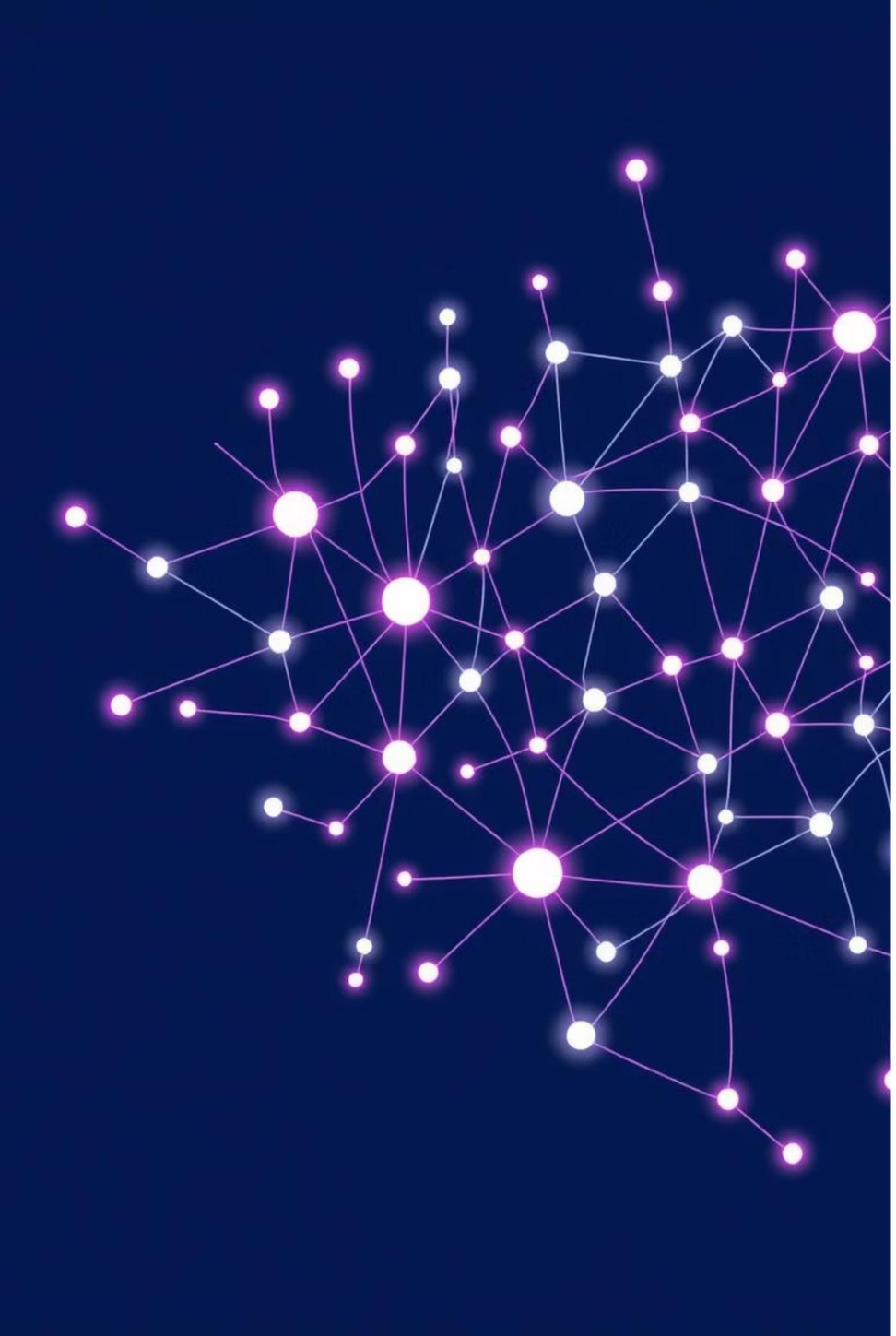
ТУСУР
TUSUR UNIVERSITY

Информатика. Введение.

Лекция 1

Доцент каф. АСУ, к.т.н:
Суханов А.Я.



Литература.

Информатика и информационные технологии : учебник для вузов / М. В. Гаврилов, В. А.Климов. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 355 с. [Электронный ресурс] : — Р е ж https://urait.ru/viewer/informatika-i-informacionnye-tehnologii-535560#page/1.

Информатика : учебное пособие для вузов / В. К. Волк. — 2-е изд., перераб. и доп. Москва : Издательство Юрайт, 2024. — 226 с.[Электронный ресурс]: — Режим доступа: https://urait.ru/viewer/informatika-534979#page/1

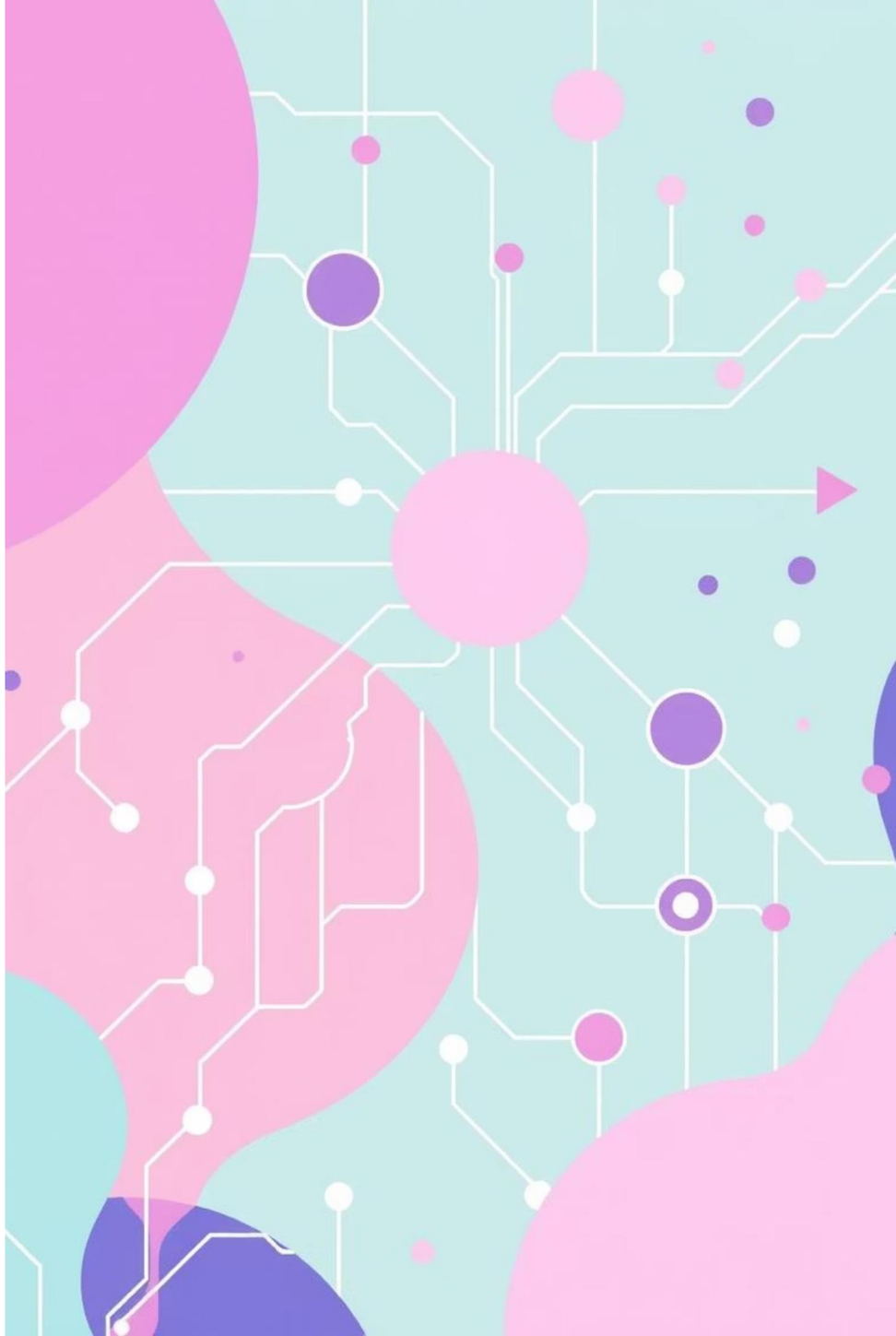
Информатика: Учебное методическое пособие по практическим и лабораторным занятиям, самостоятельной и индивидуальной работе студентов всех направлений / А. Я. Суханов - 2023. 110 с. [Электронный ресурс]: — Режим доступа: https://edu.tusur.ru/publications/10841

Информатика: Предмет и Современные Аспекты

Современная информатика — это динамичная и многогранная область, которая охватывает фундаментальные научные принципы, прикладные технологии и промышленное производство. Она является ключевой дисциплиной в эпоху цифровизации, формируя наше понимание информации и способы её обработки.

Информатика рассматривается как:

- **Фундаментальная естественная междисциплинарная наука:** Научная методология разработки информационного обеспечения процессов управления материальными объектами и интеллектуальными процессами любой природы.
- **Прикладная дисциплина:** Создание технологий и устройств обработки и передачи информации.
- **Отрасль промышленного производства:** Потребление, создание и распространение информации.



