

CS340 - Theory of Computation

What is Computation?



Programs.

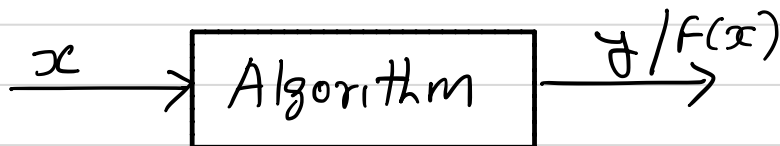
What is a program.

Binary Search - Algorithm.

What is an algorithm? Sequence of steps to implement some computation.

Algorithm - Takes an input and produces an output.

Implements a transformation



Function $F: D \rightarrow R$

Computation \equiv Functions.

Defining a function / Computing a function

$\text{prime} : \mathbb{N} \rightarrow \{\text{Yes}, \text{No}\}.$

$$\text{prime}(n) := \begin{cases} \text{Yes} & \text{if } n \text{ is a prime} \\ \text{No} & \text{if } n \text{ is not a prime.} \end{cases}$$

Which functions admit an algorithm to compute it?

Goal. Techniques to show that a function does not admit any algorithm to compute it.

Efficiency.

Propositional logic :

P - set of propositions

$$\underline{\Phi} := p \in P \mid \neg \alpha \mid \alpha \vee \beta$$

$\varphi \in \underline{\Phi}$ is a wff in propositional logic.

Valuation $\mathcal{V} : P \rightarrow \{T, F\}$

$\mathcal{V} \models \varphi$ [Semantics].

SAT: Given φ is there a valuation \mathcal{V} such that $\mathcal{V} \models \varphi$?

$\text{Voc}(\varphi)$ - Set of propositions occurring in φ .

$$2^{|\text{Voc}(\varphi)|}$$

"Hard problem"

Formal Models of Computation.

Finite state automata.

Deterministic Computation.

Non-deterministic Computation.

Regular Expression.

- Pushdown Automata.
- Turing machine.

Decision Problem. Functions with simple output - "Yes" or "No"

Set Membership Problem.

There is an underlying set S .

Question. Given an a , is $a \in S$?

$$f: D \rightarrow R$$

$$\text{graph}(f) = \{ (a, b) \mid f(a) = b \}.$$

Suppose there is no algorithm to solve

Set membership of $\text{graph}(f)$. Then f is not computable.

Suppose \exists an algorithm A_f to compute f .

$$(a, b) \in \text{graph}(f). \text{ iff } f(a) = b$$

A_f with input a - check if the output is b .

Automata and Computability - Dexter Kozen

Introduction to Automata Theory, Languages and Computation - Hopcroft, Ullman, Motwani

Introduction to the Theory of Computation - Sipser