КАФЕДРА ТЕЛЕМАТИКА

Санкт-Петербургский Государственный Политехнический Университет

Институт прикладной математики и механики

Физика вычислительных процессов

Лекция 7-1

От «науки о данных» к теории алгоритмов

28 апреля 2020 г.

Что было на прошлой лекции

- В основе «науки о данных «или Data Science лежит формула :

реальность= материя (вещество + энергия) + «данные»

«Работа» с данными требует формализации, которая связывает данные с различными структурами – таблицы, списки, «деревья» или даже «в навал», т.е. множества данных могут не иметь регулярной структуры.....

- При этом «данные» рассматриваются не только «по значению», но и с учетом таких характеристик как «**Объем** - Volume», «**Скорость** изменения или - Velocity» и «**Разнообразия** или Variety» (3V) .
- Для работы с «большими данными» или 3V объектами используется технология MapReduce — двух шаговая процедура : шаг 1 «Map» и шаг 2 «Reduce» обработки данных с помощью **большого** числа компьютеров, организованных в распределенный кластер (сеть из «нодов»). В результате обработки выходные данные структурируются по дескриптору **ключ - значение** (см. пример).

Пример на основе модели MapReduce - сколько раз различные слова встречаются в наборе документов

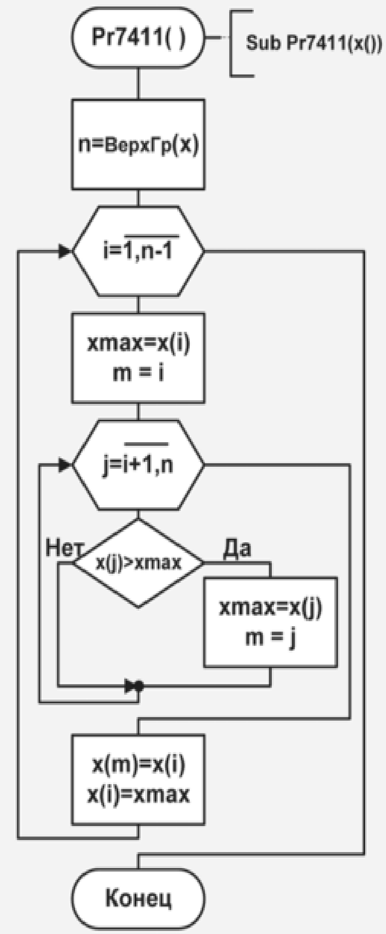
```
●     // Функция, используемая рабочими «нодами»  кластера на Map-шаге
// для обработки пар ключ-значение из входного потока
void map(String name, String document):
   // Входные данные:
   //   name - название документа
   //   document - содержимое документа
   for each word w in document:
      EmitIntermediate(w, "1");

// Функция, используемая рабочими «нодами» кластера на Reduce-шаге
// для обработки пар ключ-значение, полученных на Map-шаге
void reduce(String word, Iterator partialCounts):
   // Входные данные:
   //   word - слово
   //   partialCounts - список группированных промежуточных результатов. Количество записей в partialCounts и есть
   //    требуемое значение
   int result = 0;
   for each v in partialCounts:
      result += parseInt(v);
   Emit(AsString(result));
```

на **Map-шаге** каждый документ разбивается на слова, и возвращаются пары, где ключом является само слово, а значением — «1». Если в документе слово встречается несколько раз, то в результате обработки этого документа будет столько же этих пар, сколько раз встретилось это слово. Сформированные пары отправляются на дальнейшую обработку по ключу = само слово. Наборы объектов с одинаковым ключом попадают на **вход функции reduce**, которая складывает вхождения данного слова по всему потоку, и результат — их сумма — отправляется в виде выходных данных.

Введение в теорию алгоритмов

- Что такое алгоритм ?

набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за **конечное число** действий (программа или **текст** на каком либо языке)




```
Sub Pr7411(ByRef x() As  Single)
   Dim i, n, j, m As Integer
   Dim xmax As Single
   n = UBound(x)
  'Внешний цикл сортировки
   For i = 0 To n - 1
      xmax = x(i)
      m = i
     'Поиск  xmax  и  k
      For j = i + 1  To  n
          If x(j)  >  xmax Then
             xmax = x(j)
             m = j
          End If
      Next  j
      x(m) = x(i)
      x(i) =  xmax
   Next i
End Sub
```

Понятие алгоритма

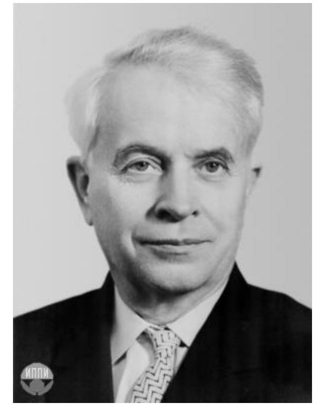
- Алгоритм относится к первоначальным **базисным понятиям математики**.
- Чтобы «понять» , что такое алгоритм надо осознать «чем занимается» **математика ?**

Имена, которые надо знать

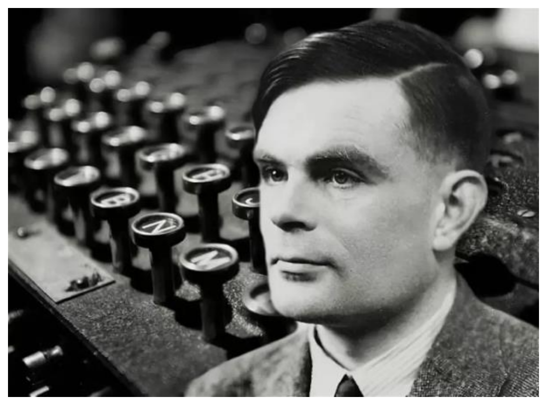
Д. Гильберт
Сторонник **полной формализации математики** и ее единства с естествознанием

