



Санкт-Петербургский  
Государственный  
Политехнический  
Университет

Институт прикладной  
математики и механики

КАФЕДРА

ТЕЛЕМАТИКА

**Физика вычислительных процессов**

Лекция 7-1

**От «науке о данных» к теории  
алгоритмов**

---

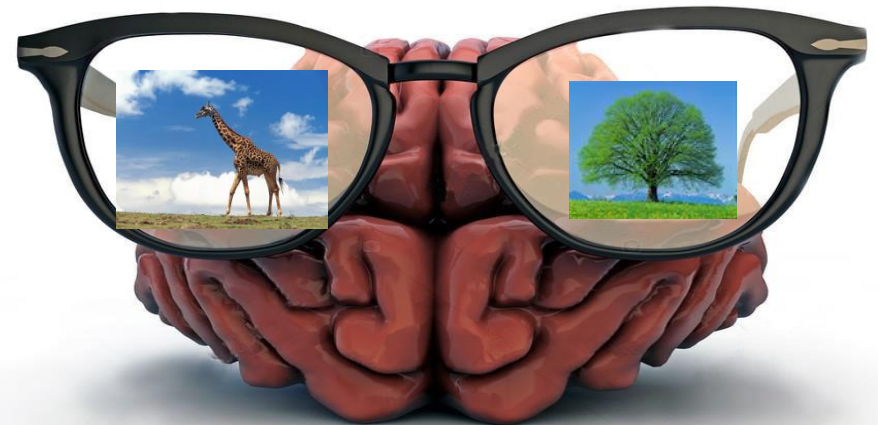
5 Мая 2020 г.

# Вопрос: данные & алгоритмы - что важнее ?

- Для интеллекта человека характерны две ключевые модальности - «знать и понимать»
  - **Знать** – это значит иметь **данные**
  - **Понимать** – это значит иметь **алгоритм** использования данных

