TechFest
the & in R&D

Microsoft Research

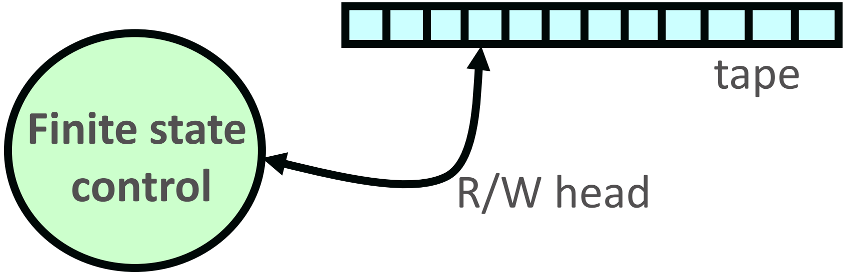
Theories of Communication & Computing

Madhu Sudan (MSR New England)

Microsoft

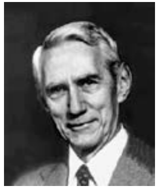
Theory of Computing

Turing (1936) → von Neumann (50s?)



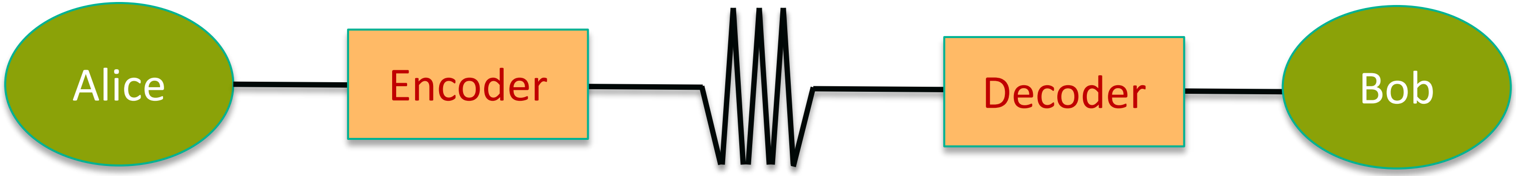
- Clean model of "universal" computer.
- One computer, that can be programmed
  (to do any computational task!)
- Separation of computing elements
  (ALU/CPU/Finite state control) from memory (RAM, tape)
- Needs reliable memory & reliable communication to/from memory.

Theory of Communication



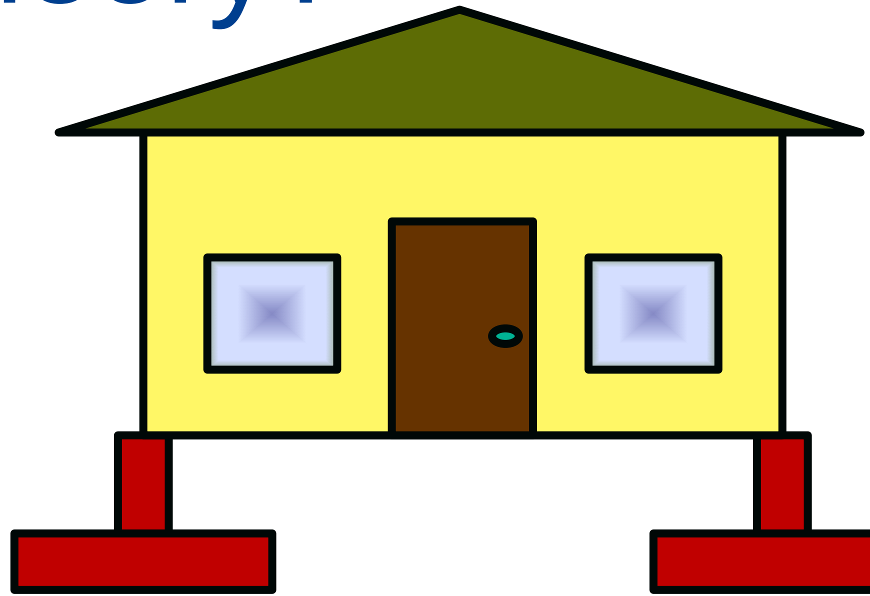
Shannon (1948)

- Clean architecture for reliable communication.



- Needs reliable encoder + decoder (two reliable computers).

Role of theory?



