Theoretical computer science: a subjective overview

Mark Braverman

Princeton University

MAY 15, 2019

The study of algorithms

Algorithm [Merriam-Webster]:

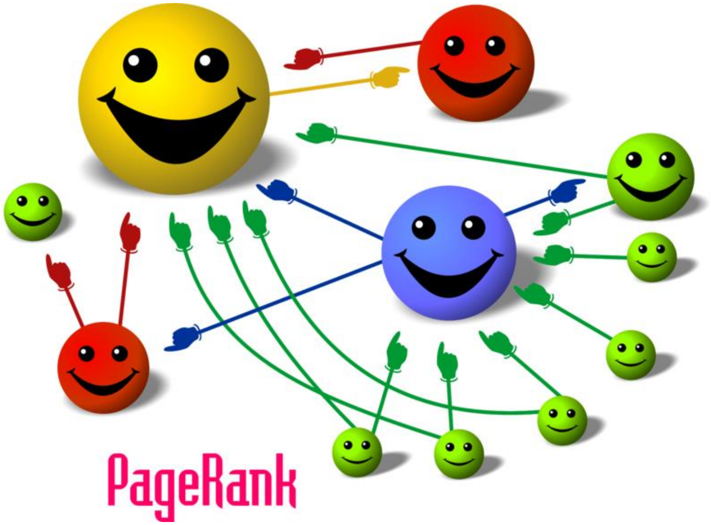
- “a procedure for solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that frequently involves repetition of an operation”

Then:

- *broadly*: **“a step-by-step procedure for solving a problem or accomplishing some end”**

Traditional view of algorithms: procedures for specific math-y tasks

- Key applications:
  - Applied mathematics
  - Statistics, data organization, search
  - Communication and signal processing

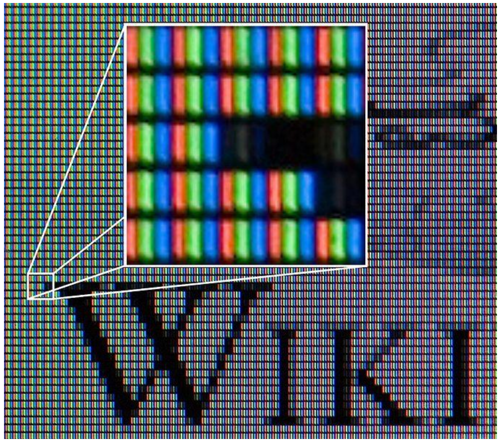


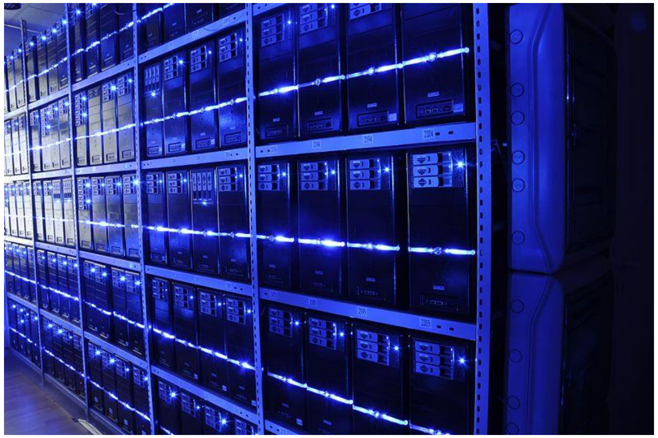
Main theme of Theoretical Computer Science:

Algorithms as an object of study

Algorithms in Engineering: Computation as a resource

- More processes and devices involve computation at various scales.
- Reducing the energy cost of computation becomes increasingly important:
  - Micro: power is a major constraint as computation becomes an important part of more devices.
  - Macro: datacenters account for ~1% of global electricity consumption.





