Chap. 1
Introduction to
the theory of computation 계산 이론.

Agenda of Chapter 1

- Main ideas from math
  - set theory, functions, and relations
  - trees and graph
  - basic proof techniques
    - deduction, induction, and by contradiction
- Central concept of languages, grammars, and automata
- Simple applications in computer science

Sets (1/2)

- Set
  - a collection of elements
  - no structure other than membership (순서 x)
- Notations
  - Set of three elements 1, 2, 3 : A= $\{1, 2, 3\}$
  - Set of all alphabets : B= $\{a, b, c, \dots, z\}$
  - Set of even integers : C= $\{0, 2, 4, 6, \dots, -2, -4, -6\}$ or $\{2n \mid n \in \mathbb{Z}\}$
  - 2 is a member of A : $2 \in A$
- Operations
  - Union : $S_1 \cup S_2 = \{x \mid x \in S_1 \vee x \in S_2\}$
  - Intersection : $S_1 \cap S_2 = \{x \mid x \in S_1 \wedge x \in S_2\}$
  - Difference : $S_1 - S_2 = \{x \mid x \in S_1 \wedge x \notin S_2\}$
  - Complementation : $\overline{S} = S^c = S' = \{x \mid x \notin S, x \in U\}$
  - Cartesian product : $S_1 \times S_2 = \{(x_1, x_2) \mid x_1 \in S_1, x_2 \in S_2\}$

ordered pair

Sets (2/2)

- Special sets
  - Universal Set U : set of all possible elements
  - Empty set (Null set) ∅ : set with no elements $\{\ \} = \emptyset$
- Relations between sets
  - 부분집합 Subset, Proper subset 진부분집합
    - ① $S_1 \subset S_2$: $\forall x \in S_1 \wedge x \in S_2$
    - ② $S_1 \subsetneq S_2$: $[\forall x \in S_1 \wedge x \in S_2] \wedge [\exists y \in S_2 \wedge y \notin S_1]$

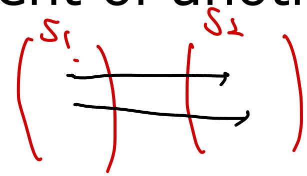
  - 서로소 Disjoint
    - $S_1 \cap S_2 = \emptyset$
- Power set 2^S $= P(S)$
  - S의 부분집합들의 집합. $S = \{1, 2\}$
  - $2^S = \{A \mid A \subset S\}$ $\quad 2^S = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$
  - $|S| = n \quad |2^S| = 2^n$
- Related rules
  - Operations with empty sets
  - Associative $(A \cap B) \cap C = A \cap (B \cap C)$
  - Distributive $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
  - DeMorgan's Law $(A \cup B)^c = A^c \cap B^c$

Functions and Relations (1/2)

- Function
  - an element of a set → a unique element of another set
  - $f : S_1 \rightarrow S_2$

    $\forall x \in S_1,\ \exists! y \in S_2,\ f(x)=y$

    unique 하게 존재한다.

    $\uparrow$ domain $\uparrow$ range

    $F\{(x_1,y_1) \ldots\ldots\}$

    $\subset \underbrace{S_1 \times S_2}$

    cartesian product's subset

  - 모든 원소가 정의 / 모든 f.