Application

Theory layer

Communication vs. Computing

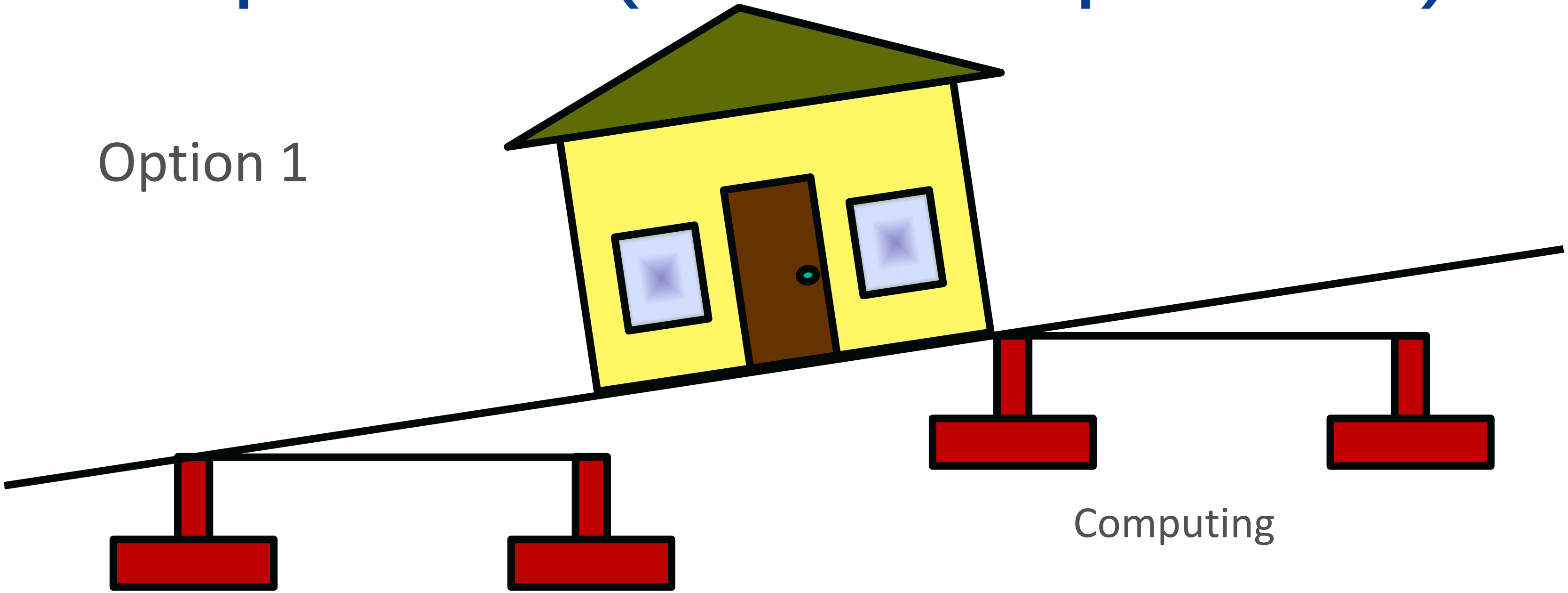
- Deeply intertwined: Each needs the other!
- Theories & (till 1990s) technologies well-separated.
- Today technologies are coming together.
- Leads to a clash of the theories!

Clash of the Theories?

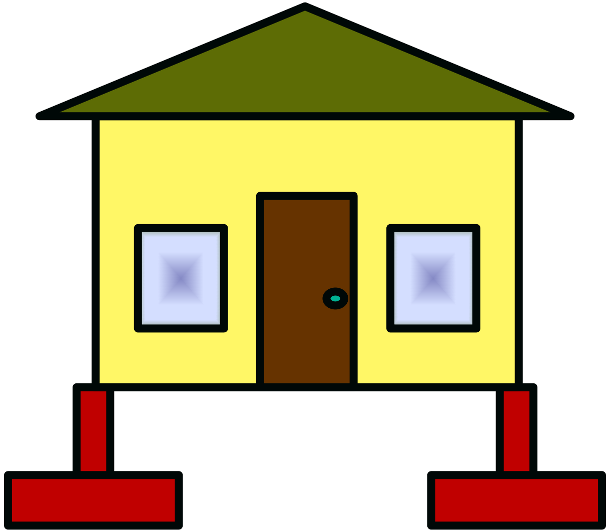
- Computing principle: Give user a programming language/operating system and let them modify device freely.
- Communication principle: Design encoder/decoder jointly. Devices at both endpoints should be designed jointly.
    - Do not let user program/alter their devices!

