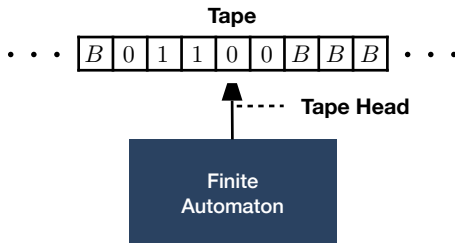# Lecture 23 – Extensions of Turing Machines
COSE215: Theory of Computation

Jihyeok Park

PLRG

2025 Spring

- 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!!**

Contents
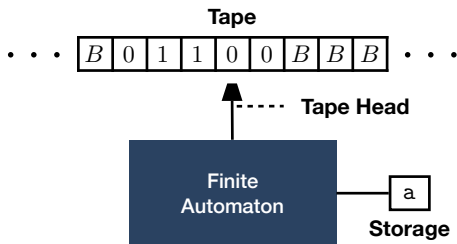
1. Extensions of Turing Machines
    TMs with Storage
    Multi-track TMs
    Multi-tape TMs
    Non-deterministic TMs (NTMs)
    More Extensions of TMs

TMs with Storage

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

Tape

$$\cdots \boxed{B}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{0}\boxed{B}\boxed{B}\boxed{B} \cdots$$

Tape Head

Finite
Automaton
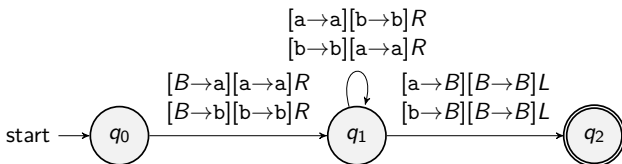
$\boxed{a}$

Storage

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

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

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

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

---

[1] https://plrg.korea.ac.kr/courses/cose215/materials/tm-storage-abn-or-ban.pdf

TMs with Storage are Equivalent to TMs
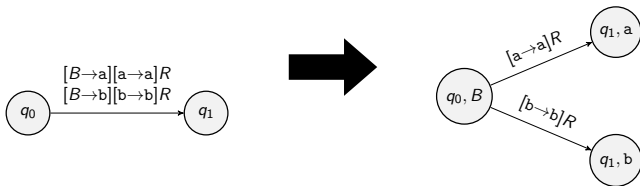
Theorem

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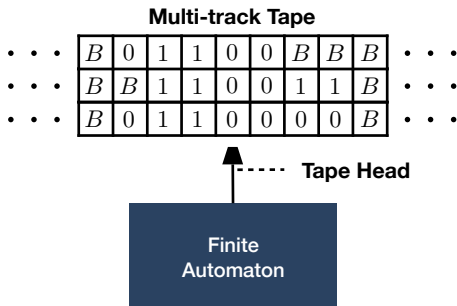
$$\delta'((q, a), b) = \delta(q, a, b)$$

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

Multi-track TMs

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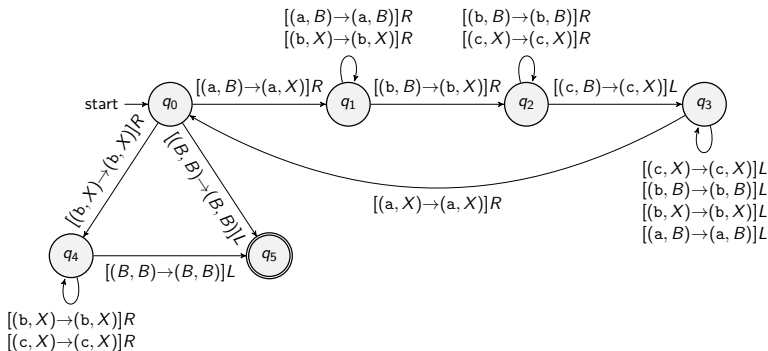
Multi-track Tape

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

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

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

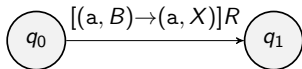The following **multi-track TM** accepts $L(M)$, and see the example for
aabbcc $\in L(M)$.[2]

---

[2] https://plrg.korea.ac.kr/courses/cose215/materials/tm-multi-track-an-bn-cn.pdf

Multi-track TMs are Equivalent to Standard TMs ◆PLRG

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 $\delta' : Q \times \Gamma' \rightharpoonup Q \times \Gamma' \times \{L, R\}$.
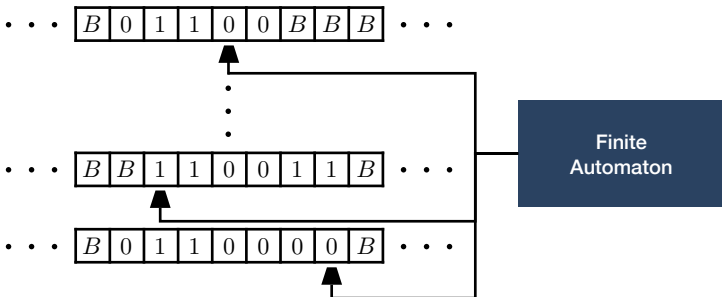For example,