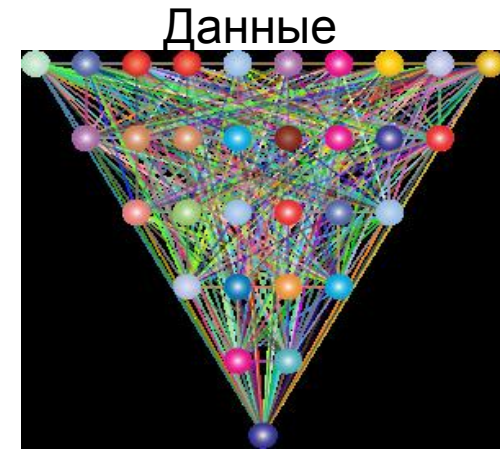
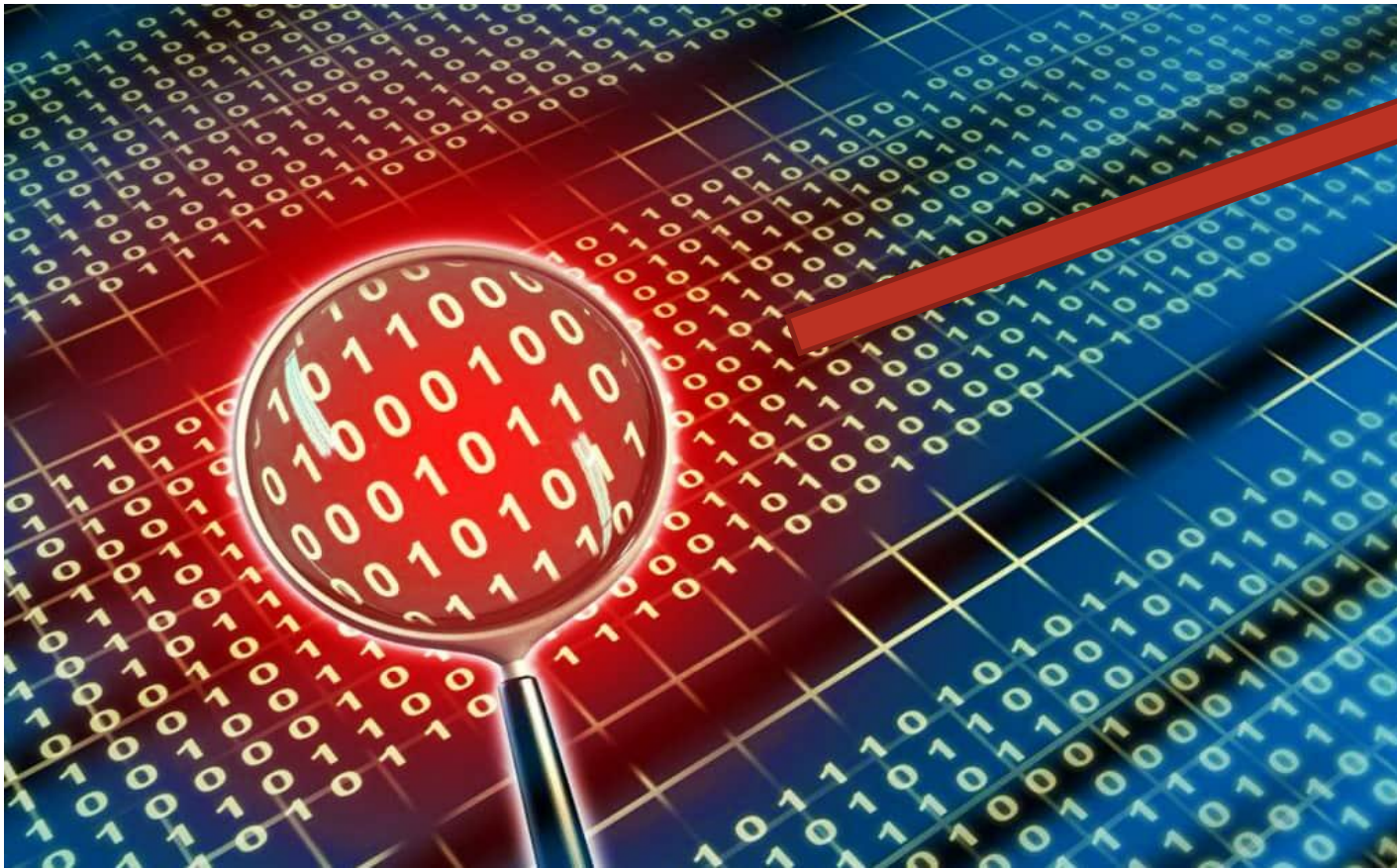
- **Интеллект** использует полученные от органов чувств данные для принятия решений на основе имеющихся знаний.
- Сами **данные** выполняют роль «интерфейса» между физическим и ментальным мирами, а часть знаний принимают «формат» алгоритмов

Человек «смотрит» на мир, обрабатывая данные



Может ли интеллект «заменить» данные ?

# Целостность данных и разрешимость алгоритмов



информация



В принципе **данные** могут содержать **больше** информации, чем **цифровой код**. Основной вопрос Data Science – как из множества полученных данных **извлечь информацию**.

# Дискретны и непрерывные данные- что важнее

"Я все больше и больше склоняюсь к мысли, что нельзя продвинуться дальше, используя теории, строящиеся на континууме".

А. Эйнштейн

Среди всех понятий физики время оказывает наибольшее сопротивление свержению мира идеального континуума в мир дискретности, информации, битов...

Дж. А. Уилер

# Математические данные о «физической» реальности

Если «единичную меру» данных в математике обозначить символом «1», то единичной мере вычислительной сложности можно сопоставить результат представления «1» в одномерном пространстве 1D с помощью 100 отдельных компонент, получим вариант «линейного приближения»:

$$1^{100} = \text{«1»}$$

В физической реальности единичной мерой сложности обладает, например, координата «время» - одномерная или 1D переменная. Если одномерную переменную заменить на двумерный 2D «вектор», который учитывает и «настоящее» и «будущее», т.е. скаляр заменить на математический объект, имеющий размерность 2, то мера «сложности» задачи возрастет и станет

$$2^{100} = 1,2676506 \times 10^{30} \approx \text{«1»} * \infty$$

Если же «вектор времени» представить как «объект» в трехмерном пространстве, описывающим «прошлое, настоящее, будущее», то вычислительная сложность «реальных» задач потребует использования «интеллектуальных» вычислительных инструментов, которые способны «учиться» и использовать для расчетов только малую часть 3D «пространства» данных.