Applications of algorithms: properties beyond performance

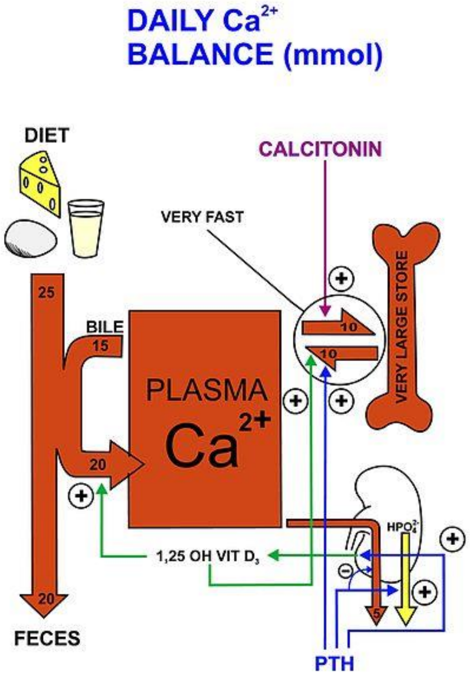
- Early examples: cryptographic protocols (difficulty is an essential feature!)
  - Digital signature scheme: easy to sign, easy to check, **hard to forge.**

Abraham Lincoln

- Algorithmic game theory:
  - Want pricing algorithm to **induce** desirable behavior from market participants.
- Algorithmic fairness and privacy:
  - Want a classification algorithm to satisfy **additional ethical or legal constraints**.

Algorithms in science: from tools to content

- Many natural processes are best understood not in terms of grand laws, but via the simple local processes that produce them.
  - Evolution in terms of natural selection.
  - Biological processes in terms of individual components and pathways.
  - Group population behavior in terms of individual behaviors.



Algorithms as an object of study

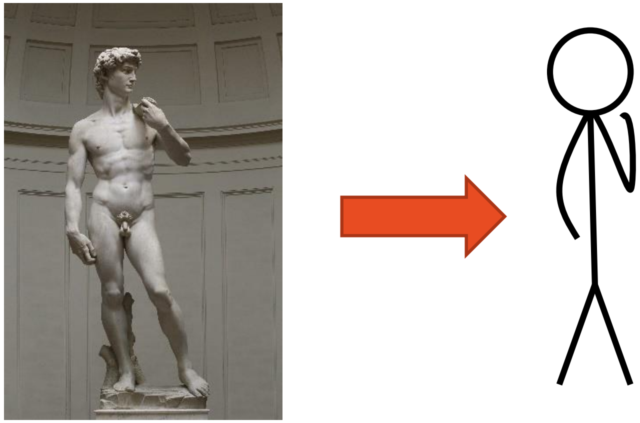
Big questions about algorithms

Big questions about algorithms

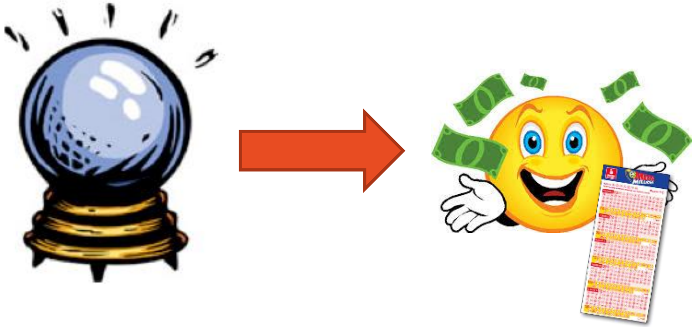
- What natural processes can be efficiently algorithmically predicted or simulated?
  - Special case: the limits of quantum computing.
- Provable limits on computation of problems we encounter:
  - A very special case: **P vs NP**.
  - Bounds known only in very abstract and very concrete regimes.
- Conservation laws governing computing and its properties?
  - Is there a “unit of computing” – similar to a unit of energy or a unit of information?

Theoretical computer science in one slide: Abstraction and reduction

- **Abstraction:** map the computational problem to a mathematical problem about a model of computation.
  - E.g. in Turing Machine in [Turing'36].