К. Гедель
Теоремы о **неполноте формальных систем**

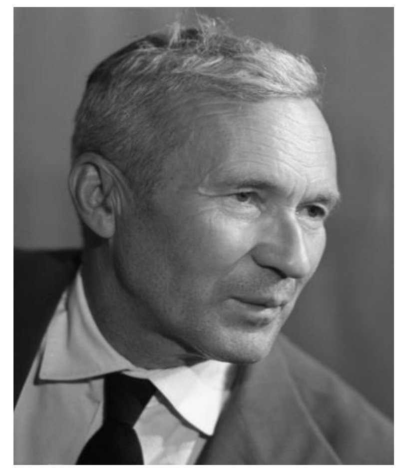
А. Черч показал существование **алгоритмически «неразрешимых задач»**



А. А. Марков - теория нормальных алгоритмов



А. Тьюринг (**абстрактная машина вычислений**, тест на ИИ)



А. Н. Колмогоров
Ввел понятие **алгоритмическая** энтропия или **сложность**

Математика – метафора реальности

Объекты реальности – неисчерпаемый «резервуар» математических аксиом

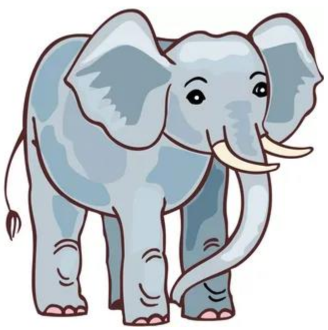
Символы или имена объектов и их свойств

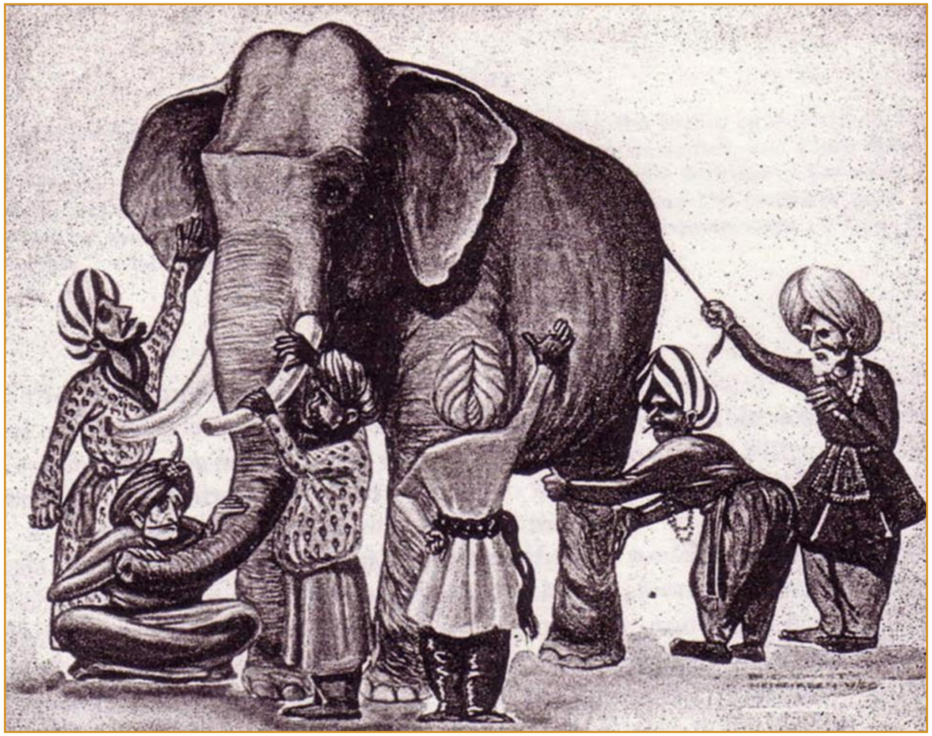
Парадоксы «прямого» восприятия или почему необходимо целостное восприятие

Проблема сложности и «неформализуемость» истины

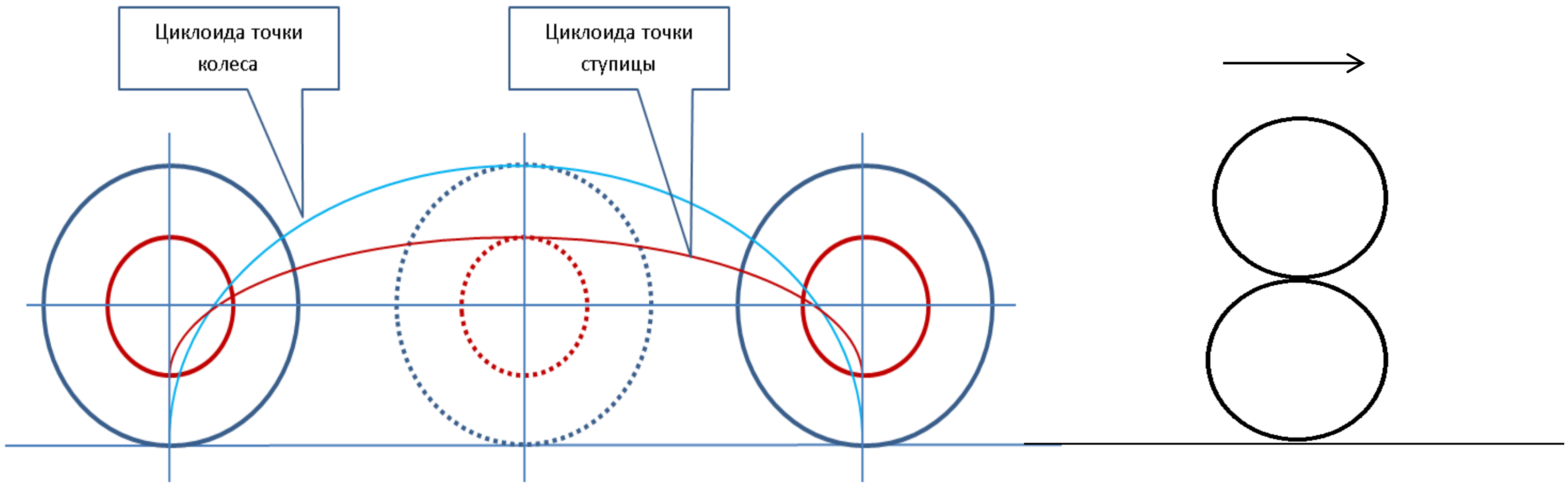
понятие **«оператор отображения»** **реальный объект и его восприятие**