# Как можно «закодировать» всю Вселенную ?

Есть оценка, что число нуклонов в видимой Вселенной  $10^{80}$

Информация в компьютере записывается в виде «нулей и единиц», так, чтобы сохранить цифровую фотографию объемом 3 мегабайта надо

$3 \times 8 \times 2^{20}$  т.е. около 25 миллионов ячеек памяти компьютера

Формально объем хранимых данных ни чем не ограничен, но ограничена скорость доступа к хранимым данным.

- **Метод «LogLog»** - преобразование массива данных произвольной длины в (выходную) битовую строку фиксированной длины:
  - $\log_2(\log_2(1000\ 000\ 000)) = 4.9$  Каждой записи присваивается т.н. хэш-значение, т.е. на исходных данных формируется «свертка». Свертка или хеш-функция менее разнообразна, чем значения входного массива (входные данные), но позволяет сформировать «адрес» того места, где хранятся сами данные.
  - Пример - хеш-функция остаток от деления числа N на число M.

# Возможности «компьютерных» алгоритмов

Современные микропроцессоры ( новое поколение контактно-релейных схем» построены на базе транзисторов. Минимально возможные линейные размеры транзистора ( $x_{\min}$ ) и максимальная рабочая частота ( $f_{\max} = 1/t_{\min}$ ) ограничиваются соотношения неопределенностей Гейзенберга ( $h$  – постоянная Планка):

$$\Delta x * \Delta p = x_{\min} * \Delta p \geq h/2\pi, \quad \text{где } \Delta p = \sqrt{2} * m * E_{\text{bit}} = \sqrt{2} * m * kT * \ln 2$$

$$\Delta t * \Delta E = t_{\min} * \Delta E \geq h/2\pi, \quad t_{\min} = h/2\pi / \Delta E, \quad \text{где } \Delta E = kT * \ln 2, \quad k - \text{постоянная Больцмана}$$

$$1,380\ 649 \cdot 10^{-23} \text{ Дж} \cdot \text{К}^{-1}$$

при  $T=300 \text{ К}$ , получим  $x_{\min} = 1.5 \text{ нм}$ , а  $t_{\min} = 4 * 10^{-14} \text{ с}$ .

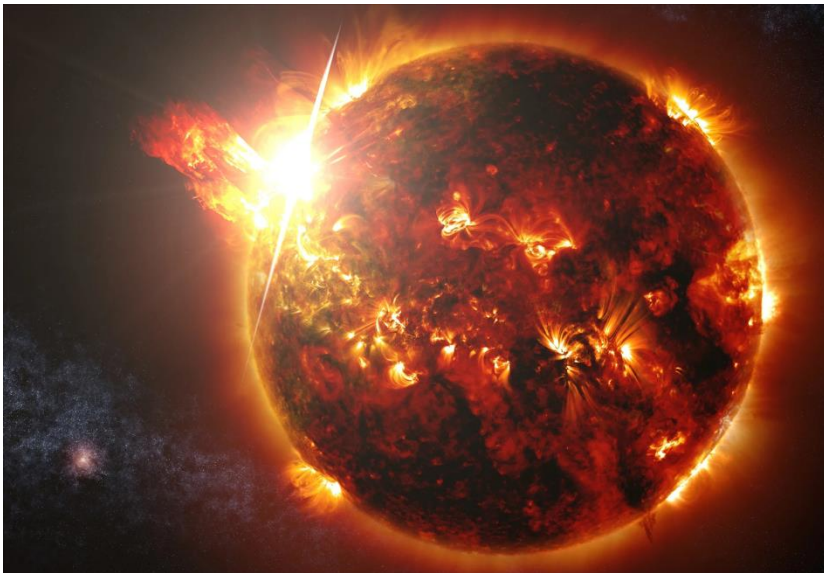
Если создать компьютер, у которого одновременно будет и самая большая плотность упаковки (определяемая  $x_{\min}$ ) и максимально возможная частота (определяемая  $t_{\min}$ ), и оценить, какая при таких условиях должна выделяться мощность  $P$  на единицу площади, то при  $T=300 \text{ К}$ , получим:

$$P = E/t = n_{\max} E_{\text{bit}}/t_{\min} = kT * \ln 2 / ((x_{\min})^2 * t_{\min}) = 3.2 * 10^6 \text{ Вт/см}^2,$$

где  $n_{\max}$  – число транзисторов, которые занимают площадь, определяемую линейными размерами канала ( $x_{\min}$ ).

