторая базируется на выборе определенного способа представления исходной информации. Наиболее распространенными среди них являются *фреймы*, *семантические сети* и *“аксиоматический” способ*. Основной постулат данной концепции : знанием является все то, что **можно получить на основе перечисленных представлений с помощью подходящих механизмов вывода.**

МОДЕЛИ ЗНАНИЙ В ИИ

1. **Измерение информации**
2. **Использование энтропии для построения деревьев классификации**
3. **Классификация моделей представления**
4. **Краткое описание моделей**

В общем случае:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048	4096

$$I(N) = \log_2 N$$

N – число событий, объектов …
(равновероятных) (Хартли)

$$I(p) = -\log_2 p$$

При $p = (N)^{-1}$

k типов

$$I = -\sum_{i=1}^{k} p_i \log_2 p_i$$

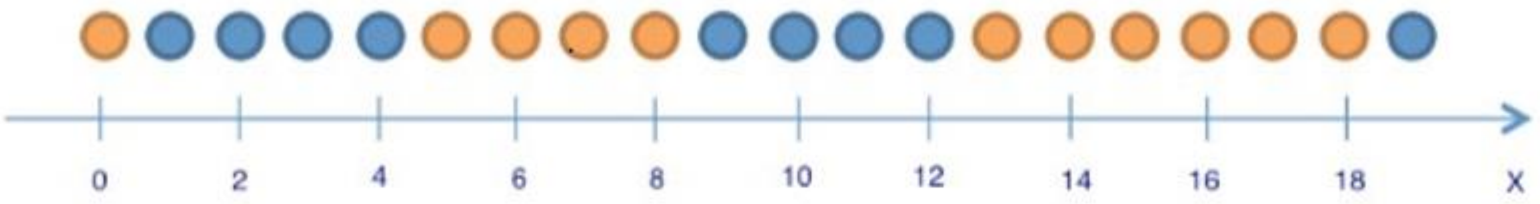
Шэннон, энтропия

Алгоритм обучения: *нужно находить правила (предикаты), на основе которых разбивать тренировочный набор данных, таким образом, чтобы уменьшалось среднее значение энтропии*.

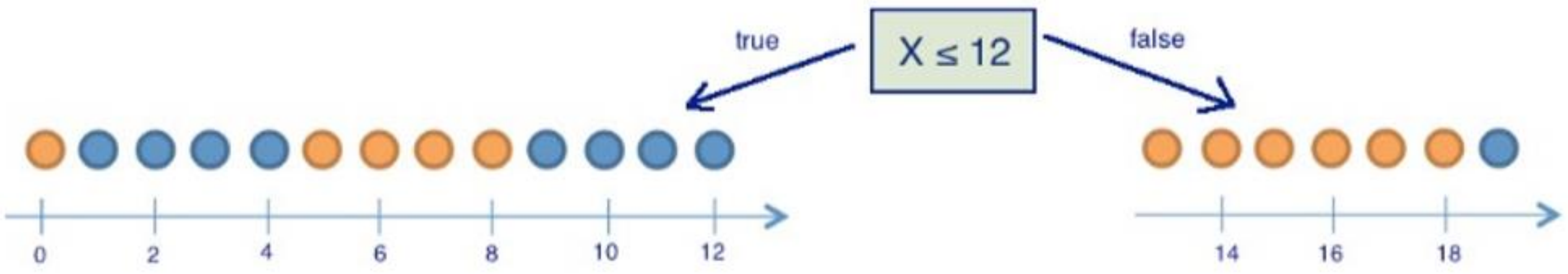
Процесс деления множества данных на части, приводящий к уменьшению энтропии, можно рассматривать как ***производство информации***.

Разбив исходный набор данных на две части **по некому предикату**, можно рассчитать энтропию каждого подмножества, после чего рассчитать среднее значение энтропии — если оно окажется меньшим чем энтропия исходного множества, значит предикат содержит некую обобщающую информацию о данных.

<u>Пример</u>: необходимо построить разбиение заданного множества шариков на подмножества, чтобы в одно подмножество попадали шарики одного цвета (предсказание цвета)



$$I_0 = -\frac{9}{20}\log_2\frac{9}{20} - \frac{11}{20}\log_2\frac{11}{20} \approx 1$$



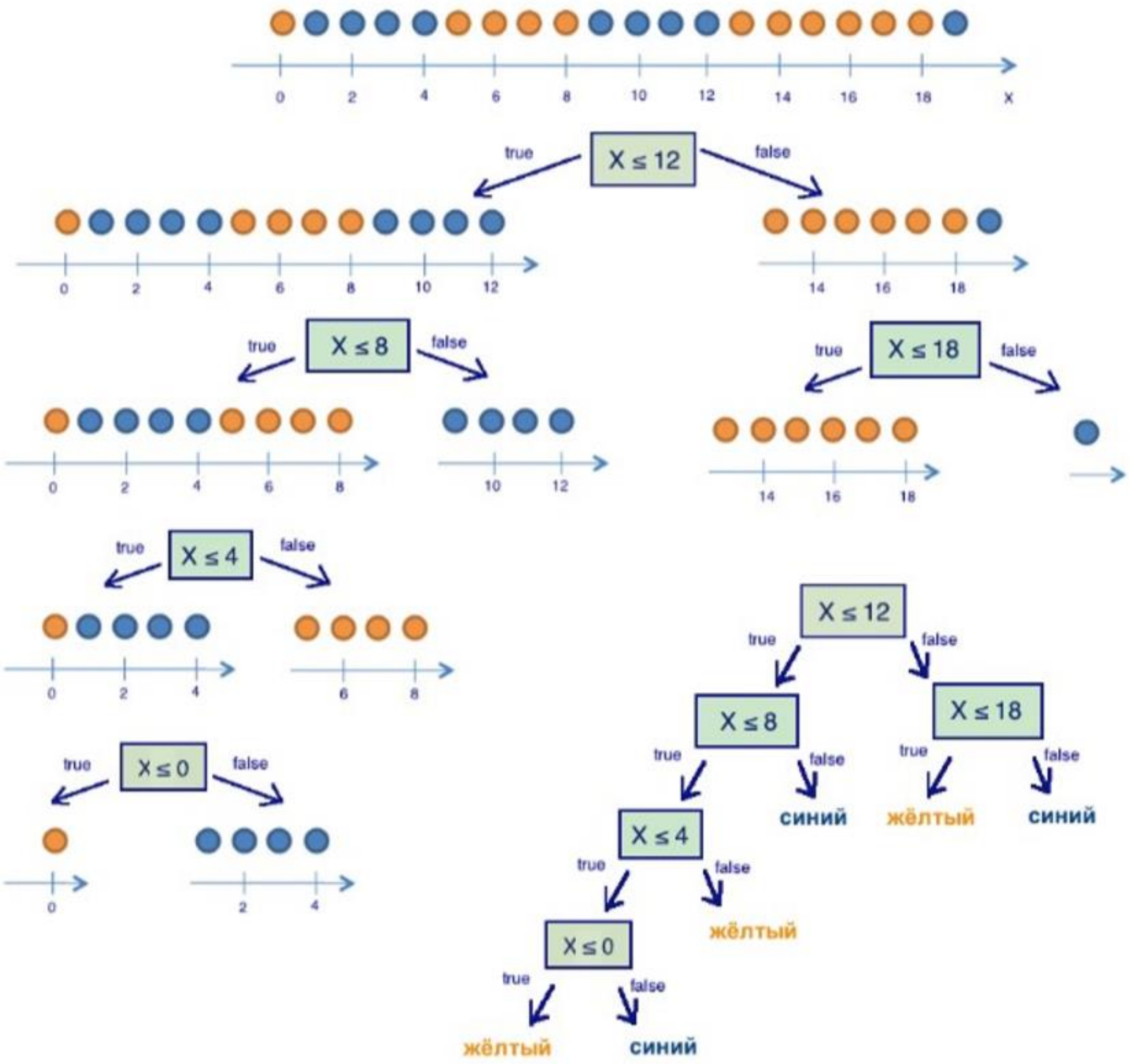
$$I_1 = -\tfrac{5}{13}\log_2 \tfrac{5}{13} - \tfrac{8}{13}\log_2 \tfrac{8}{13} \approx 0.96$$

$$I_2 = -\tfrac{1}{7}\log_2 \tfrac{1}{7} - \tfrac{6}{7}\log_2 \tfrac{6}{7} \approx 0.6$$

