# Lecture 0 – Course Overview

## COSE215: Theory of Computation

Jihyeok Park


2026 Spring

# Course Information

- **Instructor:** Jihyeok Park (박지혁)
  - **Position:** Assistant Professor in CS, Korea University
  - **Expertise:** Programming Languages, Software Analysis
  - **Office hours:** 14:00–16:00, Tuesdays (appointment by e-mail)
  - **Office:** 507, Jung Woonoh IT & General Education Center
  - **Email:** jihyeok_park@korea.ac.kr

# Course Information

- **Instructor:** Jihyeok Park (박지혁)
  - **Position:** Assistant Professor in CS, Korea University
  - **Expertise:** Programming Languages, Software Analysis
  - **Office hours:** 14:00–16:00, Tuesdays (appointment by e-mail)
  - **Office:** 507, Jung Woonoh IT & General Education Center
  - **Email:** jihyeok_park@korea.ac.kr

- **Class:** COSE215 - 02 (English)

# Course Information

- **Instructor:** Jihyeok Park (박지혁)
  - **Position:** Assistant Professor in CS, Korea University
  - **Expertise:** Programming Languages, Software Analysis
  - **Office hours:** 14:00–16:00, Tuesdays (appointment by e-mail)
  - **Office:** 507, Jung Woonoh IT & General Education Center
  - **Email:** jihyeok_park@korea.ac.kr

- **Class:** COSE215 - 02 (English)

- **Homepage:** https://plrg.korea.ac.kr/courses/cose215/

# Course Information

- **Instructor:** Jihyeok Park (박지혁)
  - **Position:** Assistant Professor in CS, Korea University
  - **Expertise:** Programming Languages, Software Analysis
  - **Office hours:** 14:00–16:00, Tuesdays (appointment by e-mail)
  - **Office:** 507, Jung Woonoh IT & General Education Center
  - **Email:** jihyeok_park@korea.ac.kr

- **Class:** COSE215 - 02 (English)

- **Homepage:** https://plrg.korea.ac.kr/courses/cose215/

- **LMS:** https://lms.korea.ac.kr/
  - Please use the **Board** > **Q&A** section for questions.

# Course Information

- **Instructor:** Jihyeok Park (박지혁)
  - **Position:** Assistant Professor in CS, Korea University
  - **Expertise:** Programming Languages, Software Analysis
  - **Office hours:** 14:00–16:00, Tuesdays (appointment by e-mail)
  - **Office:** 507, Jung Woonoh IT & General Education Center
  - **Email:** jihyeok_park@korea.ac.kr

- **Class:** COSE215 - 02 (English)

- **Homepage:** https://plrg.korea.ac.kr/courses/cose215/

- **LMS:** https://lms.korea.ac.kr/
  - Please use the **Board** > **Q&A** section for questions.

- **Teaching Assistant:** cose215@googlegroups.com
  - Hyunjoon Kim (김현준)
  - Minseok Choe (최민석)
  - Sungmin Park (박성민)

# Grading

- **4 Homework Assignments: 20%**
  - Programming assignments in Scala (submission in **LMS**)
  - **Policy on academic integrity**:
    - Do not share your code with others / Do not see others' code.
    - It's your code only if you can explain all the details of the code.
  - If violated, you get a **zero** for the assignment or an **F for the course**.
  - Late submission policy:
    - 1 day late: 20% penalty
    - 2 or more days late: no credit

# Grading

PLRG

- **4 Homework Assignments: 20%**
  - Programming assignments in Scala (submission in **LMS**)
  - **Policy on academic integrity**:
    - Do not share your code with others / Do not see others' code.
    - It's your code only if you can explain all the details of the code.
  - If violated, you get a **zero** for the assignment or an **F for the course**.
  - Late submission policy:
    - 1 day late: 20% penalty
    - 2 or more days late: no credit

- **Midterm exam: 30%**
  - April 22 (Wed.) 13:30 – 14:45 (in class, 75 min.)

# Grading

- **4 Homework Assignments: 20%**
  - Programming assignments in Scala (submission in **LMS**)
  - **Policy on academic integrity**:
    - Do not share your code with others / Do not see others' code.
    - It's your code only if you can explain all the details of the code.
  - If violated, you get a **zero** for the assignment or an **F for the course**.
  - Late submission policy:
    - 1 day late: 20% penalty
    - 2 or more days late: no credit

- **Midterm exam: 30%**
  - April 22 (Wed.) 13:30 – 14:45 (in class, 75 min.)