Основные направления информатики

Теоретическая информатика - математическая дисциплина включающая такие дисциплины и направления как:

Дискретная математика, логика высказываний, логика предикатов, теория нечетких множеств, теория алгоритмов, теория параллельных вычислений, теория автоматов, теория сетей Петри, машина Тьюринга

вычислительная математика и вычислительная геометрия, методы оптимизации

теория кодирования, изучение свойств информации, передача информации по различным каналам связи и особенности передачи информации, сигналы

системный анализ, моделирование, имитационное моделирование, теория массового обслуживания

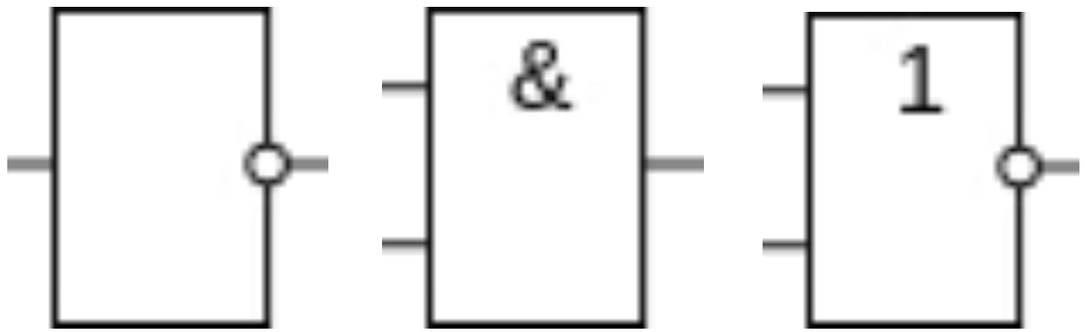
теория принятия решений, теория игр, исследование операций, математическое программирование

Дискретная Математика: Основы Логики

Дискретная математика является краеугольным камнем теоретической информатики, предоставляя инструментарий для работы с дискретными структурами и логическими операциями.

Ключевые Понятия

- Ложь и Истина: Фундаментальные значения бинарной логики.



- Логические Операции: И, ИЛИ, НЕ.
- Возможность представить любую логическую формулу посредством данных операций (СКНФ, СДНФ).

Эти концепции лежат в основе проектирования цифровых схем и алгоритмов, обеспечивая логическую строгость в информационных системах.

Логика Высказываний и Предикатов

Расширяя основы дискретной математики, логика высказываний и предикатов позволяет формализовать и анализировать более сложные утверждения и отношения.

Логика Высказываний

Выражения и атомы, которые могут принимать значения "истина" или "ложь", связываются стандартными логическими операциями (И, ИЛИ, НЕ, исключающее ИЛИ). Это позволяет строить сложные утверждения из простых.

Логика Предикатов

В логике предикатов появляются кванторы существования (∃) и всеобщности (∀), позволяющие выражать утверждения о свойствах объектов и отношениях между ними.

Пример: "Все люди смертны" — классический пример утверждения, формализуемого в логике предикатов.

Эти логические системы являются основой для искусственного интеллекта, баз данных и формальной верификации программ.

Теория Нечетких Множеств

В отличие от классической бинарной логики, теория нечетких множеств позволяет работать с неопределенностью и неточностью, что особенно важно при моделировании человеческого мышления и поведения.

Нечеткие Понятия

Часто понятия, которыми оперирует человек, нечеткие: "мягкий", "быстрый", "высокий", "умный", "красивый", "медленный", "темный", "светлый" и так далее. Классическая логика не может адекватно описать такие категории.

Функция Принадлежности

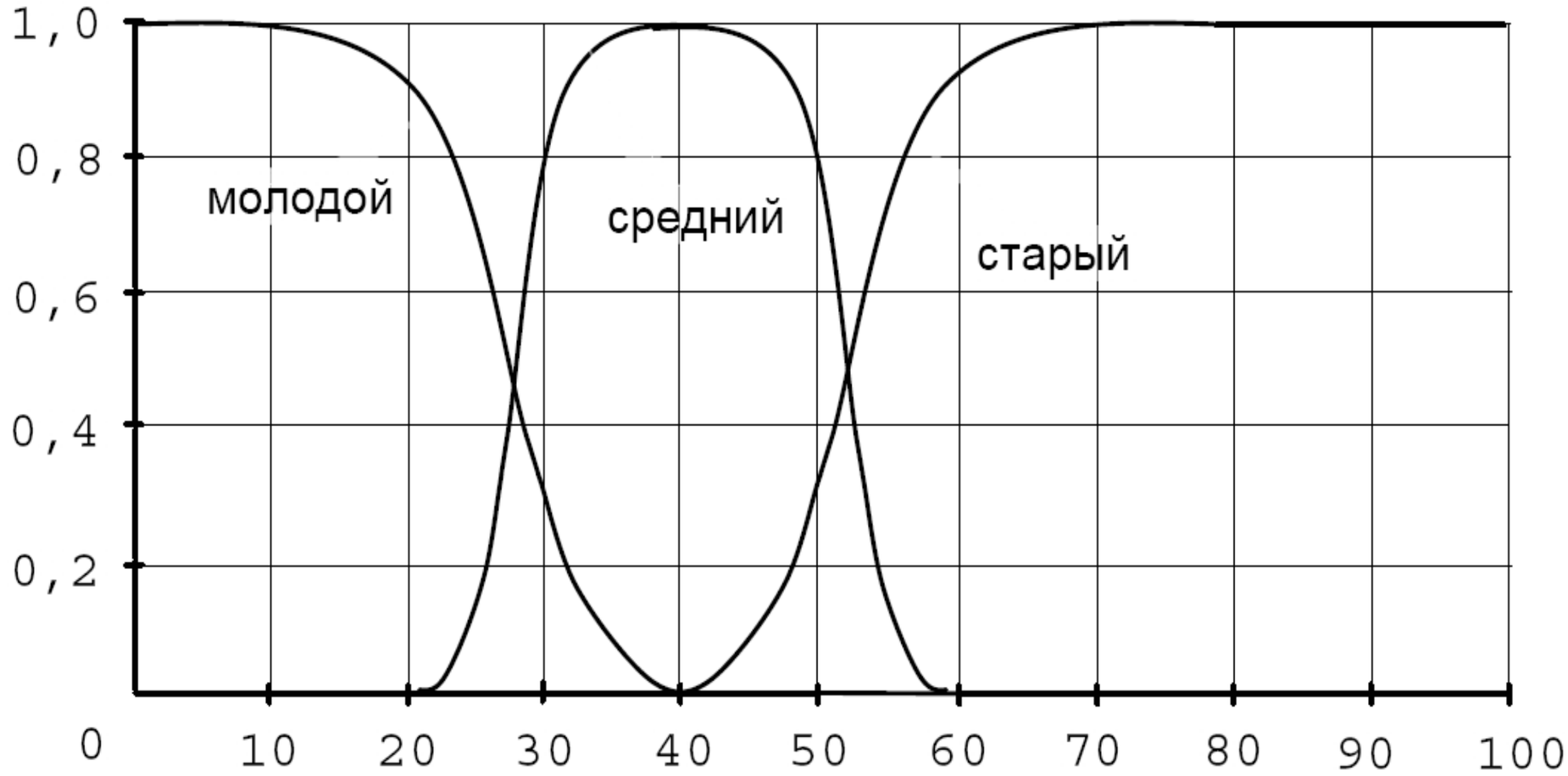
Вводится специальная функция, которая показывает, с какой силой какой-то элемент множества принадлежит данному понятию. Это позволяет количественно выразить степень принадлежности к нечеткому множеству.

Операции на Нечетких Множествах

Разработаны соответствующие операции (объединение, пересечение, дополнение) для нечетких множеств, которые учитывают степени принадлежности, позволяя выполнять логические выводы в условиях неопределенности.

Теория нечетких множеств находит применение в системах управления, распознавании образов, экспертных системах и машинном обучении.

Пример функции принадлежности нечеткому множеству



Теория Алгоритмов: Сложность и Разрешимость

Теория алгоритмов изучает фундаментальные свойства алгоритмов, их эффективность и возможность решения различных задач.
В рамках классической теории осуществляется классификация задач по классам сложности:

- **P-сложные задачи:** Могут быть решены за время, полиномиально зависящее от объема исходных данных, с помощью детерминированной вычислительной машины (например, машины Тьюринга).
- **NP-сложные задачи:** Могут быть решены за полиномиально выраженное время с помощью недетерминированной вычислительной машины, т.е. машины, следующее состояние которой не всегда однозначно определяется предыдущими. Их решение может быть проверено за полиномиальное время, но не всегда можно решить за полиномиальное время (проверка выполнимости булевой формулы, хотя бы одна интерпретация истинна).
- **NP-полные и NP – трудные:** NP полные, это задачи, которые лежат в классе NP и входят в множество трудных. Задача из класса полных может быть сведена за полиномиальное время к другой np-полной задаче. np – трудная может быть np-полной или не решаемой (например, проверка останова алгоритма).
- **Экспоненциально сложные (EXPTIME) и другие классы** (FP - найти решение которое существует, #P - посчитать число решений) .

