

집합, 관계와 그래프, 함수

집합의 표시법

조건제시법으로 집합 P 를 정의할 때 조건명제(predicate) $p(x)$ 를 이용하여,

$$P = \{x \mid p(x)\} \quad (1)$$

로 쓰기도 하나, 집합 P 의 **전체집합(universe)** U 를 특별히 드러내 표시하기도 한다.

$$P = \{x \in U \mid p(x)\} \quad (2)$$

전체집합 U 를 표시하는 조건제시법 (2)는 두 가지 중요한 의미가 있다.

(1) 집합 P 를 포함하는 전체집합(universe of discourse) U 를 특별히 밝힌다.¹⁾

$$P \subseteq U \quad (3)$$

(2) **조건명제(predicate)** $p(x)$ 에서 변수 x 를 전체제안자(**universal quantifier**) \forall 을 이용하여, $\forall x \in U$ 로 **한정(bind)**하였기 때문에, $\forall x \in U: p(x)$ 는 **명제**²⁾가 된다.

이 강의에서는 자동기계(automata), 문법(grammar) 또는 프로그램을 이용하여 집합(언어)을 정의하는 방법을 배운다. 이는 집합(언어; languages)을 기계나 컴퓨터를 이용하여 표현하는 **새로운**³⁾ 방법으로 여겨져 20세기 초 근대수학에서 매우 중요하게 받아들여졌다.

조건명제와 진리집합(truth set)

(정의) 조건명제 $p(x)$ 의 **진리집합** $P \triangleq \{x \in U \mid p(x)\}$ 로 정의한다.

부분집합과 두 집합의 같음(same; equivalent)

(정의) $A \subseteq B \triangleq \forall x \in A \Rightarrow x \in B$. (4)

(정의) $A = B \triangleq A \subseteq B \wedge B \subseteq A$. (5)

$$\equiv (\forall x \in A \Rightarrow x \in B) \wedge (\forall x \in B \Rightarrow x \in A).$$

(예) $\{a, b, c\} = \{b, a, c\}$, $\{F, T\} \neq \{0, 1\}$.

진리집합과 필요, 충분조건

(정의) 전체집합이 U 인 조건명제 $p(x)$ 에 **진리집합**이 $P = \{x \in U \mid p(x)\}$ 이고 조건명제 $q(x)$ 에 진리집합이 $Q = \{x \in U \mid q(x)\}$ 일 때, $P \subseteq Q$ 이면 $\forall x \in U: p(x) \Rightarrow q(x)$ ⁴⁾이고, 조건명제 $p(x)$ 는 조건명제 $q(x)$ 가 만족하기 위한 **충분조건**이라 부르고, 거꾸로 조건명제 $q(x)$ 는 조건명제 $p(x)$ 가 만족하기 위한 **필요조건**이라고 부른다. 또 $P \subseteq Q \wedge P \supseteq Q$, 즉 $P = Q$ 이면, 조건명제 $p(x)$ 와 조건명제 $q(x)$ 는 각각 상대조건을 만족하기 위한 **필요충분조건** 또는 동치(equivalent) 명제라 부르고, $\forall x \in U: p(x) \Leftrightarrow q(x)$ 와 같다.

1) 수학에서는 $x \in U$ 로 쓰고 U 를 전체집합이라 부르나, 프로그래밍언어 Java나 C에서는 U x 로 쓰고 U 를 변수 x 에 **타입(type)**이라 부른다.

2) 명제는 참이나 거짓이 정해진 문장(statement)이다.

3) 이 노력은 1931년 Gödel의 불완전성 정리(Incompleteness Theorem)로 1929년(?) Hilbert가 꾸었던 아름다운 꿈이 실패라는 결론으로 이어져서, 현대수학에 한계를 드러내는 것이나, 2000여년 전에 피타고라스(Pythagoras)가 절망하였던 무리수 $\sqrt{2}$ 에 비밀을 풀어주는 열쇠가 되기도 한다.