Прирост информации (Information gain, IG)

$$IG(x) = I_0 - \sum_{i=1}^{|x|} \frac{N_i}{N} \cdot I_i$$

$$IG(x \le 12) = I_0 - \frac{13}{20} \cdot I_1 - \frac{7}{20} \cdot I_2 \approx 0.16$$

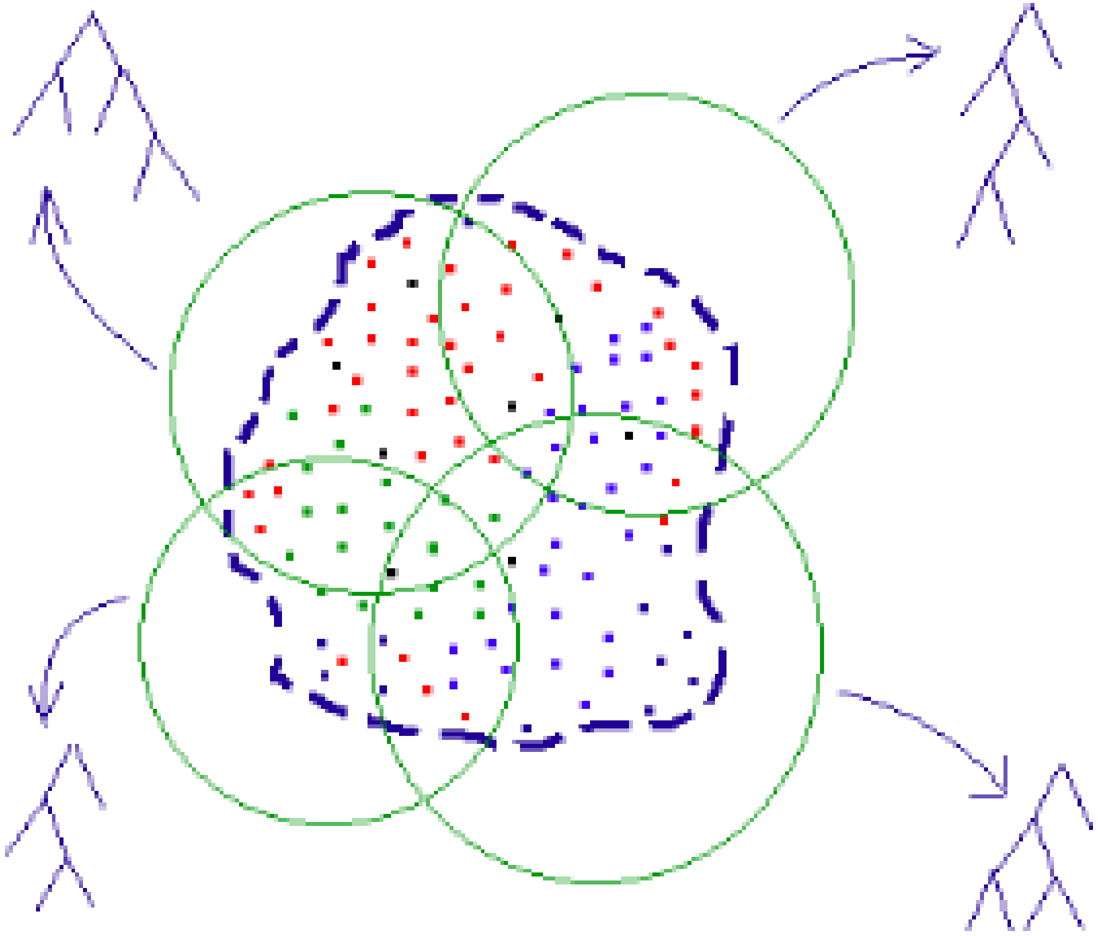


X
X ≤ 12
true
false
X ≤ 8
X ≤ 18
X ≤ 4
X ≤ 0
синий
жёлтый

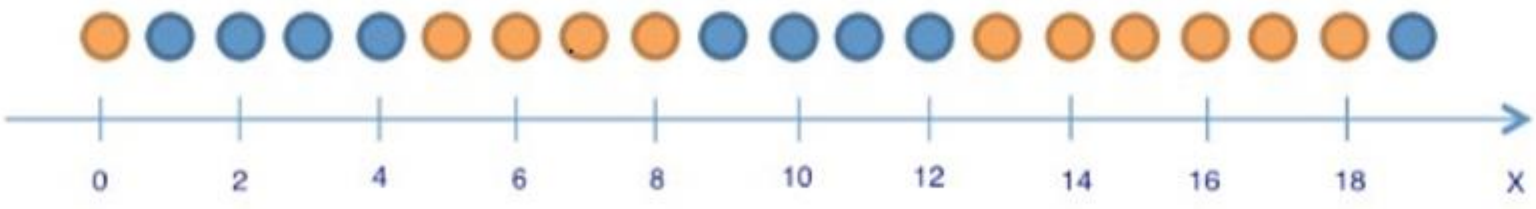
Алгоритм построения дерева классификации

```
s0 = вычисляем энтропию исходного множества

Если s0 = 0 значит:
   Все объекты исходного набора, принадлежат к одному классу
   Сохраняем этот класс в качестве листа дерева

Если s0 ≠ 0 значит:
   Перебираем все элементы исходного множества:
      На основе каждого элемента генерируем предикат, который разбивает исходное
множество на два подмножества
      Рассчитываем среднее значение энтропии
      Вычисляем ΔS
   Нас интересует предикат, с наибольшим значением ΔS
   Найденный предикат является частью дерева принятия решений, сохраняем его

   Разбиваем исходное множество на подмножества, согласно предикату

   Повторяем данную процедуру рекурсивно для каждого подмножества
```