Consequences (without equations)

Option 1

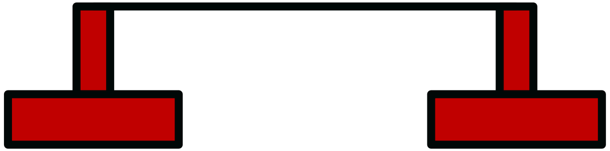


Consequences (without equations)



Communication

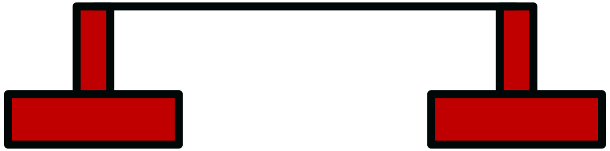
Option 2



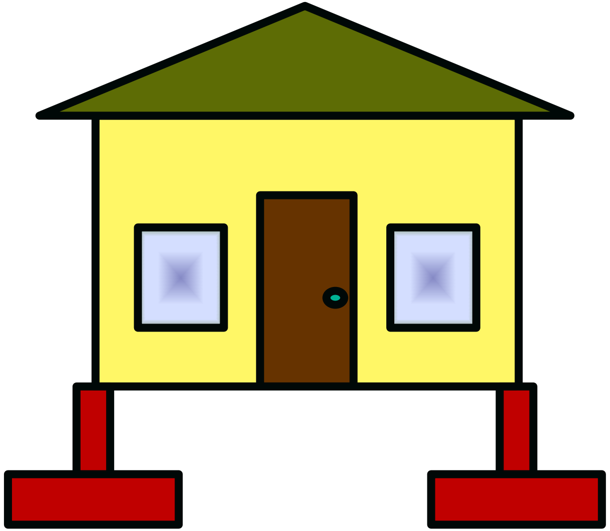
Computing

Consequences (without equations)

