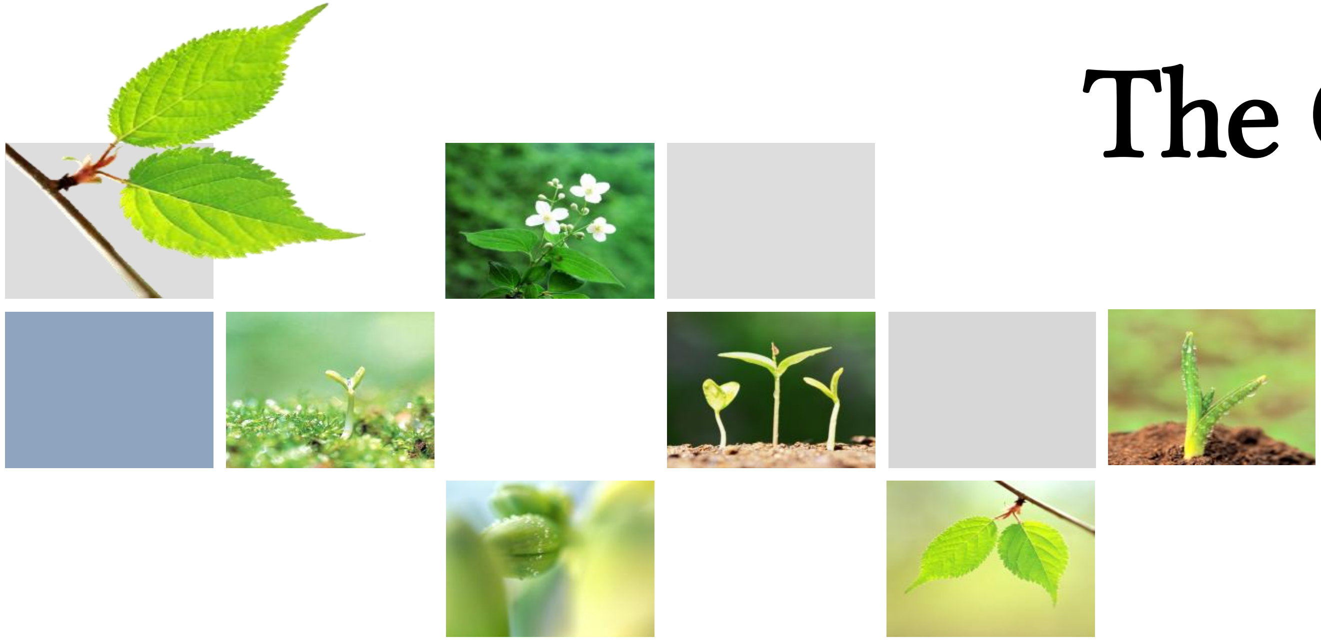
بسم الله الرحمن الرحیم



The Church-Turing Thesis

Ali Shakiba – Vali-e-Asr University of Rafsanjan – ali.shakiba@vru.ac.ir

We are going to discuss …

1. Turing Machines
2. Variants of T.M.s
3. What’s an “Algorithm”?

Turing Machine

A **Turing Machine** is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where

1. Q is the set of states
2. Σ is the **input alphabet** such that **blank symbol** $\sqcup \notin \Sigma$
3. Γ is the **tape alphabet** such that $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
4. $\delta: Q' \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ where $Q' = Q \setminus \{q_{accept}, q_{reject}\}$
5. $q_0 \in Q$ is the **start state**
6. $q_{accept}, q_{reject} \in Q$ are accept and reject states where $q_{accept} \neq q_{reject}$

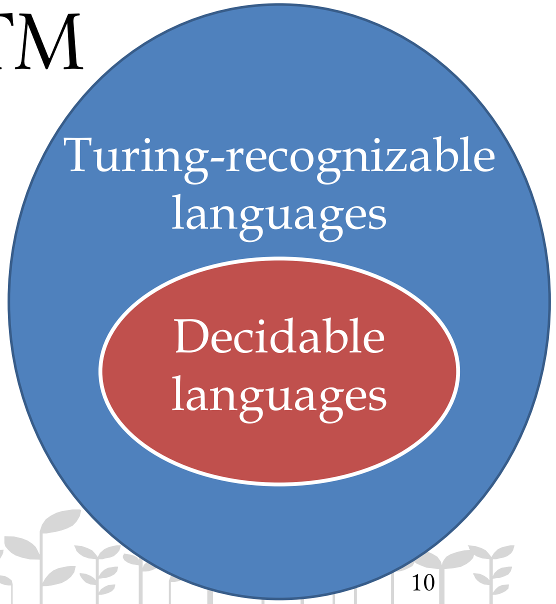
Turing–recognizable and –decidable languages

