

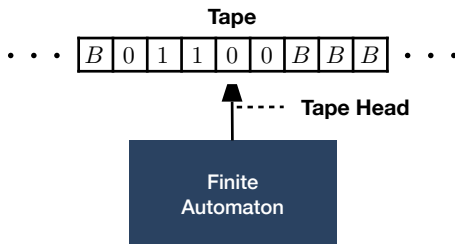
Lecture 23 – Extensions of Turing Machines

COSE215: Theory of Computation

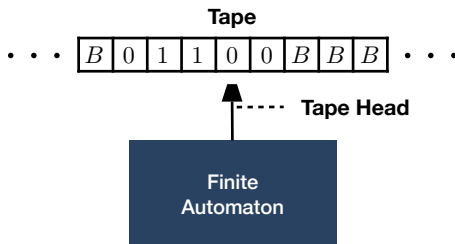
Jihyeok Park



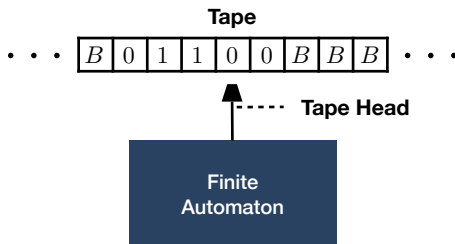
2024 Spring



- A **Turing machine (TM)** is a finite automaton with a **tape**.
- A language accepted by a TM is **Recursively Enumerable**.



- A **Turing machine (TM)** is a finite automaton with a **tape**.
- A language accepted by a TM is **Recursively Enumerable**.
- What happens if we define **other extensions** of TMs?
- Are they **more powerful** than TMs?



- A **Turing machine (TM)** is a finite automaton with a **tape**.
- A language accepted by a TM is **Recursively Enumerable**.
- What happens if we define **other extensions** of TMs?
- Are they **more powerful** than TMs? **NO!!**

1. Extensions of Turing Machines

- TMs with Storage

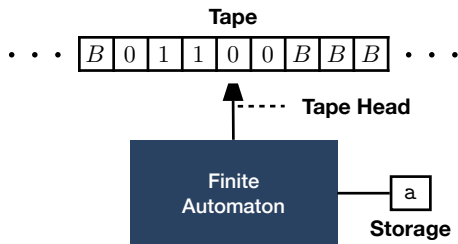
- Multi-track TMs

- Multi-tape TMs

- Non-deterministic TMs

- More Extensions of TMs

We can define a TM with a **storage**:

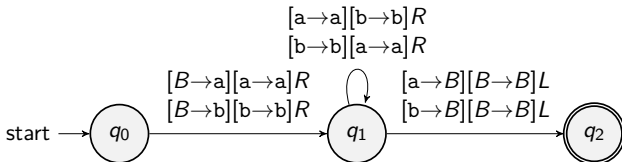


It has additional **storage** affecting the transition function:

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

$$L(M) = \{ab^n \text{ or } ba^n \mid n \geq 0\}$$

The following **TM with storage** accepts $L(M)$, and see the example for $abb \in L(M)$.¹



¹<https://plrg.korea.ac.kr/courses/cose215/materials/tm-storage-abn-or-ba>

Theorem

*A language accepted by a **TM with storage** is recursively enumerable (i.e., accepted by a standard **TM**).*

Theorem

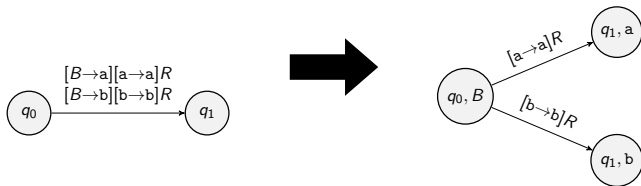
A language accepted by a **TM with storage** is recursively enumerable (i.e., accepted by a standard **TM**).

Proof) We can define an equivalent standard TM by using pairs of states and symbols in the storage as its states:

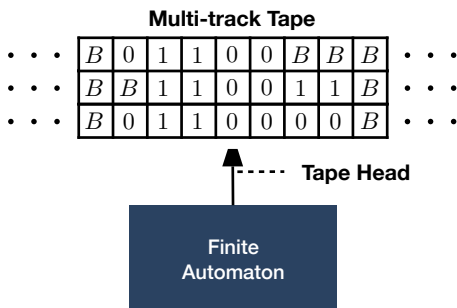
$$\delta'((q, a), b) = \delta(q, a, b)$$

where $Q' = Q \times \Gamma$ and $\delta' : Q' \times \Gamma \rightarrow Q' \times \Gamma \times \{L, R\}$.