Option 3



Communication

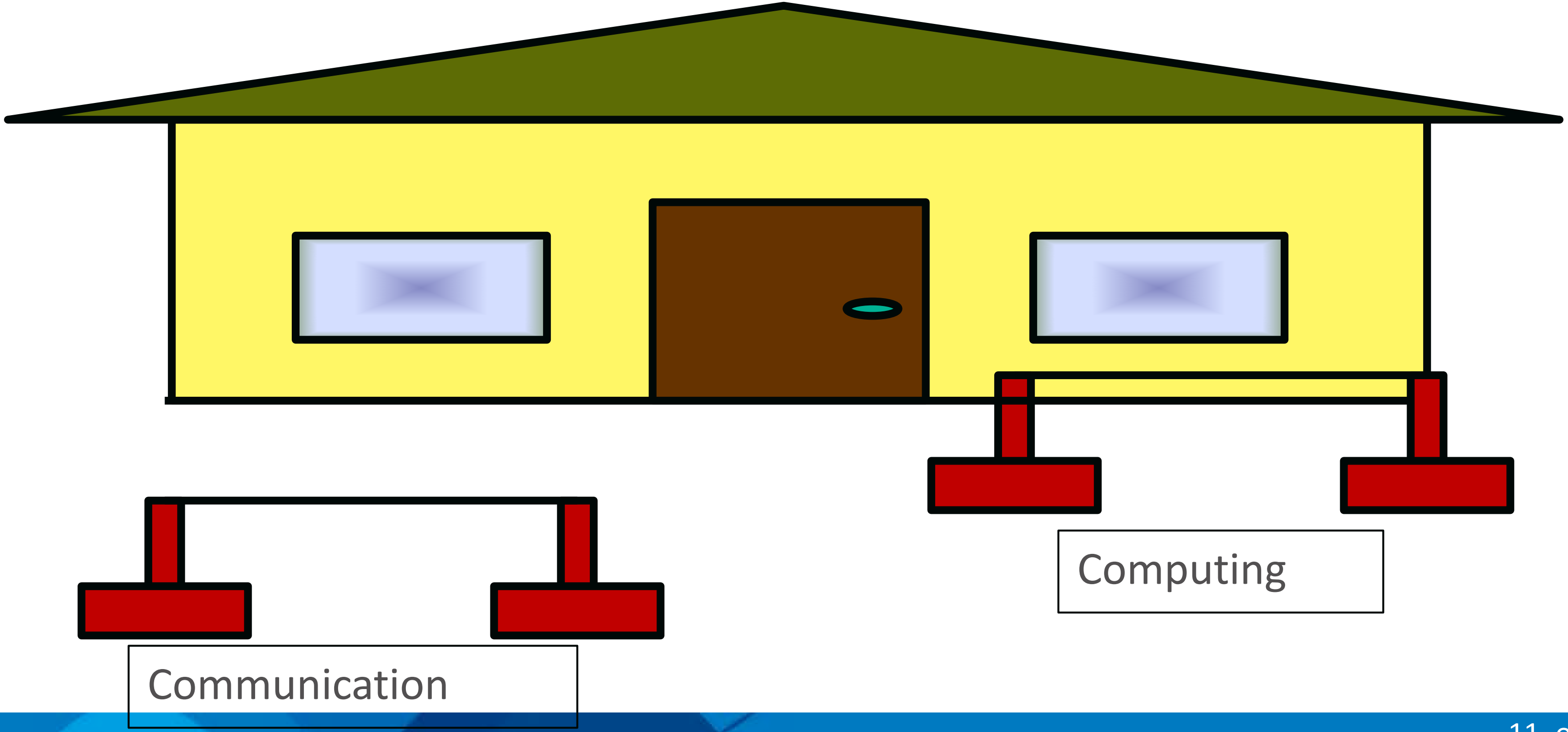


Computing

Consequences in words

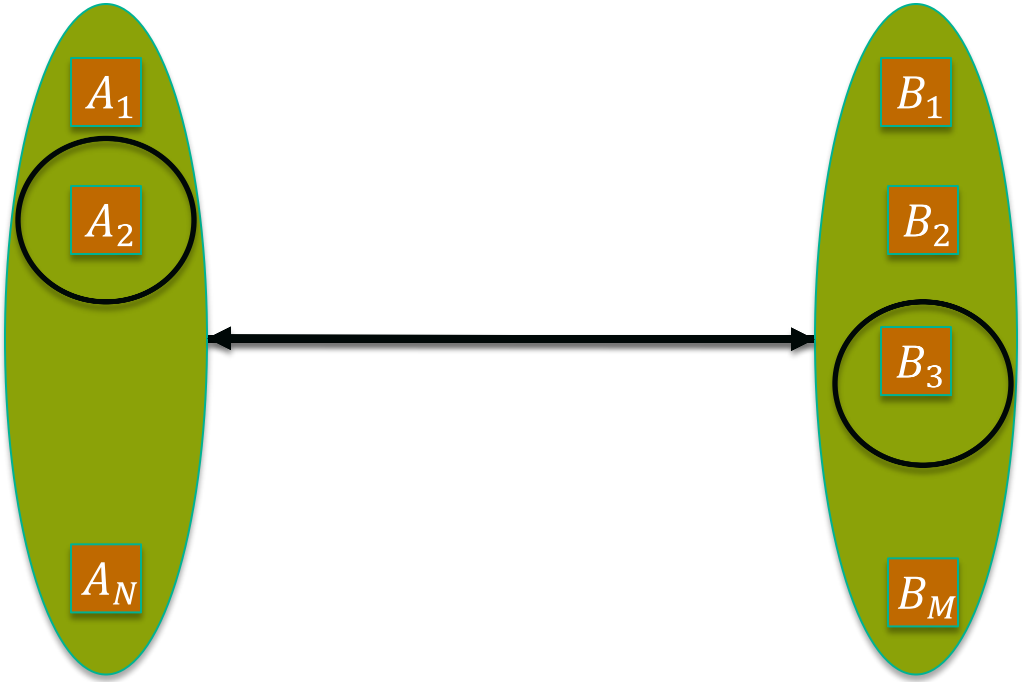
- Communicating computers are highly unstable and vulnerable.
    - They spend lots of time updating software.
    - Many are not programmable.
- Can computers communicate the way humans do?
- Long term issues:
    - What are the long term prospects of our data?
    - How will we preserve their meaning, when interpretation is changing?

A new theory?



New communication model

Classical communication
Uncertainty (about endpoints)



Aspects to study

- Understanding human-human communication:
    - why is natural language so different?
    - E.g., why is the dictionary so redundant and so ambiguous?
    - Why are (grammatical) rules made to be broken?

- Semantic Communication.
    - How can computers detect when bits are being misunderstood?
    - How can they correct errors?
    - What does “understanding” mean anyway?

Human-Human Communication

Role of dictionary? [Juba,Kalai,Khanna,S.]

$$
\begin{aligned}
M_1 &= w_{11}, w_{12}, \ldots \\
M_2 &= w_{21}, w_{22}, \ldots \\
M_3 &= w_{31}, w_{32}, \ldots \\
M_4 &= w_{41}, w_{42}, \ldots \\
&\ldots
\end{aligned}
$$