A language L is Turing-recognizable (recursively-enumerable) if some TM

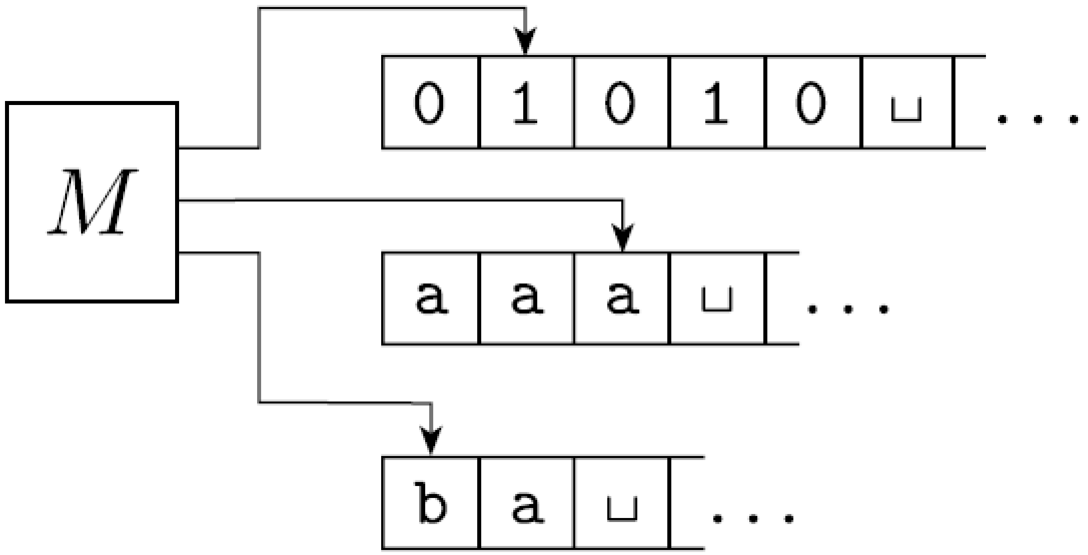
1. accept strings in L, and
2. rejects strings not in L by entering q_{reject} or looping

A language L is Turing-decidable (recursive) if some TM

1. accept strings in L, and
2. rejects strings not in L by entering q_{reject}



Multitape TMs



Multitape TMs

A **k-tape TM** is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where

1. Q is the set of states
2. Σ is the **input alphabet** such that **blank symbol** $\sqcup \notin \Sigma$
3. Γ is the **tape alphabet** such that $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
4. $\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$
5. $q_0 \in Q$ is the **start state**
6. $q_{accept}, q_{reject} \in Q$ are accept and reject states where $q_{accept} \neq q_{reject}$