For example,



We can define a TM with a **multi-track tape**:

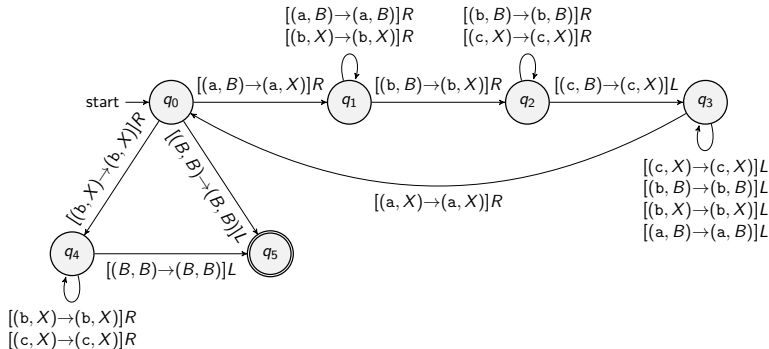


It has a tape with n **tracks** and a **single tape head**:

$$\delta : Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \{L, R\}$$

$$L(M) = \{a^n b^n c^n \mid n \geq 0\}$$

The following **multi-track TM** accepts $L(M)$, and see the example for $aabbcc \in L(M)$.²



²<https://plrg.korea.ac.kr/courses/cose215/materials/tm-multi-track-an-bn>

Theorem

*A language accepted by a **multi-track TM** is recursively enumerable (i.e., accepted by a standard **TM**).*

Theorem

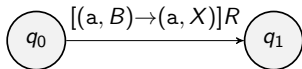
A language accepted by a **multi-track TM** is **recursively enumerable** (i.e., accepted by a **standard TM**).

Proof) We can define an equivalent standard TM by using n -tuples of symbols as a single symbol:

$$\delta'(q, \alpha) = \delta(q, \alpha)$$

where $\Gamma' = \Gamma^n$ and $\delta' : Q \times \Gamma' \rightarrow Q \times \Gamma' \times \{L, R\}$.

For example,

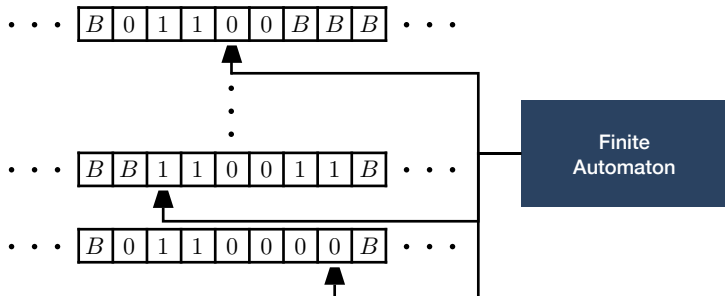


...	B	a	b	B	...
...	B	X	B	B	...



...	(B, B)	(a, X)	(b, B)	(B, B)	...
-----	--------	--------	--------	--------	-----

We can define a TM with **multiple tapes**:

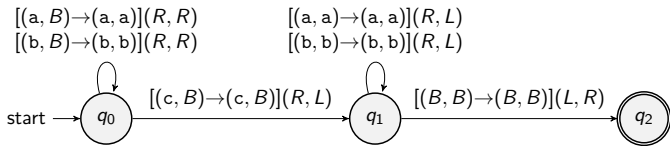


It has n **tapes**, and each tape has its **own head** that can move independently:

$$\delta : Q \times \Gamma^n \rightarrow Q \times (\Gamma \times \{L, R\})^n$$

$$L(M) = \{wcw^R \mid w \in \{a, b\}^*\}$$

The following **multi-tape TM** accepts $L(M)$, and see the example for $abbcbbba \in L(M)$.³



³<https://plrg.korea.ac.kr/courses/cose215/materials/tm-multi-tape-w-c-wr>

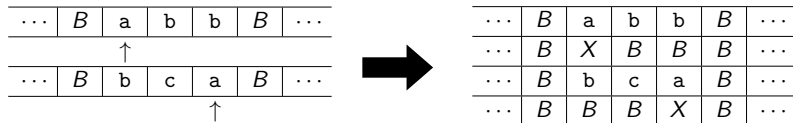
Theorem

*A language accepted by a **multi-tape TM** is recursively enumerable (i.e., accepted by a standard **TM**).*

Theorem

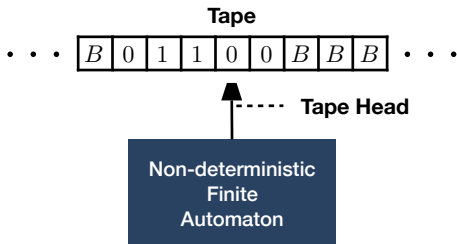
A language accepted by a **multi-tape TM** is **recursively enumerable** (i.e., accepted by a **standard TM**).