- **Reduction:** statements of the form "if we can solve problem **A**, then we can also solve problem **B**"
  - Also useful contrapositively: "If **B** is difficult, then so is **A**"



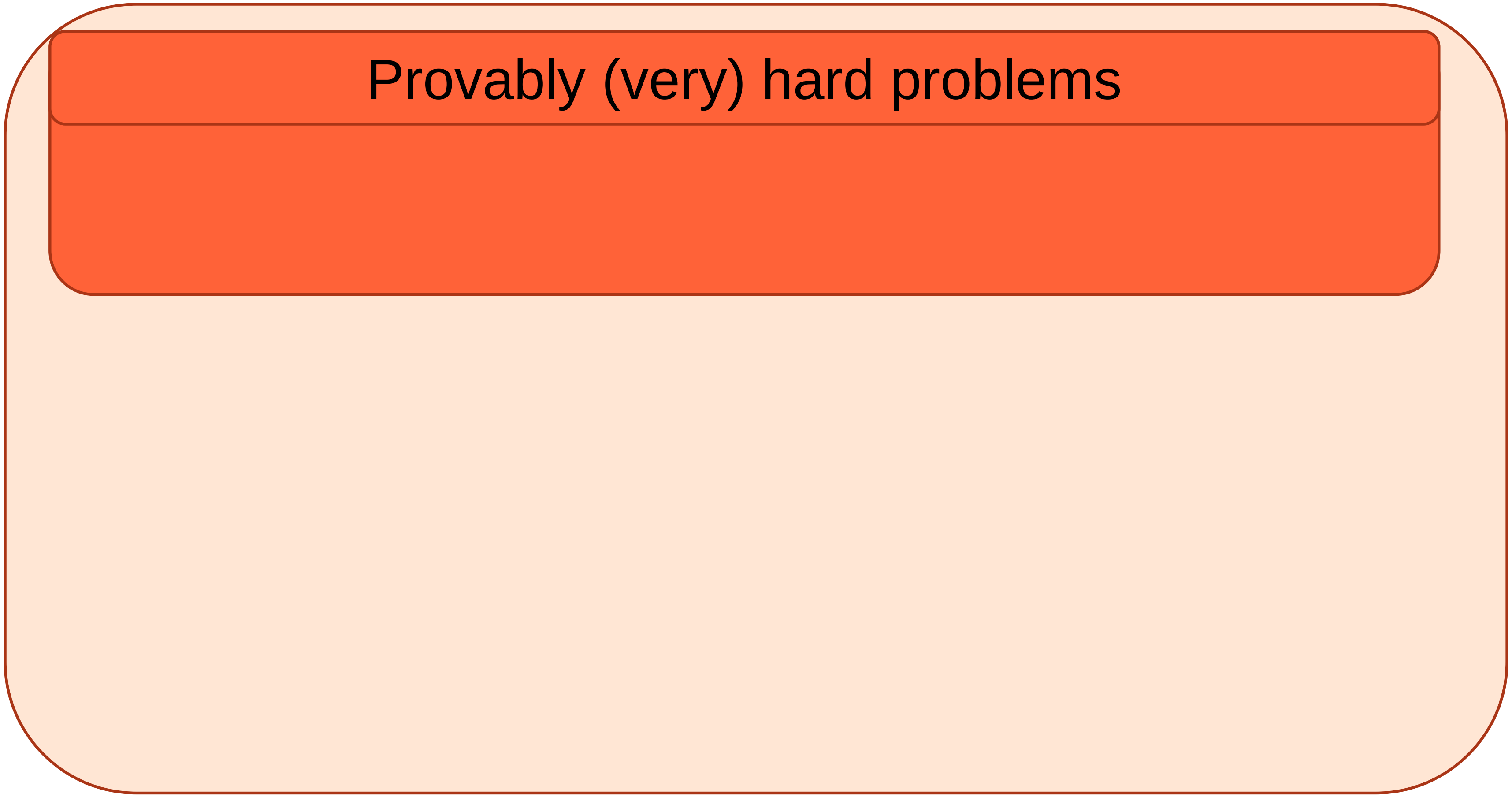
Mapping the limits of computation

Provably very difficult problems: “Will this computer program eventually halt?”; “Is this mathematical statement a theorem in this axiom system?”