Фундаментальный вопрос:
P = NP ?

Примеры задач

- **P — задачи, решаемые за полиномиальное время**
- Сортировка, поиск в графе в ширину (найти есть ли путь между вершинами, найти минимальный), проверка является ли число простым (в 2002 году найден индийскими математиками, алгоритм AKS), поиск эйлерова пути в графе (проходящий через все ребра)
- **NP — задачи, решение которых можно быстро проверить** (Но не обязательно быстро найти)
- Проверка выполнимости для КНФ (SAT), проверка существования гамильтонова цикла длиной меньше либо равно K (замкнутый гамильтонов путь проходит через все вершины по одному разу)
- **NP-полные — самые сложные задачи в NP**
- В подмножестве чисел найти есть ли такие, которые в сумме дают T. Задача о клике – множество вершин соединенных попарно (Есть ли клика размером k в графе).
- **NP-трудные**
- Задача коммивояжера (поиск минимального гамильтонова цикла), проверяется быстро, сводится к другим np-полным (ищет число, а не ответ да/нет, не относится к np-полным). Задача о клике (найти клику максимального размера).

Машина Тьюринга: Формализация Алгоритма

Машина Тьюринга — это абстрактная вычислительная машина, предложенная Аланом Тьюрингом в 1936 году для формализации понятия алгоритма. Она является теоретической основой для понимания пределов вычислений.

Абстрактный Исполнитель

Машина Тьюринга представляет собой простейшую вычислительную машину с линейной памятью, которая согласно формальным правилам преобразует входные данные с помощью последовательности элементарных действий.

Элементарность Действий

Действие меняет лишь небольшой фрагмент данных в памяти (в случае машины Тьюринга — лишь одну ячейку), и число возможных действий не бесконечно. Это подчеркивает фундаментальную природу вычислений.

Свойство Полноты

Несмотря на простоту, на машине Тьюринга можно вычислить всё, что можно вычислить на любой другой машине, осуществляющей вычисления с помощью последовательности элементарных действий. Это свойство называется **полнотой**.

Машина Тьюринга является ключевым концептом в информатике, позволяющим анализировать вычислительную мощность и ограничения различных систем.

Принцип Работы Машины Тьюринга

Понимание принципа работы машины Тьюринга позволяет глубже осознать, как формализуются и выполняются алгоритмы.

01

Задание Машины

Конкретная машина Тьюринга задаётся перечислением элементов множества букв алфавита A, множества состояний Q и набором правил, по которым работает машина.

02

Формат Правил

Правила имеют вид: `qiaj→qi1aj1dk`. Это означает: если головка находится в состоянии `qi`, а в обозреваемой ячейке записана буква `aj`, то головка переходит в состояние `qi1`, в ячейку вместо `aj` записывается `aj1`, головка делает движение `dk`.

03

Варианты Движения

Движение `dk` имеет три варианта: на ячейку влево (L), на ячейку вправо (R), остаться на месте (N).

04

Конфигурации и Правила

Для каждой возможной конфигурации `<qi, aj>` имеется ровно одно правило (для недетерминированной машины Тьюринга может быть большее количество правил).

05

Остановка Машины

Правил нет только для заключительного состояния, попав в которое, машина останавливается. Также необходимо указать конечное и начальное состояния, начальную конфигурацию на ленте и расположение головки машины.

Этот детальный механизм позволяет машине Тьюринга выполнять любые вычислимые операции, являясь универсальной моделью вычислений.

Ключевые Направления Информатики

Помимо теоретических основ, информатика включает в себя ряд прикладных направлений, формирующих современный технологический ландшафт.

Кибернетика

Наука об управлении, связи и обработке информации в сложных системах.

Программирование

Процесс создания компьютерных программ, от низкоуровневых языков до высокоуровневых фреймворков.

Искусственный Интеллект

Разработка систем, способных выполнять задачи, требующие человеческого интеллекта.

Информационные Системы

Системы для сбора, хранения, обработки и распространения информации.

Вычислительная Техника

Разработка и производство аппаратных средств для обработки информации.

Эти направления тесно взаимосвязаны и постоянно развиваются, определяя будущее технологий.

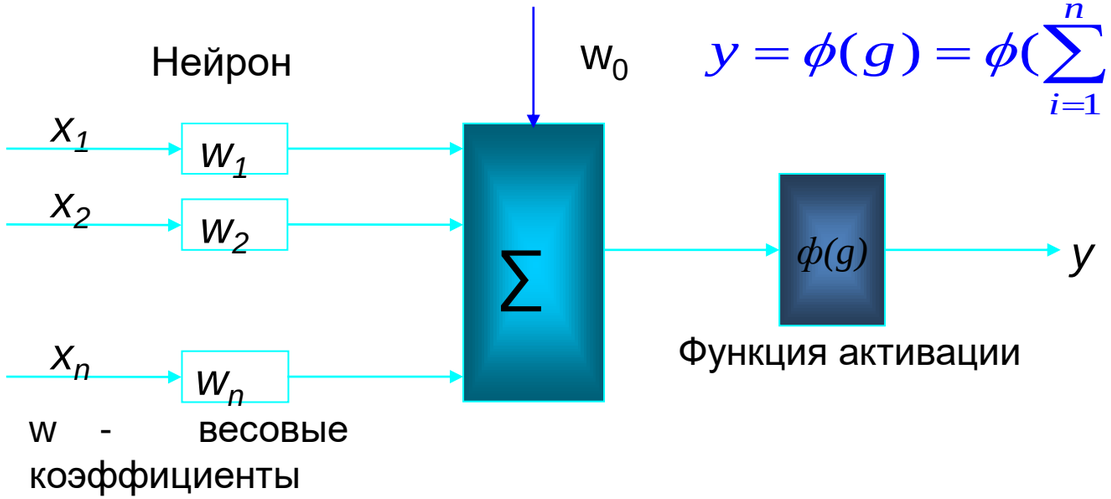
Кибернетика: Многообразие Аспектов

Кибернетика, как наука об управлении и связи, является одной из наиболее широких и междисциплинарных областей информатики, охватывающей множество теорий и приложений.

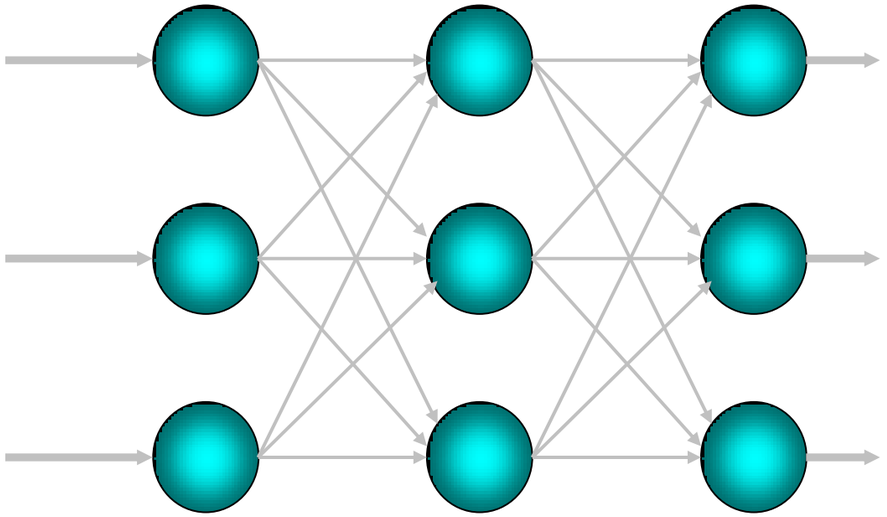
- Теория передачи сигналов
- Теория управления
- Теория автоматов
- Теория принятия решений
- Синергетика
- Теория алгоритмов
- Распознавание образов
- Теория оптимального управления
- Теория обучающихся систем
- Искусственный интеллект
- Кибернетика второго порядка
- Метакибернетика
- Компьютерное зрение
- Системы управления
- Эмерджентность
- Обучающиеся организации
- Новая кибернетика
- Теория общения
- Биоинженерия
- Биологическая кибернетика
- Биоинформатика
- Бионика
- Медицинская кибернетика
- Нейрокибернетика
- Гомеостаз
- Синтетическая биология
- Системная биология

Это обширное поле исследований подчеркивает универсальность принципов кибернетики, применимых как к техническим, так и к биологическим и социальным системам.