Proof) For a given n -tape TM, we can define an equivalent $2n$ -track TM with storage by using **odd** tracks for the original **tapes** and **even** tracks for the **tape heads**:



We can simulate one-step in the n -tape TM by gathering all the symbols pointed by the n heads into the storage, and then taking the action same as done by the n -tape TM.

We can define a TM with **non-deterministic transitions**:

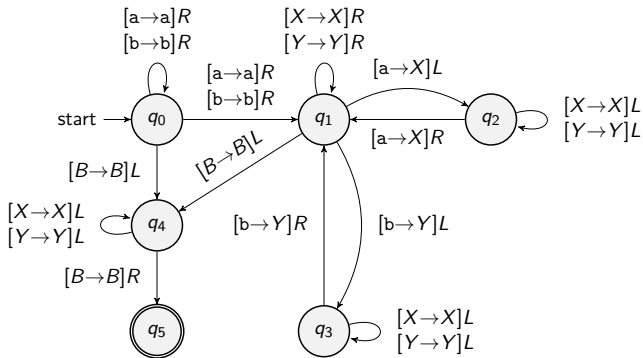


It has a **non-deterministic transition function**:

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

$$L(M) = \{ww^R \mid w \in \{a, b\}^*\}$$

The following **nondeterministic TM** accepts $L(M)$, and see the example for $abba \in L(M)$.⁴



⁴<https://plrg.korea.ac.kr/courses/cose215/materials/ntm-w-wr.pdf>

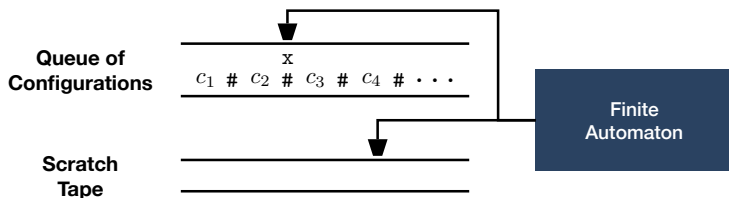
Theorem

*A language accepted by a **non-deterministic TM** is recursively enumerable (i.e., accepted by a standard TM).*

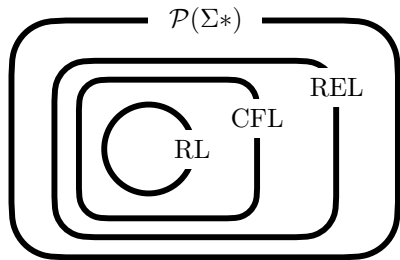
Theorem

A language accepted by a **non-deterministic TM** is **recursively enumerable** (i.e., accepted by a **standard TM**).

Proof) For a given non-deterministic TM, we can define an equivalent 2-tape TM: 1) a 2-track tape to maintain a **queue of configurations** and 2) a normal track to **simulate** the tape of the original TM.



- There are more extensions of TMs:
 - TMs with **Stay Option** – L : Left, R : Right, and S : **Stay**
 - **Queue Automata** – Automata with **Queue**
 - **Random Access Machines** – TMs with **Random Access Memory**
 - ...
- They are all **equivalent** to TMs.
- A standard **TM** is the **most powerful model of computation**.



$$\begin{array}{ccc}
 \text{TM} & = & \text{ETM} \\
 \text{(Turing Machine)} & & \text{(All Extensions of TMs)} \\
 \\
 \text{||} & & \\
 \text{REL} & & \\
 \text{(Recursively Enumerable} & & \\
 \text{Language)} & &
 \end{array}$$

1. Extensions of Turing Machines

- TMs with Storage

- Multi-track TMs

- Multi-tape TMs

- Non-deterministic TMs

- More Extensions of TMs

- Please see this document on GitHub:

<https://github.com/ku-plrg-classroom/docs/tree/main/cose215/tm-examples>

- The due date is 23:59 on Jun. 12 (Wed.).
- Please only submit `Implementation.scala` file to **Blackboard**.

- The Origin of Computer Science

Jihyeok Park
jihyeok_park@korea.ac.kr
<https://plrg.korea.ac.kr>