- **Final exam: 40%**
  - June 17 (Wed.) 13:30 – 14:45 (in class, 75 min.)

# Grading

PLRG

- **4 Homework Assignments: 20%**
  - Programming assignments in Scala (submission in **LMS**)
  - **Policy on academic integrity**:
    - Do not share your code with others / Do not see others' code.
    - It's your code only if you can explain all the details of the code.
  - If violated, you get a **zero** for the assignment or an **F for the course**.
  - Late submission policy:
    - 1 day late: 20% penalty
    - 2 or more days late: no credit
- **Midterm exam: 30%**
  - April 22 (Wed.) 13:30 – 14:45 (in class, 75 min.)
- **Final exam: 40%**
  - June 17 (Wed.) 13:30 – 14:45 (in class, 75 min.)
- **Attendance: 10%**
  - Please use **LMS** to attend the class with the code provided.
  - The first two attendance checks (Mar. 4 and 9) are just for practice.

# Schedule

| Weak | Contents | Weak | Contents |
|---|---|---|---|
| 1 | Basic Concepts | 9 | Pushdown Automata |
| 2 | Deterministic Finite Automata (DFA) | 10 | Deterministic Pushdown Automata |
| 3 | Nondeterministic Finite Automata (NFA) | 11 | Properties of Context-Free Languages |
| 4 | Regular Expressions and Languages | 12 | Turing Machines (TMs) |
| 5 | Properties of Regular Languages | 13 | Extensions of Turing Machines |
| 6 | Context-Free Grammars and Languages | 14 | Undecidability |
| 7 | Parse Trees and Ambiguity | 15 | P, NP, and NP-Completeness |
| 8 | **Midterm Exam (Apr. 22 - Wed.)** | 16 | **Final Exam (Jun. 17 - Wed.)** |

- There will be no offline lectures on following days.
  - May 25 (Mon.) – National Holiday (부처님 오신 날)
  - June 3 (Wed.) – Election Day (지방선거)
- Instead, recorded lecture videos will be uploaded to **LMS**.
- You don't need to check the attendance on these days.

# Course Materials

- **Self-contained lecture notes.**

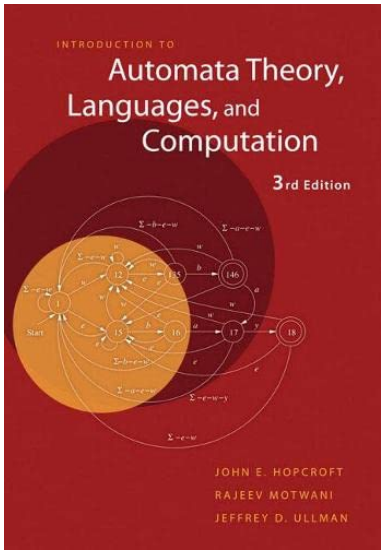
  https://plrg.korea.ac.kr/courses/cose215/

# Course Materials

- **Self-contained lecture notes.**

  https://plrg.korea.ac.kr/courses/cose215/

- Reference:




John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. Introduction to automata theory, languages, and computation. Third edition.

# Goal of This Course

- What is the *mathematical model* of computers?

# Goal of This Course

- What is the *mathematical model* of computers?

**Turing Machine!**

Let's learn **Turing Machine**

# Goal of This Course

PLRG

- What is the *mathematical model* of computers?

**Turing Machine!**

Let's learn **Turing Machine**

- Is it possible to solve *every problem* using computers?

# Goal of This Course

PLRG

- What is the *mathematical model* of computers?

**Turing Machine!**

Let's learn **Turing Machine**

- Is it possible to solve *every problem* using computers?