«слон» **мозг** **«ученые»**





«В природе» нет противоречий – все они в голове.
ПАРАДОКС - АРИСТОТЕЛЕВО КОЛЕСО

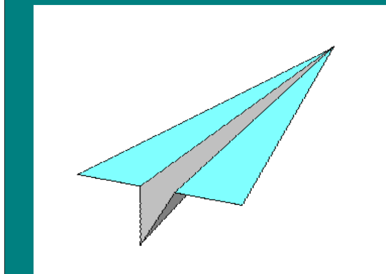


Длина расстояния между начальным и конечным пунктами, между которыми катится диск не совпадает с длинами путей (траекторий движения), проходимых точками окружностей разных диаметров $L=2*\pi*R$

Есть ли алгоритм измерения длины береговой линии ?

Парадоксы фрактальных мер !

Понятие алгоритма содержательно связано с отношением гомоморфизма реальных объектов и их математических моделей



А - модель

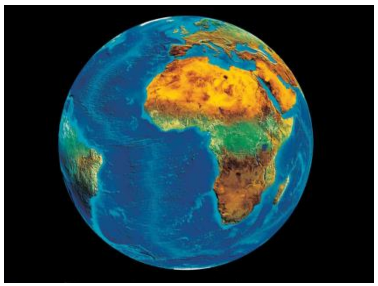
Описание	
Разработчик	ОКБ им. О.К.Антонова
Обозначение	Ан-225 «Мрия»
Кодовое обозначение NATO	Cossack (Казак)
Тип	сверхтяжелый транспортный самолет
Первый полет	21 декабря 1988
Экипаж, чел	6
Максимальное число мест, чел	60-70
Геометрические и массовые характеристики	
Длина самолета, м	84
Размах крыла, м	88,4
Площадь крыла, $м^2$	905,0
Высота, м	18,1
Размах стабилизатора, м	32,65
Габариты грузовой кабины, м	4.4 x 6.4 x 43
Максимальная длина перевозимого груза, м	70
Масса максимальная взлетная, кг	600000
Максимальная коммерческая нагрузка, кг	250000
Вес топлива, кг	
Силовая установка	
Число двигателей	6
Тип двигателя	ТРДД Д-18Т
Мощность двигателя, кгс (кН)	6x 23400 (229,5)

В – объект

Гомоморфизм - одностороннее отношение подобия между двумя системами. Систему А называют *гомоморфной* другой В системе (А является ее моделью В), если первая обладает некоторыми, но не всеми, свойствами или законами поведения другой.

Вопрос: все ли объекты реальности имеют математическую (формальную) модель ?

Пример: цепочки гомоморфизмов B ->A->R

B


A:

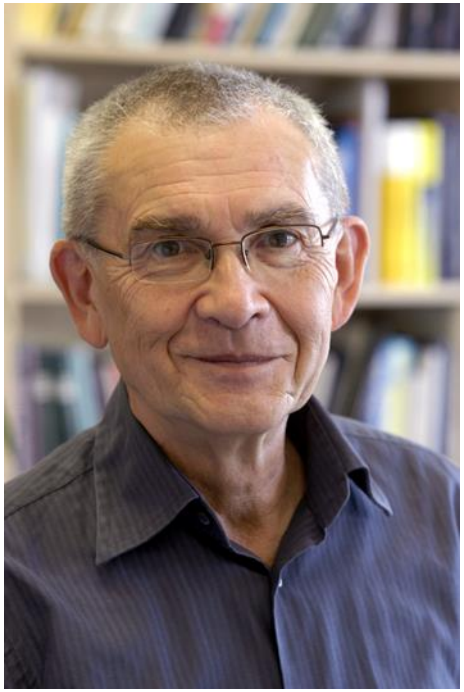

R:


Гомоморфизм (**Homomorphism**) : отображение объекта A в R таким образом, что отношения ≥ операции «+» сохраняются. В результате кратчайшее расстояние между точками уже не прямая, а … «кривая» = «геодезическая прямая»

Изоморфизм (Isomorphism) – это а 1->1 homomorphism -

- $h(0) = ab; h(1) = \epsilon$
- $h(01010) = ababab$

Ю. И. Манин - о том, чем занимается математика: книга «Математика как метафора» (2000)



Метафора есть соединение похожего с непохожим, при котором одно не может превратиться в другое.
В своей основе всякий язык имеет характер метафоры, поскольку независимо от своих намерений он всегда **остается языком** и тем самым совершенно **непохожим на то, что он описывает.**

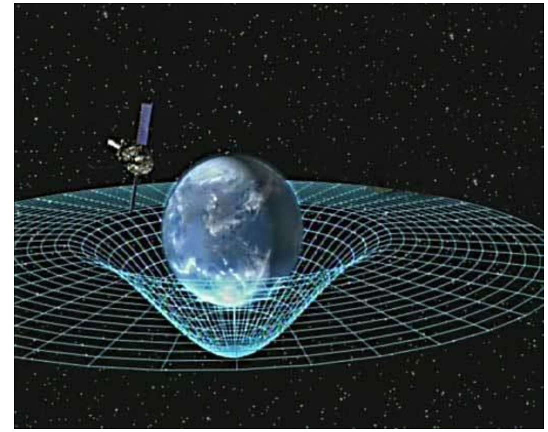
Вычислимое и невычислимое, Манин Ю.И., 1980

