Computational Complexity I: From finite automata to Turing machines

Maria-Eirini Pegia

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

Context

What is a decision problem?

Definition

In computability theory and computational complexity theory, a decision problem is a question in some formal system with a yes-or-no answer, depending on the values of some input parameters.

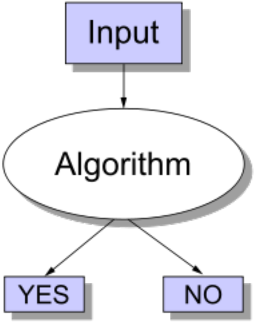


Figure: A decision problem has only two possible outputs, yes or no

Computational Theory

- Decision problems typically appear in mathematical questions of decidability, that is, the question of the existence of an effective method to determine the existence of some object or its membership in a set.

Computational Theory

- Decision problems typically appear in mathematical questions of decidability, that is, the question of the existence of an effective method to determine the existence of some object or its membership in a set.

Can we solve all the problems?

Computational Theory

- Decision problems typically appear in mathematical questions of decidability, that is, the question of the existence of an effective method to determine the existence of some object or its membership in a set.

Can we solve all the problems?

⋆ NO!!!

Computational Theory

- Decision problems typically appear in mathematical questions of decidability, that is, the question of the existence of an effective method to determine the existence of some object or its membership in a set.

Can we solve all the problems?

⋆ NO!!!

Why some problems are not solving by computers?

Why some problems aren't solving by computers?

- Hilbert (1900): Completeness and automation of mathematics
  10th Problem: Algorithm for diophantine equation

Why some problems aren't solving by computers?

- Hilbert (1900): Completeness and automation of mathematics
  10th Problem: Algorithm for diophantine equation
- Algorithm: Expression and Proof of correctness

Why some problems aren't solving by computers?

- Hilbert (1900): Completeness and automation of mathematics
  10th Problem: Algorithm for diophantine equation
- Algorithm: Expression and Proof of correctness
- Does an algorithm exist?

  Definition of "algorithm" with a computational model and the proof that the existence of an algorithm leads to contradictions.

Why some problems aren't solving by computers?

- Gödel: Mathematics are not complete!

Why some problems aren't solving by computers?

- Gödel: Mathematics are not complete!
- Turing: Mathematics are not automating!

Why some problems aren't solving by computers?

- Gödel: Mathematics are not complete!
- Turing: Mathematics are not automating!

There are problems that are not computable.

Why some problems aren't solving by computers?

- Gödel: Mathematics are not complete!
- Turing: Mathematics are not automating!

There are problems that are not computable.

Some of the most important problems in mathematics are undecidable!!!☺☺☺

Why some problems aren't solving by computers?

- Gödel: Mathematics are not complete!
- Turing: Mathematics are not automating!

  There are problems that are not computable.

  Some of the most important problems in mathematics are undecidable!!!☺☺☺

- Matijasevic (1970): There is no algorithm for the solution of every diophantine equation. For every algorithm A there is an equation that A answers false!

Computable Problem and algorithm

- (Computable) problem: defines a mapping from input data to export data.

Computable Problem and algorithm

- (Computable) problem: defines a mapping from input data to export data.
  Intuitively: is defined by a question for entry snapshots

Computable Problem and algorithm

- (Computable) problem: defines a mapping from input data to export data.
  Intuitively: is defined by a question for entry snapshots
- Snapshot: object that corresponds to entry data.

Computable Problem and algorithm

- (Computable) problem: defines a mapping from input data to export data.
  Intuitively: is defined by a question for entry snapshots
- Snapshot: object that corresponds to entry data.
  We set a question and we wait the answer.

Computable Problem and algorithm

- (Computable) problem: defines a mapping from input data to export data.
  Intuitively: is defined by a question for entry snapshots
- Snapshot: object that corresponds to entry data.
  We set a question and we wait the answer.
  Infinity set of snapshots.

Computable Problem and algorithm

- (Computable) problem: defines a mapping from input data to export data.
  Intuitively: is defined by a question for entry snapshots
- Snapshot: object that corresponds to entry data.
  We set a question and we wait the answer.
  Infinity set of snapshots.
- Algorithm: clearly defined process for the solution of a problem in finite time by a computational machine (Turing).

Computable Problem and algorithm

- (Computable) problem: defines a mapping from input data to export data.
  Intuitively: is defined by a question for entry snapshots
- Snapshot: object that corresponds to entry data.
  We set a question and we wait the answer.
  Infinity set of snapshots.
- Algorithm: clearly defined process for the solution of a problem in finite time by a computational machine (Turing).