[Turing’36]

[Gödel’31]

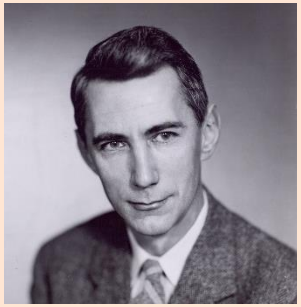
Mapping the limits of computation



Provably (very) hard problems

Mapping the limits of computation

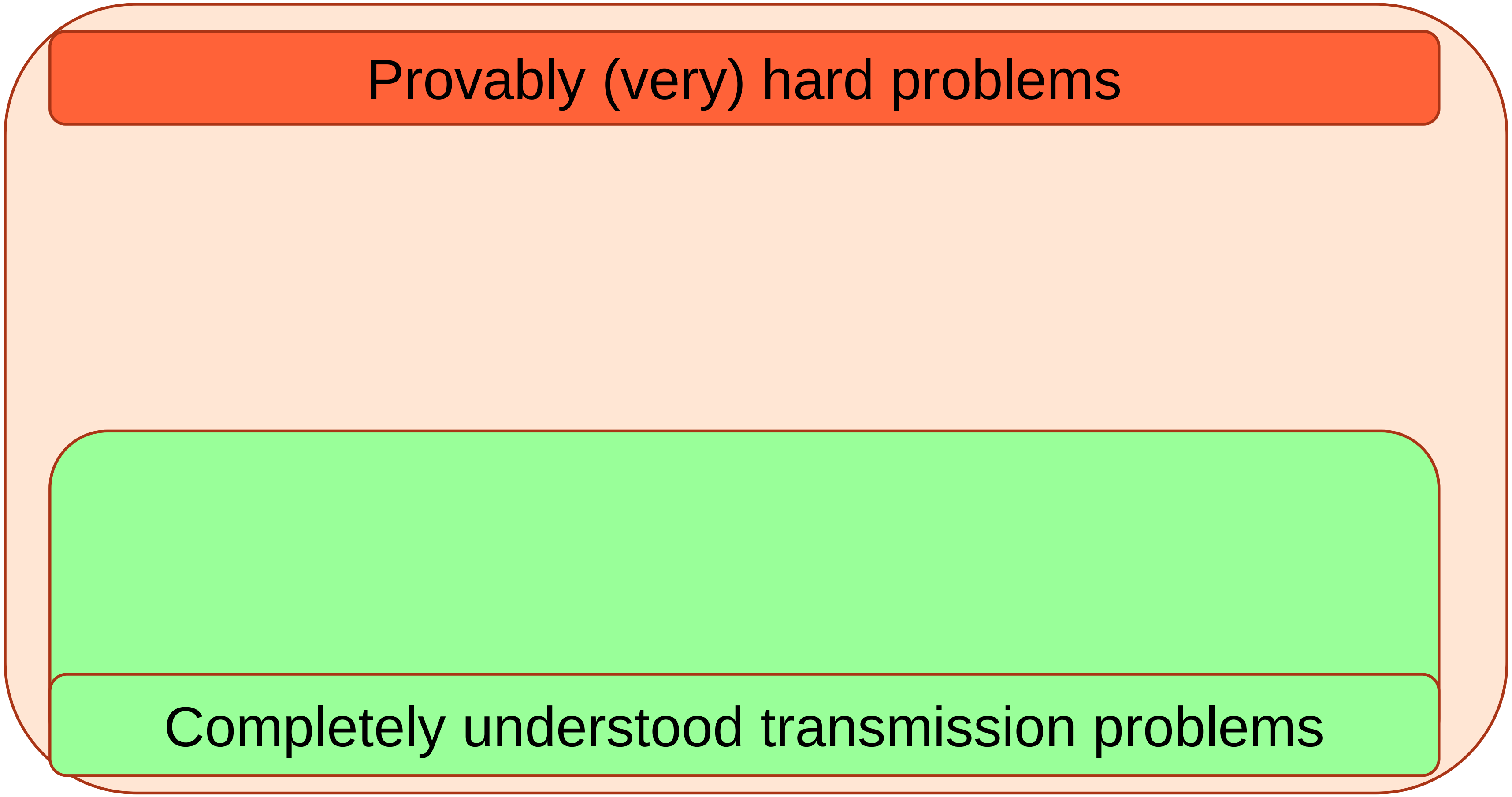
Provably (very) hard problems



[Shannon’48]

Precisely answer concrete data-transmission problems: “Transmit data over a given communication channel”. Key insight: **bit is a unit of information**. Information is a conserved quantity.