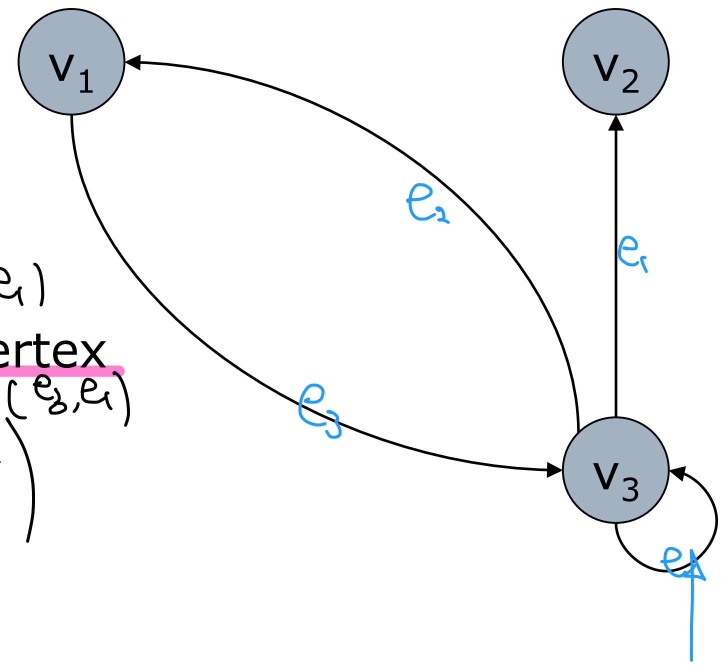
    Total function & partial function
- Relations : Arbitrary subclass of SxS
  - Set of pairs : $\{(x_1,y_1), (x_2,y_2), \ldots\}$
  - A generalization of function
  - 동치 클래스.

    Equivalence relation $x \equiv y$
    - a generalization of identity
    - reflexivity
    - symmetry
    - transitivity

Graphs and Trees (1/2)

- A graph
  - Consists of two finite sets; vertices & edges
  - Vertices $V=\{v_1,v_2,\ldots,v_n\}$
  - Edges $E=\{e_1,e_2,\ldots,e_n\}$, $e_i=(v_j, v_k)$

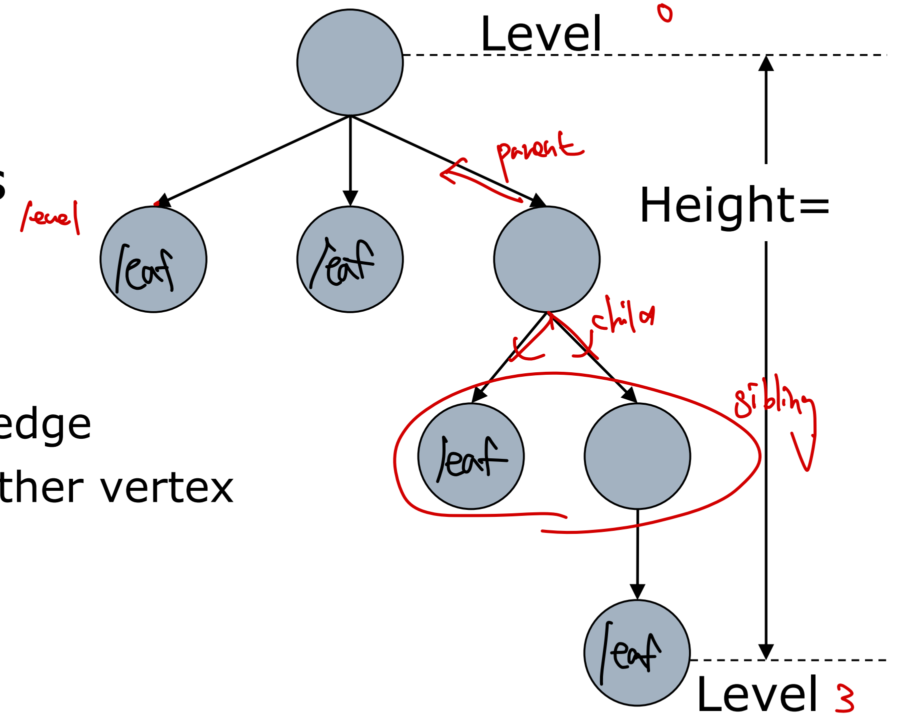
- Definitions for graph
  - walk : sequence of edge $v_1 \rightarrow v_2$ (e_3,e_2,e_3,e_4,e_1)
  - path : walk with no repeated edge (e_3,e_4,e_1)
  - simple path : path with no repeated vertex (e_3,e_1)
  - cycle : walk from a vertex to itself $(e_3,e_2\ldots)$
  - base : starting/ending point of cycle v_1
  - loop : edge from a vertex to itself e_4



Graphs and Trees (2/2)