$\Longrightarrow$ We see TM below!

Example: travelling salesman problem (TSP)

Decision Problem

Given a list of cities and the distances between each pair of cities, is there a possible route that visits each city exactly once and returns to the origin city?

Optimization Problem

Given a list of cities and the distances between each pair of cities, find the shortest (minimum cost) possible route that visits each city exactly once and returns to the origin city.

Problems and Formal Languages

Optimization Problem $\xrightarrow{\textit{reduction}}$ Decision Problem with bound B

Problems and Formal Languages

Optimization Problem $\xrightarrow{\textit{reduction}}$ Decision Problem with bound B

- Minimization: $\exists$ optimal solution with cost $\leq$ B?

Problems and Formal Languages

Optimization Problem $\xrightarrow{\textit{reduction}}$ Decision Problem with bound B

- Minimization: $\exists$ optimal solution with cost $\leq$ B?
- Maximization: $\exists$ optimal solution with gain $\geq$ B?

Problems and Formal Languages

Optimization Problem $\xrightarrow{\textit{reduction}}$ Decision Problem with bound B

- Minimization: $\exists$ optimal solution with cost $\leq$ B?
- Maximization: $\exists$ optimal solution with gain $\geq$ B?
- Optimization Problem is solved (in polynomial time)

iff

the corresponding Decision Problem is solved (in polynomial time).

Finite Automata - Recognizable Languages

Examples:

- ★ L$= (ab \cup aab)^*$
- ★ L$= (\ (ab)^* \ \cup \ (bc)^* \ )$ab
- ★ L$= ((ab \cup aba)^* a)^*$

How I can show that a language is not recognizable?

L=$\{ a^n b^n \mid n \geq 0 \}$ (is recognizable?)

How I can show that a language is not recognizable?

L=$\{\ a^n b^n \mid n \geq 0\ \}$ (is recognizable?)

(NO!!!)

How I can show that a language is not recognizable?

L={ $a^n b^n$ | n≥0 } (is recognizable?)

(NO!!!)

Can we do something better with another computational model???

How I can show that a language is not recognizable?

L={ $a^n b^n$ | n≥0 } (is recognizable?)

(NO!!!)

Can we do something better with another computational model???

YES!!!

We will see it in the next section ☺☺☺

Context-free Grammars - Context-free Languages

Definition

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

X: terminals,

V: variables,

S: axiom,

R ⊆ V x $(V \cup X)^*$ rules

Examples

★ L = { $a^n b^n$ | n≥0 } is context-free???

Examples

★ L = { $a^n b^n$ | n≥0 } is context-free??? YES!!! \(•‿•)/

Examples

- ★ L = { $a^n b^n$ | n≥0 } is context-free??? YES!!! \(•◡•)/
- ★ L = { ww^R | w ∈ $\{a, b\}^*$ }
- ★ L = { w ∈ $\{a, b\}^*$ | w = w^R }

How I can show that a language is not context-free?

L $= \{$ ww $|$ w $\in \{a, b\}^*$ $\}$ (is context-free?)

How I can show that a language is not context-free?

L = { ww | w ∈ $\{a, b\}^*$ } (is context-free?) (NO!!!)

Can we do something better with another computational model???

How I can show that a language is not context-free?

L $= \{$ ww | w $\in \{a, b\}^*$ $\}$ (is context-free?) (NO!!!)

Can we do something better with another computational model???

YES!!! We will see it in the next section ☺☺☺

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Q: set of states,

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Q: set of states,

Σ: finite set of inputs,

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Q: set of states,

Σ: finite set of inputs,

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

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Q: set of states,

Σ: finite set of inputs,

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

δ: Q x Γ $\longrightarrow$ Q x Γ x $\{L, R\}$

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Q: set of states,

Σ: finite set of inputs,

Γ: complete set of tape symbols, Σ ⊂ Γ,

δ: Q x Γ $\longrightarrow$ Q x Γ x $\{L, R\}$
- L, R standing for 'left' and 'right' direction, respectively

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Q: set of states,

Σ: finite set of inputs,

Γ: complete set of tape symbols, Σ ⊂ Γ,

δ: Q x Γ $\longrightarrow$ Q x Γ x $\{L, R\}$
- L, R standing for 'left' and 'right' direction, respectively

q_0 ∈ Q: initial state

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Q: set of states,

Σ: finite set of inputs,

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

δ: $Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$
- L, R standing for 'left' and 'right' direction, respectively