Нейронные сети



$$y = \phi(g) = \phi\left(\sum_{i=1}^{n} w_i x_i + w_0\right)$$

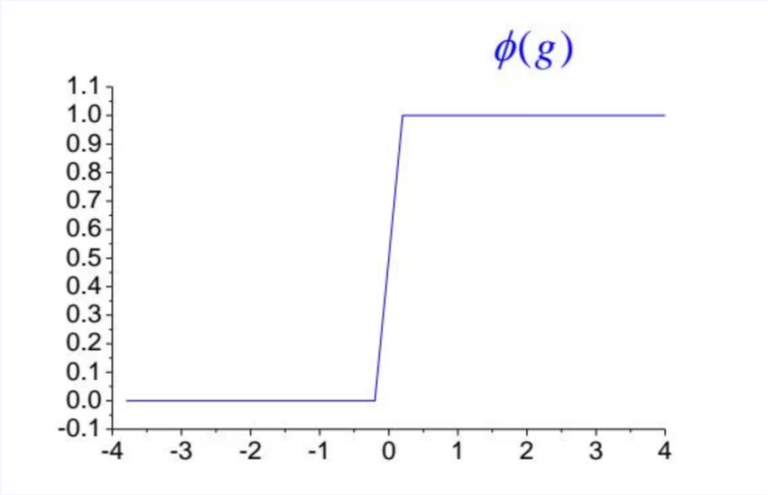


Функции Активации

Пороговая функция

Бинарный выход (0 или 1).

Простое решение.

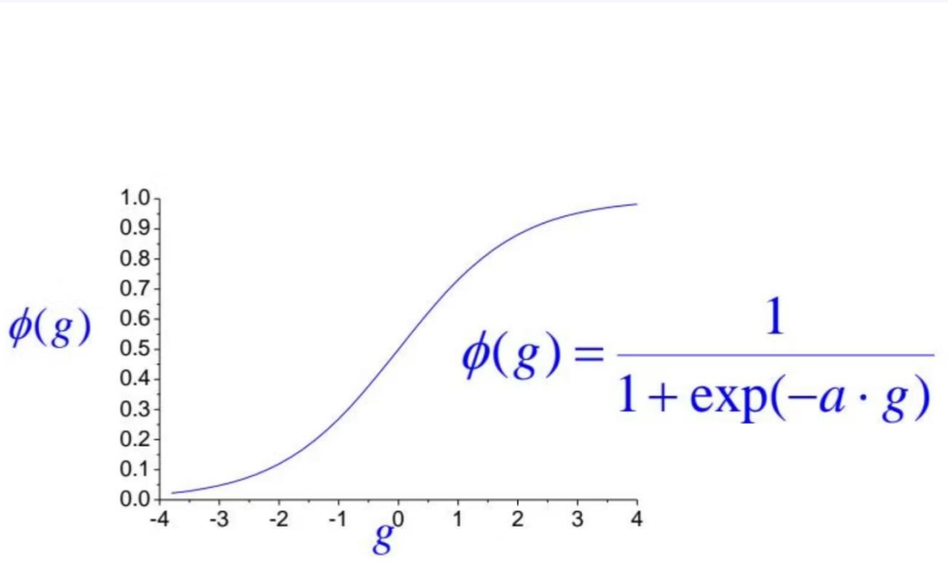


Сигмовидная функция

Плавный переход.

Выход от 0 до 1.

Используется в многослойных сетях.

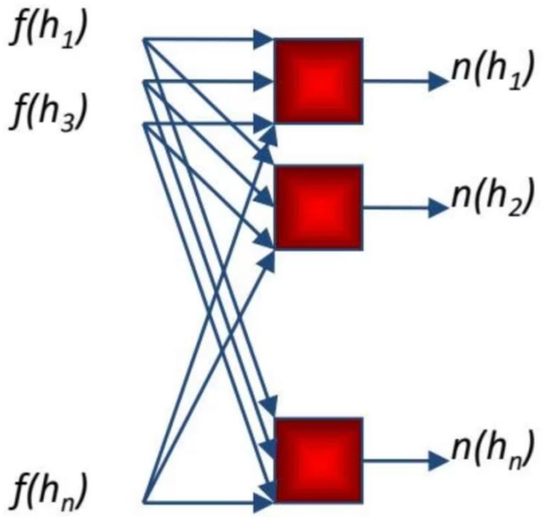


Виды Нейронных Сетей

Однослойный персептрон

Простейшая сеть.

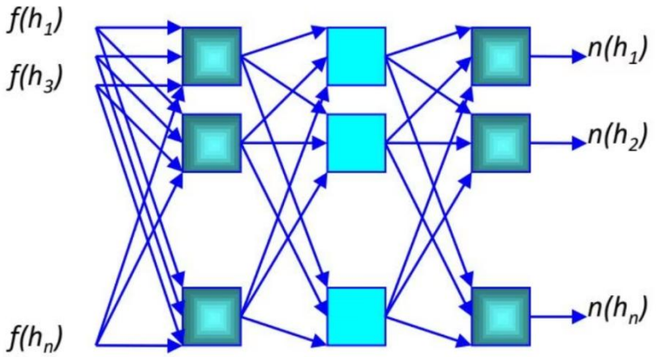
Линейная классификация.



Трехслойная сеть

Скрытые слои.

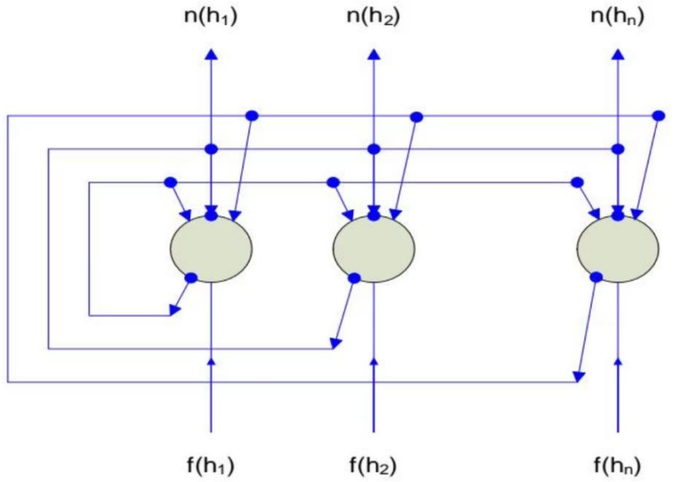
Сложные задачи.



Сеть Хопфилда

Ассоциативная память.

Оптимизация.



Методы Обучения

01

Минимизация Ошибки

Цель: уменьшить разницу между предсказанием и реальностью.

$$E = \frac{1}{2} \sum_{l=1}^{M} \sum_{i=1}^{N} (y_i(W) - d_{l,i})^2 \rightarrow \min$$

02

Обратное Распространение

Корректировка весов на основе ошибки.

$$w_{ij}(t+1) = w_{ij}(t) - h\frac{\partial E}{\partial w_{ij}}$$

03

Изменение Весов

Применение корректировок.

$$\Delta w_{ij} = -h\delta_j^{(n)} x_i^n$$

Технологии Нейронных Сетей

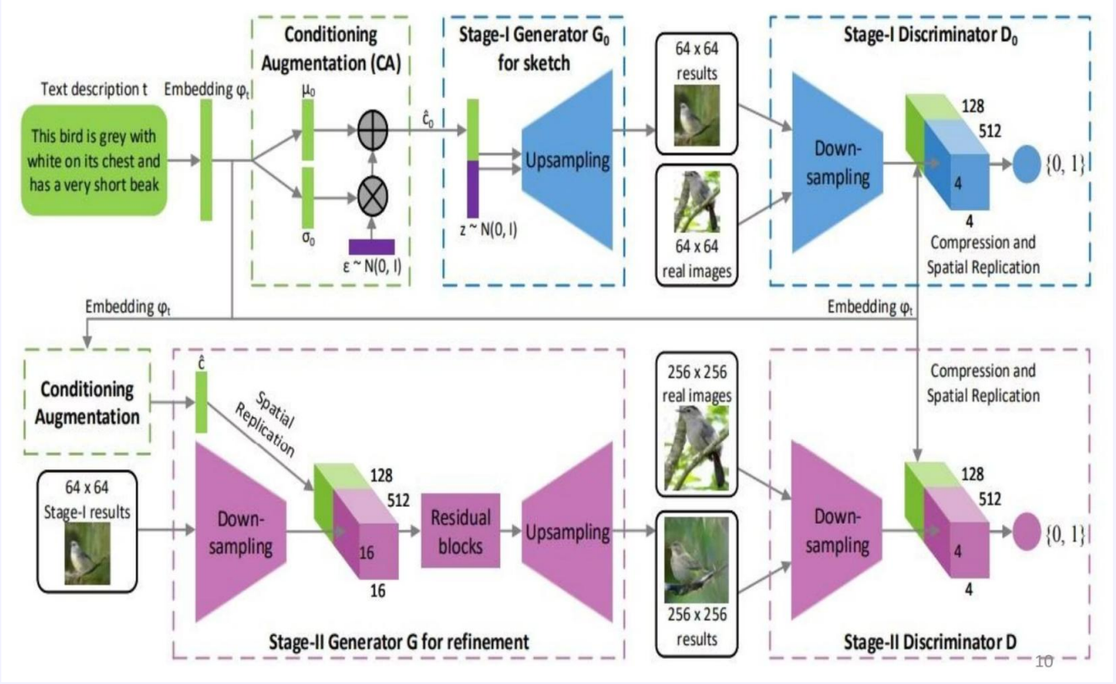
- RNN, LSTM, GRU, attention
- Fast R-CNN, Faster R-CNN (детектирование)
- Сверточные нейронные сети
- YOLO v3-5 (детектирование),
  v8-11 (сегментация), Unet

- BERT, GPT-2,3, Mamba (языковые модели)

Пример: беспилотник LoFLYTE (1996) использовал нейронные сети для обучения пилотированию.