A Tree

- A particular type of graph
- Directed graph with no cycles



Definitions

- root : a vertex with only outgoing edge
  only one path from root to every other vertex
- leaves
  Vertices without outgoing edges
- Parent, child, siblings
- level of vertex
  number of edges from root
- height of tree
  largest level of all vertices 3

Proof Techniques (1/3)

☐ Principle of Mathematical Induction

기준점

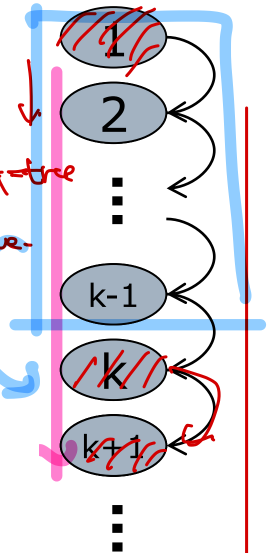
1. For some $n_0 \geq 1$, P_{n_0} is true. → 시작이 참이다.
2. For any $k \geq n_0$, truth of P_k ➔ truth of P_{k+1}.

If 1 & 2 are satisfied, then all P_1, P_2, … are true.

$n_0 = 1$

$k \geq 1$, $P_k = true$

$P_{k+1} = true$



☐ Principle of Strong Mathematical Induction

필요한 개수 만큼

1. For some $n_0 \geq 1$, $P_1, P_2, \ldots, P_{n_0}$ are true.
2. For any $k \geq n_0$, truth of $P_1, P_2, \ldots, P_k$ ➔ truth of P_{k+1}.

If 1 & 2 are satisfied, then all P_1, P_2, … are true.

일반 경우

반드시 가정

(가정이 강하다)

☐ Proof by Induction for **P_k is true for all k=1,2,…**

[basis] Show that P_1 is true.

[inductive assumption] Assume that $(P_1, P_2, \ldots,)$ P_k is true.

[inductive step] Show that $(P_1, P_2, \ldots,)$ P_k is true ➔ P_{k+1} is true

NOTE] Induction & recursive definition of functions

Proof Techniques (2/3)

$1 \sim 10 \sim 100 \cdots \infty$

$1+2+\cdots+n = \frac{(n+1)n}{2}$

Ex) Sum of all positive integers not larger than n is n(n+1)/2

S_n: $\left(\sum_{k=1}^{n} k = \frac{(n+1)n}{2} \right)$

[basis] $n=1$ 일때 $1 = \frac{(1+1)\cdot 1}{2} = 1$ (true)

[inductive assumption] $n=k$ 일때 $1+2+\cdots+k = \frac{k(k+1)}{2}$ 가정

[inductive step]

$n=k+1$: $1+2+\cdots+k+k+1 = \frac{k(k+1)}{2} + k+1 = \frac{k^2+3k+2}{2} = \frac{(k+1)(k+2)}{2}$

S_n is true for all $n \in \mathbb{Z}$

Ex) Any integer greater than 1 can be written as the product of one or more primes. ∵ Strong

[basis]

[inductive assumption]

[inductive step]

Proof Techniques (3/3)

- Proof by contradiction
  - To prove **P is true**,
    1. Assume P is false
    2. See where the assumption lead us.
    3. Arriving at a wrong conclusion, deny the assumption.

EX] $\sqrt{2}$ is irrational

Assume $\sqrt{2}$ is not rational

$\sqrt{2} = \frac{Q}{P}$ $(P, Q \in \mathbb{Z},\ P, Q$ no common factor$)$ $P \neq 0$

$2P^2 = Q^2$

$\Rightarrow$ Q^2는 짝수이다 $Q = 2k$.

Q는 짝수이다.

$2P^2 = 4k^2$

$P^2 = 2k^2$

(P도 짝수)

모순

$\sqrt{2}$ is irrational

Languages (1/3)

- Language, an informal definition
  - A system for the expression of ideas, facts, or concepts
  - a set of symbols + rules for manipulation

- Basic elements
  - Alphabet : a finite, nonempty set Σ of symbols $(a,b,c,\ldots)$ $\Sigma = \{a,b,c\}$
  - String : finite sequence of symbols $(u,v,w,\ldots)$

    e.g.) $u =$ abc. $v =$ aaa $w =$ abab
  - Operations on strings
    - Concatenation = 두개를 붙인다 $uv =$ abcaaa.
    - Reverse = 뒤집다. $w^R =$ baba
    - Length = 길이 $|w| = 4$ (symbol 의 숫자.)