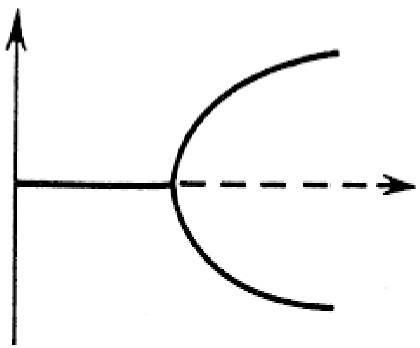
- Дано **доказательство существования невычислимых функций** и алгоритмически неразрешенных задач
- Сформулирована идея **квантовых компьютеров**

Рассматривая математику как метафору, я хочу подчеркнуть, что **интерпретация математического знания является актом в высшей степени творческим**. В некотором смысле математика—это роман о природе и человечестве. Точно **сказать, чему именно нас учит математика, невозможно** так же, как невозможно сказать, чему нас учит «Война и мир».

«Бифуркация» теории алгоритмов – проблема вычислимости

Алгоритмы «вычисления»
лишенной смысла и эмоций
физической реальности,
Движущейся под влиянием
4-х фундаментальных взаимодействий





Алгоритмы «вычисления»
осознающей себя сложно
организованной
матери, свойства которой доступны
только через движение мыслящего
субъекта

Формализации понятия алгоритма

В основе формализации лежит идея Д. Гильберта о «**выводимости**» содержательных понятий из «**правильного**» множества аксиом.

В понятие «алгоритм» заложена идея формализации свойств «вычислимости», «перечислимости» и «разрешимости». Итак:

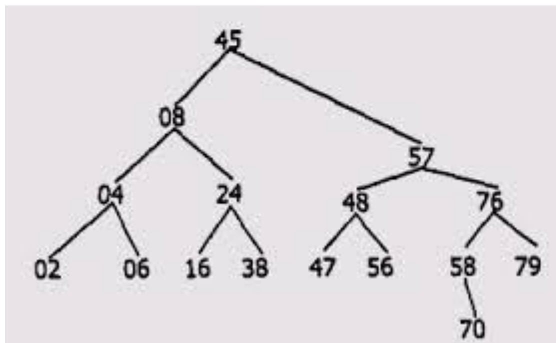
- Функция называется **вычислимой**, если существует алгоритм, позволяющий вычислять ее значения для тех наборов аргументов, для которых она определена.
- Множество **перечислимо** если все его элементы могут быть получены с помощью некоторого алгоритма.
- Множество **разрешимо** если существует алгоритм, который, получив на вход любое натуральное число, через конечное число шагов завершается и определяет, принадлежит ли оно данному множеству

Содержательная фактология:

- область определения **вычислимой** функции **перечислима**, то есть можно перебрать элементы области определения функции.
- существует **перечислимое** , но **неразрешимое** множество.

Пример: перечислимое множество

Множество X называется перечислимым (computably set), если выполняется хотя бы одно из условий:



- существует программа, перечисляющая все элементы X в произвольном порядке;
- X является областью определения вычислимой функции f;
- X является областью значений вычислимой функции f;

Разрешимость множества

- Всякому математическому утверждению можно придать вид утверждения о каких-либо множествах.
- **Пример** Рассмотрим множество квадратов натуральных чисел. Это множество **перечислимо**: для получения его элементов нужно последовательно брать числа и возводить их в квадрат. Другими словами, есть **область значений вычислимой функции** .
- Более того, это множество также может быть **разрешимым**: для проверки того, принадлежит или нет некоторое число данному множеству M, нужно построить **характеристическую**

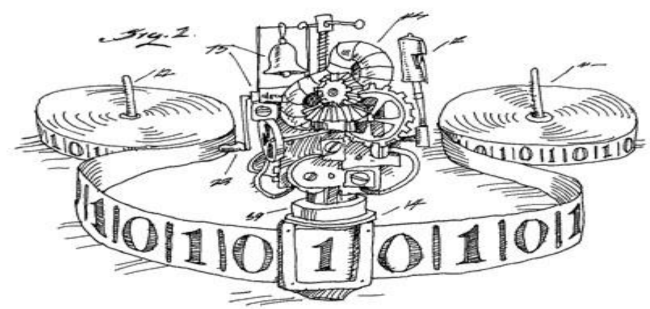
  функцию:

$$\chi_M(x_1,\dots,x_n)=\begin{cases}1,(x_1,\dots,x_n)\in M,\\0,(x_1,\dots,x_n)\notin M.\end{cases}$$

