

# Theory of Computation

## Church's Thesis

Mayank Yadav

April 10, 2025

## 1 Introduction

Church's thesis is one of the main hypotheses that laid the foundation of computation. It is also known as Turing's thesis, Church-Turing thesis. There are various ways to state this thesis.

- A function can be effectively computed (i.e computation consists of a finite number of steps, there are fixed rules for moving from one step to the next and it always outputs the correct answer) if and only if it is computable by a Turing machine.
- Every real world computation can be effectively calculated by a Turing machine.
- Any effective computational method is at most as powerful as a Turing machine.

It should be noted that Church's thesis is not a theorem but a falsiable scientific hypothesis. It cannot be proven or disproven because of the informal definition of the term "effectively computable".

## 2 Historical Background

- In 1933, Kurt Gödel gave the formal definition of general recursive functions which are partial functions  $f : \mathbb{N} \rightarrow \mathbb{N}$  that are computable.
- In 1936, Alonzo Church and his student Kleene worked on lambda calculus which is a formal system for representing computation and it consists of variables, lambda abstraction and application. For example,  $f(x) = x^2 + 5x + 1$  can be represented as  $(\lambda x.(x^2 + 5x + 1))$ . They proved that the set of all computable functions is the same as the set of all general recursive functions.
- In 1936, Alan Turing put forward the concept of a-machines(now known as Turing machines) which are abstract machines that can simulate any algorithm. He also sketched a proof of equivalence of functions defined by lambda and Turing machines.

The terms "Church's thesis" and "Turing's thesis" finally came in the book "Introduction to Meta-mathematics" by Stephen Cole Kleene in 1952.

He stated the Church's thesis as:

"Every effectively calculable function (effectively decidable predicate) is general recursive."

and Turing's thesis as:

"Every function which would naturally be regarded as computable is computable under his definition, i.e. by one of his machines."

And the culmination of these two thesis is the Church-Turing thesis.

### 3 Implications and Applications

The Church-Turing thesis has various implications for computer science and mathematics:

- **Universality:** It establishes that a single, simple model of computation (the Turing machine) can compute anything that can be computed by any algorithm.
- **Unsolvable problems:** It provides a method for proving the existence of algorithmically unsolvable problems, such as the Halting Problem.
- **Computational complexity:** It forms the foundation for complexity theory, which studies the time required in solving various computer problems.
- **Programming languages:** All modern programming languages are Turing-complete, meaning they can express any algorithm a Turing machine can compute.

### References

- [1] Michael Sipser, *Introduction to the Theory of Computation*, Cengage Learning, 2012.
- [2] Wikipedia contributors, *Church-Turing thesis — Wikipedia, The Free Encyclopedia*, 2025, [https://en.wikipedia.org/wiki/Church-Turing\\_thesis](https://en.wikipedia.org/wiki/Church-Turing_thesis),
- [3] Stanford University, *CS103: Mathematical Foundations of Computing*, Stanford University, Department of Computer Science, <https://cs103.stanford.edu/>,
- [4] Stephen Cole Kleene, *Introduction to Metamathematics*, North-Holland Publishing Company, 1952.
- [5] GeeksforGeeks, *Church's Thesis for Turing Machine*, 2023, <https://www.geeksforgeeks.org/churchs-thesis-for-turing-machine/>.