$q_0 \in Q$: initial state

$B \in \Gamma$: blank symbol

Definition

M = (Q, Σ, Γ, δ, q_0, B, F): Turing Machine (TM)

Q: set of states,

Σ: finite set of inputs,

Γ: complete set of tape symbols, Σ ⊂ Γ,

δ: Q x Γ $\longrightarrow$ Q x Γ x $\{L, R\}$
- L, R standing for 'left' and 'right' direction, respectively

q_0 ∈ Q: initial state

B ∈ Γ: blank symbol

F ⊂ Q: final states

Example 1: printing 110

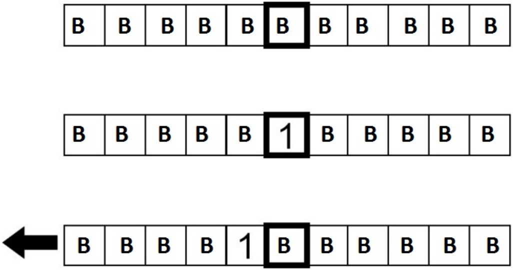
B	B	B	B	B	B	B	B	B	B	B

Example 1: printing 110

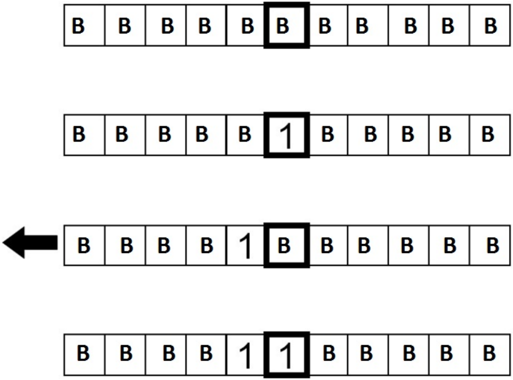
B	B	B	B	B	B	B	B	B	B	B

B	B	B	B	B	1	B	B	B	B	B

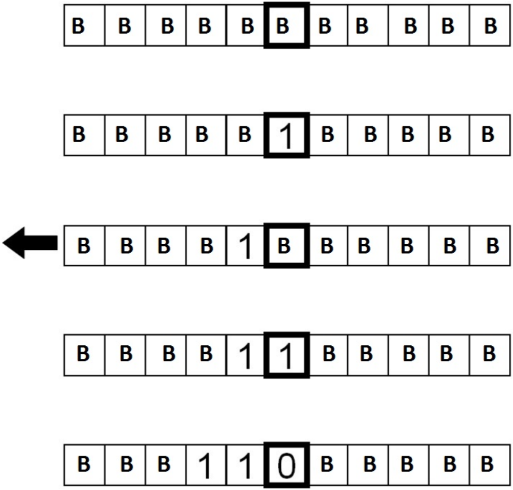
Example 1: printing 110



Example 1: printing 110



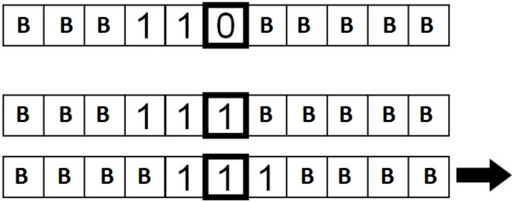
Example 1: printing 110



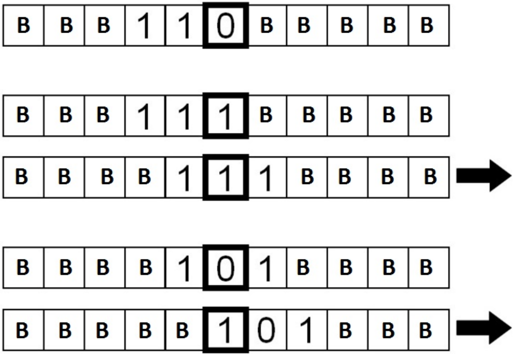
Example 2: bit inversion of 110 ☺ !!!

B	B	B	1	1	0	B	B	B	B	B

Example 2: bit inversion of 110 ☺ !!!



Example 2: bit inversion of 110 ☺ !!!



Example 2: bit inversion of 110 ☺ !!!

