# Lecture 1 – Mathematical Preliminaries

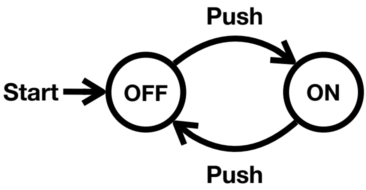
## COSE215: Theory of Computation

Jihyeok Park

PLRG

2023 Spring

# Recall




**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.**

# Contents

# Contents

# Notations in Logics

| Notation | Description |
|---|---|
| $A, B$ | arbitrary **statements**. |
| $P(x)$ | a **predicate** that involves a **variable** $x$. |
| $A \land B$ | the **conjunction** of $A$ and $B$. |
| $A \lor B$ | the **disjunction** of $A$ and $B$. |
| $\neg A$ | the **negation** of $A$. |
| $A \Rightarrow B$ | the **implication** of $A$ and $B$ (i.e., $A$ implies $B$ / if $A$ then $B$). |
| $A \Leftrightarrow B$ | $A$ **if and only if (iff)** $B$ (i.e., $A \Rightarrow B \land B \Rightarrow A$). |
| $\forall x \in X.\ P(x)$ | the **universal quantifier** (i.e, for all $x$ in $X$, $P(x)$ holds). |
| $\exists x \in X.\ P(x)$ | the **existential quantifier** (i.e., there exists $x$ in $X$ such that $P(x)$ holds). |

# Notations in Set Theory

- A **set** is a collection of elements, e.g.,
  - $\mathbb{N} = \{0, 1, 2, \cdots\}$
  - $\{x \in \mathbb{N} \mid x \text{ is even}\} = \{0, 2, 4, 6, 8, 10, 12, \cdots\}$
  - $\{x \in \mathbb{N} \mid x^2\} = \{0, 1, 4, 9, 16, 25, 36, \cdots\}$
- The **empty set** is denoted by $\varnothing$.
- The **cardinality** of a set $X$ is denoted by $|X|$.
- A **subset** $X$ of a set $Y$ is denoted by $X \subseteq Y$.

$$X \subseteq Y \iff \forall x \in X.\ x \in Y$$

- A **proper subset** $X$ of a set $Y$ is denoted by $X \subset Y$.

$$X \subset Y \iff X \subseteq Y \land X \neq Y$$

# Notations in Set Theory

- The **union** of sets

$$\begin{aligned} X \cup Y &= \{x \mid x \in X \vee x \in Y\} \\ \bigcup \mathcal{C} &= X_1 \cup X_1 \cup \cdots \cup X_n = \{x \mid \exists X \in \mathcal{C}.\ x \in X\} \end{aligned}$$

  where $\mathcal{C} = \{X_1, X_2, \cdots, X_n\}$.

- The **intersection** of sets

$$\begin{aligned} X \cap Y &= \{x \mid x \in X \wedge x \in Y\} \\ \bigcap \mathcal{C} &= X_1 \cap X_1 \cap \cdots \cap X_n = \{x \mid \forall X \in \mathcal{C}.\ x \in X\} \end{aligned}$$

  where $\mathcal{C} = \{X_1, X_2, \cdots, X_n\}$.

- The **difference** of sets

$$X \setminus Y = \{x \mid x \in X \wedge x \notin Y\}$$

## Notations in Set Theory

- The **complement** of a set $X$ is denoted by $\overline{X}$.

$$\overline{X} = \{x \mid x \in U \land x \notin X\}$$

  where $U$ is the **universal set**.

- The **power set** of a set $X$ is denoted by $2^X$ or $\mathcal{P}(X)$.

$$2^X = \mathcal{P}(X) = \{Y \mid Y \subseteq X\}$$

- The **Cartesian product** of sets $X$ and $Y$ is denoted by $X \times Y$.

$$X \times Y = \{(x, y) \mid x \in X \land y \in Y\}$$

# Contents

## Inductions on Integers

**Definition (Inductions on Integers)**

Let $P(n)$ be a predicate on integers, and if

- **(Basis Case)** $P(k)$ is hold where $k$ is an integer, and
- **(Induction Case)** for all $n \geq k$, $P(n) \Rightarrow P(n+1)$,

then $P(i)$ is hold for all $i \geq k$.

$P(n)$ is called **induction hypothesis**.

# Inductions on Integers

## Example

Prove that $\forall n \geq 0.\ \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

**Proof)**

# Inductions on Integers

## Example (Exercise)

Prove that $\forall n \geq 0.\ \sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$.

**Proof)**

# Structural Inductions

In CS, we often define somethings as **inductively-defined sets**.

## Example (Inductive Definition of Trees)

A **tree** is defined as follows:

- **(Basis Case)** A single **node** $N$ is a tree.
- **(Induction Case)** If $T_1, \cdots, T_n$ are trees, then a graph defined with a new node $N$ and edges from $N$ to $T_1, \cdots, T_n$ is a tree as well.

## Example (Inductive Definition of Arithmetic Expressions)

An **arithmetic expression** is defined as follows:

- **(Basis Case)** A **number** or a **variable** is an arithmetic expression.
- **(Induction Case)** If $E$ and $F$ are arithmetic expressions, then so are $E + F$, $E \times F$, and $(E)$.