Multitape TM ? TM

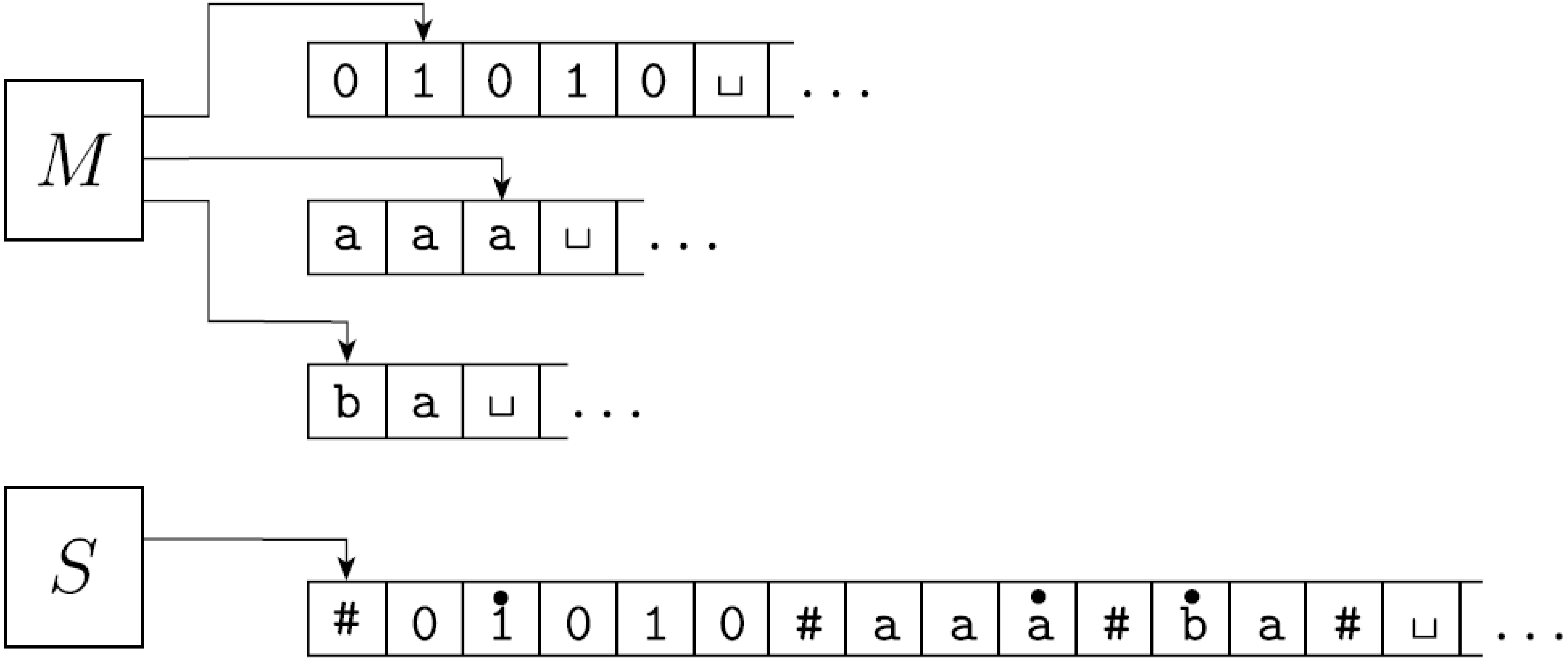
Theorem 3.13: Every multitape TM has an equivalent single-tape TM.

Proof on P. 177

Multitape TM ? TM

Theor

TM.



77

Multitape TM ? TM

Theorem 3.13: Every multitape TM has an equivalent single-tape TM.

Proof on P. 177

Corollary 3.15: A language is Turing-recognizable if and only if some multitape TM recognizes it.

Nondeterministic TMs

A **nondeterministic TM** is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ where

1. Q is the set of states
2. Σ is the **input alphabet** such that **blank symbol** $\sqcup \notin \Sigma$
3. Γ is the **tape alphabet** such that $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
4. $\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$
5. $q_0 \in Q$ is the **start state**
6. $q_{accept}, q_{reject} \in Q$ are accept and reject states where $q_{accept} \neq q_{reject}$

Nondeterministic TM ? TM

Theorem 3.16: Every nondeterministic TM has an equivalent TM.

Proof on PP. 178-179

Corollary 3.18: A language is Turing-recognizable if and only if some nondeterministic TM recognizes it.

Corollary 3.19: A language is Turing-decidable if and only if some nondeterministic TM decides it.