Prefix

aabcbc suffix

substring

Languages (2/3)

- More definitions
  - empty string λ : a string with no symbols $|\lambda| = 0$
  - substring, prefix, suffix
  - Σ^* : set of strings obtained by concatenating zero or more symbols in **Σ**
  - λ를 뺀 값 $\Sigma^+ : \Sigma^* - \{\lambda\}$
- Language L : a subset of Σ^*. → Σ^* 중에 내가 원하는 것만 뽑아서 만든거.
  - Sentence : a string(element) in a language L

EX1.9]

- $\Sigma = \{a,b\}$, Σ^* ∞개. $= \{ \lambda, a, b, aa, ab, ba, bb, aaa, \dots \}$
- subset of Σ^*. $\{a,aa,aab\}$: a finite language on Σ =같다.
- $L=\{ a^n b^n \mid n \geq 0 \}$: an infinite language on Σ

~~subset of~~ $\{ \lambda, ab, aabb, aaabbb \dots \}$ = 규칙이 주어진 경우.

Languages (3/3)

L에는 포함X
Σ^*에는 포함.

- Operations on Language

집합의 연산이 모두 가능해진다.

  - Complement =여집합 $\bar{L} = \{ w \mid w \notin L,\ w \in \Sigma^* \}$
  - Reverse $L^R = \{ w^R \mid w \in L \}$
  - Concatenation $L_1 L_2 = \{ w_1 w_2 \mid w_1 \in L_1,\ w_2 \in L_2 \}$

새로운 string의 집합
확장 가능성↑

  - L^n : L concatenated with itself n times

순서대로 하는 것이 아님

$LL \cdots L = \{ w_1 w_2 \cdots w_n \mid w_i \in L,\ i = 1, 2, \cdots n \}$

  - Star-closure $L^* = L^0 \cup L \cup L^2 \cup L^3 \cdots = \bigcup_{i=0}^{\infty} L^i$
  - Positive closure L^+ - L^*에서 L^0을 뺀 값. $= \bigcup_{i=1}^{\infty} L^i$

$L^+ = L^* - \{\lambda\}$

EX1.10]

- $L = \{ a^n b^n \mid n \geq 0 \}$
- $L^2 = LL = \{ a^n b^n a^m b^m \mid n \geq 0,\ m \geq 0 \}$

* $LL \neq \{ a^n b^n a^n b^n \mid n \geq 0 \}$

- $L^R = \{ b^n a^n \mid n \geq 0 \}$

$L = \{ \lambda,\ ab,\ aabb,\ aaabbb \ \ldots \}$

L^n

$L^2 = \{ \lambda,\ ab,\ abab,\ aabbab \ \ldots \}$

Grammars (1/6)

= Language를 정의하는 수단.

Grammar

- a language-definition mechanisms

A typical rule of English grammar

명사구 서술어

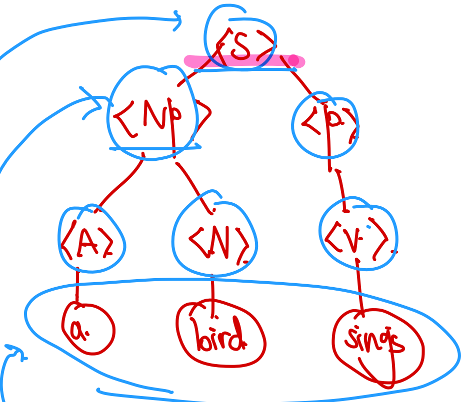
<Sentence> → *<noun_phrase> <predicate>*

관사 명사

<noun_phrase> → *<article><noun>*

동사.

<predicate> → *<verb>*



4가지 요소로 정의가 됨.

