ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ

ВШ ИСКУСТВЕННОГО ИНТЕЛЛЕКТА

Введение в профессиональную деятельность

ЛЕКЦИЯ 4 : ШАХМАТНАЯ ДОСКА ПРИРОДЫ

3.03.2022

ПОЛИТЕХ

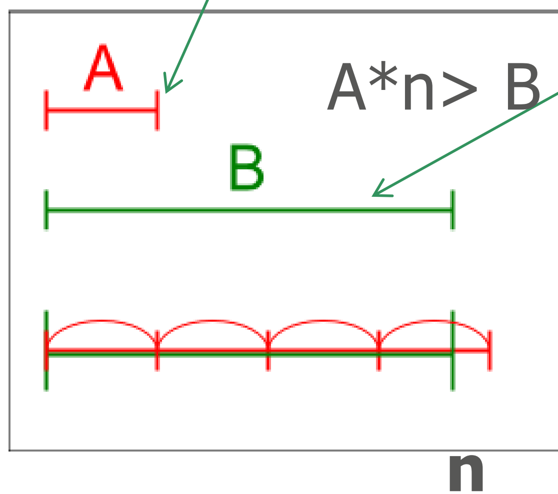
Что обсуждали на прошлой лекции

- Обсуждали аксиому Архимеда – основу **счетной аддитивности** свойств физической реальности и гипотезу **«it from bit»** - основу концепции цифровой физики

- Было показано, что в природе отсутствуют бесконечно малые и бесконечно большие величины. При этом природный объект МОЖЕТ **одновременно** находится в нескольких логически несовместных состояниях (состояние суперпозиции), которые физически невозможно разделить без разрушения самого объекта!
- Суперпозиция носит информационный характер и может реализовываться без взаимного физического (энергетического) влияния одного состояния на другое. В состоянии суперпозиции информация о свойствах объекта не «рассеиватся» (не теряется), поэтому не происходит диссипация энергии и увеличение энтропии.

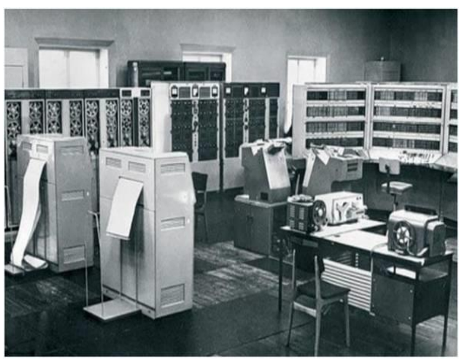
Напоминание: Аксиома Архимеда

Аксиома Архимеда : если даны отрезки А (масштаб) и В (объект измерения) , то можно отрезок А отложить несколько раз так, что сумма будет равна или «немного» превосходить отрезок В ,



Утверждение: ПРОСТРАНСТВО-ВРЕМЯ физической реальности архимедово то есть **одномасштабно**, гладко, «делимо» и «однородно». На языке математики аксиома Архимеда утверждает буквально следующее: **В природе не существует бесконечно малых и бесконечно больших величин.**

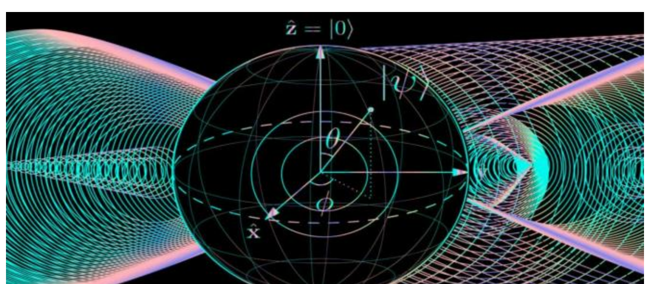
Напоминание: компьютер - «цифровой паровоз»



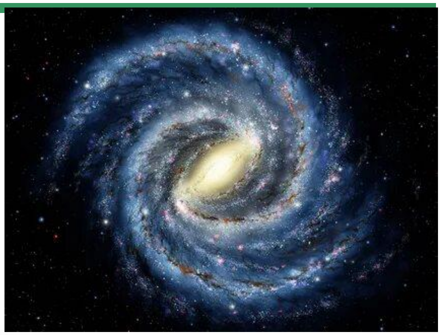
- Правило 1: шахматные операции необратимы….