RANDOM FOREST



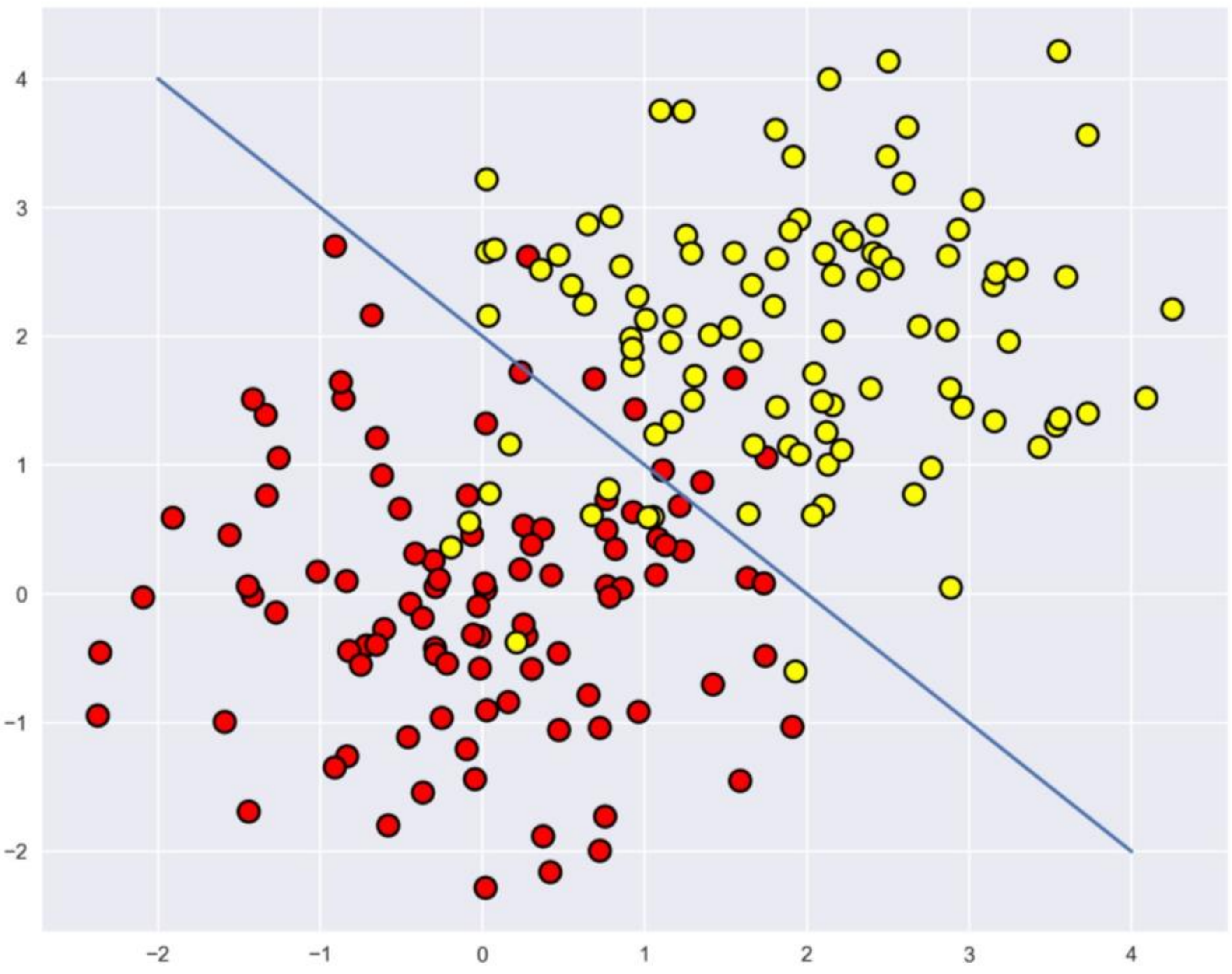
Продолжение примера:

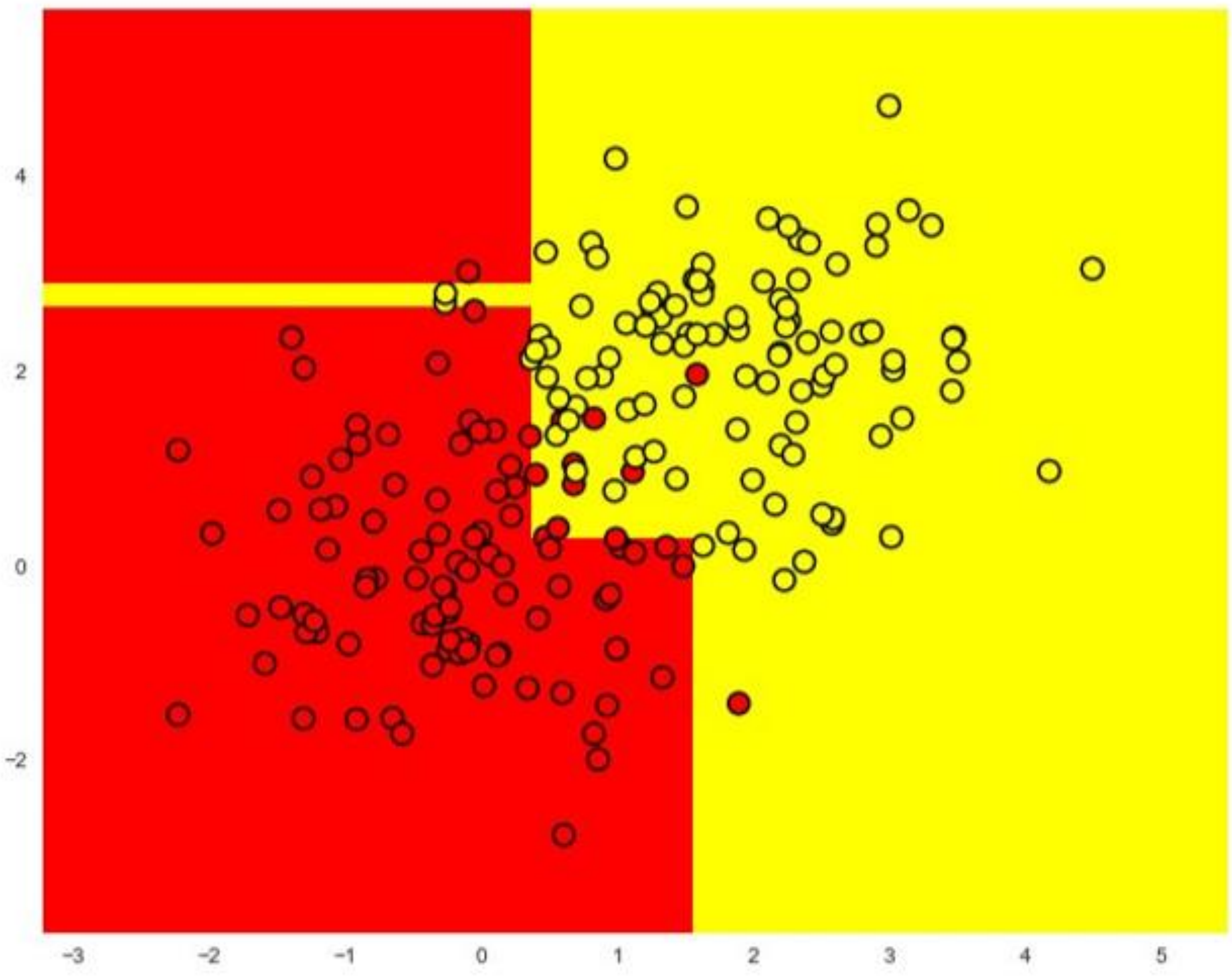


Введем множество $X = \{X_1, X_2\}$
X_1 − подмножество шариков желтого цвета
X_2 − подмножество шариков синего цвета
Оценим количество **знаний** об X с помощью формулы Хартли:

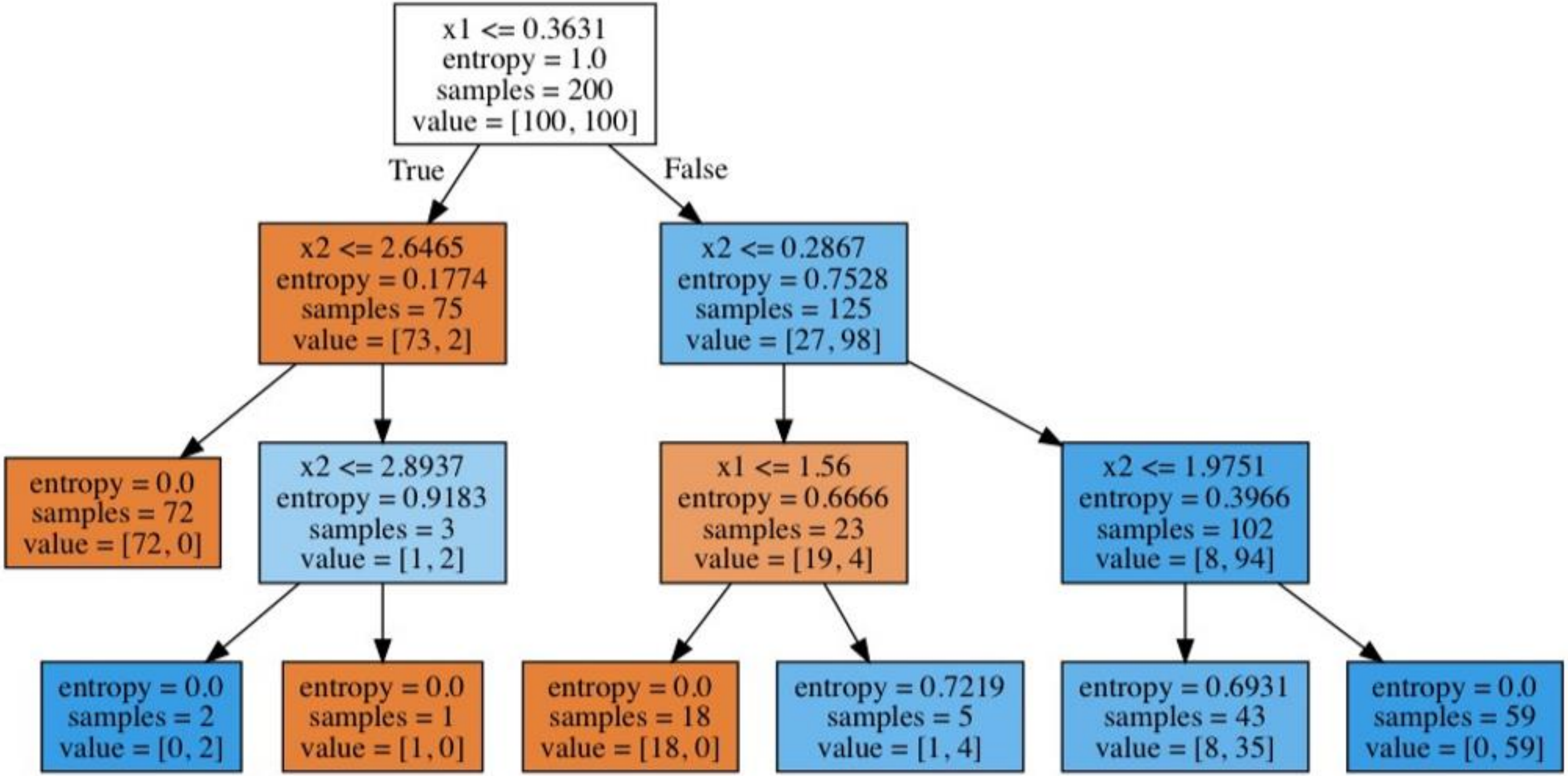
$$\mathrm{I(X)} = log_2|\mathrm{X}|$$

Пример: задача распознавания с обучением





4
2
0
−2
−3
−2
−1
0
1
2
3
4
5



x1 <= 0.3631
entropy = 1.0
samples = 200
value = [100, 100]
True
False
x2 <= 2.6465
entropy = 0.1774
samples = 75
value = [73, 2]
x2 <= 0.2867
entropy = 0.7528
samples = 125
value = [27, 98]
entropy = 0.0
samples = 72
value = [72, 0]
x2 <= 2.8937
entropy = 0.9183
samples = 3
value = [1, 2]
x1 <= 1.56
entropy = 0.6666
samples = 23
value = [19, 4]
x2 <= 1.9751
entropy = 0.3966
samples = 102
value = [8, 94]
entropy = 0.0
samples = 2
value = [0, 2]
entropy = 0.0
samples = 1
value = [1, 0]
entropy = 0.0
samples = 18
value = [18, 0]
entropy = 0.7219
samples = 5
value = [1, 4]
entropy = 0.6931
samples = 43
value = [8, 35]
entropy = 0.0
samples = 59
value = [0, 59]


Методы представления знаний
степень структурированности знания
Правила
Объекты
степень определенности (формальности)
степень операционности
Логическая модель(предикаты)
Продукционная модель
Семантическая сеть
степень динамичности
Нечеткая логика
Условные вероятности
Фреймы
степень динамичности
Статическая
Динамическая
Объектно-ориент. модель

Логическая модель предполагает унифицированное описание объектов и действий в виде **предикатов первого порядка**. Под предикатом понимается логическая функция на N - аргументах (признаках), которая принимает истинное или ложное значение в зависимости от значений аргументов. Для объектов соответствующие отношения задаются явно в виде фактов, а действия описываются как правила, определяющие логическую формулу вывода фактов из других фактов.

Продукционные модели используются для решения более сложных задач, которые основаны на применении эвристических методов представления знаний, позволяющих настраивать механизм вывода на особенности проблемной области и учитывать неопределенность знаний.

Для ее реализации в описание продукций вводятся предусловия и постусловия в виде:

$$A, B, C \Rightarrow D, F$$

$C \Rightarrow D$ - продукционное правило

A - предусловие выбора класса правил

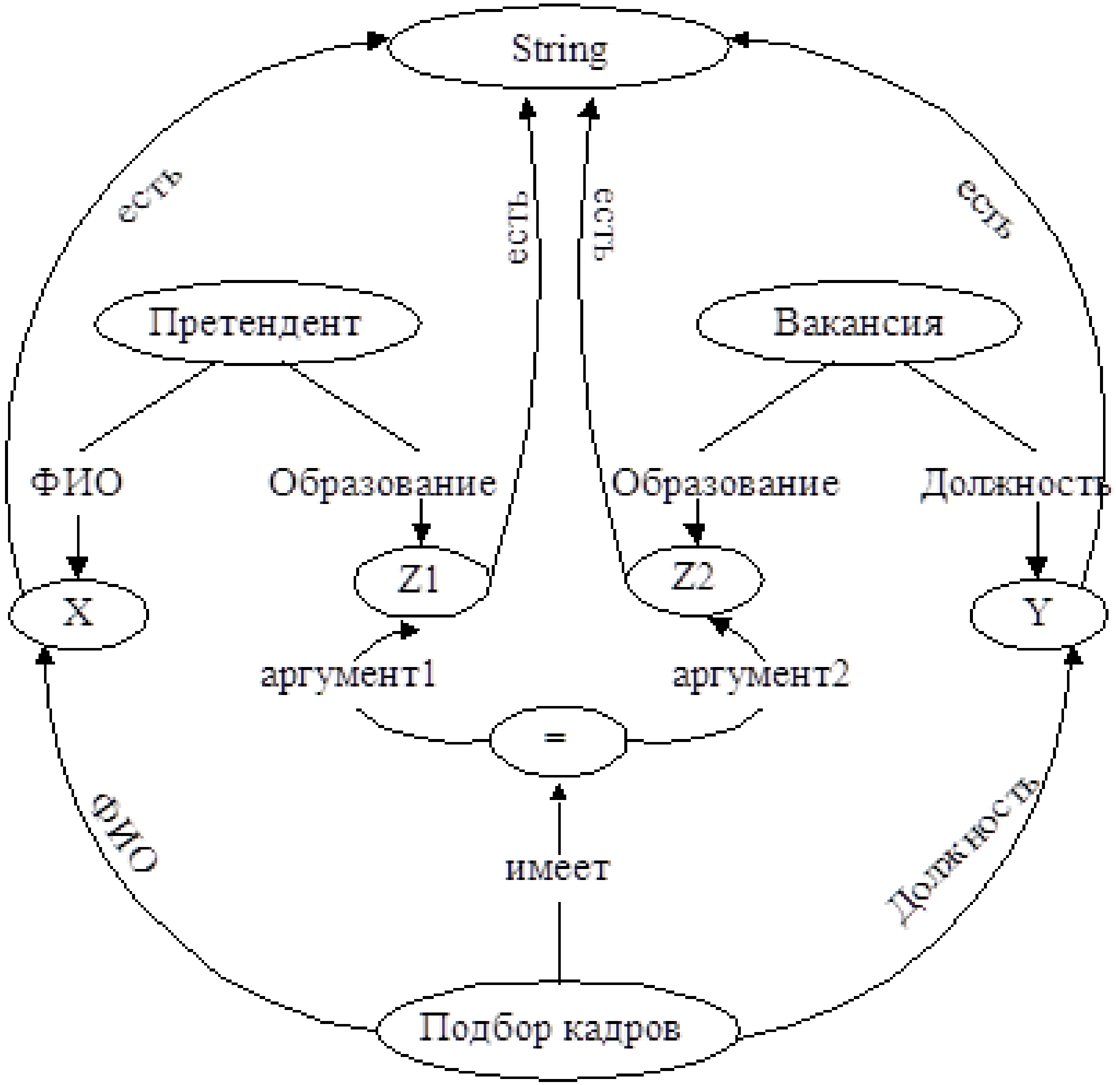
B - предусловие выбора правила в классе

F - постусловие правила, определяющее переход

Языки логического программирования ПРОЛОГ, ЛИСП.