Definition of **Grammar** *G*

- G defined as a quadruple $G=(V,T,S,P)$
- ① V : variable, a finite set of objects ※. 명사 앞에는 많은 것이 들어갈 수 있는 변수와 같은 것.
- ② T : terminal symbols, a finite set of objects ※. 더 쪼개지지 않는 상태.
- ③ $S \in V$: start variable, a special symbol 시작점 <S>이 start variable이 됨
- ④ P : a finite set of production 규칙 다양한 나누어 지는 규칙이 있을 수 있기 때문. (문장을 생성하는 규칙 (P ⊂ production)).
- assumption) V and T are non-empty and disjoint

(V and T are non-empty and disjoint)

Grammars (2/6)

- Production rule
  - Specify how the grammar transforms one string into another
  - Define a language associated with the grammar
  - $x \rightarrow y$ 규칙
  - $x \in (V \cup T)^+, \quad y \in (V \cup T)^*$

x자리에는 λ가 안됨.

Variable 과 Terminal 을 막 붙여서 만든 것

$\langle S \rangle \rightarrow \langle NP \rangle \langle P \rangle$

★ 화살표 1개와 2개는 다름

유도

- Derivation of string
  - $w \Rightarrow z$: w derives z (z is derived from w) 변환된다
  - $\overset{*}{\Rightarrow}$

$\langle S \rangle \Rightarrow \langle NP \rangle \langle P \rangle \Rightarrow \langle a \rangle \langle N \rangle \langle P \rangle \Rightarrow \langle a \rangle \langle N \rangle \langle V \rangle \Rightarrow a \langle N \rangle \langle V \rangle \overset{*}{\Rightarrow} a$ bird sings.

- Language defined (generated) by the grammar
  - set of all terminal string generated by production rule
  - Let $G = (V, T, S, P)$
  - Language $L(G) = \{ w \mid S \overset{*}{\Rightarrow} w, w \in T^* \}$

set $L(G) = \{ w \mid S \overset{*}{\Rightarrow} w, w \in T^* \}$

Grammer 에 의해 생성되는 Language.

Grammars (3/6)

- A derivation of the sentence w & sentential form
  - When $w \in L(G)$, $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots\cdots w_n \Rightarrow w \in L(G)$

    sentential form   sentence

- Equivalence of two grammars
  - G_1 and G_2 are equivalent
    - G_1 and G_2 generate the same language
  - $L(G_1) = L(G_2)$

$L(G) = \{ w \mid S \overset{*}{\Rightarrow} w,\ w \in T^* \}$

EX1.11]

V T

- $G = (\{S\},\{a,b\},S,P),\ \ P:\ S \rightarrow aSb,\ \ S \rightarrow \lambda$
- $L(G) = \{ a^n b^n \mid n \geq 0 \}$

$S \Rightarrow aSb \Rightarrow ab$

$\Rightarrow aaSbb \rightarrow aabb$

$\Rightarrow aaaSbbb \rightarrow aaabbb$

EX1.12] Find a grammar generating $L = \{ a^n b^{n+1} \mid n \geq 0 \}$

$S \rightarrow aSb$

$S \rightarrow b$

production

$= S \rightarrow aSb \mid b$

$G = (\{S\},\{a,b\},S,P)$

Grammars (4/6)

language가 G에 의해 생성되었는지 how?

- To show that a language L is generated by G
  - (a) every $w \in L$ can be derived from S using G
  - (b) every string derived using G is in L

두가지 조건

모든 $w \in L$ 은 S로부터 유도되어야 함.

EX1.11 revisited] $G = (\{S\},\{a,b\},S,P),\ P:\ S \rightarrow aSb,\ S \rightarrow \lambda$

무한히 많은 string

– $L(G) = \{a^n b^n \mid n \geq 0\}$ ← 수학적 개념

$a^0b^0 \quad a^1b^1$

$S \rightarrow aSb$, a^nb^n

(a) $\forall w \in L,\ S \overset{*}{\Rightarrow} w$

[basis] $n=0$, $w = a^0b^0 = \lambda$ $\quad S \Rightarrow \lambda$

$n=1$, $w = a^1b^1$, $\quad S \Rightarrow aSb \Rightarrow ab$) 둘다 성립.