Правило А: цифровые вычисления в компьютере не обратимы. Вентили «И-НЕ», принимая на вход два бита, выдают результат размером всего **один бит**. По полученному результату, нельзя однозначно восстановить значения двух исходных аргументов. Вычисления необратимы….Каждая логическая операция (вычисление) вентиля «И-НЕ» уменьшает информационную энтропию системы (на 1.189 бита), и, соответственно, рассеивает не менее ~0.02 эВ тепла.

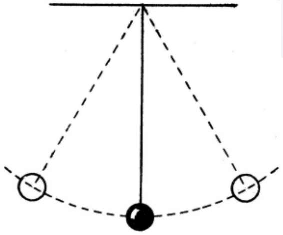
Аналогично, любая запись в ячейку ОЗУ приводит к уничтожению предыдущих значений. На физическом уровне «старая» информация она рассеивается в пространстве в виде тепла или эл/маг излучений.



«ШАХМАТНОЕ» ИНФОРМАЦИОННОЕ ПРОСТРАНСТВО ПОЧЕМУ НЕЛЬЗЯ МЕНЯТЬ ПРАВИЛА ВО ВРЕМЯ ИГРЫ ?



Уменьшение энтропии



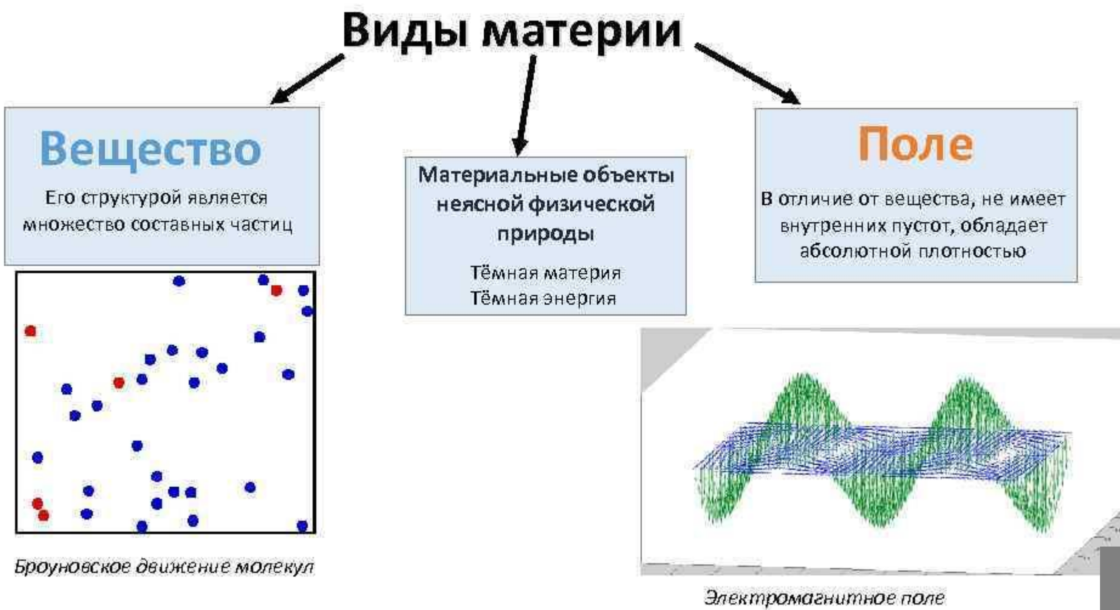
Усложнение правил

ПРИНЦИП «ШАХМАТНОЙ ДОСКИ»

Благодаря «сепарабельности» топологического «пространства-время», окружающего человека, в нем можно выделить классы эквивалентности, т.н. фактор-множества (например, черные/белые клетки). На основе классов эквивалентности можно сформировать конечное подмножество состояний (траекторий), компоненты которого суть последовательности элементов, состоящих из компонент базиса топологически рассматриваемого пространства, а их мера образует инвариант, отражающий законы сохранения , актуальные для физических законов

Для описания базиса топологии подходят различные «цифровые меры», которые выражаются через рациональные или вещественные числа (всякое вещественное число можно представить в виде предела последовательности из рациональных чисел) Такая «цифровая» реальность порождает явления, которые можно наблюдать (измерять с помощью приборов-инструментов), получая **информацию** о том, какие числовые характеристики могут быть сопоставлены этим явлениям.