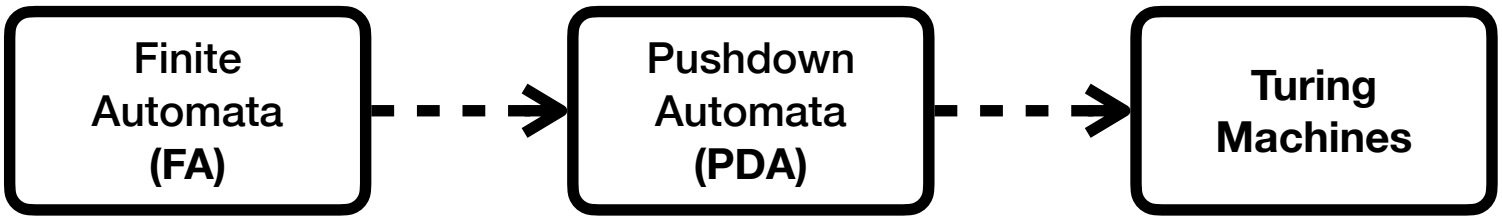
**No!**

Let's learn **Undecidability** and **Intractability**

# Roadmap: Towards Turing Machine

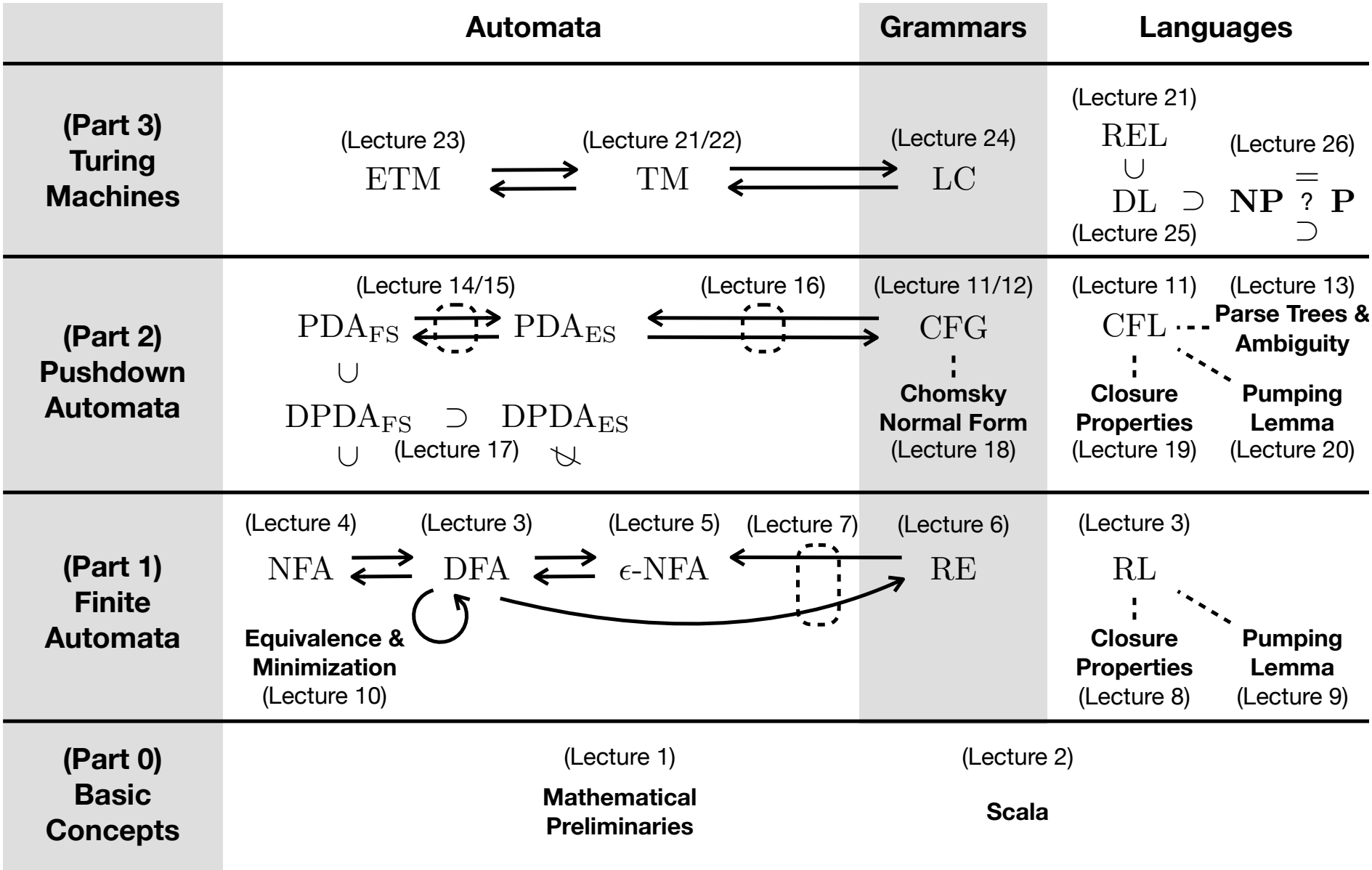
PLRG

A Turing machine is a specific kind of **automaton**.