[assumption] $n=k$ 일때 $w = a^kb^k$. $S \overset{*}{\Rightarrow} a^kSb^k \Rightarrow a^kb^k$ 임을 가정

[Inductive step] $n=k+1$ 일때 $w = a^{k+1}b^{k+1}$ (가정에 의해) $S \overset{*}{\Rightarrow} a^kSb^k \Rightarrow a^k aSb\, b \Rightarrow a^{k+1}Sb^{k+1} \Rightarrow a^{k+1}b^{k+1}$ 로 성립.

(b) $\forall w,\ \text{if } S \overset{*}{\Rightarrow} w \text{ then } w \in L$

terminal string 전체를 대상으로 함.

Let n: # of derivation step.

[basis] $n=1$ 일때, $S \Rightarrow \lambda = a^0b^0 \in L$, $n=2$ 일때 $S \Rightarrow aSb \Rightarrow ab \in L$ 로 성립.

[assumption] $n=k$ 일때, 모든 가능한 derivation: $S \overset{k-1}{\Rightarrow} a^{k-1}Sb^{k-1} \Rightarrow a^{k-1}b^{k-1} \in L$ 임을 가정

[Inductive step] $n=k+1$ 일때. 가능한 derivation 들은 다 $S \Rightarrow a^{k-1}Sb^{k-1}$ (그런데) $\Rightarrow a^{k-1}aSb^{k-1}b \Rightarrow a^kb^k \in L$ 로 성립.

Grammars (5/6)

주어진 G → 맞는 L

production 들

EX1.13] Σ= $\{a,b\}$, Grammar G : $S \to SS,\ S \to \lambda\ \ S \to aSb,\ S \to bSa$

↓ 대칭

– $L(G) = \{a^n b^n, b^n a^n, a^n b^n a^m b^m,$
$a^n b^n b^m a^m, b^n a^n a^m b^m, b^n a^n b^m a^m$
$\cdots a^n b^n b^m a^m a^k b^k \cdots\cdots$

$b^n a^n$

$S \Rightarrow SS \Rightarrow a^n b^n a^m b^m.$

SS
$\Rightarrow SSS$
$\Rightarrow SSSS$

핵심 성질을 파악할 필요가 있다. = "a와 b의 갯수가 같아진다."

" $\{w \mid n_a(w) = n_b(w)\}.$

(a) $\forall w \in L \quad S \overset{*}{\Rightarrow} w$ let $n = n_a(w) = n_b(w)$

$n=0 \quad a^0 b^0$
$n=1. \quad ab, ba.$
$n=k \ ? \cdots$

교과서

(b) $\forall w.$ if $S \overset{*}{\Rightarrow} w.$ then $w \in L.$

terminal는 derivation 할 때 항상 a,b가 같음

자명하다.

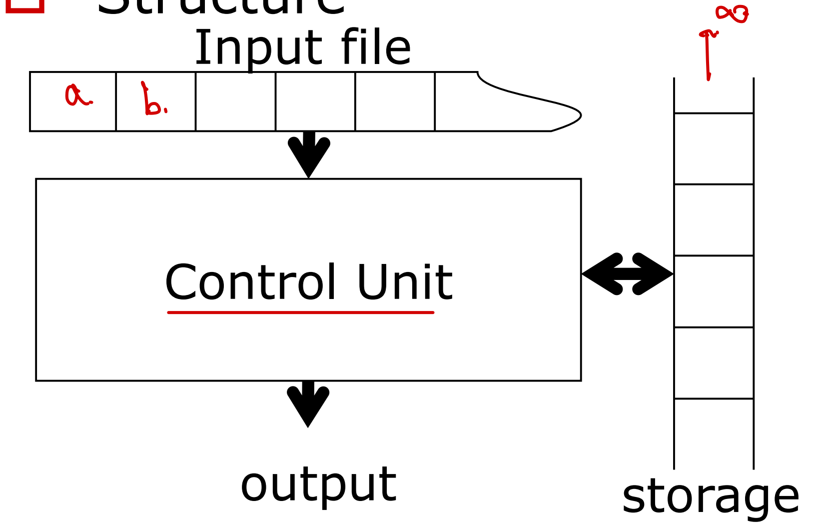
Grammars (6/6)