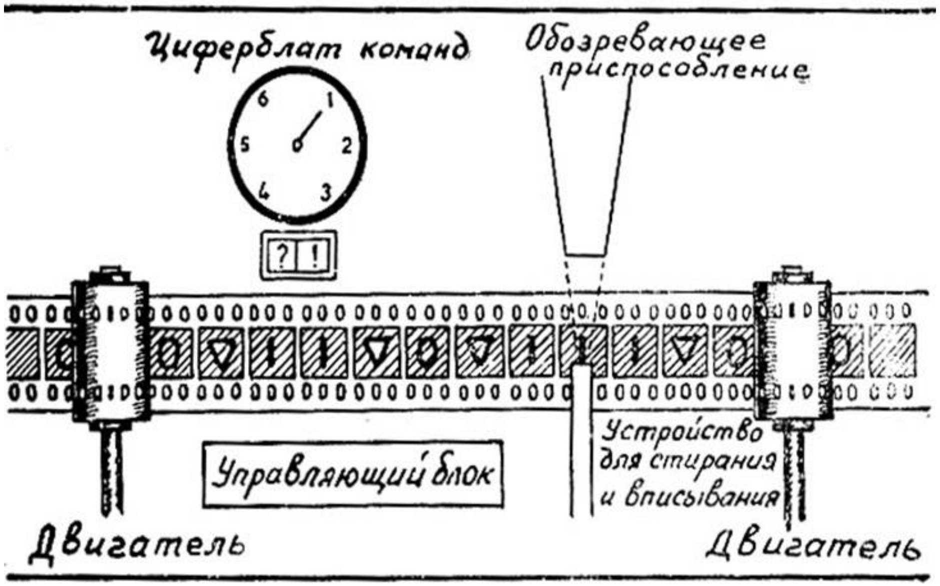
Модель вычислений Черча-Тьюринга

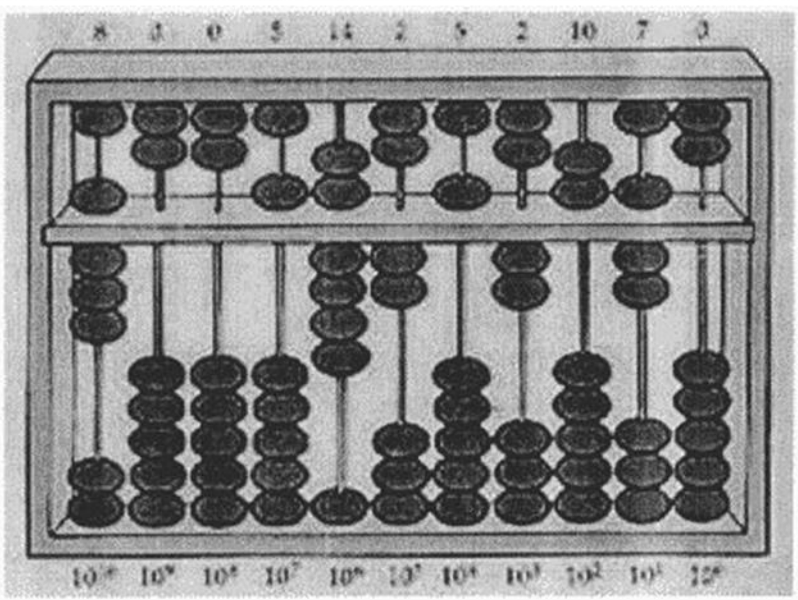
- **Идея:** Все вычислительные операции, которые «чувствительны» только к **синтаксическим правилам** , могут быть смоделированы механически, т.е. с помощью простых «механических перемещений» этих символов и использования памяти для хранения «вычисленного» символа.
- Вычисление нового символ основано на правилах вывода, которые записаны в терминах синтаксических свойств самих обрабатываемых символов (тезис Маркова).

Пример операций: сложение целых чисел



Организация вычислений в «машине Тьюринга» (МТ)





МТ - это автомат с «бесконечной цифровой лентой, движение которой есть «вычисление» нового символа, который кодирует полученный результат.

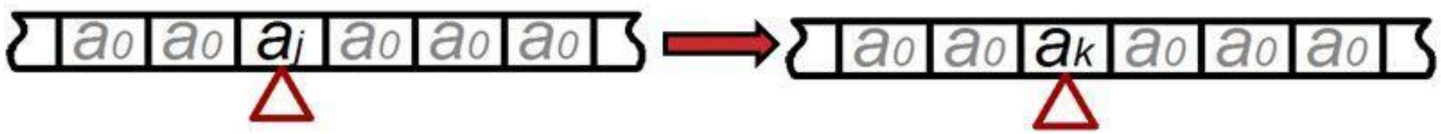
Физическое «поведение» МТ – основано на работе двигателей, которые осуществляют «перемещение ленты», на которой расположены символы «алфавита» , подсчете количества перемещений, устройства «обозревающего» результат и запоминания, полученных результатов.

Механизм процесса алгоритмических вычислений или

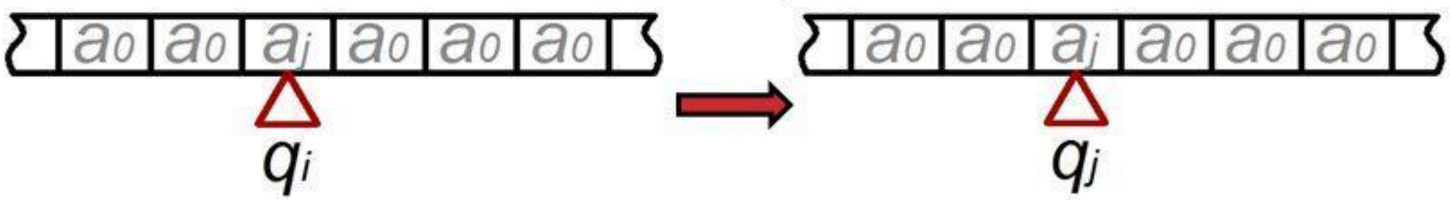
Действия машины Тьюринга

За один такт своей работы машина Тьюринга может:

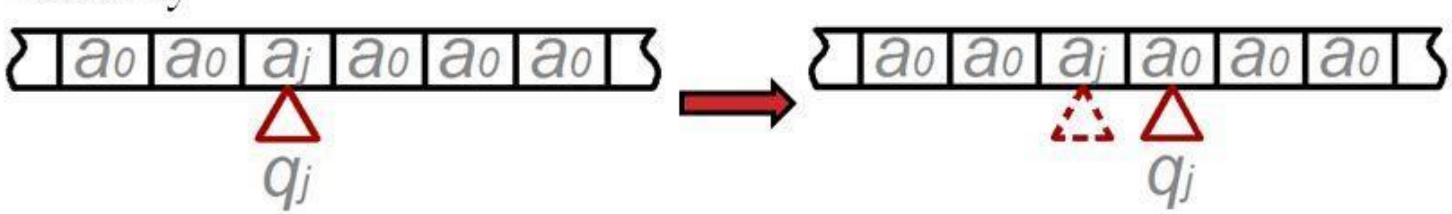
1) Изменить / не изменить символ, записанный на ленте