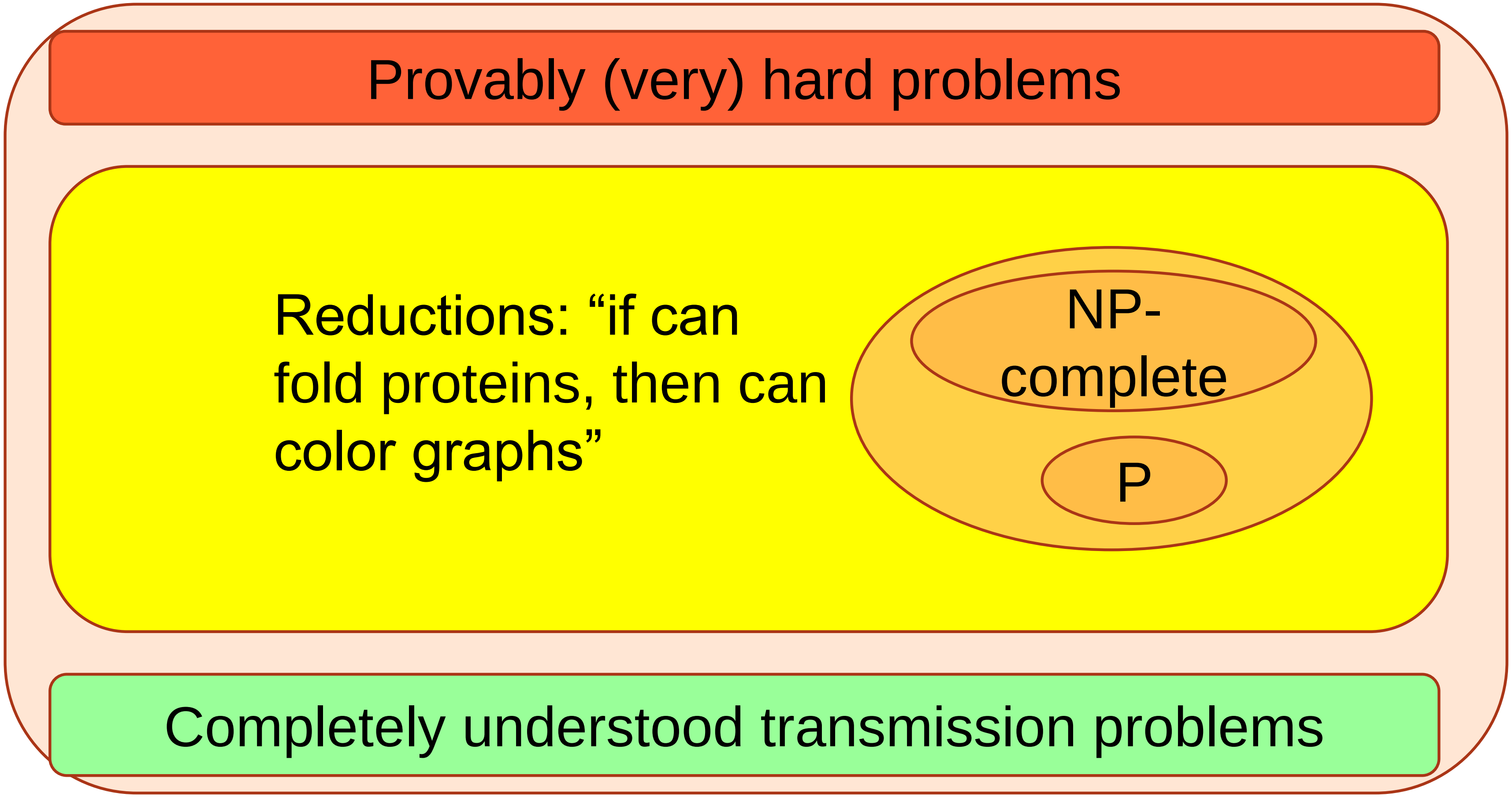
Mapping the limits of computation



Provably (very) hard problems

Completely understood transmission problems

Mapping the limits of computation



Provably (very) hard problems

Reductions: “if can fold proteins, then can color graphs”

NP-complete

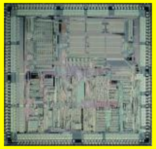
P

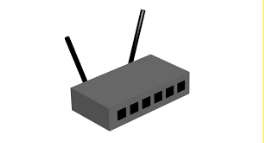
Completely understood transmission problems

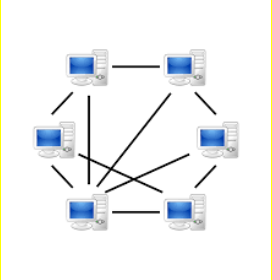
Mapping the limits of computation

Provably (very) hard problems

Restricted models: distributed models; interactive communication; models where data $\gg$ memory