$\cdots$	B	a	b	B	$\cdots$
$\cdots$	B	X	B	B	$\cdots$

➡️

$\cdots$	(B, B)	(a, X)	(b, B)	(B, B)	$\cdots$

Multi-tape TMs

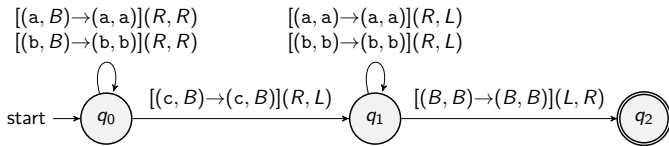We can define a TM with **multiple tapes**:

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

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

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

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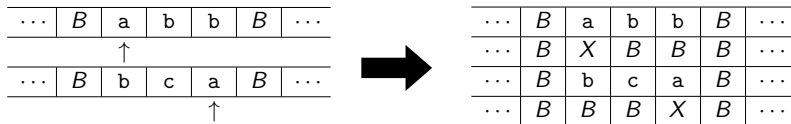abbcbba $\in L(M)$.[3]

---

[3] https://plrg.korea.ac.kr/courses/cose215/materials/tm-multi-tape-w-c-wr.pdf

Multi-tape TMs are Equivalent to Standard TMs   🅰PLRG

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 a storage by using **odd** tracks for the original **tapes** and **even** tracks for the **tape heads**:
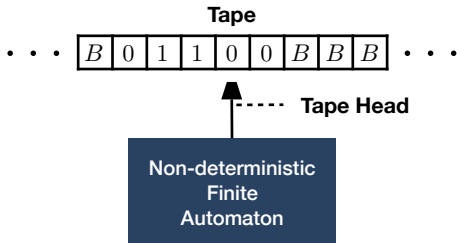
$\cdots$	B	a	b	b	B	$\cdots$
		$\uparrow$				
$\cdots$	B	b	c	a	B	$\cdots$
				$\uparrow$		

➡

$\cdots$	B	a	b	b	B	$\cdots$
$\cdots$	B	X	B	B	B	$\cdots$
$\cdots$	B	b	c	a	B	$\cdots$
$\cdots$	B	B	B	X	B	$\cdots$

We can simulate one-step in the n-tape TM by 1) scanning the tape to store the symbols under the n heads into the storage, and then 2) scanning the tape again to update the symbols and move the heads.

However, it is **inefficient** because we need to scan all the symbols on the tape to simulate a single step in the n-tape TM.

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

Tape

$$\cdots \;\; \boxed{B}\;\boxed{0}\;\boxed{1}\;\boxed{1}\;\boxed{0}\;\boxed{0}\;\boxed{B}\;\boxed{B}\;\boxed{B}\;\; \cdots$$
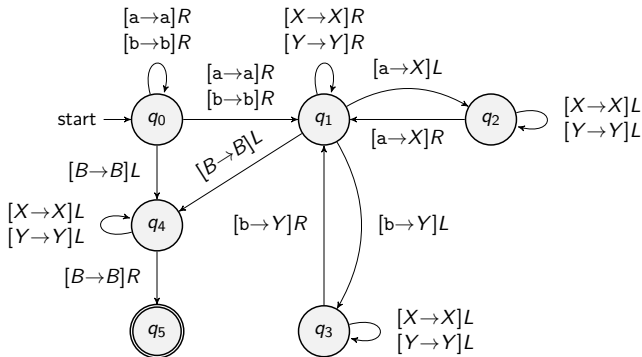
Tape Head

Non-deterministic
Finite
Automaton

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

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

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

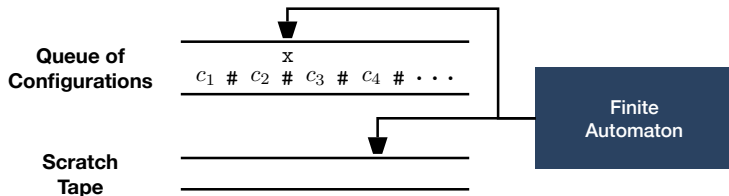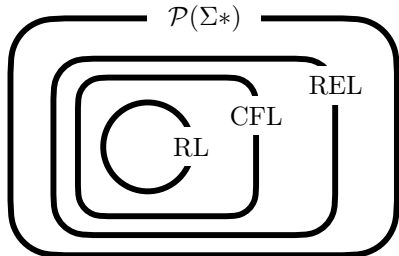The following **nondeterministic TM** accepts $L(M)$, and see the example for abba $\in L(M)$.[4]

Non-deterministic TMs are Equivalent to Standard TMs

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.

Similarly, it is **inefficient** because we need to capture all the configurations of the non-deterministic TM to simulate a single step.

- 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**
  - **Quantum TMs** – TMs with **Quantum States**
  - · · ·

- They are all **equivalent** to TMs.

- A standard **TM** is the **most powerful model of computation**.

$$\mathcal{P}(\Sigma*)$$

REL

CFL

RL

TM **=** ETM

(Turing Machine)   (All Extensions of TMs)

‖

REL

(Recursively Enumerable
Language)

Summary

1. Extensions of Turing Machines
    TMs with Storage
    Multi-track TMs
    Multi-tape TMs
    Non-deterministic TMs (NTMs)
    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. 16 (Mon.).

- Please only submit `Implementation.scala` file to **LMS**.

- The Origin of Computer Science

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