4) 전체집합 U 가 명백하면 $p \Rightarrow q$ 로 짧게 쓰기도 한다.

곱집합(Cartesian product)과 관계(Relation) 그래프(Graph)

(정의) 집합 A 와 B 의 **곱집합(Cartesian product)**을

$$A \times B \triangleq \{(a, b) | a \in A, b \in B\} \quad (6)$$

로 정의한다. 이 때 곱집합의 원소 (a, b) 를 **순서쌍(ordered pair)**⁵⁾이라 부르고, 집합 $A \times B$ 의 크기는 집합 A 와 B 의 크기의 곱이다

$$|A \times B| = |A| \times |B| \quad (6).$$

(정의) 집합 A 에서 B 로 가는(from A to B) **관계(relation)** R 은 두 집합 A 와 B 의 곱집합의 **부분집합**으로 정의한다.

$$R \subseteq A \times B \quad (7)$$

이때 집합 A 를 관계 R 의 **정의역(domain)**, B 를 **치역(range; co-domain)**이라 부른다. $(a, b) \in R$ 일 때 aRb 로 쓰기도 한다.

(정의) 집합 A 에서 B 로 가는 관계 R 의 **역관계** $R^{-1} \subseteq B \times A$ 를 아래로 정의한다.

$$R^{-1} = \{(b, a) \in B \times A | (a, b) \in R\}$$

(정의) 관계 R 의 정의역과 치역이 같은 집합 A 일 때 **관계 R 은 집합 A 에서 정의한(on A) 관계라 부른다.**

$$R \subseteq A \times A \quad (8)$$

(정의) 집합 A 에서 정의한 관계 $E \subseteq A \times A$ 를 에지로 표현한 순서쌍 $G = (A, E)$ 을 집합 A 의 **그래프**라고 정의한다.

관계에 성질(property)과 그래프

집합 A 에서 정의한 관계 R 을 에지로 하는 그래프 $G = (A, R)$ 에서, 관계 R 이 특별한 성질을 만족할 때 그래프를 간단히 그릴 수 있다.

(1) 관계가 **reflexive**⁷⁾하면, 모든 노드에서 자기 자신으로 가는 에지(self edge)가 **항상 있고**, **irreflexive**하면 self edge가 **항상 없으므로**, 관계가 reflexive인지 irreflexive인지를 미리 알고 있다면, 그래프에서 self edge는 따로 표시하지 않을 수 있다.

(2) **Symmetric**하면 에지가 양방향으로 모두 있으므로 에지에 화살표를 따로 표시할 필요가 없다(undirected graph). **Antisymmetric** 또는 **asymmetric**하면 한 방향으로만 에지가 있는 것이므로, 에지에서 화살표를 뺀 방향 없는(undirected) 그래프로 간단하게 표현할 수도 있다. 다만 양방향으로 에지가 있는 symmetric의 경우는 문제가 없으나, 한 방향으로만 에지가 있는 antisymmetric이나 asymmetric일 때는 에지가 출발하는 노드를 위쪽에 그려서 그래프로 표현하기도 한다(예: 뿌리 깊은 나무(rooted tree)).

(3) **Transitive**하면, 처음 노드에서 마지막 노드로 중간 노드를 통하여 에지가 연결하면, 반드시 처음 노드에서 마지막 노드로 에지가 직접 있는 경우이다⁸⁾. 모든 연결을 다 표시하는 것은 낭비이고 기본적인 도시 연결 상황만을 나타내면 충분하고, 이 기본적인 연결 상

5) $(a, b) \neq (b, a)$ 이고 $\{a, b\} = \{b, a\}$ 인 집합과는 다르므로 순서쌍이라고 부른다.

6) **합집합** $A \cup B$ 의 크기는 $|A \cup B| \leq |A| + |B|$ 임에 유의하라.

7) 관계의 성질인 i) reflexive와 irreflexive, ii) symmetric과 asymmetric antisymmetric, iii) transitive에 정확한 정의는 TP나 이산수학 교과서를 참조하시오.