Основные Направления

Генерация Изображений
GAN, вариационные автоэнкодеры.



Обучение с Подкреплением
DQN, A2C, PPO.

Классификация Изображений
Распознавание объектов.

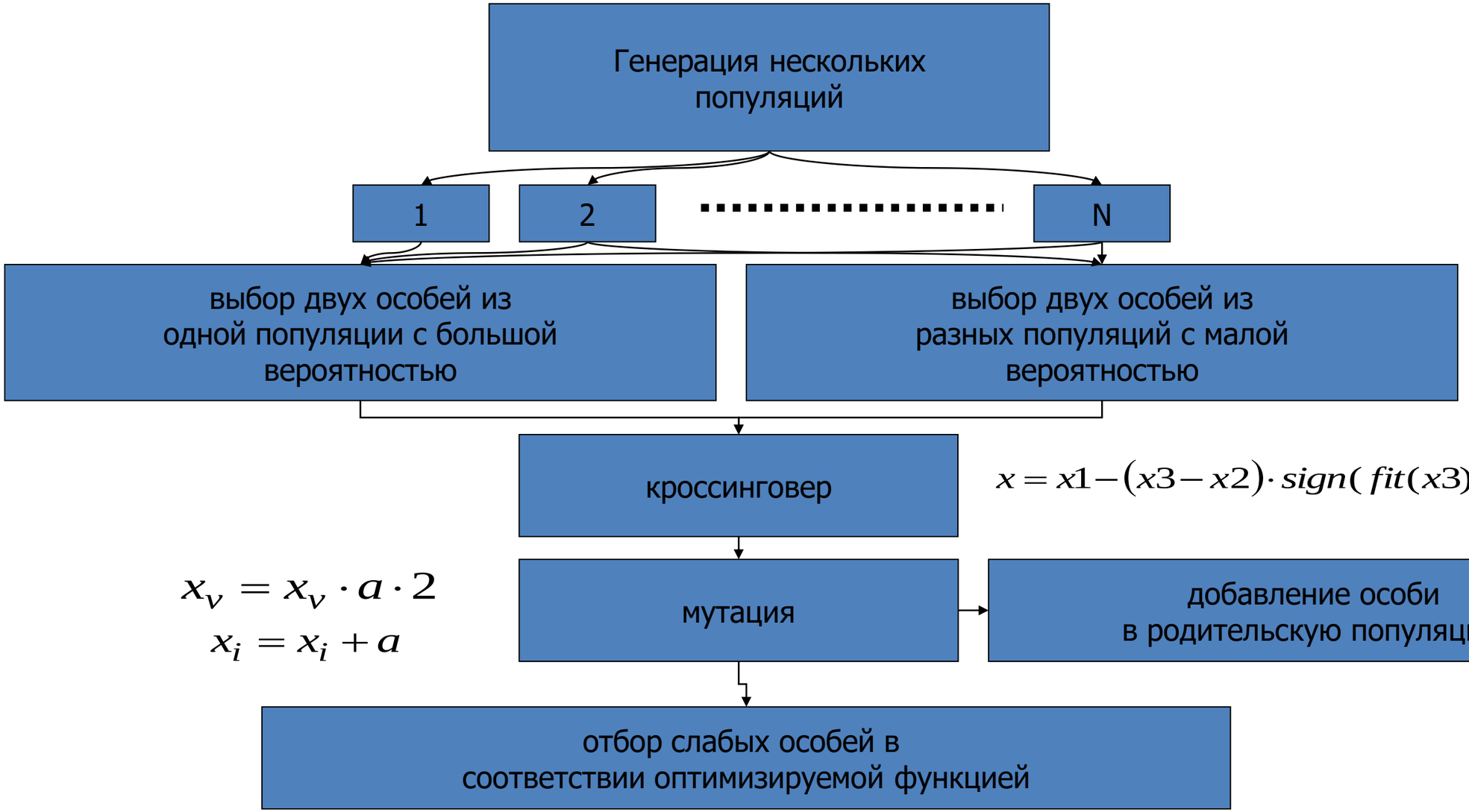
NLP
Обработка естественного языка.

Библиотеки для Нейронных Сетей

JAX

Инструменты для создания и обучения сетей, распараллеливания вычислений на CPU и GPU.

Оптимизация с использованием генетического алгоритма



$$x = x1 - (x3 - x2) \cdot sign(\, fit(x3) - fit(x2)) \cdot 0.001$$

$$x_v = x_v \cdot a \cdot 2$$

$$x_i = x_i + a$$

Эволюционно настраивается животное, которое учится двигаться.
Модель движения задается полиномиальными уравнениями.
С помощью эволюционного алгоритма ищутся параметры агента.

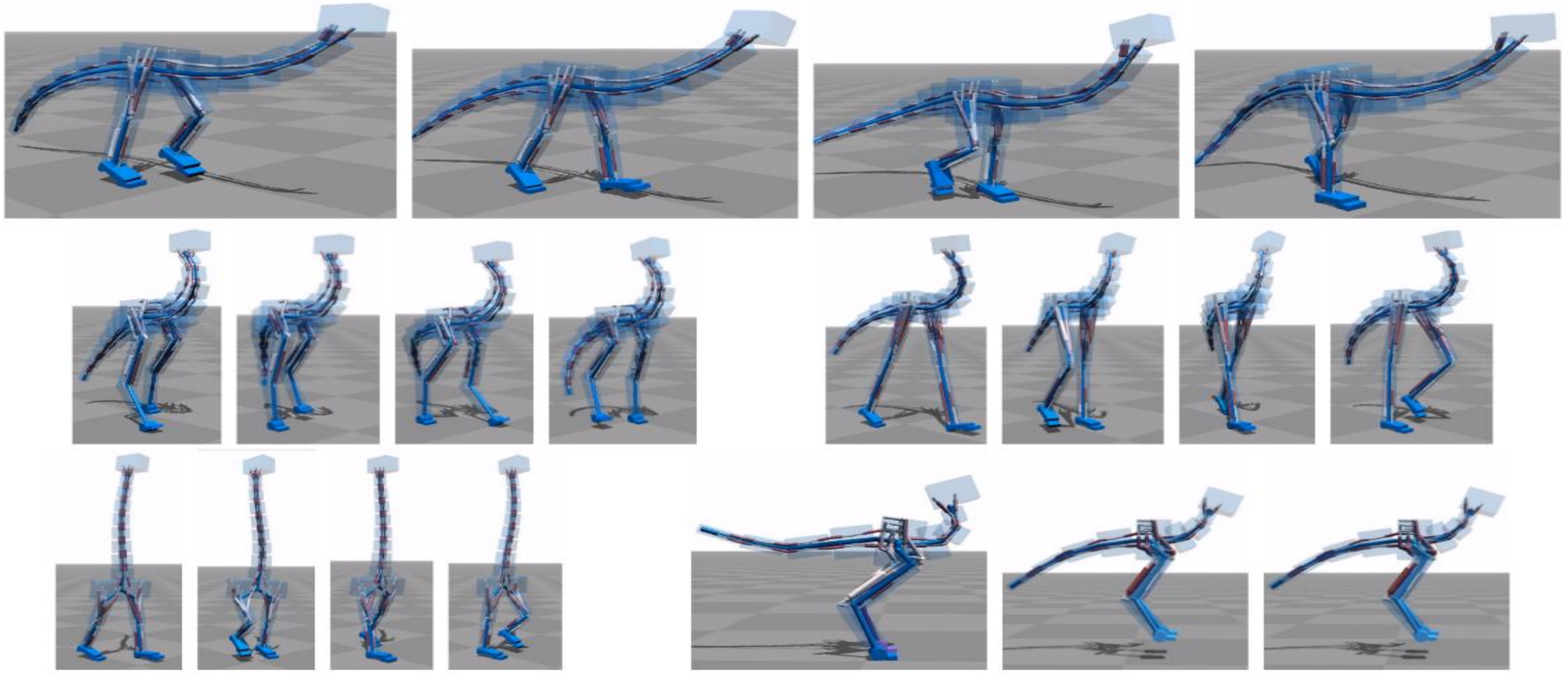
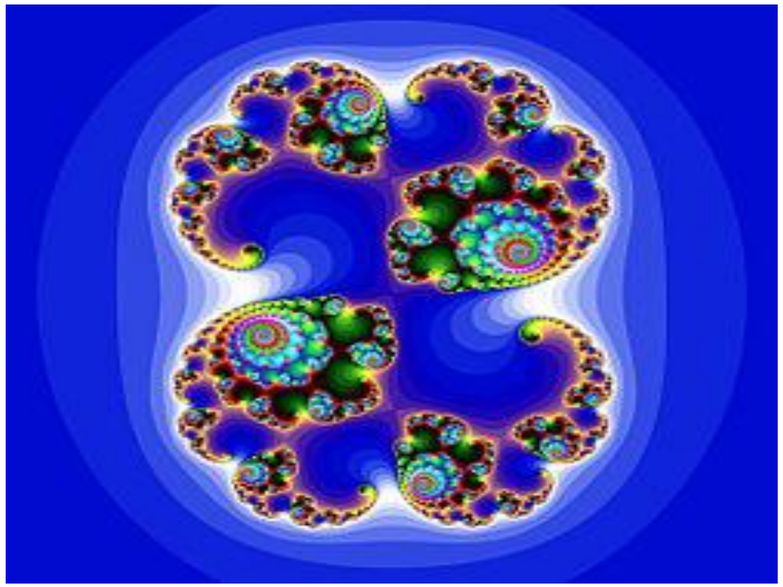
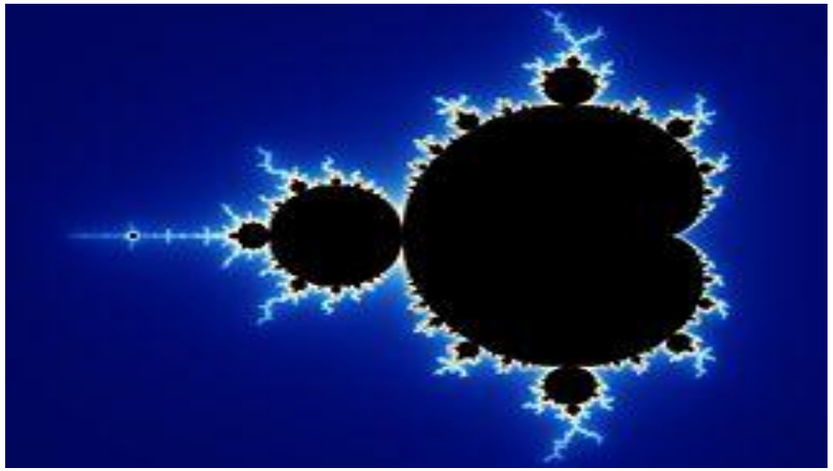