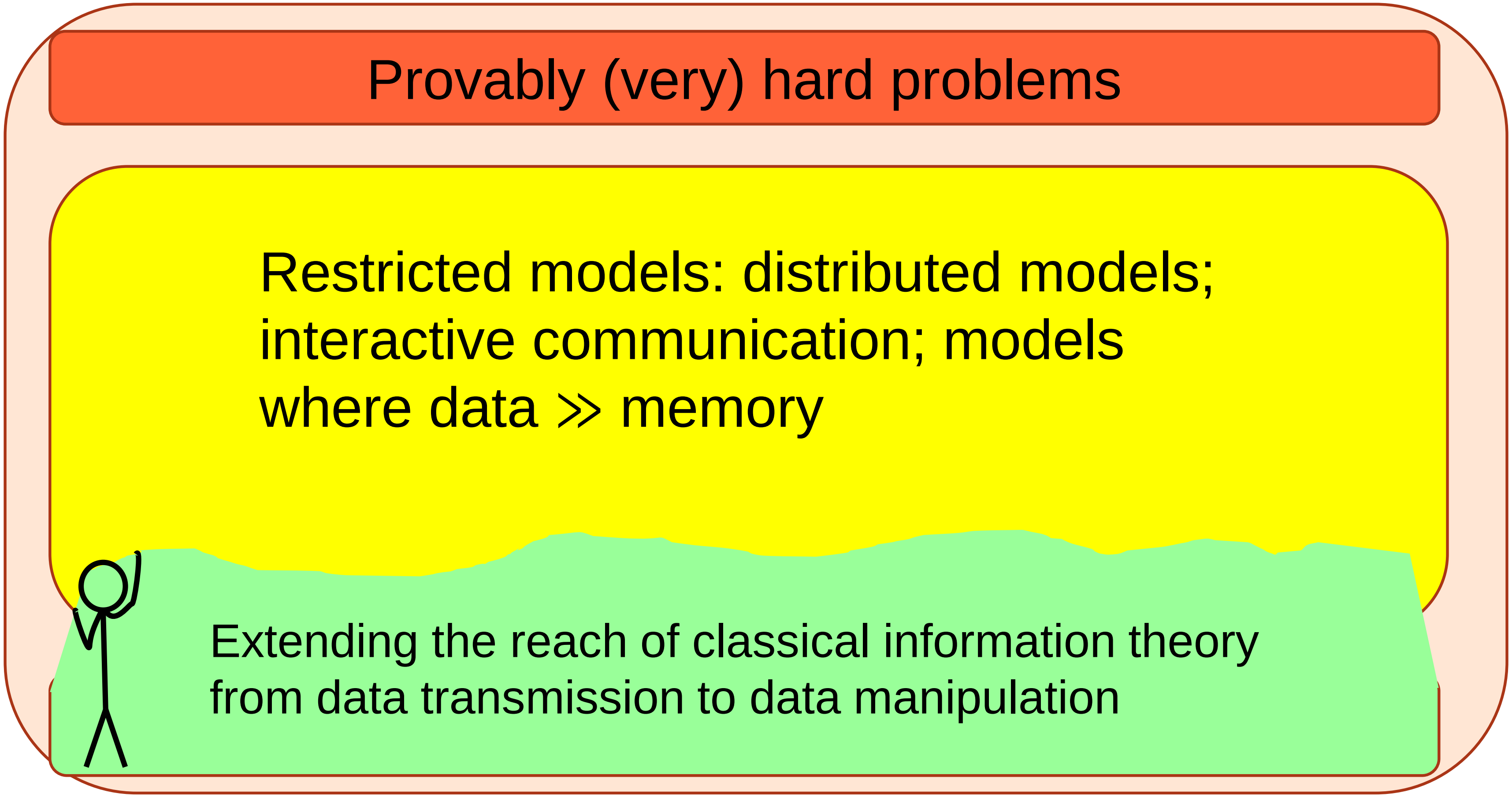






Completely understood transmission problems

My agenda: Information Complexity



Provably (very) hard problems

Restricted models: distributed models; interactive communication; models where data ≫ memory

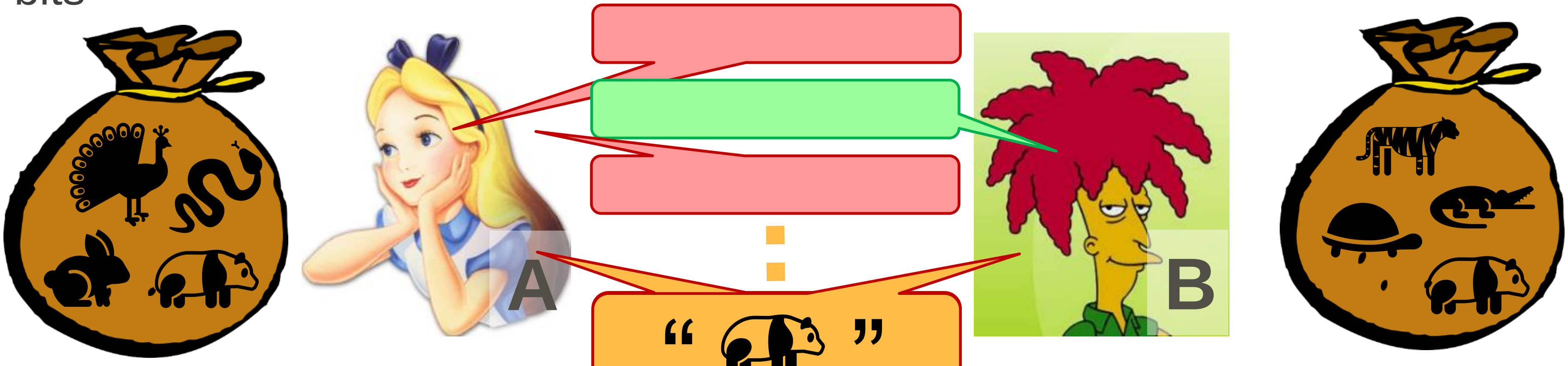
Extending the reach of classical information theory from data transmission to data manipulation

Finding a common element

- Alice and Bob each has n elements. Goal: find a common element if one exists (with high probability)

- A non-trivial theorem from late 1980s [Kalyanasundaram-Schnitger, Razborov]: need linear (in n) number of bits of communication

- With information complexity [B Garg Pankratov Weinstein'13]: $\approx \left(\frac{2}{\ln(2)}\right) n \pm o(n)$ bits