8) R is transitive. $\triangleq \forall a, b, c \in A : aRb \wedge bRc \Rightarrow aRc$.

향을 Hasse diagram이라 부른다.

(예) 자연수의 집합 N 에서 정의한 관계의 예($=, <, \leq$)

$$= \stackrel{\text{def}}{=} \{(a, b) \in N \times N \mid a = b\}$$

reflexive, symmetric, transitive

$$< \stackrel{\text{def}}{=} \{(a, b) \in N \times N \mid a < b\}$$

irreflexive, asymmetric, antisymmetric, transitive

$$\leq \stackrel{\text{def}}{=} \{(a, b) \in N \times N \mid a < b \vee a = b\} = (< \cup =)$$

reflexive, antisymmetric, transitive

(정의) 집합 정의역과 치역이 같은 집합 A 에서 정의한 관계 R 은 **합성**(composition 또는 **곱**(product))이 쉽게 정의되므로 관계 R 이 여러(n)번 계속 곱하여지는 **반복곱** R^n ($n \geq 0$)을 아래와 같이 **recursive**하게 정의한다.

$$R^0 \stackrel{\text{def}}{=} id_A \quad \text{단 } id_A = \{(a, a) \mid a \in A\}^9.$$

$$R^n \stackrel{\text{def}}{=} R \circ R^{n-1} \quad n \geq 1. \quad (9)$$

관계의 p-closure

집합 A 안에 있는 관계 R 과 관계 R 의 성질의 집합 $P = \{reflexive, symmetric, transitive\}$ 의 원소 $p \in P$ 의 p -closure R' 을 아래와 같이 정의한다.

$R' \supseteq R$ 을 만족하는 집합 A 안에 있는 관계 중에 성질 $p \in P$ 를 만족하는 가장 적은 관계집합이다.

(정의) $A \subseteq B$ 일 때 집합 A 가 집합 B 보다 적거나 같다고 한다.

(예) Reflexive closure of R , $R' = R \cup id_A$,

Symmetric closure of R , $R' = R \cup R^{-1}$.

Transitive closure of R , $R^\dagger = R \cup R^2 \cup R^3 \cup \dots = \bigcup_{i \in N_1} R^i$.

Reflexive and transitive closure of R , $R^* = R^0 \cup R^1 \cup R^2 \cup \dots = \bigcup_{i \in N_0} R^i$.

함수(function)

(정의) 집합 A 에서 B 로 가는 관계 f 중 (1) A 의 **모든**(total) 원소 $a \in A$ 에 대하여 관계 afb (또는 $(a, b) \in f$)가 있는 $b \in B$ 가 반드시 (2) **하나씩**(unique) 있는 경우, 관계 f 를 **집합 A 에서 B 로 가는 함수** f 라 부르고,

$$f: A \rightarrow B$$

로 표시하며 특히 afb 또는 $(a, b) \in f$ 대신 $f(a) = b$ 로 표시하기도 한다. 이 때 집합 A 를 함수 f 의 정의역(domain) 집합 B 를 함수 f 의 치역(range)이라 한다.

$$f: A \rightarrow B, \text{ if } (f \subseteq A \times B) \wedge (\forall a \in A : \exists_1 b \in B : afb)$$

9) $\forall R \subseteq A \times A : R \circ id_A = id_A \circ R = R$. id_A 는 합성(\circ)에 관한 항등원(identity element)이다.