Семантические сети. Вариант объектно-ориентированных методов представления знаний, в которых типизируются отношения между объектами. Элементарной единицей знаний в семантической сети служит триплет, в котором имя предиката представляет помеченную дугу между двумя узлами графа, соответствующими двум связанным объектам.

Пример.


String
есть
есть
есть
есть
Претендент
Вакансия
ФИО
Образование
Образование
Должность
X
Z1
Z2
Y
аргумент1
аргумент2
=
ФИО
имеет
Должность
Подбор кадров

Фреймы. Являются развитием семантических сетей. Методы представления знаний во фреймах базируются на сборе в одну структуру всех атрибутов (поименованных отношений) объектов. Такая структура называется фреймом. Причем в качестве значений слотов (атрибутов) могут выступать как обычные значения данных, так и действия, направленные на получение этих значений. Действия реализуются в виде присоединенных процедур или процедур-демонов, вызываемых по определенным условиям.

Пример.

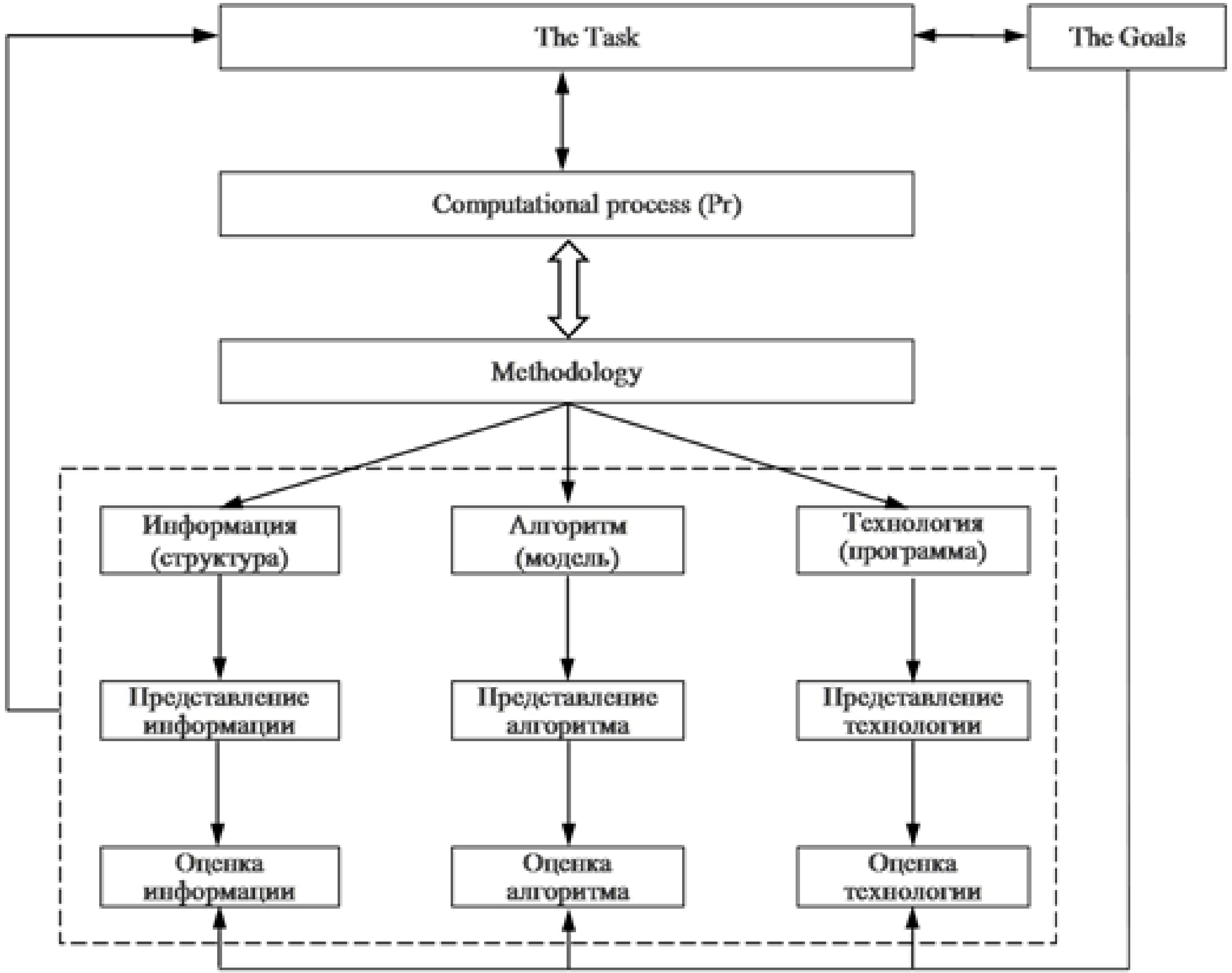
Фамилия	Год рождения	Специальность	Стаж, число лет
Попов	1965	Слесарь	5
Сидоров	1946	Токарь	20
Иванов	1925	Токарь	30
Петров	1937	Сантехник	25

значение фрейма

(Список работников:
Фамилия (Попов - Сидоров - Иванов - Петров);
Год рождения (1965 - 1946 - 1925 - 1937);
Специальность (слесарь - токарь - токарь - сантехник);
Стаж (5 - 20 - 30 - 25)).

фрейм

(Список работников:
Фамилия (значение слота 1);
Год рождения (значение слота 2);
Специальность (значение слота 3);
Стаж (значение слота 4)).