ШАХМАТНАЯ ДОСКА ПРИРОДЫ…. «МАТЕРИЯ-ИНФОРМАЦИЯ»



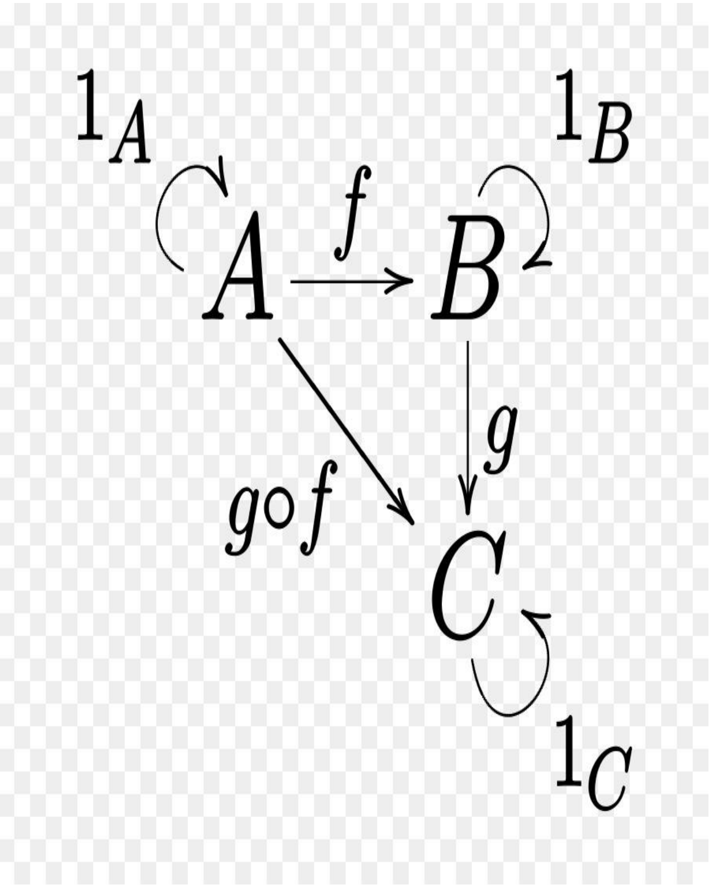
Можно не только переставлять по правилам фигуры, но и «переставлять правила»

Математическая основа игры на такой шахматной доске: **теория категорий**

Раздел математики, изучающий свойства отношений между математическими объектами, не зависящие от внутренней структуры объектов.

Категория — простая концепция. ЕЕ суть в том, что есть функция (морфизм) f, которая принимает аргумент типа A и возвращает значение типа B. Есть другая функция (морфизм) g, которая принимает B и возвращает C. Эти операции образуют композицию из f в g. Композиция ассоциативна. Если у вас есть три морфизма, f, g и h, которые могут быть составлены (то есть, их типы согласованы друг с другом), вам не нужны скобки, чтобы составить их. Математически, это записывается так:

$h \circ (g \circ f) = (h \circ g) \circ f = h \circ g \circ f$



Формальные определения

Имеется класс объектов , обозначаемый как Ob_C ;

Для каждой пары объектов A, B из этого класса **Ob$_c$** задано множество морфизмов (или стрелок), которые обозначаются как

$Home_c$ (A,B), причем каждому морфизму соответствуют единственные A и B

Имеется единичная стрелка объекта A, которая называется idA

Монáда — особый тип данных в языках программирования, для которого возможно задать последовательность выполнения операций над хранимыми значениями…. с побочными эффектами (например, с помощью функциональных языков можно осуществлять императивные операции)

$$\begin{tikzcd} \Phi(X) \arrow[r, "\Phi(f)"] \arrow[d, "\eta_X"'] & \Phi(Y) \arrow[d, "\eta_Y"] \\ \Psi(X) \arrow[r, "\Psi(f)"] & \Psi(Y) \end{tikzcd}$$