위의 조건 (1) 정의역에 모든 ...은 만족하지 않지만 (2) 치역에 하나만 ...은 만족할 때 관계 f 를 부분함수(partial function)라 부르고,

$$f: A \mapsto B, \text{ if } (f \subseteq A \times B) \wedge (\forall a \in A: [(\exists_1 b \in B: a f b) \vee (\nexists b \in B: a f b)])^{10)}$$

(1) (2)를 모두 만족하는 함수를 전체함수(total function)라 부르기도 한다.

$$f: A \rightarrow B, \text{ if } (f \subseteq A \times B) \wedge (\forall a \in A: \exists_1 b \in B: a f b)$$

함수 $f: A \rightarrow B$ 의 역함수 $f^{-1}: B \rightarrow A$ 는 함수가 아니고 관계다. f^{-1} 가 함수가 되려면,

(1) 정의역 B 에 모든 ...을 만족하면, 즉

$$\forall b \in B: \exists a \in A: a f b \text{ 또는 } \forall b \in B: \exists_1 f^{-1}(b) \in A \text{ 이면}$$

함수 $f: A \rightarrow B$ 는 onto 함수(correspondence) 또는 surjection(전사함수)라 부르고,

$$|A| \geq |B| \text{ 이다.}$$

(2) 치역 A 에 하나만 ...을 만족하면, 즉

$$\forall a \in A: \exists_1 b \in B: a f b \text{ 또는 } b \in B: \exists_1 f^{-1}(b) \in A \text{ 이면}$$

함수 $f: A \rightarrow B$ 는 one-to-one(1:1) 함수 또는 injection(단사함수)라 부르고.

$$|A| \leq |B| \text{ 이다.}$$

함수 $f: A \rightarrow B$ 의 역함수 $f^{-1}: B \rightarrow A$ 도 함수이면 함수 $f: A \rightarrow B$ 는 one-to-one onto 함수(1-1 correspondence; 짝짓기) 또는 bijection(전단사함수)이라 부르고

$$|A| = |B| \text{ 이다.}$$

(정의) $|A| = |B| \iff A \cong B$.

두 집합이 동등할 때(\cong) 두 집합이 크기(cardinality)가 같다고 정의한다.

집합에 동등(isomorphism¹¹)

(정의) $A \cong_f B \iff (\exists f: A \rightarrow B \wedge \exists f^{-1}: B \rightarrow A)$.

$$\equiv \exists f: A \leftrightarrow B.$$

$A \cong_f B$ 일 때, $|A| = |B|$ 이고 집합 A 와 집합 B 는 함수 f 를 통하여(with respect to: w.r.t.) 동등(isomorphic)하다고 한다. 집합 A 와 집합 B 가 짝짓기 함수 f 를 통하여 동등하다는 말은, $B = f(A)$ 로 $A = f^{-1}(B)$ 로 나타낼 수 있기 때문이다.

짝짓기 함수¹²) f 를 생략하고 $A \cong B$ 로 간단히 쓸 수도 있다.

집합의 동등함(\cong)은 집합의 같음(=)을 포함하는($\cong \supseteq =$) 더 큰 개념이다.

(예) $\{a, b, c\} \neq \{1, 2, 3\}$ 이지만 $\{a, b, c\} \cong \{1, 2, 3\}$ 이다. 이 때 짝짓기 함수 f 는 무엇일까?
 $|\{a, b, c\}| = |\{1, 2, 3\}| = 3$ 이다.

멱집합(Power Set)

(정의) 집합 A 의 부분집합의 집합을 집합 A 의 멱집합(power sets)이라 부르고 2^A 로 쓴다.

10) 부분함수의 조건을 $|f(a)| \leq 1$ 로 쓰는 책도 있다.

11) Equivalent(같음; same)와 구분하기 위하여 isomorphism(동등)이라는 다른 용어를 사용하였다.

12) 짝짓기 함수는 1-1 onto 함수를 뜻한다.

$$2^A \triangleq \{B \mid B \subseteq A\}.$$

(예) $2^{\{0,1,2\}} = \{ \emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{1,2\}, \{0,2\}, \{0,1,2\} \}.$

(사실) $|2^A| = 2^{|A|}.$

(증명) 생략.