EX1.14] $G=(\{A,S\}, \{a,b\},S,P_1)$, $P_1 : S \rightarrow aAb \mid \lambda$, $A \rightarrow aAb \mid \lambda$

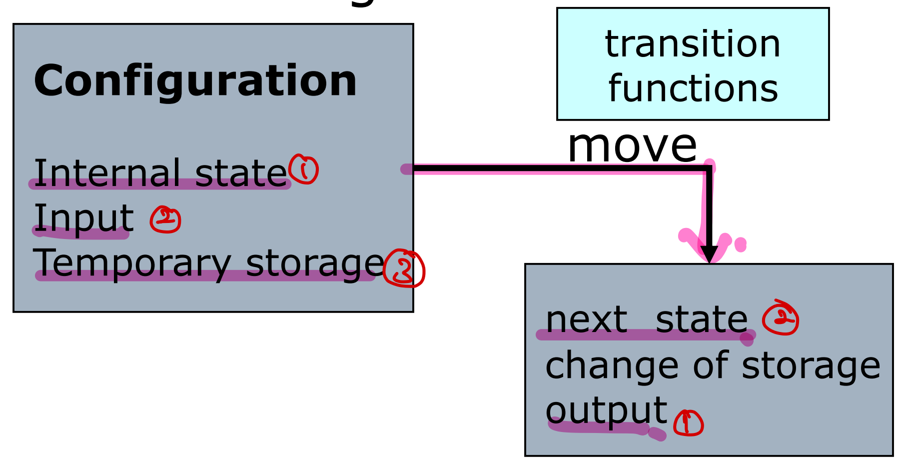
- $L(G) =$

Automata (1/2)

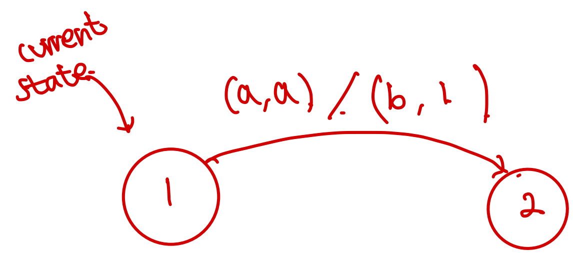
- Structure



- Processing



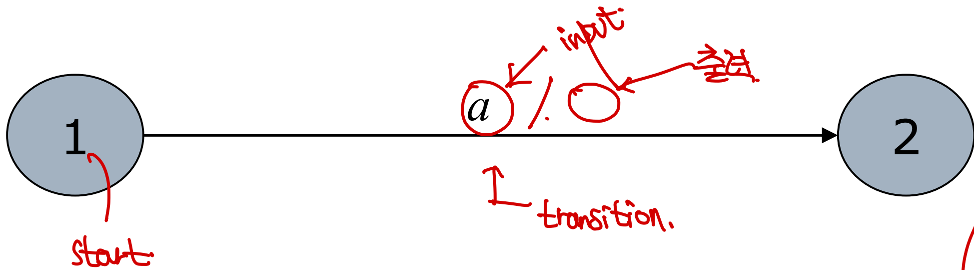
- Input file
  - automata can read but not change
  - one symbol for one cell, one cell at a time