Пример

Допустим, у нас есть три унарные функции: f1, f2 и f3, принимающие число и возвращающие его увеличенным на 1, 2 и 3 соответственно. Также каждая функция генерирует сообщение, представляющее собой отчёт о выполненной операции.

def f1(x): **return** (x + 1, str(x) + "+1")
def f2(x): **return** (x + 2, str(x) + "+2")
def f3(x): **return** (x + 3, str(x) + "+3")

Надо объединить их в цепочку для обработки параметра x, иначе говоря, мы хотели бы вычислить x+1+2+3. В идеале, программа должна выглядеть как простая цепочка функций, вроде **f3(f2(f1(x))).** К сожалению, типы данных, возвращаемых f1 и f2, не соответствуют типам параметров f2 и f3

Следующая диаграмма показывает вычислительный процесс, происходящий при x=0, где v1, v2 и v3 – значения, полученные в результате вызовов unit и bind

x = 0 —*unit(x)*→ v1 = (0, "Ops:") —*bind(v1, f1)*→ v2 = (1, "Ops:0+1;") —*bind(v2, f2)*→ v3 = (3, "Ops:0+1;1+2;") —*bind(v3, f3)*→ (6, "Ops:0+1;1+2;3+3;")

Функция unit преобразует входной параметр x в кортеж из числа и строки. Функция bind вызывает функцию, переданную ей в качестве параметра, и накапливает результат в промежуточной переменной t. Теперь если появится функция f4, мы просто включим её в цепочку:

bind(f4, bind(f3, ...))

Что стоит прочитать и понимать

АКАДЕМИЯ НАУК СССР

Научный совет по комплексной проблеме «Кибернетика»

Ю. А. ГАСТЕВ

ГОМОМОРФИЗМЫ И МОДЕЛИ

Логико-алгебраические аспекты моделирования

ИЗДАТЕЛЬСТВО «НАУКА»

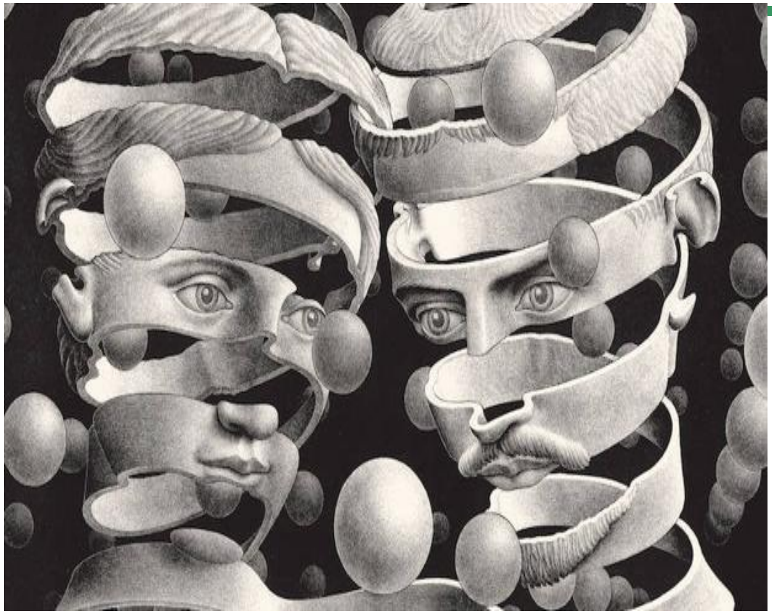
Москва 1975

Морфизм - морфизм представляет собой сохраняющее структуру **отображение** одной математической структуры в другую того же типа

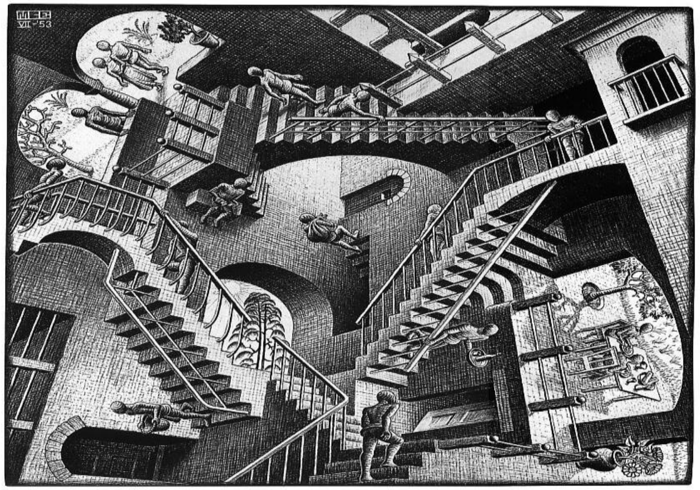
Изоморфизм (от греч. ἴσος - равный, одинаковый и μορφή - форма) - **отношение** между объектами, выражающее в некотором смысле тождество их структуры (строения)