# Удельные показатели энерго-эффективности природных и вычислительных процессов:

- Солнце излучает 0.1вт с 1 тонны своей массы или 0,1 мВт/кг, при этом температура поверхности 6000 К, удельная плотность -  $1,4 \text{ г/см}^3$ , площадь поверхности  $6,08 \times 10^{18} \text{ м}^2$ , масс  $1,9891 \cdot 10^{30} \text{ кг}$ ;  $10^{-4} \cdot 1,99 \cdot 10^{30} / 6,08 \cdot 10^{18} = 3,3 \cdot 10^7 \text{ Вт/М}^2 = 3,3 \cdot 10^3 \text{ Вт/см}^2$
- Thermal Design Power — расчетная тепловая мощность микропроцессора  $70 \text{ Вт/см}^2$ .
- Человек излучает 100-500 ватт с 100 кг своей массы –  $1 \text{ Вт/кг} = 0.5 \text{ Вт/м}^2 = 5 \cdot 10^{-5} \text{ Вт/см}^2$



Сравним эти цифры с величиной  $3.2 \cdot 10^6 \text{ Вт/см}^2$ ,



- Новые идеи или даже области в математике приходят из физики. Физические теории дают неожиданные наблюдения и гипотезы, которые сложно сделать «изнутри» чистой математики.
- Совершенно разные области знаний связаны между собой через «объект-данные» - например удельные показатели вычислительной эффективности. Это свойство называется супервентность.
- Супервентность физических процессов и энергетических затрат на вычисления, с помощью которых можно их моделировать, дает возможность рассматривать различные научные теории с общих «информационно-вычислительных» позиций.

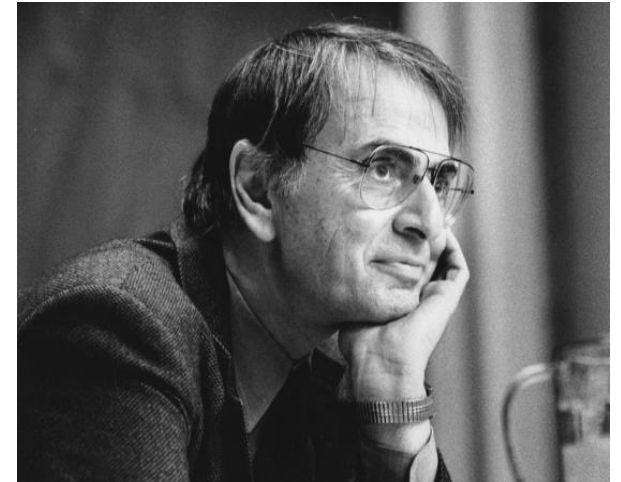
**Вопрос – нужна ли «большо» НАУКЕ алгоритмическая строгость и вычислительная разрешимость ?**

# Алгоритмы обработки «неполной информации»

- В реальной жизни мы **не обладаем полнотой информации** (см. проблему сложности) .
- Требуется умение работать с приблизительными, а не с абсолютно точными данными. Надо исходить из того, что данные могут оказаться недостоверными, но при этом результаты вычислений должны отвечать «физическим законам» или имеющиеся данные должны «породить» ответ – решения пока ... нет.
- Надо признать, что проводя вычисления мы вынуждены сказать, что **не знаем наверняка** можем ли мы доверять полученным результатам.

# Истина вычислений в «нефальсифицируемости» (К. Саган)?

- Что важнее точность или скорость вычислений? Для компьютерных наук необходимо добиваться нефальсифицируемости результатов.
- При этом **нефальсифицируемость** не означает строгую математическую истинность результатов. Часто принципиально не известно ложно или истинно некое математическое утверждение до того момента, пока не завершим вычисления. Как сказал **Карл Саган**: "экстраординарные утверждения требуют экстраординарных доказательств".
- Для доказательства истинно или ложно утверждение надо сравнить его с теоретическими выводами и понять насколько вычисленный результат можно проверить с помощью реальных экспериментов