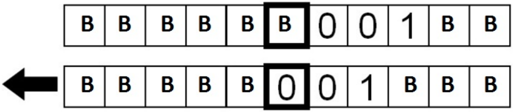
B	B	B	1	1	**0**	B	B	B	B	B

B	B	B	1	1	**1**	B	B	B	B	B
B	B	B	B	1	**1**	1	B	B	B	B

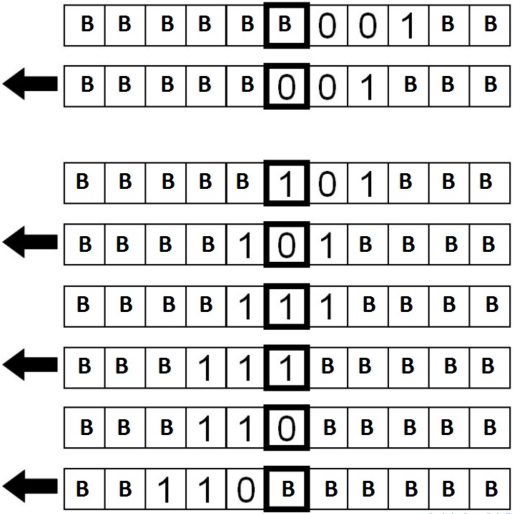
B	B	B	B	1	**0**	1	B	B	B	B
B	B	B	B	B	**1**	0	1	B	B	B

B	B	B	B	B	**0**	0	1	B	B	B
B	B	B	B	B	**B**	0	0	1	B	B

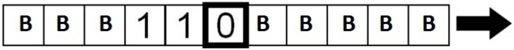
Example 3: bit inversion of 001



Example 3: bit inversion of 001



Example 3: bit inversion of 001



Example 3: bit inversion of 001

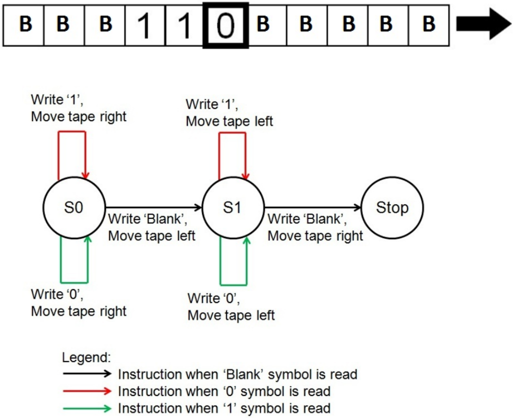


Figure: Example 1 and 2 'together'

Example 4

M = (Q, Σ, Γ, δ, q_0, B, F)

Q $= \{q_0, q_1, q_2, q_3, q_4\}$, Σ $= \{0, 1\}$, Γ $= \{0, 1, X, Y, B\}$, F $= \{q_4\}$,

δ: Q x Γ $\longrightarrow$ Q x Γ x $\{L, R\}$

State	Symbol 0	1	X	Y	B
q_0	(q_1, X, R)	–	–	(q_3, Y, R)	–
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)	(q_2, Y, L)	–
q_3	–	–	–	(q_3, Y, R)	(q_4, B, R)
q_4	–	–	–	–	–

Example 4

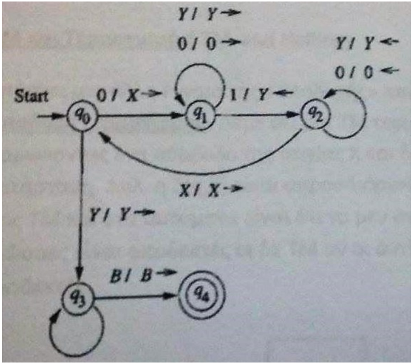


Figure: L $=\{ 0^n 1^n \mid n \geq 1 \}$

Example 4

- $q_0 0011 \vdash^* Xq_1 011 \vdash^* X0q_1 11 \vdash^* Xq_2 0Y1 \vdash^* q_2 X0Y1 \vdash^* Xq_0 0Y1 \vdash^* XXq_1 Y1 \vdash^* XXYq_1 1 \vdash^* XXq_2 YY \vdash^* Xq_2 XYY \vdash^* XXq_0 YY \vdash^* XXYq_3 Y \vdash^* XXYYq_3 B \vdash^* XXYYBq_4 B$

Example 4

- $q_0 0011 \vdash^* \mathrm{X} q_1 011 \vdash^* \mathrm{X}0 q_1 11 \vdash^* \mathrm{X} q_2 0\mathrm{Y}1 \vdash^* q_2 \mathrm{X}0\mathrm{Y}1 \vdash^* \mathrm{X} q_0 0\mathrm{Y}1 \vdash^* \mathrm{XX} q_1 \mathrm{Y}1 \vdash^* \mathrm{XXY} q_1 1 \vdash^* \mathrm{XX} q_2 \mathrm{YY} \vdash^* \mathrm{X} q_2 \mathrm{XYY} \vdash^* \mathrm{XX} q_0 \mathrm{YY} \vdash^* \mathrm{XXY} q_3 \mathrm{Y} \vdash^* \mathrm{XXYY} q_3 \mathrm{B} \vdash^* \mathrm{XXYYB} q_4 \mathrm{B}$