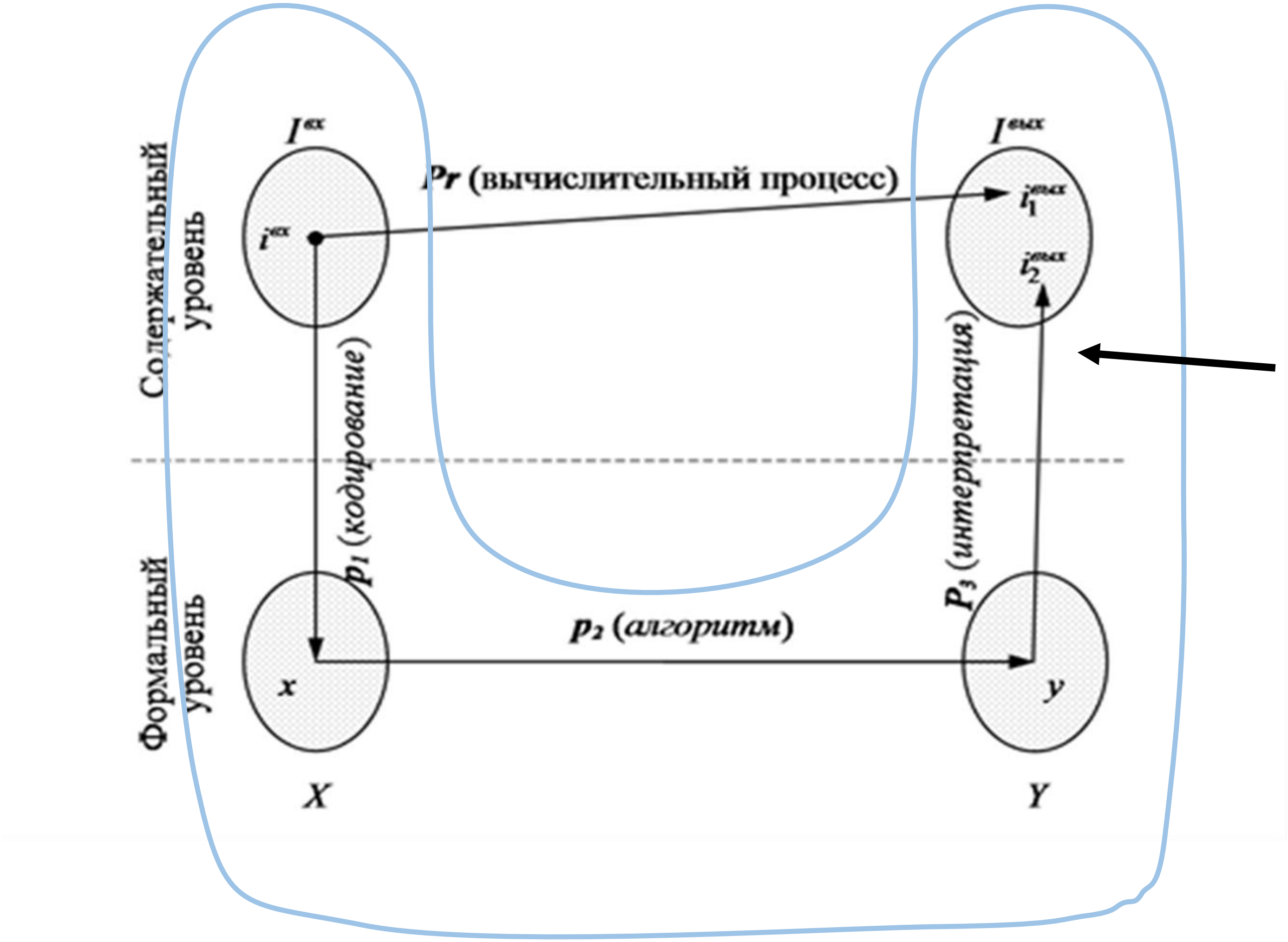


Методология решения задач ИИ

ПРОГРАММЫ ДЛЯ ЗАДАЧ ИИ

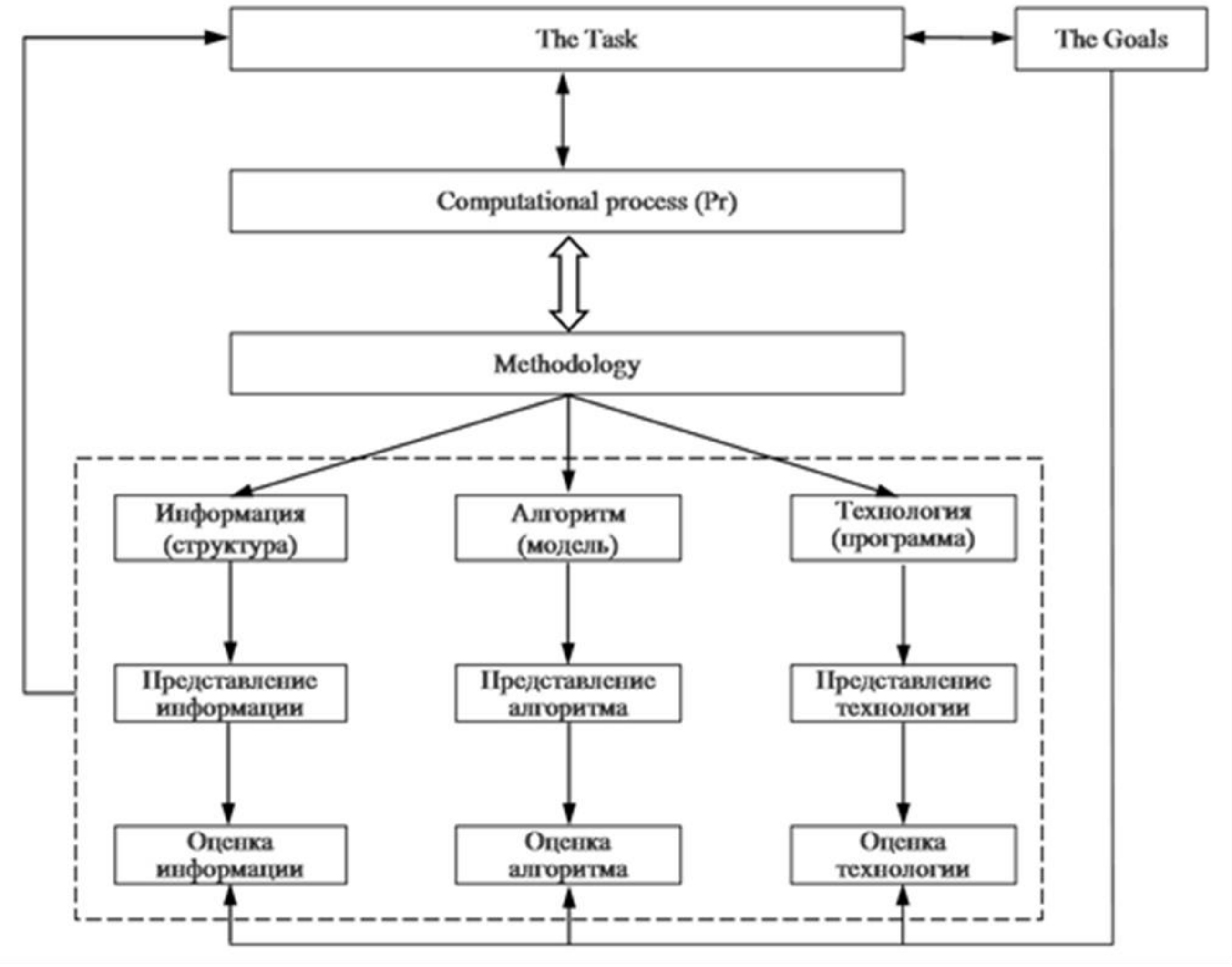
1. **Задачи ИИ и программы**
2. **Концепция системы, основанной на знаниях**
3. **Структура Knowledge Base System (KBS)**

Место программы в процессе решения задачи



Программа – среда и средства, с помощью которых поддерживается решение задачи

Программа должна поддерживать методологию решения



<u>Свойства</u>:

1. Индуктивность
2. Возможность работы со знаниями (и данными)
3. Эволюция программы

ПРИМЕР:

Место программы с точки зрения автоматизации процесса решения задачи

- с точки зрения пользователя ЭВМ и ППО неотделимы и должны предоставлять пользователю средства общения на языке максимально приближенном к естественному;
- схема решения задачи, которая поддерживается ППО, не зависит от технологии и имеет следующий вид:

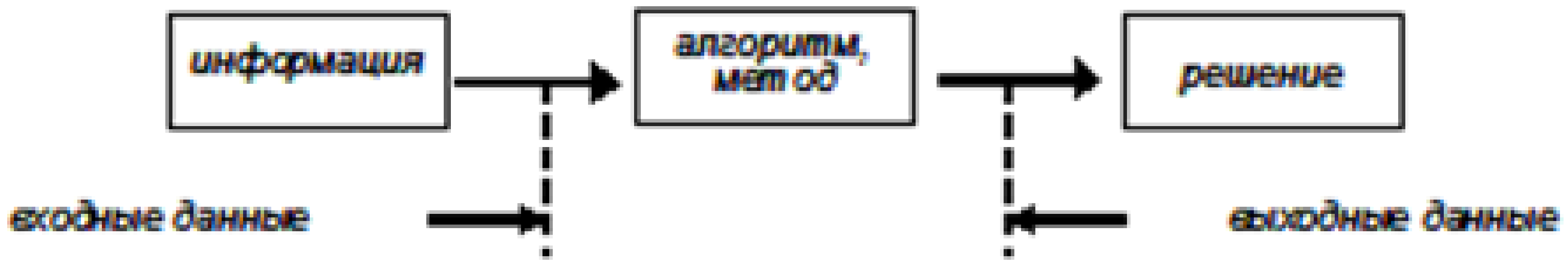


Рис.3

Технологические предпосылки

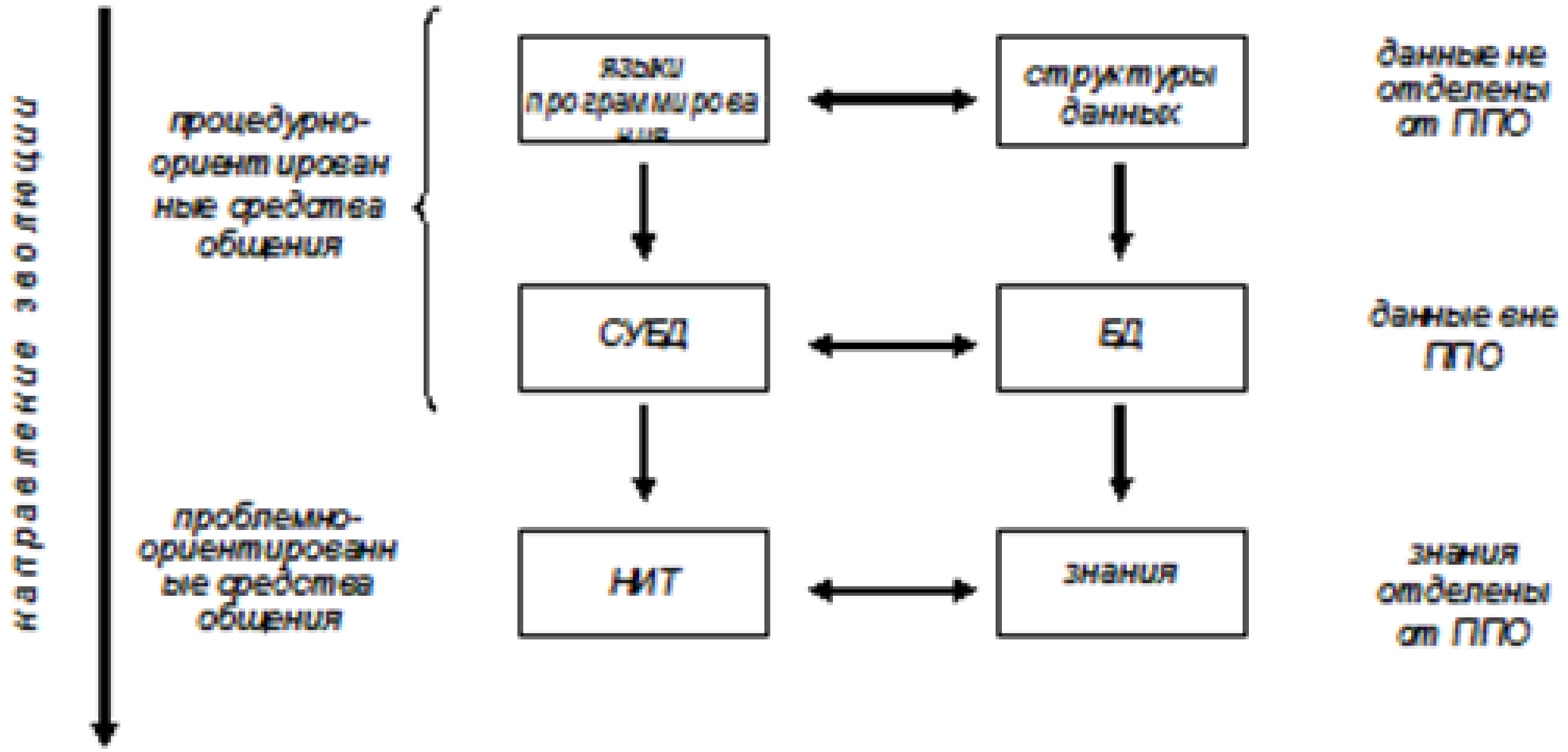


Рис. 4

Место KBS и ее функции

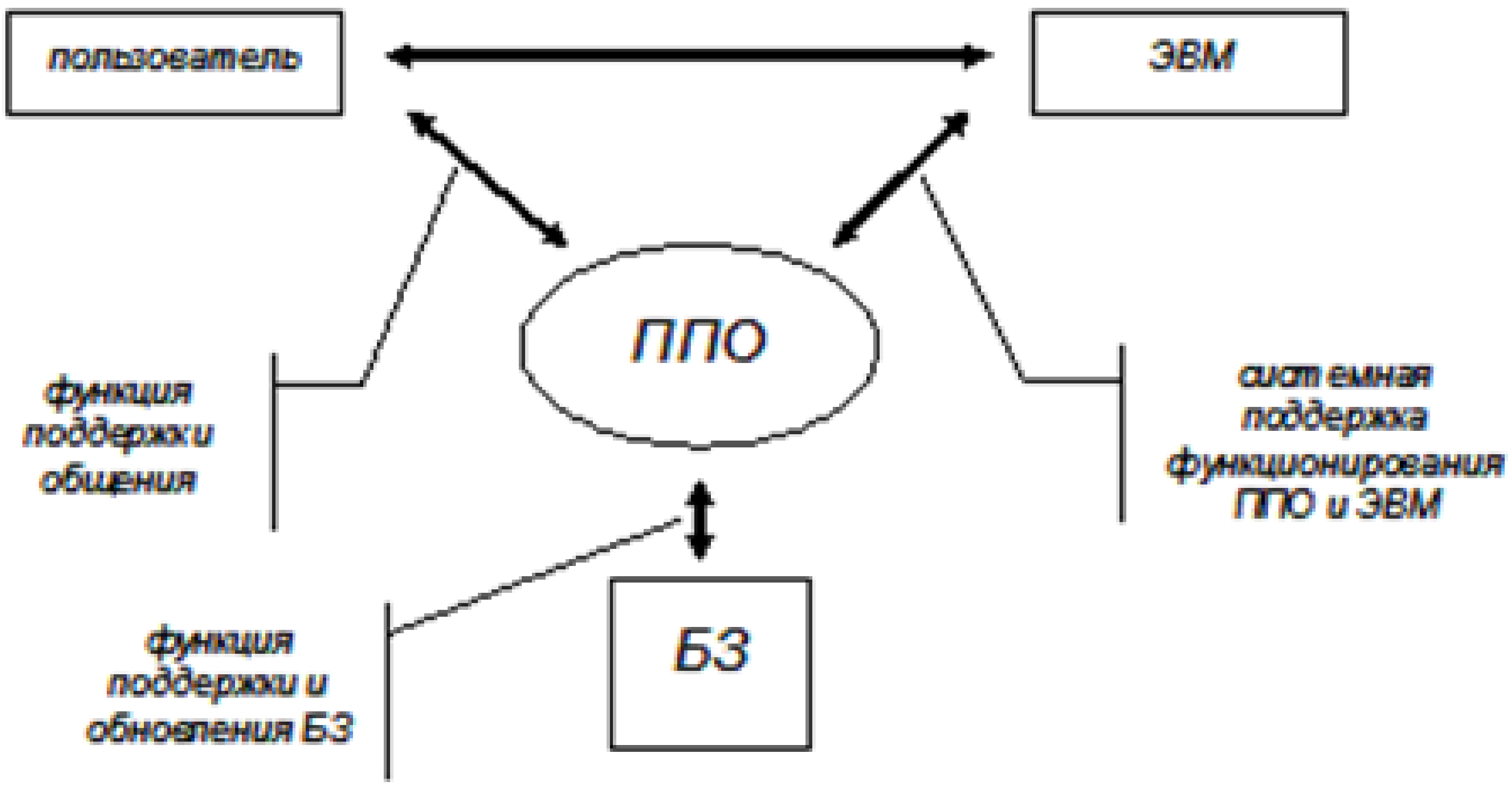


Рис. 5

KBS

Функции:

- **ведение диалога**. Определение его структуры и той роли, которую пользователь и интерфейс выполняют на текущем шаге диалога;
- **понимание**. Преобразование поступающих от пользователя высказываний во внутреннее представление;
- **обработка высказываний**. Формирование и определение заданий на данном шаге диалога;
- **интерпретация**. Формирование выходных высказываний на ЕЯ.