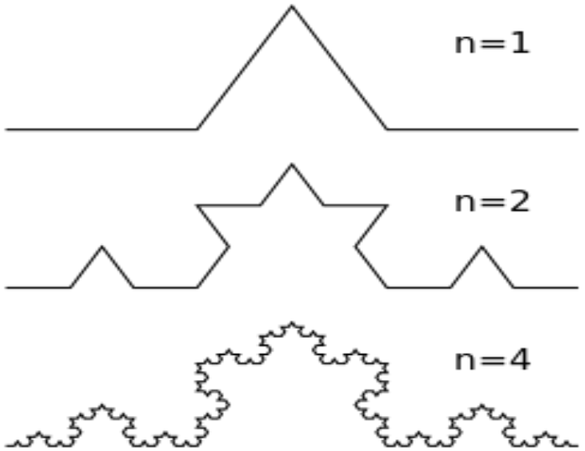
Figure 9: *Example synthesized walking and hopping gaits.*

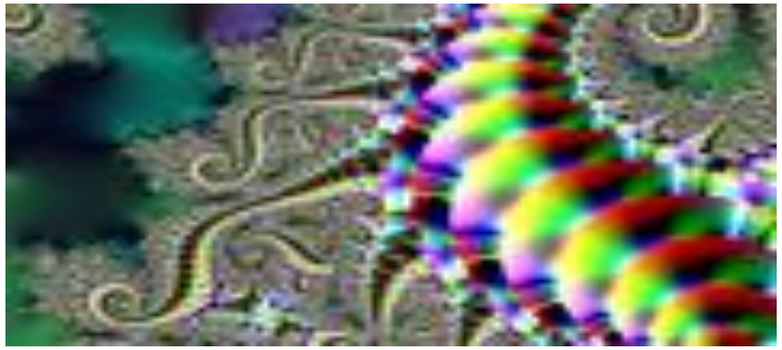
- Аппроксимация изображения полигонами

- Фракталы









Искусственный интеллект

Экспертные системы

Заменяют человека эксперта в некоторой предметной области, состоят из машины вывода и базы фактов, а также системы объяснений выводов

Базы знаний

Модели представления знаний

Семантические сети, Фреймы, Продукционные модели

Биологический интеллект (ГА, НС)

Методы познания

- Дедукция

Все люди смертны, Сократ человек, Сократ смертен

- Индукция

Очередная машина желтого цвета оказалась такси, значит такси в городе желтого цвета