- $q_0 0010 \vdash^* \mathrm{X} q_1 010 \vdash^* \mathrm{X}0 q_1 10 \vdash^* \mathrm{X} q_2 0\mathrm{Y}0 \vdash^* \mathrm{X} q_0 0\mathrm{Y}0 \vdash^* \mathrm{XX} q_1 \mathrm{Y}0 \vdash^* \mathrm{XXY} q_1 0 \vdash^* \mathrm{XXY}0 q_1 \mathrm{B}$

$$L(M) = \{\ w \in \Sigma^* \mid q_0 w \vdash^* upv \text{ for some } p \in F \text{ and any } u,v \in \Gamma\ \}$$

recursively enumerable languages: the recognizable languages in TM

An equivalent model: Unrestricted Grammars

Definition

G=(X,V,S,R): unrestricted grammar

An equivalent model: Unrestricted Grammars

Definition

G=(X,V,S,R): unrestricted grammar

X: terminals

An equivalent model: Unrestricted Grammars

Definition

G=(X,V,S,R): unrestricted grammar

X: terminals

V: set of nonterminal symbols, $X \cap V = \varnothing$

An equivalent model: Unrestricted Grammars

Definition

G=(X,V,S,R): unrestricted grammar

X: terminals

V: set of nonterminal symbols, $X \cap V = \varnothing$

$S \in V$: axiom

An equivalent model: Unrestricted Grammars

Definition

G=(X,V,S,R): unrestricted grammar

X: terminals

V: set of nonterminal symbols, $X \cap V = \varnothing$

$S \in V$: axiom

R: rules of the form $u \implies w$, $\quad u, w \in (V \cup X)^*$, $w \neq \epsilon$

Example

L $= \{$ ww $|$ w $\in \{a, b\}^*$ $\}$ is recursively enumerable???

Example

L $= \{$ ww | w $\in \{a, b\}^* \}$ is recursively enumerable???

YES!!! \(•‿•)/

Example

L = { ww | w $\in \{a, b\}^*$ } is recursively enumerable???

YES!!! \(•‿•)/

G = ($\{S, F, M, A, B\}$, $\{a, b\}$, S, R): unrestricted grammar

S $\longrightarrow$ FM, F $\longrightarrow$ FaA, F $\longrightarrow$ FbB, Aa $\longrightarrow$ aA, Ab $\longrightarrow$ bA, Ba $\longrightarrow$ aB, Bb $\longrightarrow$ bB, AM $\longrightarrow$ Ma, BM $\longrightarrow$ Mb, F $\longrightarrow$ ε, M $\longrightarrow$ ε

Example

- $S \Longrightarrow \varepsilon M \Longrightarrow \varepsilon\varepsilon = \varepsilon$
- $S \Longrightarrow (FaA)M \Longrightarrow Fa(Ma) \Longrightarrow \varepsilon aMa \Longrightarrow a\varepsilon a \Longrightarrow aa = a^2$
- $S \Longrightarrow (FaA)M \Longrightarrow F(aAM) \Longrightarrow (FbB)(aAM) \Longrightarrow Fb(Ba)AM \Longrightarrow Fb(aB)BAM \Longrightarrow FbaB(AM) \Longrightarrow FbaB(Ma) \Longrightarrow Fba(BM)a \Longrightarrow Fba(bM)a \Longrightarrow baba = (ba)^2$

Thesis Church - Turing

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

Thesis Church - Turing

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

- extended models of TM
  - DTM
  - Nondeterministic TM
  - Restricted TM
  - Multitape TM

  .
  .
  .

Halting Problem is undecidable

¯_(ツ)_/¯

★ In computability theory, the halting problem is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running or continue to run forever.

Halting Problem is undecidable

¯_(ツ)_/¯

★ In computability theory, the halting problem is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running or continue to run forever.

★ Alan Turing (1936) proved that a general algorithm to solve the halting problem for all possible program-input pairs cannot exist.

References

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

Next

¯_(ツ)_/¯

¯_(ツ)_/¯

¯_(ツ)_/¯ Asymptotic Notation!!!

¯_(ツ)_/¯

¯_(ツ)_/¯

¯_(ツ)_/¯

¯_(ツ)_/¯ Classification Algorithms!!!

¯_(ツ)_/¯

¯_(ツ)_/¯

Thank you!!!