Founding fathers of computing theory → Alan Turing, Alonzo Church, Kurt Gödel

Wanted to have philosophical clarification ~~but~~ they had a lot of technological payoffs.

Problems we wanted to compute → Computable but not so efficiently.

Rise of computational complexity theory ←

Complexity theory does not solve these questions, but only contributes useful perspectives.

Philosophy ∩ Theoretical computer science

Semantics of prog. languages → philosophy of language

distributed systems theory → Reasoning about knowledge

Kolmogorov complexity → length of shortest computer program for some functionality

Complexity closer to actual sciences than computability theory since all relevant problems are computable.

→ More relevant than computability theory for philosophy

Relevance of polynomial time

→ Entscheidungsproblem ["decision problem"] → By Hilbert

- An algorithm to verify if the statement is universally true.
- Incompleteness theorem → destroyed it for complete systems
- Church Turing thesis → Not doable for even incomplete systems

[Can't group into "yes", "no" or "undecidable"]

[Gödel letter to Von Neumann]

→ A way to check : Brute Force through all combinations

- Suppose the proof will be n bits ⇒ check for all 2^n combinations

↓

could have been done if it was poly : P

Even if "decision problem" was unsolvable it would reduce the requirement of mental strength of mathematician to find the proof

for large enough n → if machine says no then stop worrying about it

If P=NP ⇒ We can find algorithms to solve the rest of the millenium problems.

Finding proof of some statement being true in a formal system $\in$ NP (-complete too)

P=NP ⇒ reasonable length proof can be found in reasonable time

Evolution → Optimization problem where steps grow polynomially wrt to number of variables.

Godel knew given exponential time, life can be formed from pure randomness, but is it polynomial? it had to considering the geological time.

Theory of evolvability ? → Speed of evolution

What does known mean ? Computable ? Can be listed ? Exists ?

[Largest prime is 30k pages long, classifies as known?]

Turing Test

- If something passes the Turing Test does it get the "conscious" title?
- Can you even write such a program?

Penrose argument ⇒ Incompleteness theorem means machines can not check the consistency of their own rules, but humans somehow can. So machines can not simulate Human brain. (Why?)

Look up table argument ⇒ → No point of generalizing due to human limitation

- A interacts with B for finite time, hence we can construct every single possible output required. Why? because it is finite.

- Although its not computable with the space in universe, its still finite.

Searle's Chinese room argument ?

lookup table can only be conscious if its polynomial in time.

Can humans solve NP complete problems in polynomial time ?

Based on factoring problem → NO, Computers are infact more efficient at it.

Reasons why computers can't simulate human brains

- Creativity → Forget about creating, even recognizing good art is not polynomial [if it was then making it would be NP].

- Humans are much better at solving special cases of NP-complete problems, by proving theorems or developing algorithms specific to that case.

Proof complexity & pigeon hole principle ?

Account of knowledge → Concern in philosophy

Logically omniscient → If they know certain facts then they know all its implications.

Is knowledge just a list of facts we know? There are problems when formalizing this.

→ Boundary between facts we know when asked v/s we know without being asked.

→ Boundary between facts we have to think v/s the ones we don't have to.

→ Major problem: Answering 1 out of 2 questions even though both mean the same.
[multiplication v/s factorization]

- Kids knowing commutativity of addition → Ask directly & they will be confused. But ask them about height increase when a stack is shuffled they will implicitly use the commutativity of addition.

Preventing computational omniscience → define what are computable functions.

Church-Kleene → listed axioms for computable functions rather than having a machine defined (Turing).

$f(x), g(x)$ → computable → $f(g(x))$ computable

FP → function polynomial time → polynomial time on deterministic TM.

Cobham's axioms → list of computable functions

- Function which are non zero finitely often
- Pairing $f(x), g(x)$ computable then $f(x) * g(x)$ also → pairing function for $\mathbb{N}$
- Composition → $f(g(x))$
- Grab bag → $x+y$ $x \cdot y$

$\lfloor \log_2 x \rfloor + 1$ $x \,\&\, 2<<i$ → i^{th} bit $2^{|x|^2}$ → map n to n^2 bits & soon

$\Pi^1_1 \langle x, y \rangle = x/y$ $x \wedge 2<<i$ → flip i^{th} result

- Bounded recursion

$$g(\langle x, k \rangle) = \begin{cases} f(g(\langle x, \lfloor k/2 \rfloor \rangle)) & k > 1 \\ x & k = 1 \end{cases}$$

→ run for $\log_2 k$ steps

FP is the smallest class that satisfies these axioms.

