Computational Complexity I: Computability vs Complexity

Maria-Eirini Pegia

School of Informatics Thessaloniki

Seminar on Theoretical Computer Science and Discrete Mathematics
Aristotle University of Thessaloniki

Context

Computational Theory

Computational Theory

What is a decision problem?

Computational Theory

What is a decision problem?

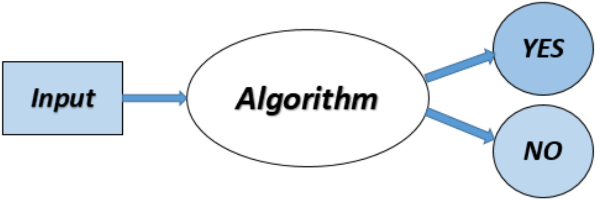
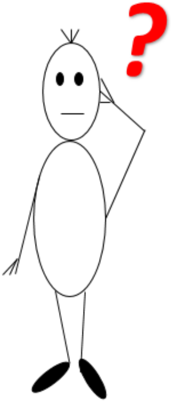
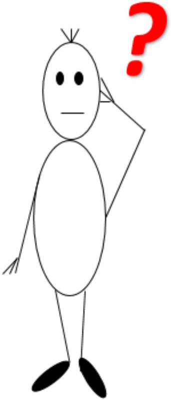


Figure: A decision problem has only two possible outputs, YES or NO.

Computational Theory

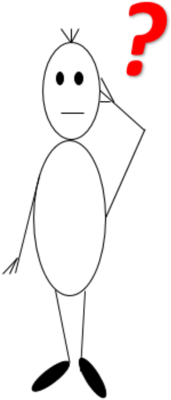


Computational Theory



- Can we solve all the problems?

Computational Theory



- Can we solve all the problems?

- Why some problems are not solving by computers?

History Overview

Figure: Hilbert

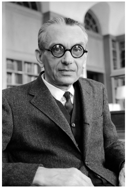


Figure: Gödel

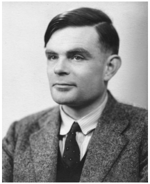


Figure: Turing

History Overview

Figure: Hilbert

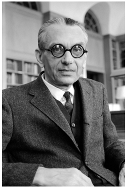


Figure: Gödel

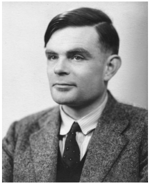


Figure: Turing

- Completeness and automation of mathematics (1900).

History Overview

Figure: Hilbert

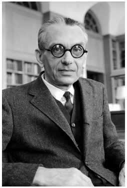


Figure: Gödel

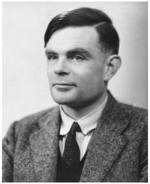


Figure: Turing

- Completeness and automation of mathematics (1900).
- Mathematics are not complete!

History Overview

Figure: Hilbert

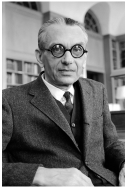


Figure: Gödel

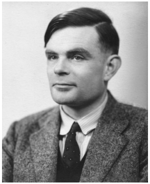


Figure: Turing

- Completeness and automation of mathematics (1900).
- Mathematics are not complete!
- Mathematics are not automating!

History Overview

Figure: Hilbert

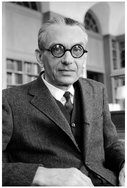


Figure: Gödel

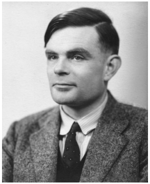
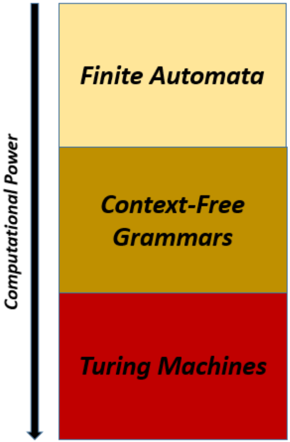


Figure: Turing

- Completeness and automation of mathematics (1900).
- Mathematics are not complete!
- Mathematics are not automating!

There are problems that are not computable!!!

Models of Computation



- Finite Automata and Regular Expressions
- Grammars and Pushdown Automata
- Turing Machines

Context

Finite Automata

- Automaton = an abstract computing device

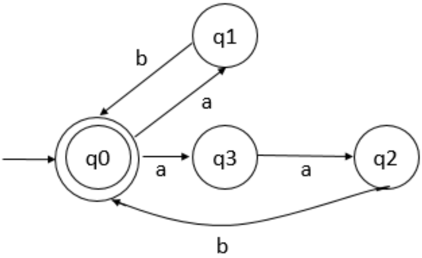


Figure: $L = (ab \cup aab)^*$

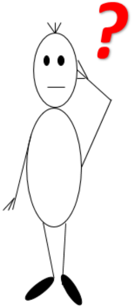
Recognizable Languages

$$L = \{ \; a^n b^n \mid n \geq 0 \; \}$$

Recognizable Languages

$L = \{\ a^n b^n \mid n \geq 0\ \}$ NOT recognizable…
(pumping lemma)

What can we do now?