Гомоморфизм — это морфизм в категории алгебраических систем, то есть **отображение** алгебраической системы A, сохраняющее основные операции и основные отношения.

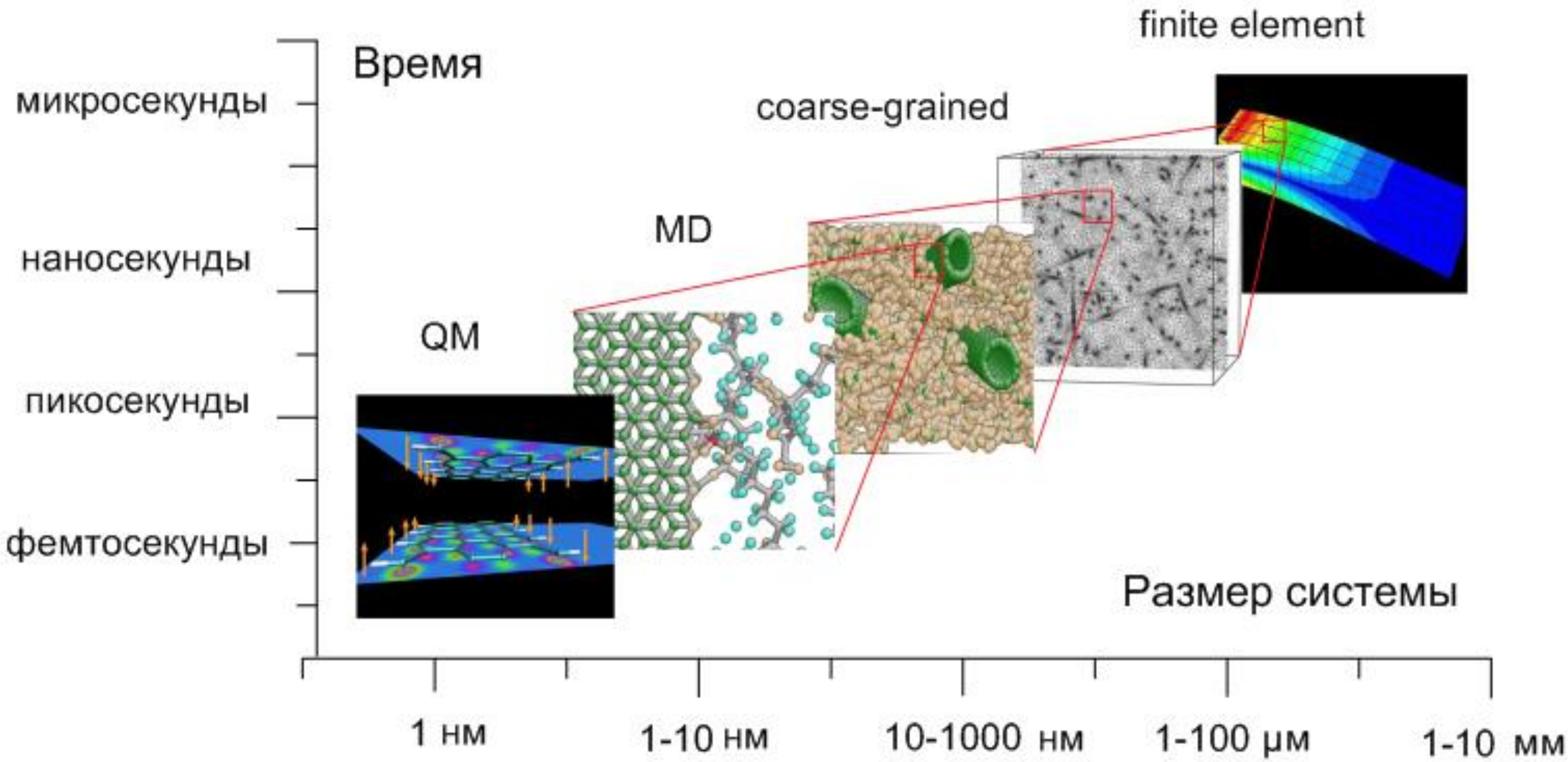
Моде́ль (от лат. modulus — «мера, аналог, образец») — это упрощенное **представление** реального устройства и/или протекающих в нем процессов, явлений.