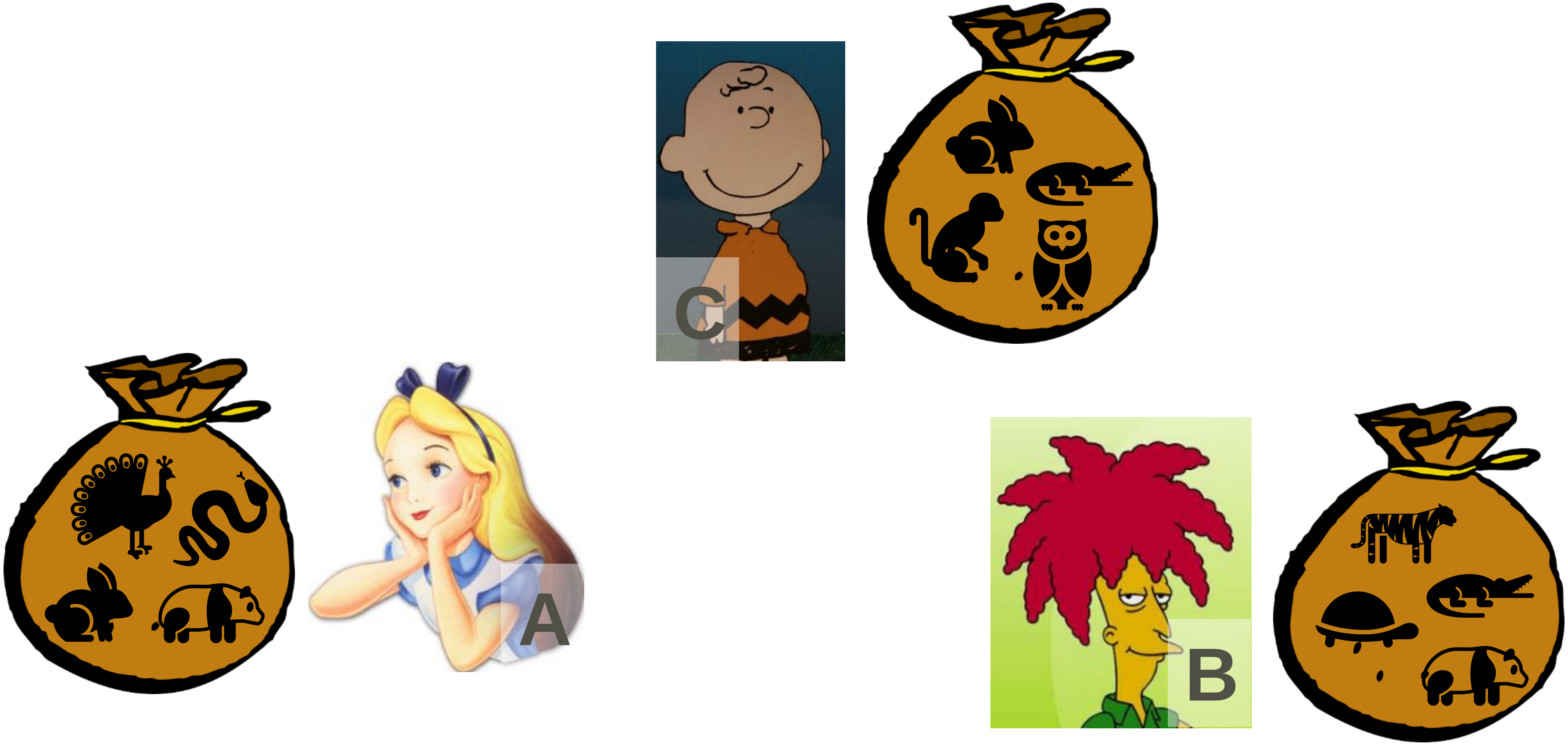


n animals

n animals

Finding a common element

- Challenge: same problem with 3 players.



Three problems

Power Washington, DC with a single diesel generator:

- Not going to happen, because…
- Conservation of energy.

Three problems

Back up world’s Facebook photos on my thumb drive:

- Not going to happen, because…
- Conservation of information.

Three problems

Do NSA cryptoanalysis on a smartphone:

- Not going to happen, because…
- Conservation of ??

Thank You!