2) Изменить / не изменить своё внутреннее состояние



3) Переместить головку по ленте влево / вправо / не перемещать головку



Обобщения: Тезис Черча-Тьюринга (Ч-Т) и тезис Маркова

Тезис **Черча-Тьюринга**: Вычисления положительных целых чисел – это формализуемая механическая процедура, реализуемая с помощью «логической» машины Тьюринга (МТ).

Тезис **Марков**: «Нормальный» алгоритм вычислений это правила подстановки символов алфавита, с помощью которого описывается результат вычислений

Итого:

1. Вычисления в смысле тезиса Ч-Т рассматриваются как механический процесс и процедура механической подстановки, которые реализуются с помощью абстрактной логической , а не физической машины.

2. Большинство физических систем могут быть описаны с помощью универсальных вычислительных механизмов с использованием абстрактной машины Тьюринга и нормальных алгоритмов Маркова.

Может ли МТ «вычислить» большую физическую систему ?

Возникает фундаментальный вопрос: почему **физические процессы** д.б. эквивалентны процессам вычислений (отношение гомоморфизма) в смысле тезиса Ч-Т?

- МТ – это лишь logical computing machine , а не «физическая» машина. С помощью МТ вычисляются «точки» в пространстве N (натуральных чисел), а не свойства физических систем. Связь между «точками» и свойствами - область сознания.
- Физические системы в процессе своего «существования» формально не вычисляют ни каких «точек» и ни когда не останавливаются (does not calculate a function and it is not expected to stop). Точка в физическом смысле не существует и д.б. заменена некой протяженной структурой или «струной».

Итак,

1) С точки зрения физики МТ выбирает точки в области значений вычислимой функции. Достижение это точки осуществляется с помощью «вычисление» некой **траектории движения ленты**.
2) Законы такого движения задаются «кодами символов, которые являются «носителями» элементарных действий, осуществляемых «двигателями» под **контролем системы управления МТ**.
3) «Сеть из МТ» как модель поведения физической системы не может быть эффективно реализована с помощью одной МТ . В сети из МТ нет информационного обмена между «лентами»

Вычисления как процесс физической реализации алгоритма

Итак, МТ исполняет предопределенный алгоритм, записанный как текст с помощью символов алфавита. Вопрос,

- может ли машин Тьюринга адекватно вычислять поведение объектов, которые находятся в «вечном движении» ?
- Как можно вычисления в МТ изменить, чтобы «любое множество данных о физических объектов» было бы алгоритмически разрешимо ?

Хотя современные компьютерные системы :

- На макро уровне имеют большое число процессорных ядер объединяются в локальный кластер, распределенный грид или глобальное «облако».
- Компьютеры работают как «алгоритмически замкнутые» системы, которые состоят из отдельных частей и , хотя и взаимодействуют с реальными физическими объектами, сенсорами и актуаторами , но …**не способны к обучению,** т.е. «на лету» менять «грамматику» реализуемых алгоритмов.

Три основных класса алгоритмических моделей

- **Первый класс** моделей основан на арифметизации алгоритмов - любые данные можно закодировать числами, и как следствие — всякое их преобразование становится в этом случае арифметическим вычислением - рекурсиями
- **Второй класс** - модели, которые выполнимы (вычислимы) с помощью абстрактной машина, к которой предъявляются требования простоты и универсальности (т.н. машина Тьюринга).
- **Третий класс** алгоритмов задается с помощью алфавита, над символами которых определено конечное множество допустимых подстановок и порядок их применения (нормальные алгоритмы Маркова -суть подстановками , так слово «слон» превращается в слово «муха» :

  «слон» —> «суон» —» «муон» —> «мухн» —> «муха».)