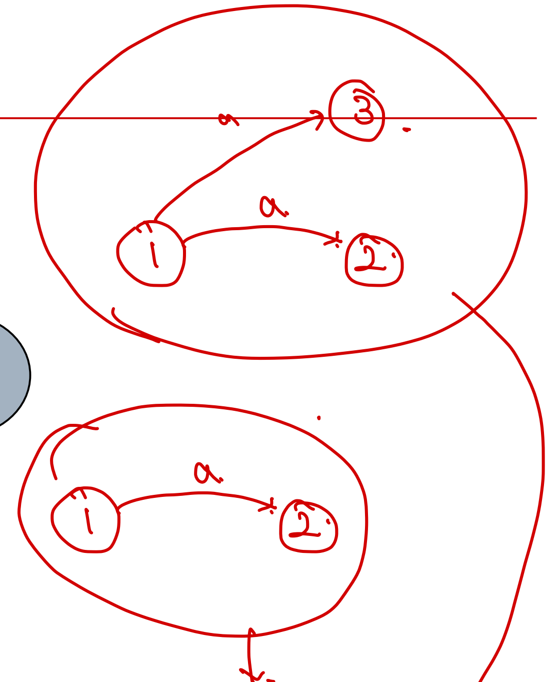


- Storage
  - unlimited number of cells
  - automata can read and change
- Control unit
  - finite number of internal state
  - change a state to another state at a time

Automata (2/2)

Automata as a graph





Types of Automata

- 결정적
- Deterministic vs. Nondeterministic
  - deterministic automata : each move is uniquely defined
  - Nondeterministic automata : several possible moves
- 출력 Accepter vs. Transducer
  - Accepter : output is yes/no
  - Transducer : strings of symbols of output

유한한 symbol

Example of Language and Accepter

- Identifiers in a programming language (변수의 이름)

$$\langle id\rangle \rightarrow \langle letter\rangle\langle rest\rangle$$
$$\langle rest\rangle \rightarrow \langle letter\rangle\langle rest\rangle \mid \langle digit\rangle\langle rest\rangle \mid \lambda$$
$$\langle letter\rangle \rightarrow a \mid b \mid \ldots \mid z$$
$$\langle digit\rangle \rightarrow 0 \mid 1 \mid \ldots \mid 9$$

Grammar

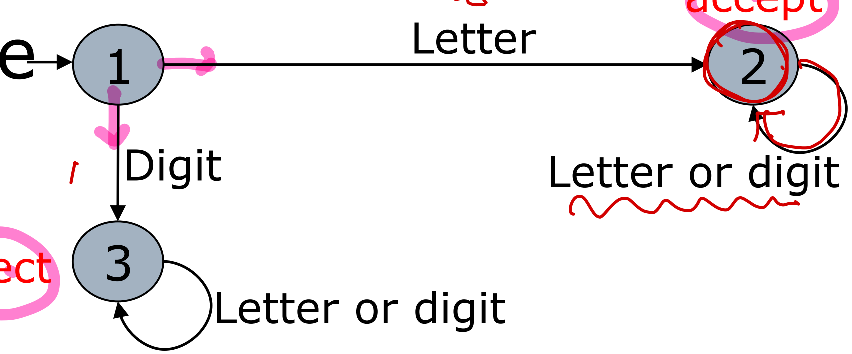
Accepter.

Identifier → a12 (O) ← accept; 숫자로 시작 X

- Variables :
- Terminals :

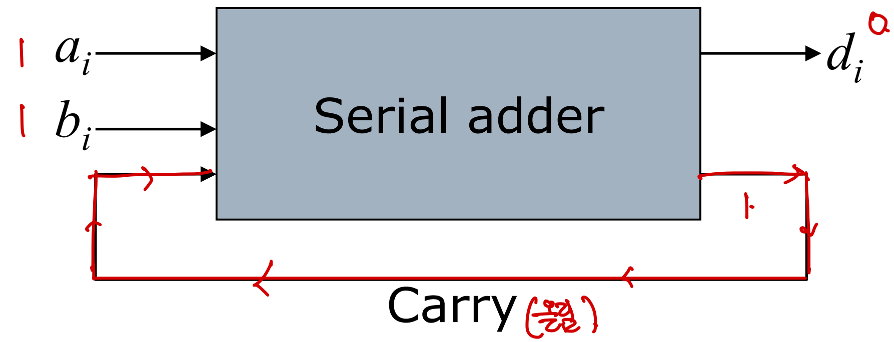
a123 → accept or reject.

- Accepter of the language



Example of Digital Logic

- A binary adder can be represented by Transducer
  - $x = a_0 a_1 \ldots a_n$, $y = b_0 b_1 \ldots b_n$
  - Calculate $x+y = d_0 d_1 \ldots d_n$



  - A Transducer

⇒ Turing Machine