- **Part 1: Finite Automata (FA)**
  - Regular Expressions (REs)
  - Regular Languages (RLs)
  - Applications: text search, etc.
- **Part 2: Pushdown Automata (PDA)**
  - Context-Free Grammars (CFGs)
  - Context-Free Languages (CFLs)
  - Applications: programming languages, natural language processing, etc.
- **Part 3: Turing Machines (TMs)**
  - Lambda Calculus (LC)
  - Recursively Enumerable Languages (RELs)
  - Undecidability and Intractability

# Roadmap: Towards Turing Machine



# Introduction of Automata

A Turing machine is a specific kind of **automaton**.

# Introduction of Automata

A Turing machine is a specific kind of **automaton**.

Then, what is an **automaton**?

# Introduction of Automata

A Turing machine is a specific kind of **automaton**.

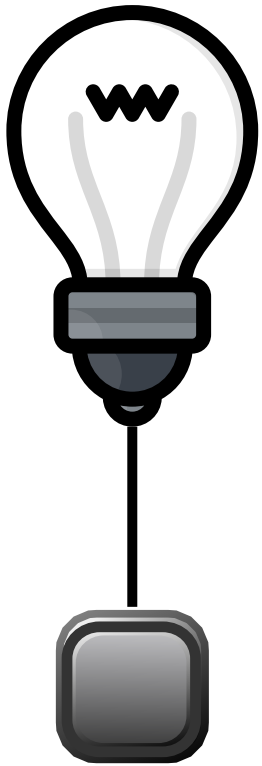
Then, what is an **automaton**? A **state transition system** that takes an **input** and changes its **state** based on the input.

# Introduction of Automata

A Turing machine is a specific kind of **automaton**.

Then, what is an **automaton**? A **state transition system** that takes an **input** and changes its **state** based on the input.

For example,

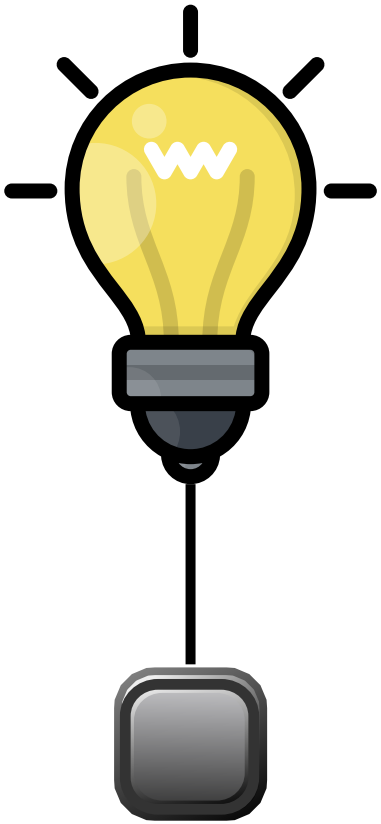


# Introduction of Automata

A Turing machine is a specific kind of **automaton**.

Then, what is an **automaton**? A **state transition system** that takes an **input** and changes its **state** based on the input.