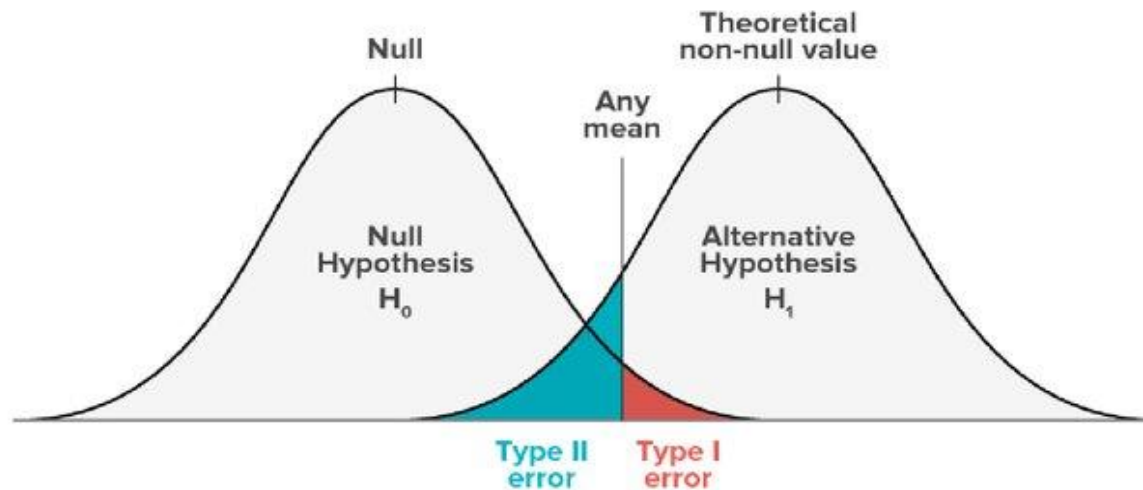


# Компьютерные «базы знаний» vs «базы данных» ?

- Базы знаний – это компьютеры структуры, которые устроены таким образом, чтобы «вычислять» закономерности и делать предсказательные прогнозы, основанные на использовании достоверных данных.
- Суть требований «нефальсифицируемости» состоит в том, чтобы избегать т.н. ошибок второго рода ( есть ошибки первого рода их суть - отвергнута правильная гипотеза (невиновный обвинен – отвергаем верную гипотезу  $H_0$ ) т.е. это ложно-положительная ошибка (false positive - FP); ошибки второго рода (виновный освобожден – принимаем неверную гипотезу  $H_0$ ) это ложно-отрицательные (false negative - FN) ошибки.
- Нефальсифицируемость – это требование совершаешь незначительные по вероятности ошибки первого рода.

## Ошибки первого и второго рода

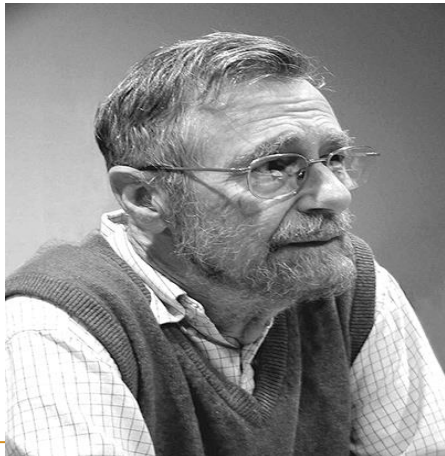
1. Ошибка первого рода ( альфа-ошибка) – мы находим связь там, где ее нет. ( 0 – 5% )
2. Ошибка второго рода ( бетта-ошибка) – мы не находим связи там, где она есть. ( 0 – 20% )



# Алгоритм – это программа , у которой ошибки первого рода минимальны

- Программа – суть последовательность физически реализуемых операций, результат которой минимизирует ошибки первого рода
- Большинство вычислительных алгоритмов имеют двойственный характер: сочетают в себе абстрактную форму - текст, описывающий последовательность машинных операций и некое физическое обличие, которое характеризует «механический» в смысле «машины Тьюринга» процесс обработки цифровых данных.
  - Поэтому надо понимать, что «программист разрабатывает алгоритмы, предназначенные для механического исполнения существующим или мыслимым компьютерным оборудованием»