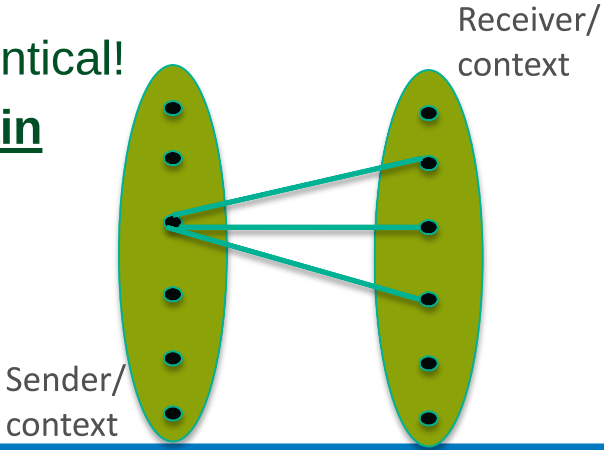
- Dictionary: gives list of words representing a message
  - words appear against multiple messages
  - multiple words per message.
- How to decide which word to use? Context!
  - Encoding: Given message, use shortest unambiguous word in current context.
  - Decoding: Given word, use most likely message in current context among messages whose list includes uttered word.
- Context = ???. Probability distribution on messages!
  - $P_i = \text{Prob}\,[\text{message} = M_i]$

Human Communication - 2

$$
\begin{aligned}
M_1 &= w_{11}, w_{12}, \ldots \\
M_2 &= w_{21}, w_{22}, \ldots \\
M_3 &= w_{31}, w_{32}, \ldots \\
M_4 &= w_{41}, w_{42}, \ldots \\
&\ldots
\end{aligned}
$$

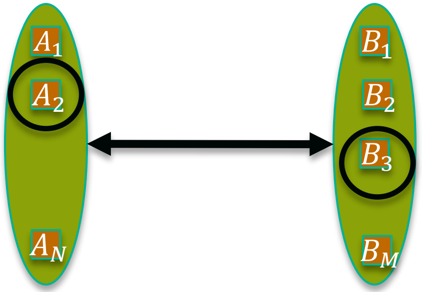
Role of dictionary? [JKKS]

- Good (Ideal?) dictionary
  - Should compress messages to entropy of context: $H(P = \langle P_1, \ldots, P_N \rangle)$.
- Even better dictionary?
  - Should not assume context of sender/receiver identical!
  - Compression should work even if sender **uncertain** about receiver (or receivers' context).



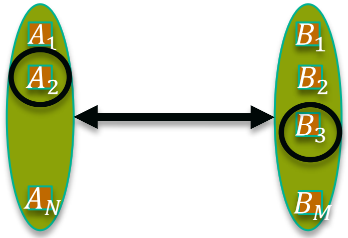
Theorem [JKKS]: If dictionary is "random" then compression achieves message length $H(P) + \Delta$, if sender and receiver distributions are " Δ-close".

Meaning of Bits



- Meaning?
  - Bits ↔ Instructions (Algorithm/Computer Program)
  - Whither **uncertainty**?
    - Receiver may not know programming language of sender.
    - Uncertainty arises from diversity! Else intelligent Alice can adapt!
- Communicate meaning?
  - Can we send language first? Or a compiler?
    - Compile to which language?
    - Need to have common ground – some common language first?
  - But humans don't seem to need this?

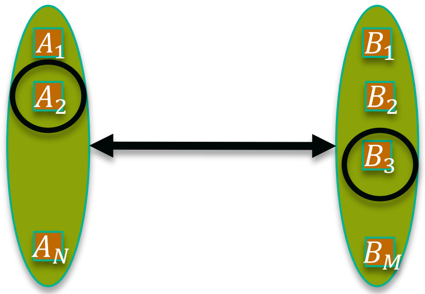
Semantic Communication



[Goldreich+Juba+S., Juba+S.]

- Theorem 1: In sufficient diversity, can not communicate meaning!
  - Main issue: generically, can not detect misunderstanding.
  - If you can't detect misunderstanding, shouldn't be communicating.
- Why communicate at all?
  - Computer scientist: To get *useful* data.
  - Systems scientist: To exert remote control.
  - Economist: To gain strategic advantage.
  - Common theme: Communication must have a goal.
- Theorem 2: If we can sense progress towards the goal, then can learn meaning.
  - Main insight: Absence of progress towards goal signals misunderstanding.
- Warning: Learning of meaning can be "exponentially slow"

Conclusions



- For Practice:
  - Communication and computation can be bridged differently.
  - Computers can communicate, and maintain reliability, but we have to be explicit about goals of communication, and be able to sense progress in achieving such goal.
- For Theory:
  - Robust communication leads to new class of problems.
  - Rich opportunity to bring together Math, CS, Information Theory, Economics, while learning from linguists, philosophers and communication (media) scholars.

Thank You