Феномен самореференции (self-reference), - это «наведенное» (индуцированное) свойство, возникающее в сложных системах наделенных strange loop или циклической структурой, которая проходит через несколько уровней иерархии целостной системы и попадает в исходную точку (Эшер: moving only up or down the levels of the hierarchical system, **one** finds oneself where one began).



Отношение «время протекания процессов – масштаб системы»



Основная проблема компьютерных наук – сложность описания свести к сложности вычислений

Суть проблемы – определить сколько и каких базовых вычислительных операций требуется для решения задачи ?

Пример: сложность решения задача «коммивояжёра» - который должен посетить n городов.
Формальное точное решение требует n! Операций

Нужно ли нам на практике «точное решение» ?

Если n=49 ,то число операций > числа атомов во Вселенной.

Варианты сложности вычислительных задач:

1) Полиномиальное «время» (от размерности) –> n^3, n^7, …
2) Экспоненциальное «время» –> 2^n, n!.. .

Сопутствующие проблемы

- Редукция - выявление условий, при выполнении которых появляется возможность использовать для решения полиномиальные алгоритмы
- Эффективность - сокращение объема используемой памяти, параллельность процессов вычислений,…..
- Аппроксимация - поиск приближенного ответа, который оптимален “most of the time and cases”,
- Рондомизация - разработка вероятностных алгоритмов поиска решений, протоколов, моделей или оценок.

ФОРМАЛИЗАЦИЯ КЛАССОВ «СЛОЖНОСТИ» ЗАДАЧ

- Класс P - сложно решить, легко проверить (быть творческим)
- Класс NP - легко решить , сложно проверить (оценить творчество)

Если удастся доказать, что «P= NP» , то это будет эквивалентно тому, что всегда можно найти «удовлетворительное» решение прикладной задачи за полиномиальное время

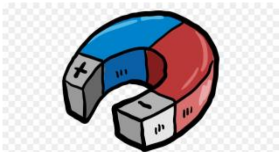
ПРИОРИТЕТНАЯ ЗАДАЧ - ПОИСК «УДОВЛЕТВОРИТЕЛЬНОГО» РЕШЕНИЯ

Дано: множество “ограничений”
Требуется: вычислить n-bit слово, которое кодирует “решение” и, которое удовлетворяет всем этим ограничениям

«Решение», должно быть таким, чтобы его можно было **легко проверить** на «удовлетворение заданным ограничениям»

Пример задачи, которую трудно решить: «поиск иголки в стогу сена»

- Почему «трудно» ?
- Как можно радикально «облегчить» поиск ?



Поиск «удовлетворительного» решения - задача Boolean Satisfiability

Формализация задачи: сведение задачи к разрешимости предиката

$$F=(A + B + C) \cdot (D + F + G) \cdot (A + G + K) \cdot (B + P + Z) \cdot (C + U + X)$$

где A,B,C,D,....X – логические переменные, F={1,0}

Вопросы:

- Имеется ли «простой» алгоритм поиска «satisfying assignment»?
  - Что делать, если переменных будет не 3 , а 100 или 1000 ?
    - Сколько «времени» надо затрать, чтобы найти «удовлетворительное назначение» и …. как такое решение закодировать с помощью булевых переменных ?