There is no closure in Lobhomis axioms since functions $f: \mathbb{N} \to \mathbb{N}$ are uncountably infinite, so you can't include all of them!

Breaking the old problem → Answering multiplication & not factoring

- It is not about knowing a fact, here its about how to solve, we dont know a fast factoring algo but we know for multiplication.

Quantum computing → Implications on philosophy?

- Same as quantum mechanics itself → many world interpretation.

large overhead due to Entanglement → n entanglements → 2^n vector.

1000 spin states → $2^{1000} >> 10^{80}$ → number of atoms in universe

- Just to keep track of 1000 spins → insane computation.

Classical with high accuracy can account for equal amount of info.
Quantum limits the continuity, Planck time & length.

Bell's inequality when broken had a lot of philosophical implications. in time quantum computing will also end up having it.

Breaking quantum mechanics in an attempt to construct a quantum computer >> Actually constructing 1

Many world interpretation

Deutsch → Quantum mech implies existance of parallel universes

- His argument is amount of resources required to factor a large number Exceed even the number of atoms in the universe.

→ Still unproven that there is no fast classical algorithm.

If there Exists such algorithm → brings local variables into picture & Bells inequality can no longer prove its absence

Predict QM results classically in polynomial time.

- Why have parallel universes rather than 1 quantum universe?

Fiction is less stronger than the actual reality

Some counters against parallel universes

Holevo's theorem → sets an upper bound on how much information can be stored in qubits. So that means qubits have some capacity as normal bits.

Parallel computation → Shor's algorithm doesnot just check all possibilities parallely. If it could then we could extend it to solve other NP problems in BQP.

Computational notion of proof

- A computational process that terminates in a certain way iff true.

Zero knowledge proof (& multiplayer games connection)

Proving that 2 Graphs G & H are not isomorphic to eachother

- Give any instance of G or H, prover will give you what its isomorphic to.

→ Features

- Proof is probabilistic at the end.

- Proof is interactive ⇒ But much faster than traditional proofs.
- limited to lack of prover's ability that they can't directly know what the other person is going to ask for.

→ Exactly like the Turing Test!

- It's 0 knowledge part → The prover just confirms what the asker asked. The asker learns nothing new.
- The asker can not even replicate the proof.

GMW protocol → Every conventional proof has a 0knowledge proof.

↳ Internet crypto implications

Complexity, space & time

- Space reusable but not time ⇒ Making space superior

What if you reuse time? Closed timelike curves (& NP-hardness)

→ Results in grandfather paradox

Deutsch's solution ⇒ • CTC admits a fixed point

S → mapping of quantum states after going through CTC 1x

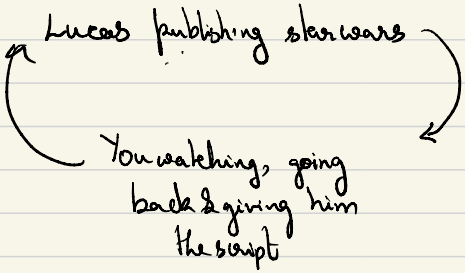
There exists $S(P) = P$

- The probability that you are born = 1/2, You go & kill your grandfather. giving you being born a 1/2 probability.

The Evolutionary principle

- Finding this fixed point is an astronomically hard computational problem.

- Consider this situation →



- Breaks the Evolutionary principle:

Knowledge requires causal process to bring it into existence

↳ Complexity theory analogue

↓

There is no physical mean of solving NP complete problem in Poly time

(obeying laws of physics)

Proving it

If we could create CTC & find the fixed point then NP-complete problems could be solved in polynomial resources.

$f: \{0, \ldots 2^n - 1\} \longrightarrow \{0, 1\} \longrightarrow$ need to find a 1

↓ binary string of n bits into number

↳ yes

↳ no

- Just need a single evaluation.

If $f(x) = 1$ output x ⟶ fixed point.
else output $(x+1) \% 2^n$

Complexity of finding Equilibria ⟶ PPAD class
↳ Not NP-complete since it always exists

Problem of induction in the paper?

Zero knowledge proofs & 2 player games

Simple rules giving rise to complex problems (Game of life)