Context

Context-free Grammars

Definition

$G = (X, V, S, R)$ context-free grammar

X: terminals,

V: variables,

S: axiom,

$R \subseteq \mathrm{V} \times (V \cup X)^*$

Context-free Languages

Context-free Languages

$$L = \{ \ a^n b^n \mid n \geq 0 \ \}$$

Context-free Languages

$L = \{ a^n b^n \mid n \geq 0 \}$ is context-free

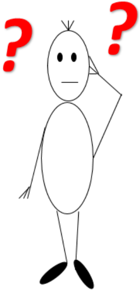
Context-free Languages

$$L = \{\ ww \mid w \in \{a, b\}^*\ \}$$

Context-free Languages

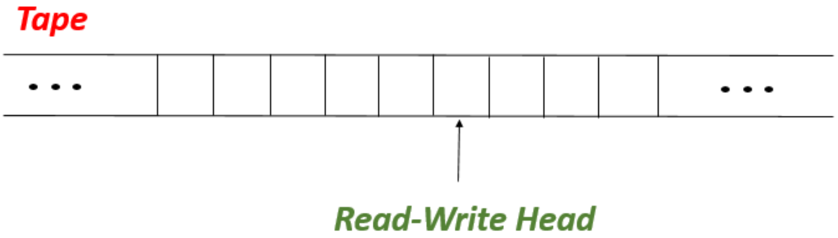
$L = \{\ ww \mid w \in \{a, b\}^*\ \}$ NOT context-free...

What can we do now?



Context

Turing Machine



Turing Machine

Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ Turing Machine

Turing Machine

Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ Turing Machine
Q: set of states,

Turing Machine

Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ Turing Machine
Q: set of states,
Σ: finite set of inputs,

Turing Machine

Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ Turing Machine

Q: set of states,

Σ: finite set of inputs,

Γ: complete set of tape symbols, $\Sigma \subseteq \Gamma$,

Turing Machine

Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ Turing Machine
Q: set of states,
Σ: finite set of inputs,
Γ: complete set of tape symbols, $\Sigma \subseteq \Gamma$,
δ: $Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$,

Turing Machine

Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ Turing Machine
Q: set of states,
Σ: finite set of inputs,
Γ: complete set of tape symbols, $\Sigma \subseteq \Gamma$,
δ: $Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$,
$q_0 \in Q$: initial state,

Turing Machine

Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ Turing Machine

Q: set of states,

Σ: finite set of inputs,

Γ: complete set of tape symbols, $\Sigma \subseteq \Gamma$,

δ: $Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$,

$q_0 \in Q$: initial state,

$B \in \Gamma$: blank symbol,

Turing Machine

Definition

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ Turing Machine

Q: set of states,

Σ: finite set of inputs,

Γ: complete set of tape symbols, $\Sigma \subseteq \Gamma$,

δ: $Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$,

$q_0 \in Q$: initial state,

$B \in \Gamma$: blank symbol,

$F \subset Q$: final states

Thesis Church-Turing

If a problem can be solved with an algorithm, then there is a TM which solves the problem.

Thesis Church-Turing

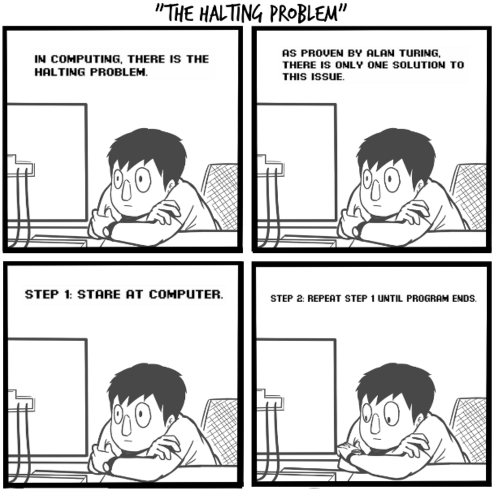
If a problem can be solved with an algorithm, then there is a TM which solves the problem.

Extended model of TM:

- DTM
- Nondeterministic TM
- Restricted TM,
- Multitape TM, ...

Halting Problem

Halting Problem



Context

Synopsis

- Computability
- Finite Automata
- Context-free Grammars
- Turing Machines

References

- Hopcroft, J. E., Ullman, J. D.. Introduction to Automata Theory, Languages, and Computation. Boston: Addison-Wesley, c2001.
- Papadimitriou, C. H.. Computational Complexity. Reading, Mass.: Addison-Wesley, 1994.

Thank you!