Если все таки P = NP, то

Это будет научная «революция» 5.0

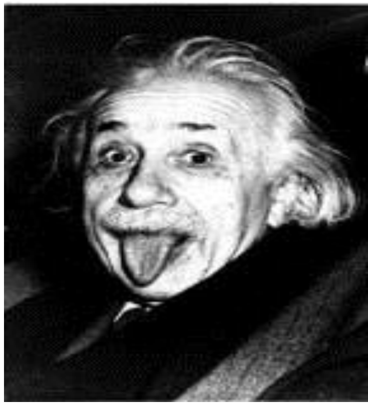
- Доказательство любых математических теорем может быть найдено за полиномиальное время
- Любые последовательности (кодовые образцы) в множестве экспериментальных данных могут быть найдены за полиномиальное время от длины рассматриваемой последовательности данных
- Проблем Искусственного Интеллекта будут иметь полиномиально- эффективные алгоритмы.

ПОЛИТЕХ

«Арсенал» современных компьютерных наук: программирование, вычисления машинное обучение

P=NP или:
Можно ли научиться решать сложные задачи?
Как научиться хорошо играть в шахматы ?

Может ли робот «обучится» до…



С чем связать процесс обучения - с навыками выполнения отдельных операций, следование алгоритму или практическим опытом ?

Темы для доклада: утверждения о сложности как мере физической целесообразности

(A. Yao) Computational complexity of physical theories (e.g., general relativity)?

(Denek and Douglas): Computational complexity as a possible way to choose between various solutions (“landscapes”) in physical theory.

Пример: Вычислительная сложность физических проблем

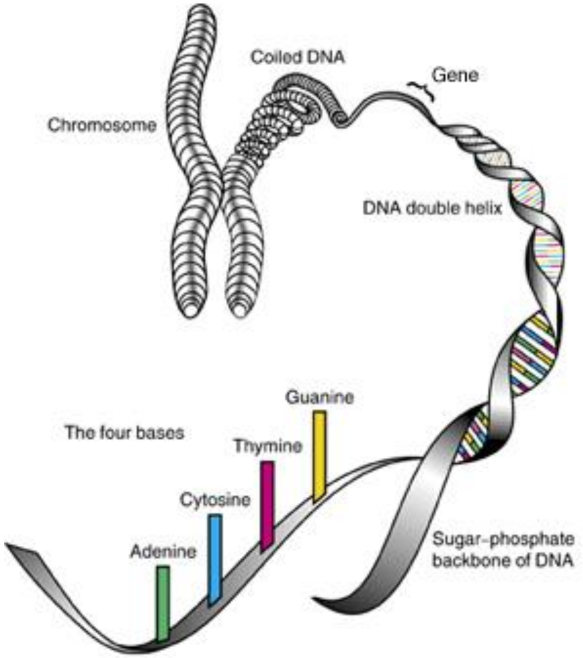
расшифровка генома (long sequence of A,C,T,G)

Метод:

•Extract many random fragments of selected sizes (2, 10, 50 150kb)

• For each fragment, read first and last 500-1000 nucleotides (*paired reads*)

• Вычислительно собрать геном из парных считываний.



adapted from the DOE Joint Genome Institute website
www.jgi.doe.gov
used with permission

Новое понятие - Наука, основанная на алгоритмах или“Algorithm driven science”

ПОЛИТЕХ

Аксиомы «новой науки»

Аксиома 1:
если **вероятность** обнаружить частицу в состоянии $|k\rangle$ **равна 1**, то частица *действительно* находится в состоянии $|k\rangle$.

Аксиома 2:
любое унитарное (обратимое) преобразование можно «воплотить» в виде физического устройства**.**

Аксиома 3 (об отсутствии скрытых параметров):
если система находится в состоянии, которое описывается волновой функцией - набором независимых физических величин, задающих вектор **состояние системы**, то вся информация о результатах измерений содержится в векторе состояния этой системы**.**

Аксиома 4:
невозможно передать информацию от одной системы к другой без физического взаимодействия, скорость которого не больше **c** в вакууме.

Заключение: методы описания и понимания реальности

Научные модели процессов и явлений, основанные на **данных измерений** (физических процессов) .

Научные принципы использования данных (19-20 вв) :

Редукционизм: описание процессов как феномена, который сводится к свойствам «простых» его составляющих

Научные принципы использования данных (21 век) :

Системная сложность: понимание процессов требует рассмотрения их как неделимого целого, свойства которого появляются в разных условиях по разному, но эти свойства у отдельных частей принципиально **отсутствуют.**