# Structural Inductions

### Definition (Structural Inductions)

Let $P(x)$ be a predicate on a **inductively-defined set** $X$, and if

- **(Basis Case)** $P(x)$ is hold for all base cases $x$, and
- **(Induction Case)** for all $x \in X$,

$$P(x_1) \land \cdots \land P(x_n) \Rightarrow P(x)$$

  where $x_1, \cdots, x_n$ are the **sub-structures** of $x$.

then $P(x)$ is hold for all $x \in X$.

$P(x_1), \cdots, P(x_n)$ are called **induction hypotheses**.

## Example

Prove that for all tree $T$, the number of nodes in $T$ is equal to the number of edges in $T$ plus one.

**Proof)**

### Example (Exercise)

Prove that for all arithmetic expression $E$, the number of left parentheses in $E$ is equal to the number of right parentheses in $E$.

**Proof)**

# Mutual Inductions

## Definition (Mutual Inductions)

Let $P(x)$ and $Q(x)$ are predicates on a **inductively-defined set** $X$, and if

- **(Basis Case)** $P(x)$ and $Q(x)$ are hold for all base cases $x$, and
- **(Induction Case)** for all $x \in X$,

$$P(x_1) \wedge \cdots \wedge P(x_n) \wedge Q(x_1) \wedge \cdots \wedge Q(x_n) \Rightarrow P(x) \wedge Q(x)$$

  where $x_1, \cdots, x_n$ are the **sub-structures** of $x$.

then $P(x)$ and $Q(x)$ are hold for all $x \in X$.

$P(x_1), \cdots, P(x_n)$ and $Q(x_1), \cdots, Q(x_n)$ are called **induction hypotheses**.

# Mutual Inductions

**Theorem**

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

**Proof)**

# Contents

# Symbols

We first define a finite and non-empty set of **symbols**.

- $\Sigma = \{0, 1\}$ – binary symbols.
- $\Sigma = \{\mathsf{a}, \mathsf{b}, \cdots, \mathsf{z}\}$ – lowercase letters.
- $\Sigma = \{a \mid a \text{ is an ASCII character}\}$ – ASCII characters.

# Words

A **word** $w \in \Sigma^*$ is a sequence of symbols:

- $\Sigma = \{0, 1\}$ – $\epsilon, 0, 1, 00, 01, 10010, \cdots$
- $\Sigma = \{a, b, \cdots, z\}$ – $\epsilon$, a, b, abc, hello, cs, students, $\cdots$

Notations:

| Notation | Description |
|---|---|
| $\epsilon$ | the **empty word**. |
| $w_1 w_2$ | the **concatenation** of $w_1$ and $w_2$.<br>($w_1$ is a **prefix** of $w_1 w_2$ and $w_2$ is a **suffix** of $w_1 w_2$) |
| $w^R$ | the **reverse** of $w$. |
| $\|w\|$ | the **length** of $w$. |
| $\Sigma^k$ | the set of all words of length $k$. |
| $\Sigma^*$ | the set of all words (the **Kleene star**).<br>(i.e., $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \cdots = \bigcup_{k=0} \Sigma^k$) |
| $\Sigma^+$ | the set of all words except $\epsilon$ (the **Kleene plus**).<br>(i.e., $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \cdots = \bigcup_{k=1} \Sigma^k$) |

# Languages

A **language** $L \subseteq \Sigma^*$ is a set of words. When $\Sigma = \{0, 1\}$, we can define the following languages:

- $L = \{\epsilon, 0, 1\}$ – the empty word, zero, and one.
- $L = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \cdots\}$ – all binary words.
- $L = \{0^n 1^n \mid n \geq 0\}$ – equal number of consecutive zeros and ones.
- $L = \{10, 11, 101, 111, 1011, \cdots\}$ – prime numbers in a binary format.

# Languages

- The **union**, **intersection**, and **difference** of languages:

$$L_1 \cup L_2 \qquad L_1 \cap L_2 \qquad L_1 \setminus L_2$$

- The **reverse** of a language:

$$L^R = \{w^R \mid w \in L\}$$

- The **complement** of a language:

$$\overline{L} = \Sigma^* \setminus L$$

- The **concatenation** of languages:

$$L_1 L_2 = \{w_1 w_2 \mid w_1 \in L_1 \land w_2 \in L_2\}$$

# Languages

- The **power** of a language:

$$
\begin{aligned}
L^0 &= \{\epsilon\} \\
L^n &= L^{n-1}L \qquad (n \geq 1)
\end{aligned}
$$

- The **Kleene star** of a language:

$$L^* = L^0 \cup L^1 \cup L^2 \cup \cdots = \bigcup_{n \geq 0} L^n$$

- The **Kleene plus** of a language:

$$L^+ = L^1 \cup L^2 \cup L^3 \cup \cdots = \bigcup_{n \geq 1} L^n$$

# Summary

1. Mathematical Notations
   - Notations in Logics
   - Notations in Set Theory
2. Inductive Proofs
   - Inductions on Integers
   - Structural Inductions
   - Mutual Inductions
3. Notations in Languages
   - Symbols
   - Words
   - Languages

# Next Lecture

- Basic Introduction of Scala

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