For example,

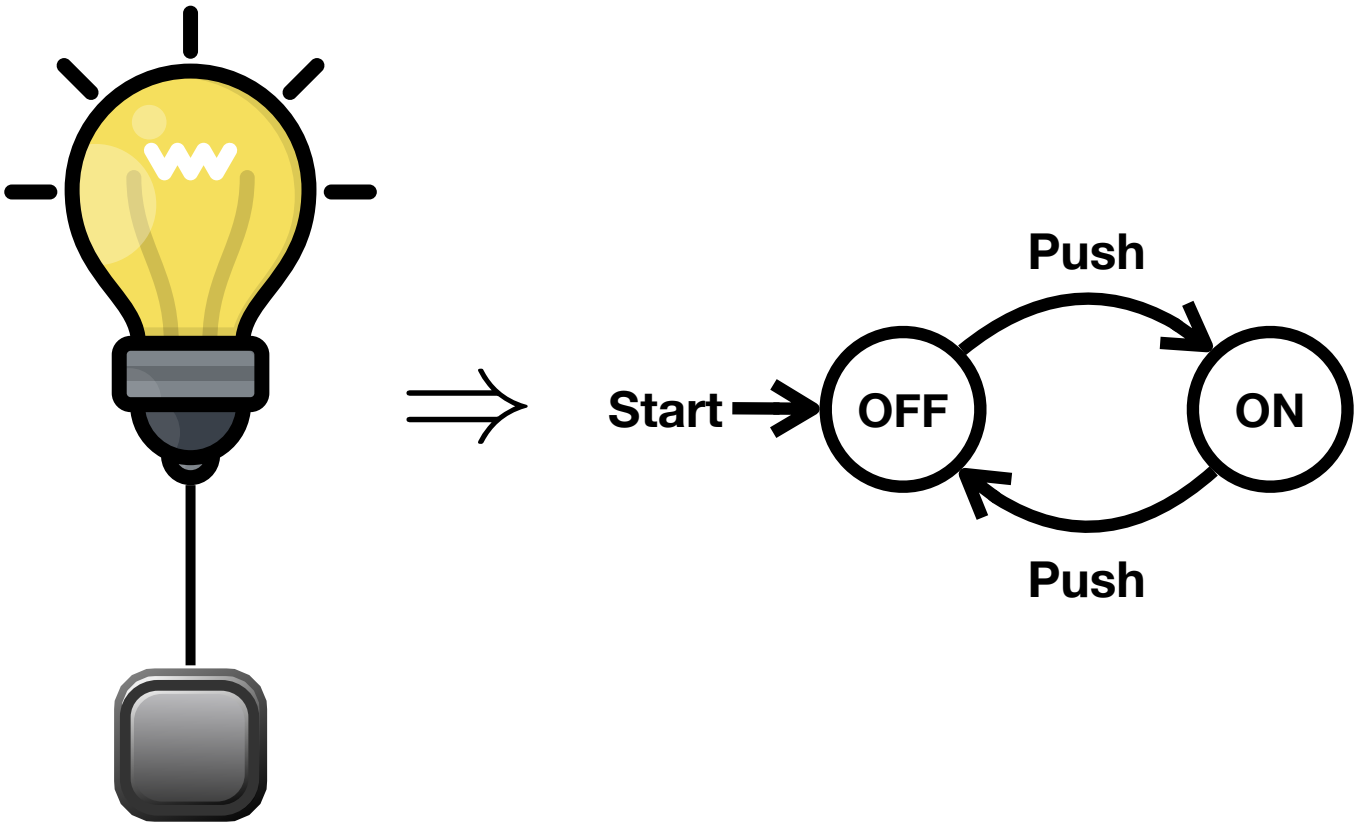


# Introduction of Automata

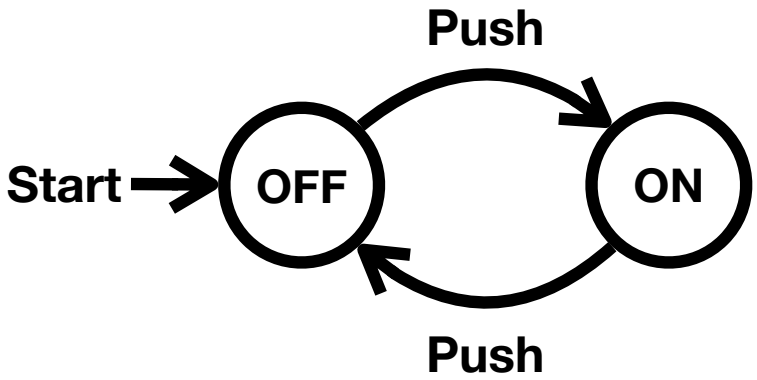
A Turing machine is a specific kind of **automaton**.

Then, what is an **automaton**? A **state transition system** that takes an **input** and changes its **state** based on the input.