Фундаментальные ограничения: принцип неопределенности

Принцип неопределенности **сформулирован Гейзенбергом для квантовой механики:**

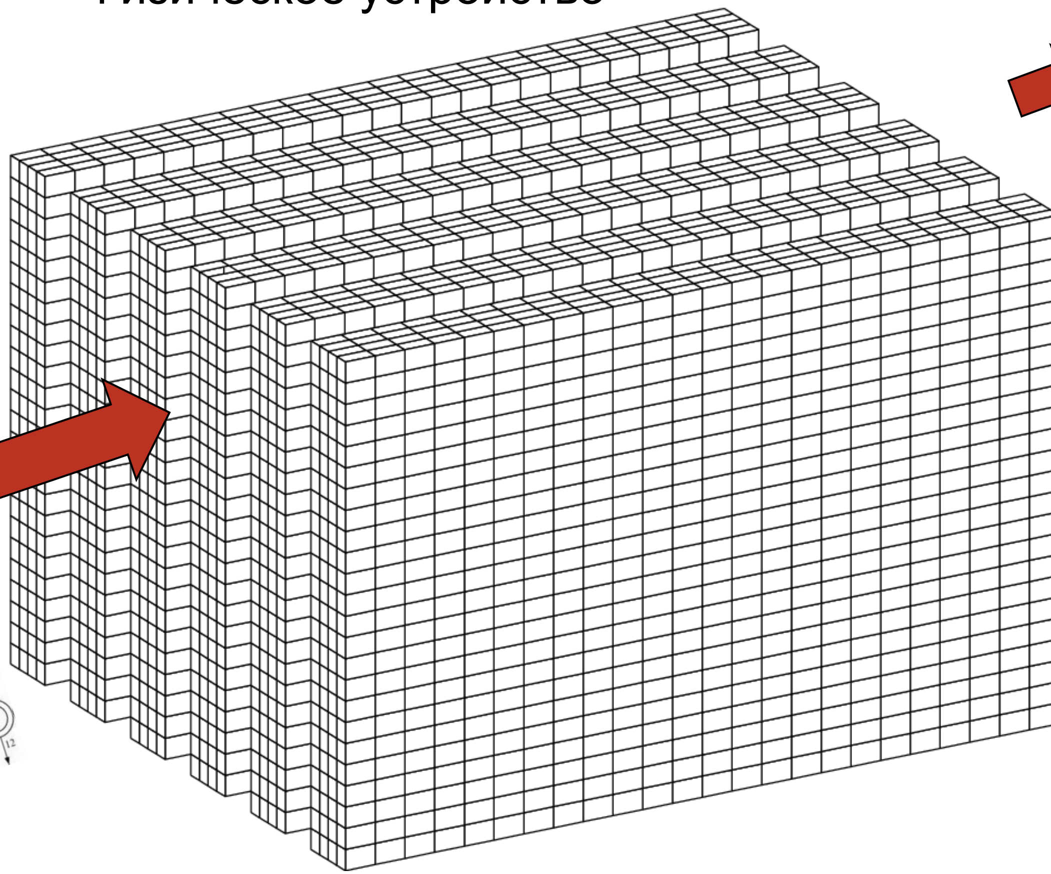
- **«невозможно одновременно точно измерить импульс и координаты квантового объекта»**

но в *теории алгоритмов это означает* :

- **невозможно одновременно обеспечить перенос семантики (смысла) алгоритма (текста) с одного языка описания на другой и сохранив при этом число используемых для описания алгоритма символов (сохранив синтаксические параметры)»**

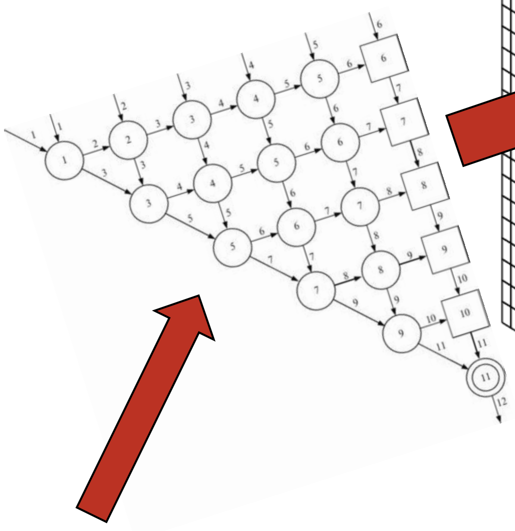
Представления алгоритма на языке параллельных потоков вычислений – много инструкций – много данных (MIMD)

