Theory of Formal Languages and Automata

Lecture 17

Mahdi Dolati

Sharif University of Technology

Fall 2023

December 1, 2023

Variants of TMs

- Variants of the Turing machine model:
  - All have the same power,

- **Robustness**: Invariance to certain changes in the definition:
  - Turing machines have an astonishing degree of robustness.

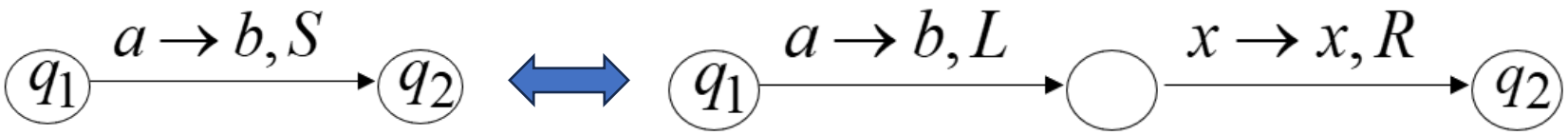
Variants of TMs

Stay Put

- The transition function:

$$\delta: Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\},$$

- We can convert any TM with the "**stay put**" feature to one that does not have it:
  - Replace stay put transition with two transitions:
    - Moves to the right,
    - Move back to the left.



Variants of TMs
Equivalence

- Two machines are equivalent if they recognize the same language.
- The key to showing the equivalence of TM variants:

To show that two models are equivalent, we simply need to show that one can **simulate** the other.

Variants of TMs

Multitape TM

- A multiple TM: Has several tapes:
  - Each tape has its own head for read and write,
  - Input in on tape 1 and others are filled with blank.

- Transition function:

$$\delta: Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, S\},$$

$$\delta(q_i, a_1, \dots, a_k) = (q_j, b_1, \dots, b_k, d_1, \dots, d_k), \mathrm{d_i} \in \{L, R, S\}$$

- In state q_i if heads 1 to k read a_1 to a_k, then
- Go to state q_j, write b_1 to b_k on tapes, and change heads according to d_1 to d_k.

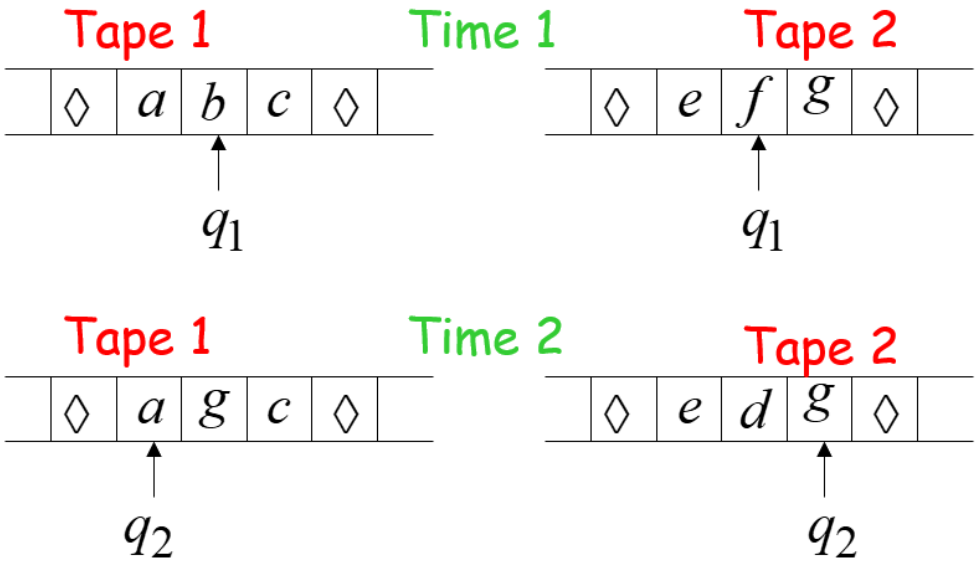
Variants of TMs

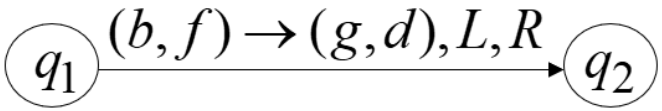
Multitape TM

- Transition function:

$$\delta: Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, S\},$$

$$\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, d_1, \ldots, d_k), \mathrm{d_i} \in \{L, R, S\}$$