For example,



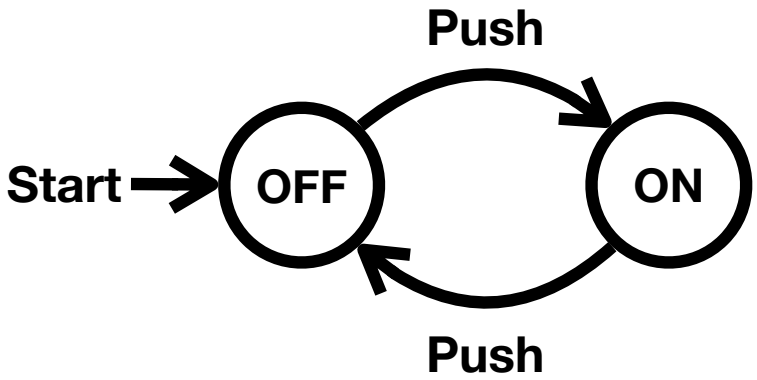
# Introduction of Automata




**Theorem**

*The current state is* OFF *if and only if the button is pushed even times.*

# Introduction of Automata

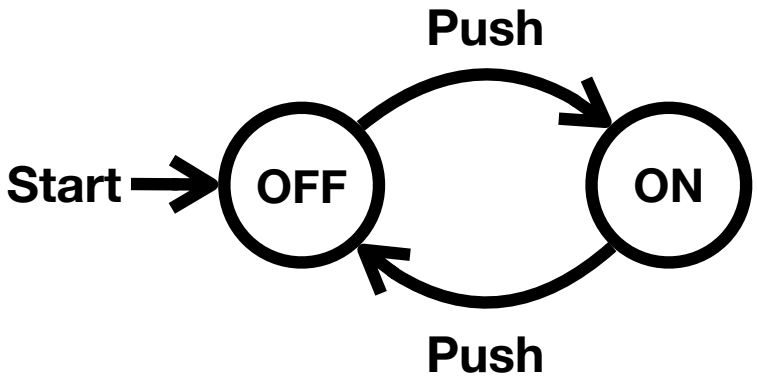



**Theorem**

*The current state is* OFF *if and only if the button is pushed even times.*

- Is it possible to prove it?

# Introduction of Automata



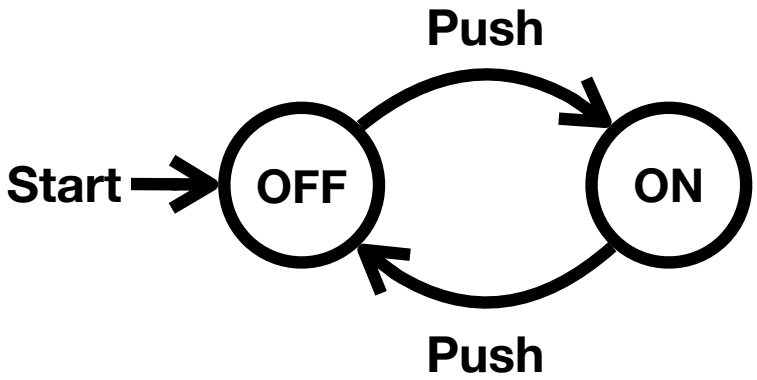

**Theorem**

*The current state is* OFF *if and only if the button is pushed even times.*

- Is it possible to prove it?

  Let's learn **mathematical background and notation.**

# Introduction of Automata




**Theorem**

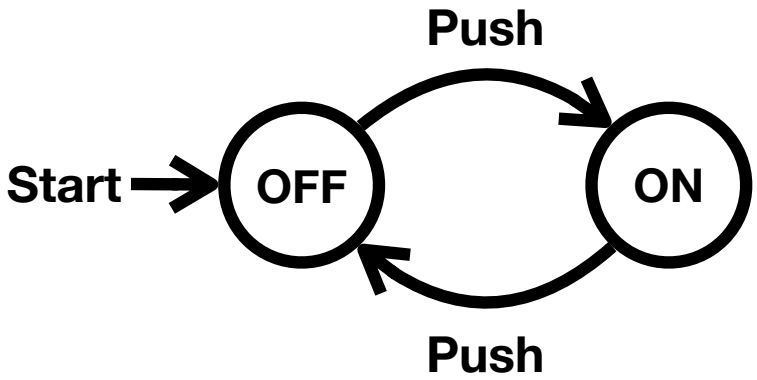
*The current state is* OFF *if and only if the button is pushed even times.*

- Is it possible to prove it?

  Let's learn **mathematical background and notation.**

- Is it possible to implement the automaton?

# Introduction of Automata




**Theorem**

*The current state is* OFF *if and only if the button is pushed even times.*

- Is it possible to prove it?

  Let's learn **mathematical background and notation.**

- Is it possible to implement the automaton?

  Let's learn **Scala** as an implementation language.

# Next Lecture

- Mathematical Preliminaries

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