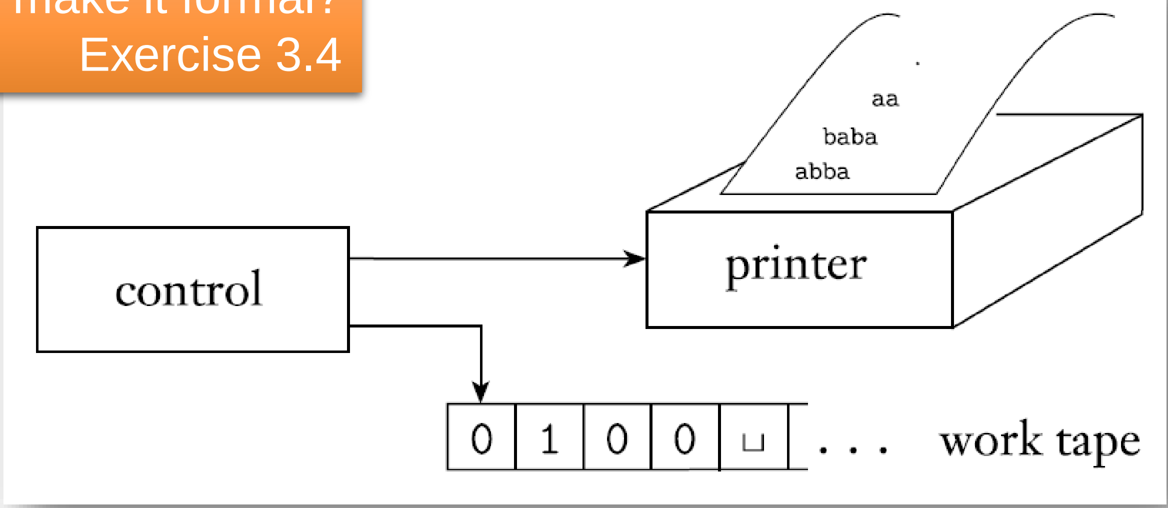
Exercise 3.3

NTM is a **decider** if all branches halt on all inputs.

Enumerators

- Starts with a blank tape
- Print strings

Can you make it formal?
Exercise 3.4



- The language of enumerator E is the set of strings it prints.
- It may generate strings in any order or with repetition

Theorem 3.21: A language is Turing-recognizable if and only if some enumerator enumerates it.

Proof on P. 181

TMs Robustness

- All reasonable variants of TM recognize the same class of language that a standard TM can recognize.
  - k-PDA for k Problem 3.9
  - write-once TM Problem 3.10
  - TM with
    - doubly infinite tape Problem 3.11
    - left reset, i.e. {R,RESET} Problem 3.12
    - ~~stay instead of left, i.e. {R,STAY}~~ Problem 3.13
  - Queue automaton Problem 3.14
  - Nondeterministic, Multi-tape

Hilbert's 10th Problem

Is there "a process according to which it can be determined by a finite number of operations" that a polynomial has an integral root?

Is there "an algorithm" that a polynomial has an integral root?

NO!

Yuri Vladimirovich Matijasevich, 1970

Church-Turing Thesis

- 1936
  - Alonzo Church: λ-calculus
  - Alan Turing: Turing Machines

Intuitive notion of algorithms	equals	*Turing machine algorithms*

Further Exercises

- Read Page 184, that D is TM-undecidable
  - Try to solve Problem 3.21

$D = \{p |\ p \text{ is a polynomial with an integral root}\}$

- Try to turn the "Proof Ideas" into formal "Proofs" and compare with the proofs in the book.
- Try to solve the following Problems:
  - 3.9-3.14
  - 3.17-3.20

Further Readings

- The following book annotates the elaborating paper by A. Turing in 1936:
  - Petzold C. **The annotated Turing: a guided tour through Alan Turing's historic paper on computability and the Turing machine**. Wiley Publishing; 2008 Jun 16.
  - Turing AM. **On computable numbers, with an application to the Entscheidungsproblem**. J. of Math. 1936 Nov;58(345-363):5.
- How about "Random Access TMs"?
  - Lewis HR, Papadimitriou CH. **Elements of the Theory of Computation**. Prentice Hall PTR; 1997 Aug 1.
    - Section 4.4, pp. 210-221
  - Cook SA, Reckhow RA. **Time-bounded random access machines**. InProceedings of the fourth annual ACM symposium on Theory of computing 1972 May 1 (pp. 73-80). ACM.

Further Readings (cont'd)

- Undecidability of Hilbert's 10^{th} problem
  - Matijasevich IV. **Hilbert's tenth problem**. MIT press; 1993.
  - Davis M. **Hilbert's tenth problem is unsolvable**. The American Mathematical Monthly. 1973 Mar 1;80(3):233-69.
- Read these two papers to enjoy!
  - Goldreich O. **Invitation to complexity theory**. ACM Crossroads. 2012 Mar 1;18(3):18-22.
  - Horswill I. **What is computation?**. XRDS: Crossroads, The ACM Magazine for Students. 2012 Mar 1;18(3):8-14.