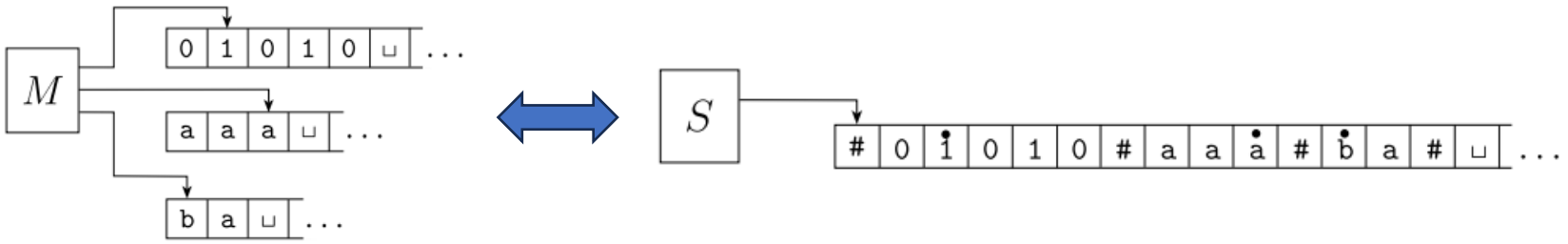




Variants of TMs
Multitape TM

Theorem: Every multitape Turing machine has an equivalent single-tape Turing machine.

- Proof:
  - Consider TM **M** with **k** tapes.
  - Convert **M** to an equivalent TM **S** with a single tape.
  - **S** stores the information of **k** tapes on a single tape:
    - Use symbol **#** to separate different tapes,
    - Use a dotted version of tape symbols to mark the location of tape heads. These are *virtual* tape heads.



Variants of TMs
Multitape TM

- Proof (Cont.):
  - Consider TM **M** with **k** tapes. Convert **M** to an equivalent TM **S** with a single tape.
  - Working of S on input $w = w_1 \dots w_n$:
    - Prepare the tape to represent all k tapes of M:

$$\#\dot{w}_1 w_2 \cdots w_n \#\dot{\sqcup}\#\dot{\sqcup}\# \cdots \#.$$

    - Scan the tape and determine all symbols under the virtual tape heads.
    - Make a second pass and update the tapes.
    - Each time a virtual head moves onto a #:
      - Write a blank,
      - Shift the rest of the tape to right.

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

Variants of TMs

Nondeterministic TM

- Definition:

$$\delta: Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\}).$$

- Computation:
  - A tree,
  - Each branch correspond to a different possibility,
  - If one branch leads to the accept state, then the input is accepted.

Variants of TMs

Nondeterministic TM

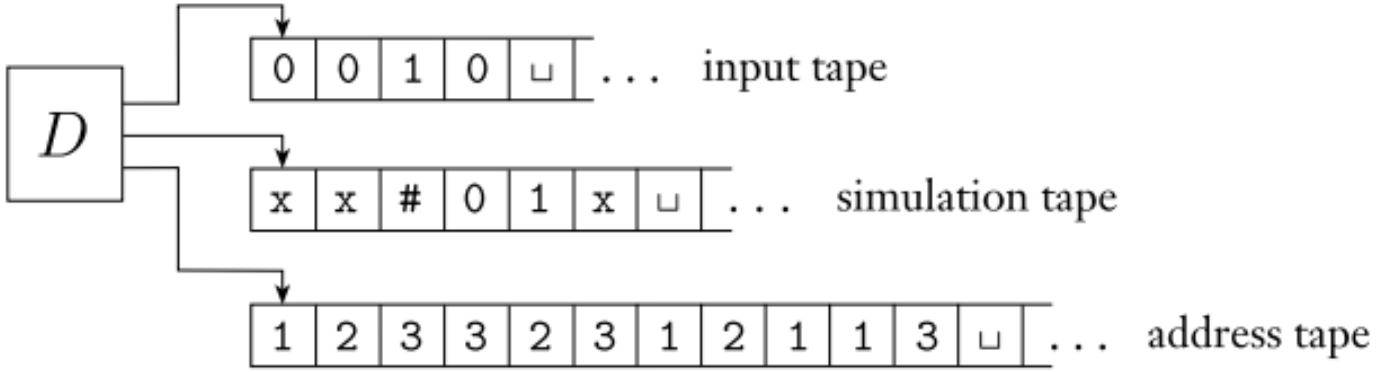
Theorem: Every nondeterministic Turing machine has an equivalent deterministic Turing machine.