- Абдукция

Все люди смертны, Сократ смертен, значит Сократ человек

- Анализ

Разложение на детали

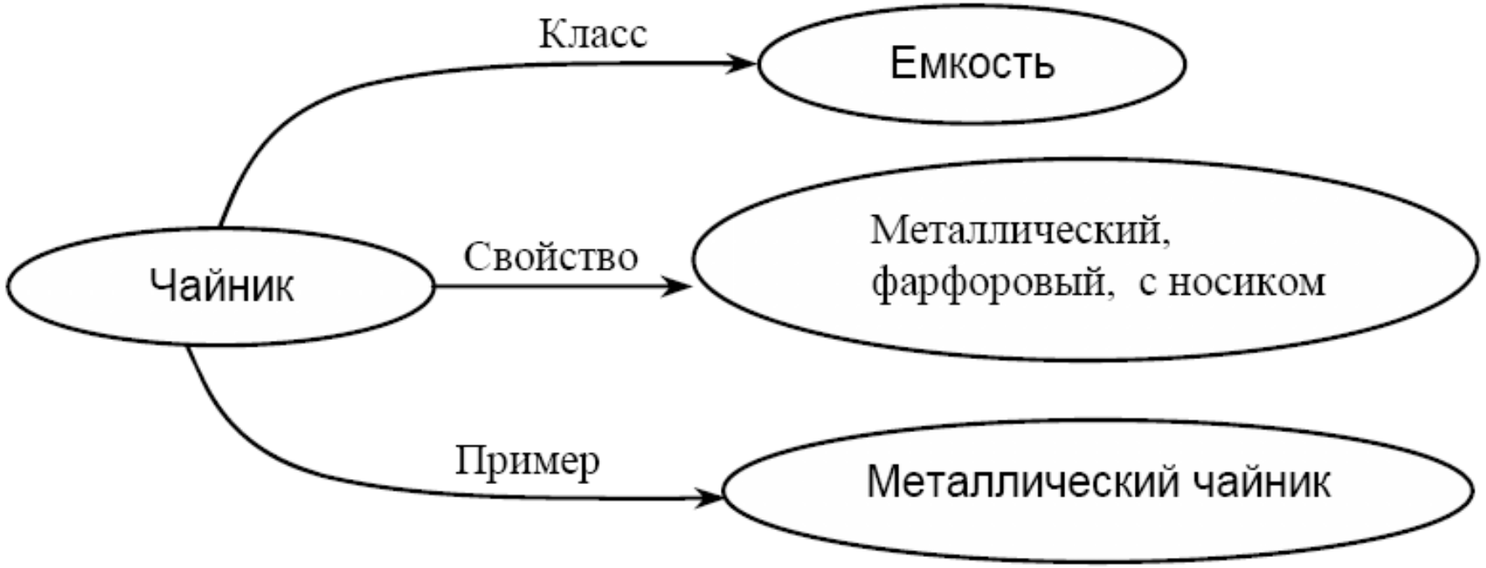
- Синтез

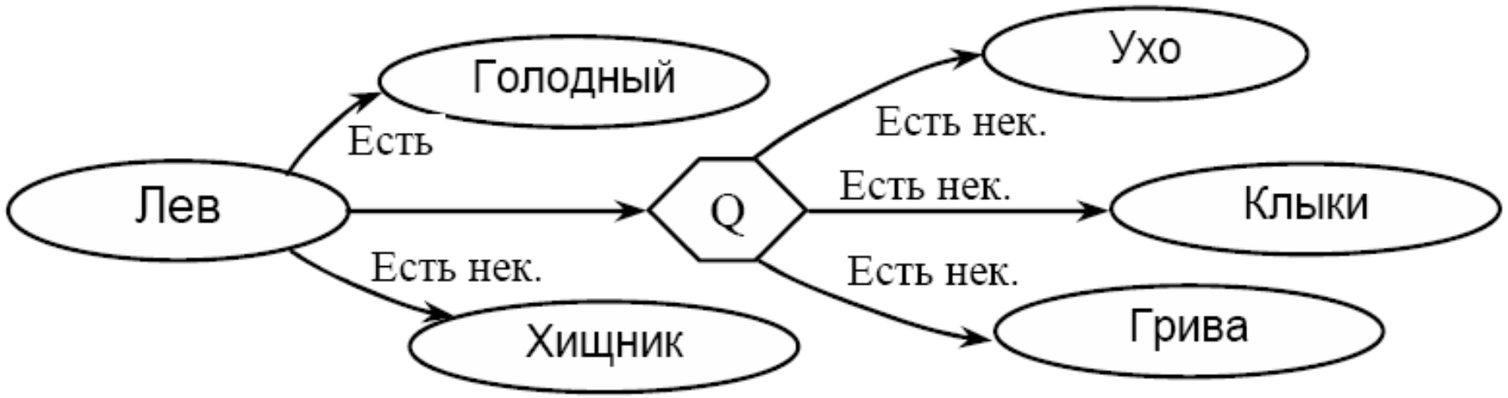
Объединение в общее

- Перебор вариантов
- Эвристика

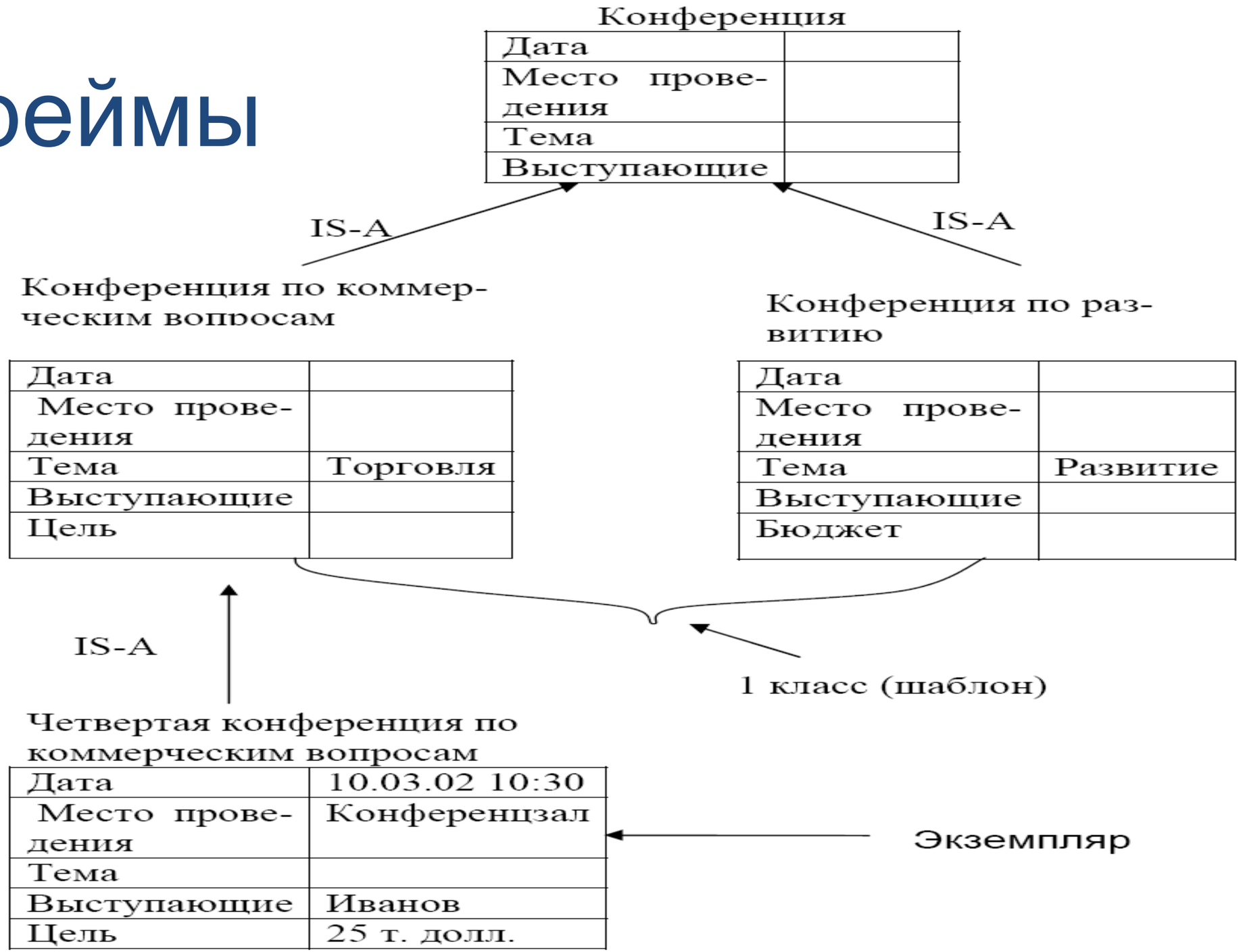
Способ или алгоритм который с высокой вероятностью приведет к результату близкому к требуемому

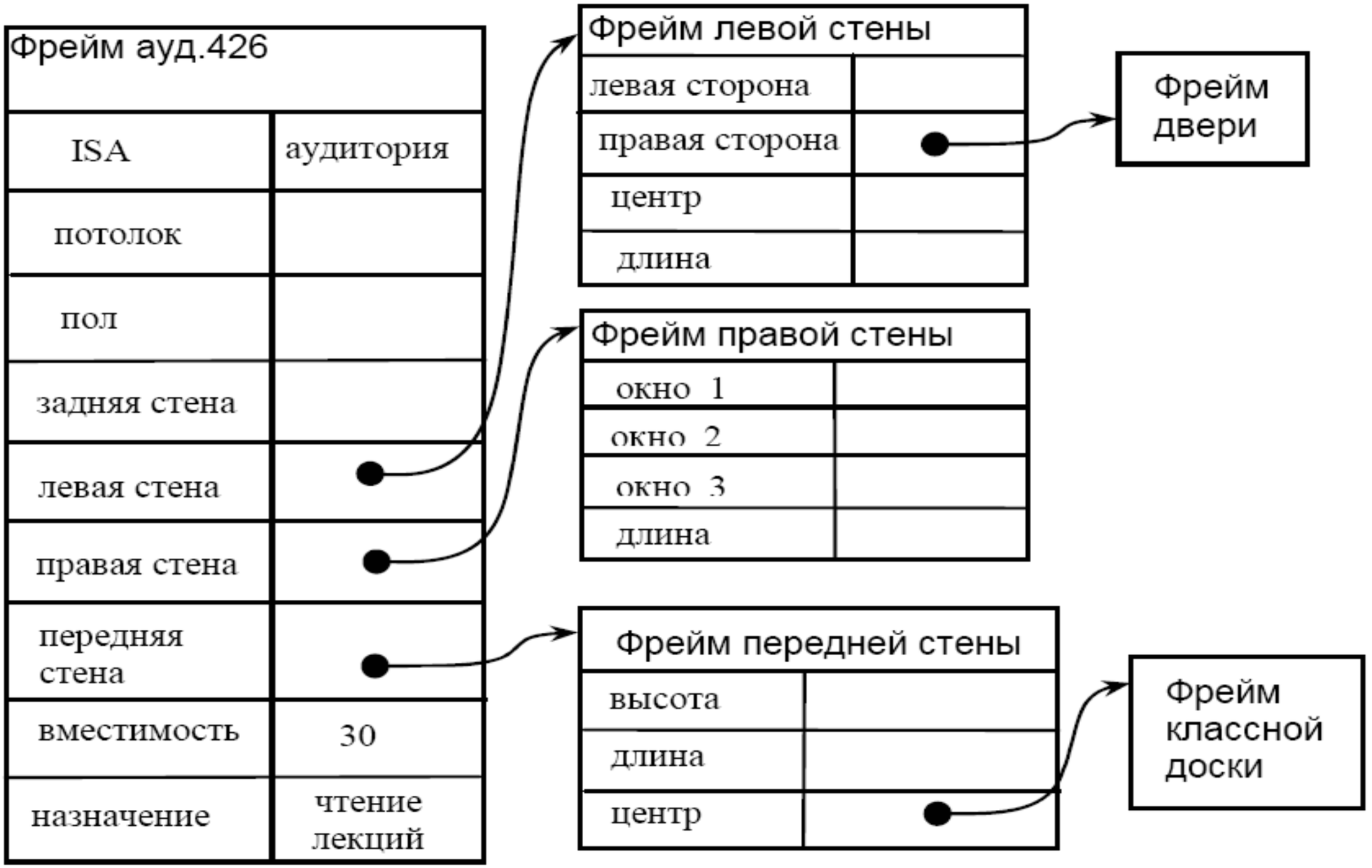
Семантическая сеть





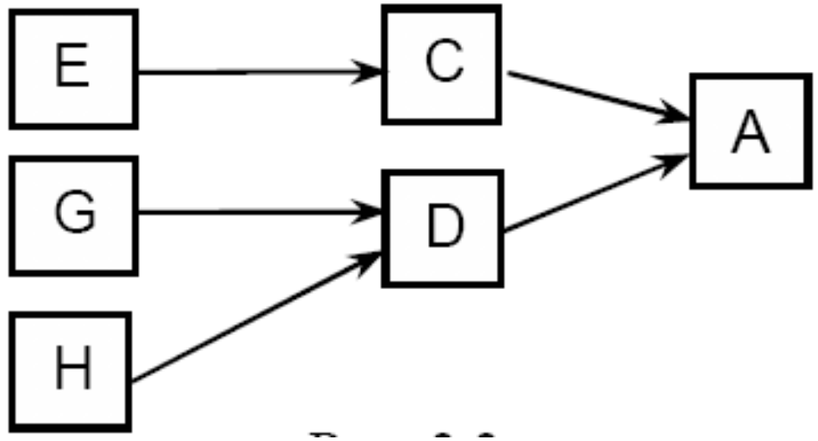
Фреймы


Конференция
Дата
Место прове-дения
Тема
Выступающие
IS-A
IS-A
Конференция по коммер-ческим вопросам
Конференция по раз-витию
Дата
Место прове-дения
Тема
Торговля
Выступающие
Цель
Дата
Место прове-дения
Тема
Развитие
Выступающие
Бюджет
IS-A
1 класс (шаблон)
Четвертая конференция по коммерческим вопросам
Дата
10.03.02 10:30
Место прове-дения
Конференцзал
Тема
Выступающие
Иванов
Цель
25 т. долл.
Экземпляр