обобщенная схема ЕЯ – интерфейса

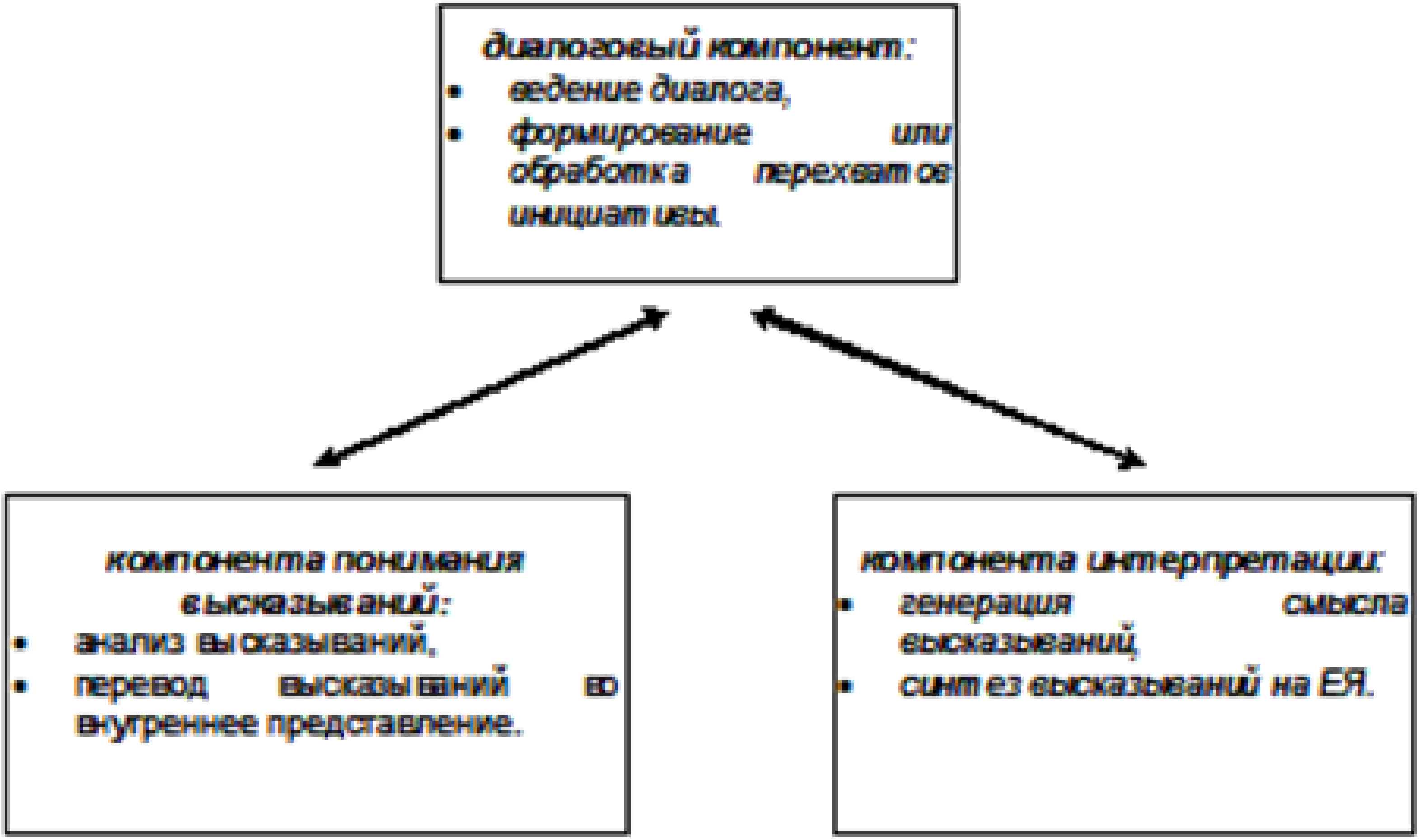


Рис. 6

Ключевым элементом KBS является БЗ

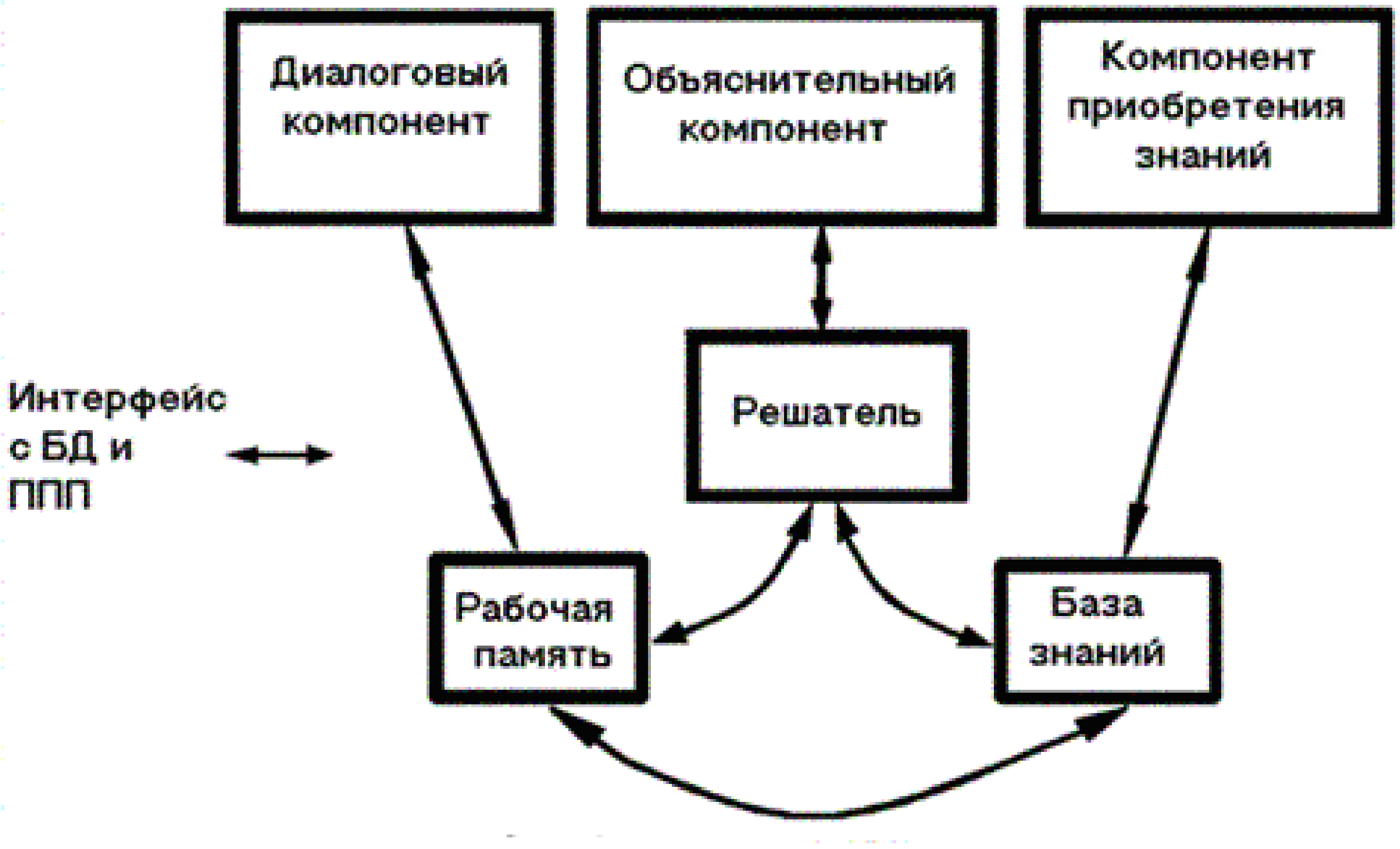
Для построения БЗ:

- проблему представления знаний;
- проблему манипуляции знаниями.

с функциональной (рис. 5) и проблемной (рис. 3) точек зрения БЗ должна содержать:

- компоненту, поддерживающую ЕЯ – интерфейс;
- компоненту, поддерживающую общие знания о предметной области и обеспечивающую хранение и ввод входной информации;
- компоненту, поддерживающую хранение и необходимые манипуляции выходной информацией;
- компоненту, обеспечивающую необходимыми знаниями системную поддержку функционирования KBS и ЭВМ.

ЭКСПЕРТНАЯ СИСТЕМА



ЭТАПЫ РАЗРАБОТКИ ЭС