Turing Machine Algorithms

... 

Wait... isn’t Algorithms next year?!!?
Intuitively speaking, understanding of algorithms has existed for thousands of years.

Formally speaking, algorithms were not defined until the 20th century.

Without a formal definition, it is almost impossible to prove that an algorithm can’t be created.
Historical Context

David Hilbert
Address at the International Congress of Mathematicians in Paris, 1900

Identified 23 mathematical problems as a challenge for new century.

10th problem: Create an algorithm to determine if a polynomial had an integral root.

Assumption was that an algorithm existed, it just needed to be found.

We now know that this problem is not algorithmically solvable.
Historical Context

Alonzo Church
Alan Turing
Church-Turing Thesis

1936, Church published a notation called \( \lambda \)-calculus to define algorithms.

1936, Turing published the specifications for an abstract “machine” to define algorithms.

1952, Equivalence of models shown by Kleene

1970, Proof published that no algorithm exists for testing if a polynomial has integral roots.
How to describe a Turing Machine

For fun and profit.

Formal Description
- Explicitly state everything
- Most detailed
- Avoid at all costs!!!

Implementation Description
- English prose describing movement of head and storage of data on tape.
- No state details

High-Level Description
- English prose describes an algorithm
- No implementation details (no head or tape details)
- Use this unless instructed otherwise
Formal Notation for Turing Machines

Input is always a **string**.

If input is an object, it must be represented as a string.
- Polynomials, graphs, grammars, automata, etc.
- Input can be combinations of different types of objects.

An object $O$ encoded as a string is $\langle O \rangle$.

For several objects $O_1, O_2, \ldots, O_k$, they are encoded as $\langle O_1, O_2, \ldots, O_k \rangle$.

Algorithm is given with **text**, indented as needed for block structure.
Example: \( A = \{ \langle G \rangle \mid G \text{ is a connected undirected graph} \} \)

High-level description:

\[ M = \text{“On input } \langle G \rangle, \text{ the encoding of a graph } G: } \]

1. Select the first node of \( G \) and mark it.
2. Repeat the following stage until no new nodes are marked:
   3. For each node in \( G \), mark it if it is attached by an edge to a node that is already marked.
4. Scan all nodes of \( G \) to determine whether they all are marked. If they are, accept; otherwise, reject.”

\[ \langle G \rangle = (1, 2, 3, 4) ((1, 2), (2, 3), (3, 1), (1, 4)) \]
History of Undecidability - Segue Into Chapter 4

Videos from Computerphile on YouTube