Фрейм ауд.426
ISA
аудитория
потолок
пол
задняя стена
левая стена
правая стена
передняя стена
вместимость
30
назначение
чтение лекций
Фрейм левой стены
левая сторона
правая сторона
центр
длина
Фрейм двери
Фрейм правой стены
окно 1
окно 2
окно 3
длина
Фрейм передней стены
высота
длина
центр
Фрейм классной доски

Продукционная модель

посылка p,
правило $p \rightarrow q$,
заключение q;



$C \wedge D \rightarrow A, \quad B \rightarrow A, \quad E \rightarrow C, \quad F \rightarrow C, \quad G \wedge H \rightarrow D, \quad J \wedge K \wedge L \rightarrow B.$

Программирование

- Логические языки (Prolog, Меркурий)
- Функциональные языки (Lisp, Haskell)
- Процедурные (Си, Паскаль и т.д.)
- Объектно-ориентированные (Java, C#, Ruby)
- Машинные

Пример на языке пролог

- предок(X, Z) :- родитель(X, Z).
- предок(X, Z) :- родитель(X, Y), предок(Y, Z).
- родитель(пам, боб). % Пам - родитель Боба
- родитель(том, боб).
- родитель(том, лиз).
- родитель(боб, энн).
- родитель(боб, пат).
- родитель(пат, джим).
- ?- предок(пам, X).
- X = боб;
- X = энн;
- X = пат;
- X = джим

Пример расчета факториала на haskell и Си и Паскале (рекурсия)

```
program factorial;
function fact(n: integer): longint;
begin
    if (n = 0) then
        fact := 1
    else
        fact := n * fact(n - 1);
end;

var
    n: integer;

begin
    for n := 0 to 16 do
        writeln(n, '! = ', fact(n));
end.
```

```
factorial :: Integer -> Integer
factorial 0 = 1
factorial n = n * factorial (n - 1)
```

```
#include <stdio.h>
unsigned long long factorial(unsigned long long n)
{
    if (n == 0) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}
int main(void)
{
    int n;
    for (n = 0; n <= 16; n++) {
        printf("%i! = %lld\n", n, factorial(n));
    }
```

Пример на Лиспе

```
(defun factorial (n)
    (if (= n 0)
        1
        (* n (factorial (- n 1))) ) )

(loop for i from 0 to 16
    do (format t "~D! = ~D~%" i (factorial i)) )
```

ООП

- Инкапсуляция
- Наследование
- Полиморфизм

Информационные системы

- Базы данных
- ГИС
- Поисковые системы

Вычислительные системы

- Компьютеры на базе транзисторных элементов
- Компьютеры на базе оптических элементов
- Биокомпьютеры
- Квантовые компьютеры

- Биокомпьютер — компьютер, который функционирует как живой организм или содержит биологические компоненты. Создание биокомпьютеров основывается на направлении молекулярных вычислений. В качестве вычислительных элементов используются белки и нуклеиновые кислоты, реагирующие друг с другом.

- Молекулярные компьютеры — вычислительные системы, использующие вычислительные возможности молекул (преимущественно, органических). Можно сказать, что молекулярные компьютеры — это молекулы, запрограммированные на нужные свойства и поведение. Молекулярные компьютеры состоят из сетевых нано-компьютеров. В работе обычной микросхемы используют отдельные молекулы в качестве элементов вычислительного тракта.

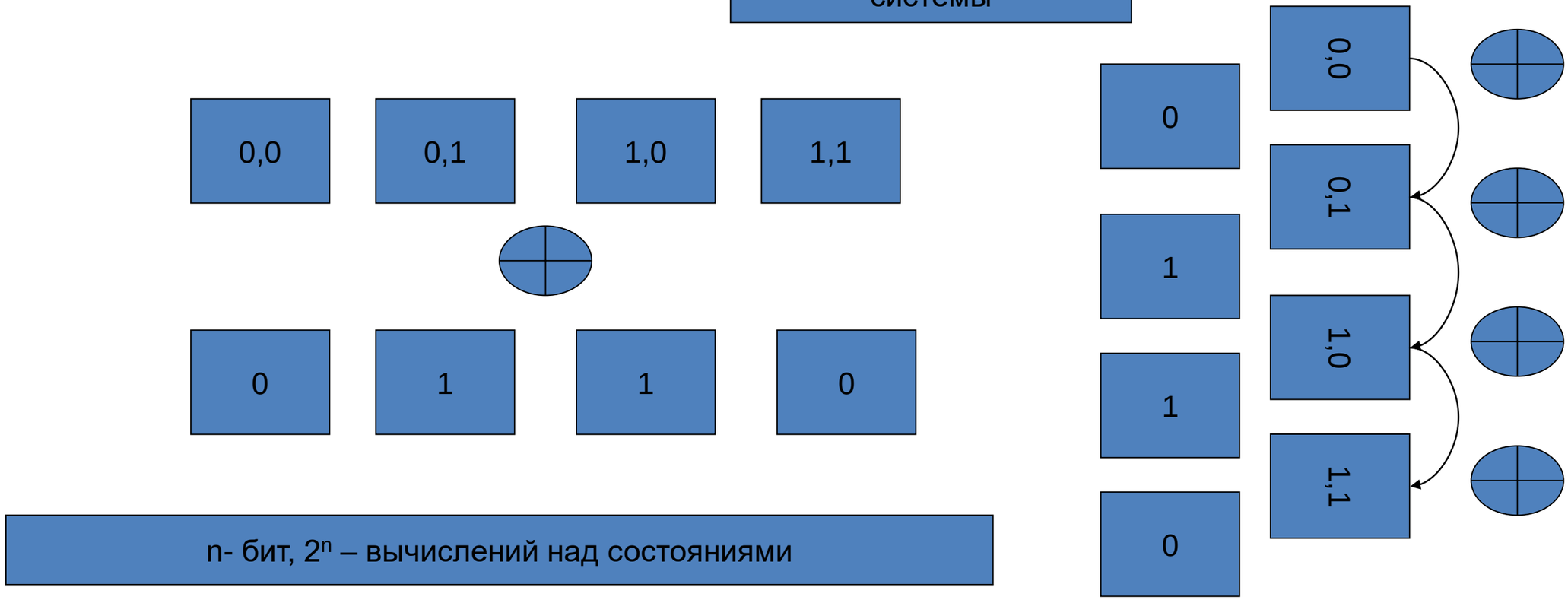
- В частности, молекулярный компьютер может представлять логические электрические цепи, составленные из отдельных молекул; транзисторы, управляемые одной молекулой, и т. п. В микросхеме памяти информация записывается с помощью положения молекул и атомов в пространстве.

- Одним из видов молекулярных компьютеров можно назвать ДНК-компьютер, вычисления в котором соответствуют различным реакциям между фрагментами ДНК. От классических компьютеров ДНК-компьютеры отличаются тем, что химические реакции происходят сразу между множеством молекул независимо друг от друга.

Квантовый компьютер

- Кубиты, кубайты
- Вычисления проводятся сразу на всеми состояниями
- Квантовые криптографические алгоритмы

$$a_1\langle 0,0| + a_2\langle 0,1| + a_3\langle 1,0| + a_4\langle 1,1|$$



Оптические или **фотонные вычисления** — вычисления, которые производятся с помощью фотонов, сгенерированных лазерами или диодами. Используя фотоны, возможно достигнуть более высокой скорости передачи сигнала, чем у электронов, которые используются в современных нам компьютерах. Большинство исследований фокусируется на замене обычных (электронных) компонентов компьютера на их оптические эквиваленты. Результатом станет новая цифровая компьютерная система для обработки двоичных данных. Такой подход дает возможность в краткосрочной перспективе разработать технологии для коммерческого применения, поскольку оптические компоненты могут быть внедрены в стандартные компьютеры, сначала создавая гибридные системы, а впоследствии и полностью фотонные. Однако опто-электронные приборы теряют 30% энергии на конвертацию электронов в фотоны и обратно. Это также замедляет передачу информации. В полностью оптическом компьютере надобность преобразования сигнала из оптического в электронный и обратно в оптический полностью исчезает