Физическое устройство



Текст

текст

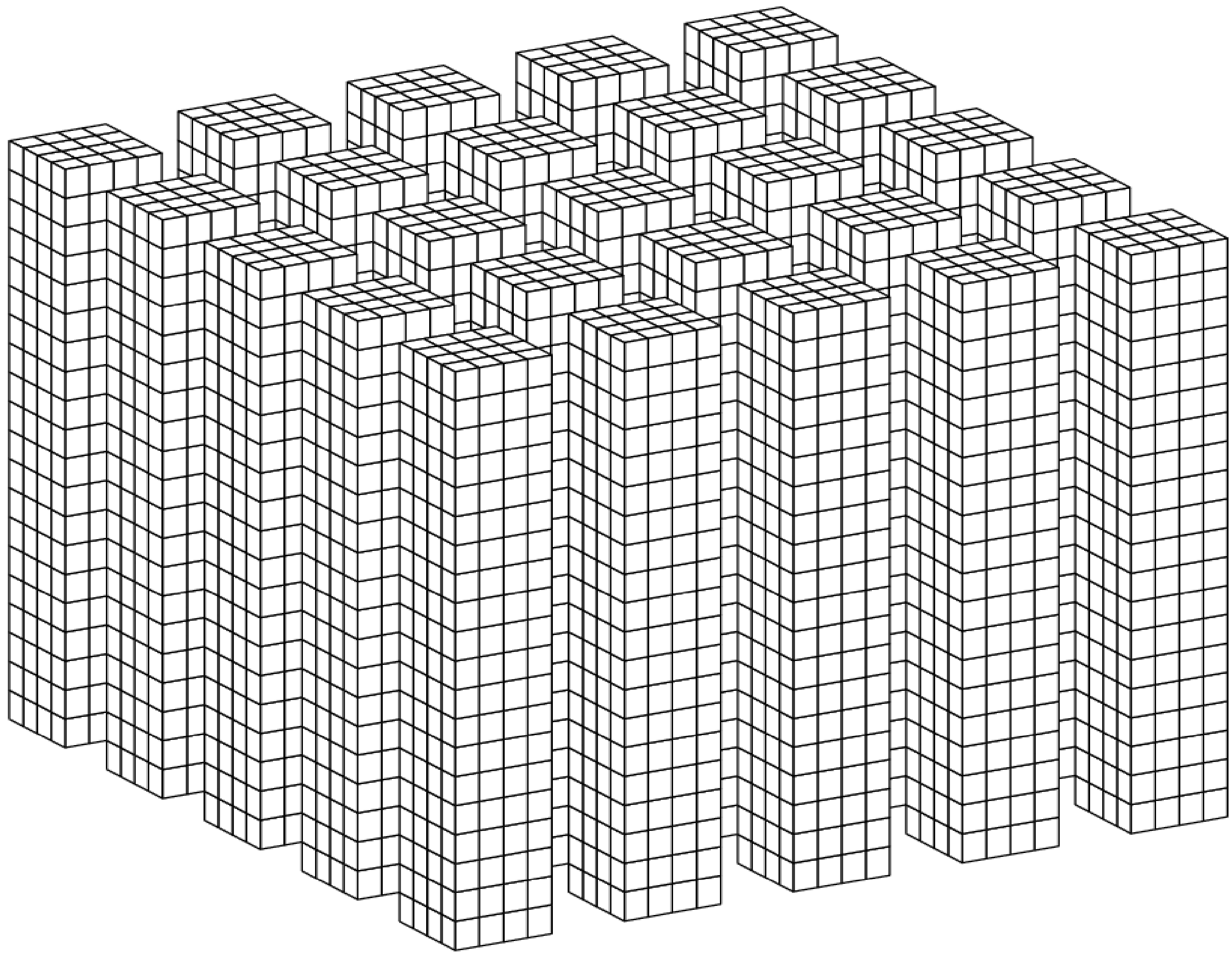


Алгоритм

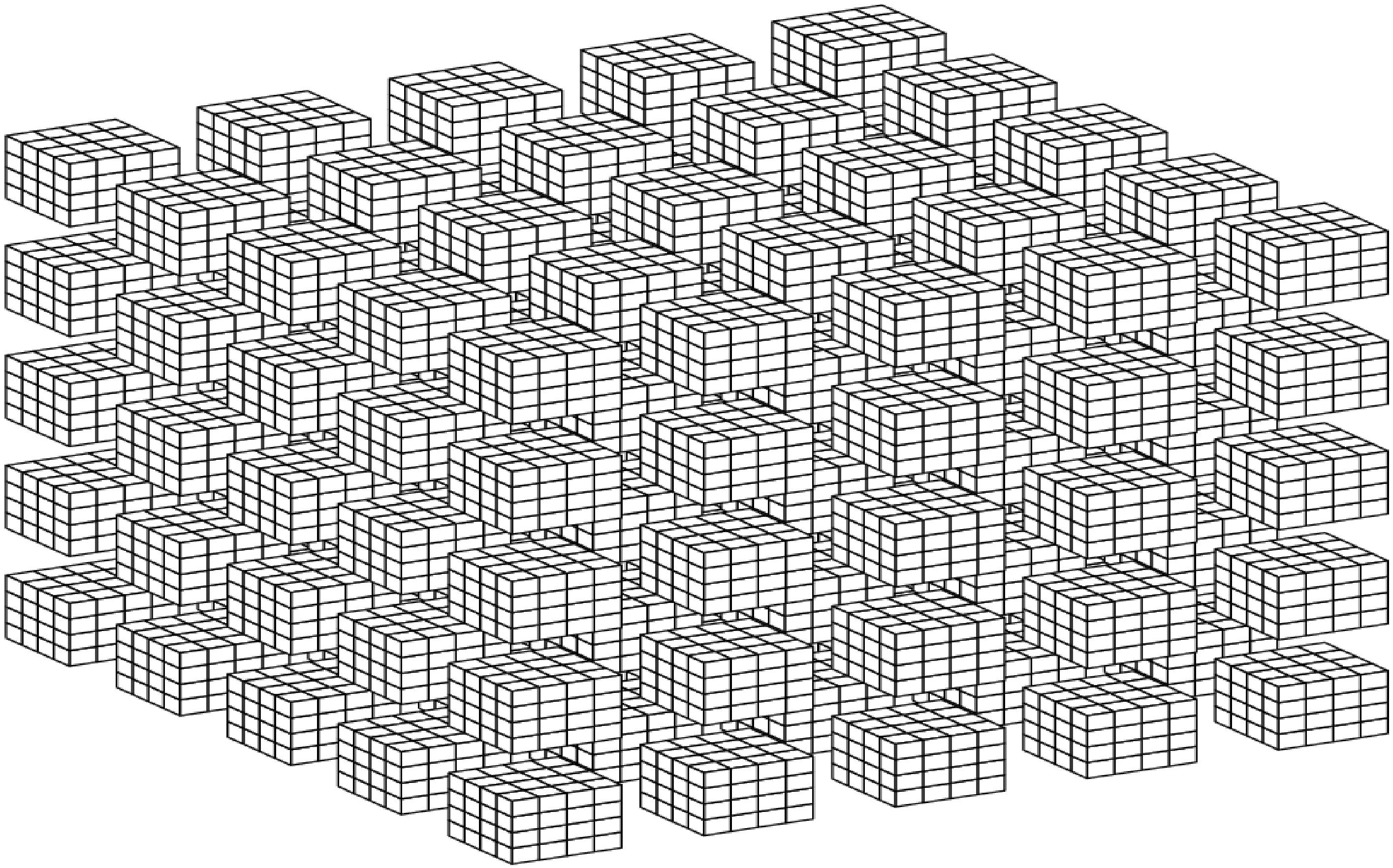
Вычислитель MIMD – Multiply Instruction-Multiply Data

Результат

Вариант реализации: Метод суперскалярных операций - различные операции выполняются над разными данными (SIMD)



Вариант реализации: Вычислительная система с одиночным потоком команд и одиночным потоком данных (SISD)



Выводы

«Целые числа сотворил господь бог, а все прочее – дело людских рук».
Леопольд Кронекер (1823 – 1891)

С позиций теории алгоритмов и тезиса Ч-Т, чтобы процесс можно было назвать вычислением его описание должно включать :

- Алфавит
- Грамматику/онтологию построения слов из символов алфавита
- Алгоритм решения как последовательность операций над словами
- Узлы обработки, где над словами производятся операции
- Механизм, который позволяет реализовать эти операции, формируя последовательность «состояний» рассматриваемого процесса.