- Proof idea:
  - Consider nondeterministic TM N,
  - Construct a deterministic TM D that simulates N.
  - D tries all possible branches of N.
  - Computation of N is a tree:
    - Searching the tree **depth-first** is not a good idea: A branch may never end.
    - Thus, D must explore the tree **breadth-first**.

Variants of TMs
Nondeterministic TM

Theorem: Every nondeterministic Turing machine has an equivalent deterministic Turing machine.

- Proof:
  - D is a TM with 3 tapes:
    - One tape to keep the input,
    - One tape to keep a copy of N's tape,
    - One tape to keep the location in the computation tree of N.



Variants of TMs
Nondeterministic TM

- Proof (Cont.):
  - Assume each configuration leads to at most b configurations:
    - Choices of N's transition function,
  - Define $\Gamma_b = \{1,2,3, \dots, b\}$.
  - Address is a string over Γ_b that shows the choices of transition.
    - For example: 231 means follows the 2nd, 3rd, and finally 1st choices.
    - Empty string is the address of the initial configuration.

Variants of TMs

Nondeterministic TM

- Proof (Cont.):
  1. Initially, tape 1 contains the input w, and tapes 2 and 3 are empty.
  2. Copy tape 1 to tape 2 and initialize the string on tape 3 to be ε.
  3. Use tape 2 to simulate N with input w on one branch of its nondeterministic computation. Use tape 3 to determine which choice to make among those allowed by N's transition function.
     1. No more symbols on tape 3 or invalid: Go to stage 4.
     2. A rejecting configuration is encountered: Go to state 4.
     3. An accepting configuration is encountered: Accept.
  4. Replace the string on tape 3 with the next string in the **string ordering**. Simulate the next branch of N's computation by going to stage 2.

Variants of TMs
Nondeterministic TM

- Two corollaries:

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

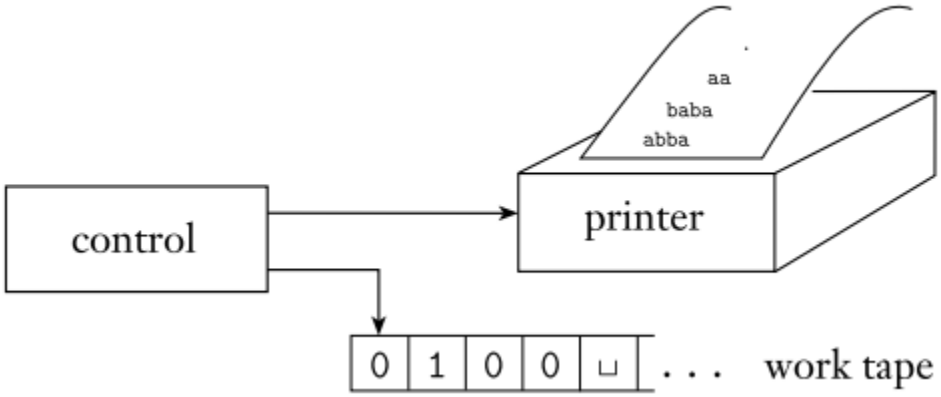
- We call a nondeterministic Turing machine a **decider** if all branches halt on all inputs.

Corollary: A language is decidable if and only if some nondeterministic Turing machine decides it.

Variants of TMs

Enumerators

- **Enumerator**: A Turing machine with an attached printer. The Turing machine can use that printer as an output device to print strings.
- The language enumerated by enumerator E is the collection of all the strings that it eventually prints out (in any order, possibly with repetitions).
- The name **recursively enumerable** originates from this variant.



Variants of TMs

Enumerators

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

- Proof:
  - Consider enumerator E and TM M.
  - **Direction 1:**
  - M can run E,
  - Every time E prints a string, M compares it with the input:
    - If matched: Accept,
    - If not matched: Continue with running E.
  - M accepts all strings that E prints out.

Variants of TMs

Enumerators

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

- Proof (Cont.):
  - Consider enumerator E and TM M.
  - **Direction 2:**
  - Assume s1, s2, … is a list of all possible strings in Σ^*.
  - Run M for i=1,2, … steps on each input s1, s2, …, si.
  - Print any input that is accepted by M.

  - If M accepts string s in K steps, then E eventually prints s (in step K, K+1, …).

Variants of TMs
Other Models

- There are many other models of general purpose computation:
  - All share the essential feature of **unrestricted** access to **unlimited** memory.
- All these models are equivalent when satisfy reasonable requirements (finite amount of work in a single step).

- Similar to different programming languages: Python, Java, …
- They can all program any given algorithm: They describe the same class of algorithms.