**Дейкстра, 1974**



## Еще раз о компьютерных науках

- Итак, компьютерная наука **не является абстрактной математической дисциплиной**, независимой от возможностей «физического мира». Чтобы быть использованными на практике, компьютерные программы должны иметь физический «носитель» и «физический механизм исполнения».
- Спецификации физического носителя «компьютерных» вычислений могут выражаться различными способами. В компьютерных науках используются формальные и точные способы описания программ. Языки программирования, например, UML имеет богатую онтологию (онтология - спецификация концептуализации) и большое **разнообразие механизмов исполнения**.
- Используемая онтология языка может включать такие понятия, как компоненты, соединители, интерфейсы и конфигурации, которые определяются той или иной концептуализацией и соответствующей **семантикой**.

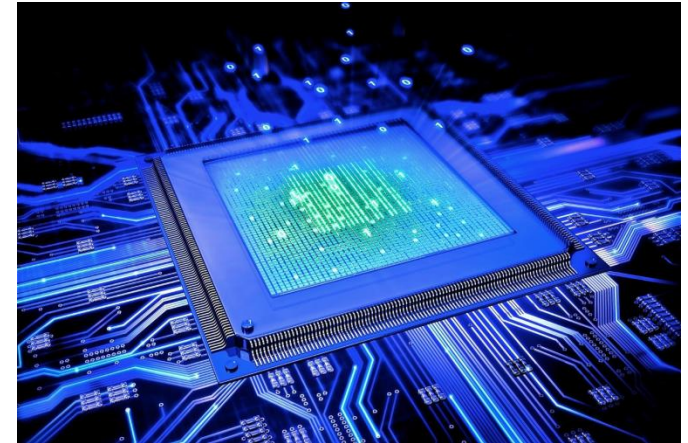
# Что важнее: определение или спецификация

- Взятое само по себе **определение** не обязательно должно быть спецификацией
- Вопрос, когда определение выступает в качестве спецификации зависит от того, возможно ли на основе определения реализовать акт управления устройством или системой. Можно сказать, что спецификации является нормативным определением.
- Только определение, функционирующее как **спецификация**, позволяет оценить состояние объекта с точки зрения выполнения им тех или иных действий. На любом уровне абстракции логическая роль спецификации всегда одна и та же - определить **критерий** правильности и неправильности работы устройства.



# Алгоритмы и физические процессы

Фундаментальный вопрос - всякий ли физический процесс является результатом некоторого вычислительного процесса? Например:



В Природе реализуются весьма «сложная» модель эволюционных вычислений, в которой используются самореферентные процессы, порождающие **не коды новых числа, а новые физические состояния объектов**, которые управляют процессом вычислений.

Природный «компьютер» как бы вычисляет сам себя: управляет процессами вычислений с помощью «**программ**», которые – суть его «**аппаратное обеспечение**», которое формируется в процессе этих вычислений.

## «натуральные» vs компьютерные вычисления

Итак, результат вычислений м.б. не только символы, кодирующие числа и обладающие заранее заданной синтаксической структурой, но и новые материальные объекты и процессы, которые порождаются в процессе вычислений.

- Вопрос – можно ли описать окружающий нас мир в терминах процессов вычислений и обработки информации

в основе эволюции материи лежит «вычислительное» пространство ,



**Конрад Цузе:**

«Природа» постоянно вычисляет свое будущее, основываясь на текущем состоянии материи. Все природные явления – продукт вычислений, реализуемых с помощью **«прото» вычислительных устройств**, обрабатывающих прото-информацию или информационный код материи.

# Выводы

- Компьютерные науки не является абстрактной математической дисциплиной, независимой от возможностей «физического мира».
- Чтобы быть использованными на практике, компьютерные программы должны иметь физический «носитель» и «физический механизм исполнения».
- Удельные показатели энерго-эффективности природных и вычислительных процессов:
  - Предельно достижимое значение =  $3.2 * 10^6$  Вт/см<sup>2</sup>
  - Солнце:  $3,3 * 10^3$  Вт/см<sup>2</sup>
  - Thermal Design Power современных микропроцессоров :  $70$  Вт/см<sup>2</sup>.
  - Человек =  $5 * 10^{-5}$  Вт/см<sup>2</sup>
- (задание – проверить эти значения с помощью вычислений)