(정의) 집합 A 에서 B 로 가는 함수의 집합을 B^A 라 쓴다.

$$B^A \triangleq \{f \mid f: A \rightarrow B\}.$$

(예) $\{T, F\}^{\{0,1,2\}} \triangleq \{$
 $\{(0, F), (1, F), (2, F)\}, \{(0, F), (1, F), (2, T)\}, \{(0, F), (1, T), (2, F)\}, \{(0, F), (1, T), (2, T)\},$
 $\{(0, T), (1, F), (2, F)\}, \{(0, T), (1, F), (2, T)\}, \{(0, T), (1, T), (2, F)\}, \{(0, T), (1, T), (2, T)\} \}$
 $\cong \{ (F_0F_1F_2), (F_0F_1T_2), (F_0T_1F_2), (F_0T_1T_2),$
 $(T_0F_1F_2), (T_0F_1T_2), (T_0T_1F_2), (T_0T_1T_2) \}$
 $\cong \{ 000_2, 001_2, 010_2, 011_2, 100_2, 101_2, 110_2, 111_2 \}$

(사실) $|B^A| = |B|^{|A|}.$

(증명) 생략.

(정의) 집합 A 가 유한하면 $B^A \cong B^{|A|}$ 이므로, B^n (단 $|A| = n$)으로 간단히 쓰기도 한다.

(예) $\{T, F\}^{\{0,1,2\}} \cong \{T, F\}^{\{a,b,c\}} \cong \{0,1\}^{\{1,2,3\}} \cong \{0,1\}^3$

(사실) B^n 는 $|B|$ -진수¹³⁾ 길이 n 짜리 문자열(string)¹⁴⁾이다.

(사실) 집합 A 에 멱집합 2^A 은 집합 A 에서 집합 $\{0,1\}$ 로 가는 함수와 동등하다.

$$2^A \cong \{0,1\}^A.$$

(예) $A = \{0,1,2\}$ 일 때, $2^A = \{ \emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\} \}$
 $\leftrightarrow_f \{ 0_00_10_2, 0_00_11_2, 0_01_00_2, 0_01_01_2, 1_00_10_2, 1_00_11_2, 1_01_00_2, 1_01_01_2 \}$
 $\emptyset \leftrightarrow_f 0_00_10_2, \{0\} \leftrightarrow_f 1_00_10_2, \{1\} \leftrightarrow_f 0_01_00_2, \{2\} \leftrightarrow_f 0_00_11_2,$
 $\{0,1\} \leftrightarrow_f 1_01_00_2, \{0,2\} \leftrightarrow_f 1_00_11_2, \{1,2\} \leftrightarrow_f 0_01_01_2, \{0,1,2\} \leftrightarrow_f 1_01_01_2.$

(증명) $A = \{1, 2, \dots, n\}$ 라 하자, $b_1b_2 \dots b_n \in \{0,1\}^n$ (단 $1 \leq \forall i \leq n: b_i \in \{0,1\}$)
 $b_1b_2 \dots b_n \leftrightarrow_f \{i \in A \mid 1 \leq \forall i \leq n: b_i = 1\}$

관계를 함수로 표기하기

관계 $R \subseteq A \times B$ 이고 $a \in A$ 에 대하여 $aRb_1, aRb_2, \dots, aRb_k$ 일 때,

$$R(a) = \{b_1, b_2, \dots, b_k\} \subseteq B \text{로 쓰기도 한다.}$$

단 $k \geq 0$ 이고, $k=0$ 이면 $\{b_1, b_2, \dots, b_k\} = \emptyset$ 이라 본다.

이 때 $R: A \rightarrow 2^B.$

즉 관계 $R \subseteq A \times B$ 을 집합 A 를 정의역으로 집합 B 의 **부분집합**을 치역으로 하는 **함수** (**set valued function**)로 보고 있다¹⁵⁾.

13) n -진수는 n -진법을 말하는 것으로 위에 예는 이진수(binary number) 문자열이다.

14) 문자열(string)은 1-3에서 다시 자세히 다룬다.

15) 관계 R 은 함수인가 함수가 아닌가?