# 집합, 관계와 그래프, 함수

집합의 표시법

집합을 정의하는 원소나열법 다음에, 두 번째 방법인 조건제시법에서

$$A = \{x|\ p(x)\} \tag{1}$$

집합 $A$의 universe 인 집합 $U$를 명시하여 표현하는 방법

$$A = \{x \in U \mid p(x)\} \tag{2}$$

은 두 가지 의미를 가지고 있다.

(1) 집합 $A$를 포함하는 전체집합(universe of discourse; 혹은 집합 $A$의 원소 $a$의 타입(type)[1]) $U$를 명확히 표현한다.

$$A \subseteq U \tag{3}$$

(2) 집합 $A$를 포함하는 전체집합 $U$를 명시하여 predicate(조건명제) $p(x)$를 한정(bind)한다. 집합 $A$를 조건명제 $p(x)$를 만족하는 진리집합(truth set)이라 부른다.

집합을 표현하는 세 번째 방법인 automaton, grammar 또는 프로그램이 이 강의에서 다루어 질 집합(언어; languages)을 표현하는 **새로운** 방법이다.

두 집합의 같음(same)

(정의) $A \subseteq B \overset{\text{def}}{=} \forall x \in A \Rightarrow x \in B.$

(정의) $A = B \overset{\text{def}}{=} A \subseteq B \wedge A \subseteq B.$

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

관계(Relation) 와 그래프(Graph)

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

$$A \times B \overset{\text{def}}{=} \{(a,b)|a \in A,\ b \in B\}$$

로 정의한다. 이 때 **곱집합**의 원소 $(a,b)$를 순서쌍(ordered pair)[2]이라 부르고, 집합 $A \times B$의 크기는 집합 $A$와 $B$의 크기의 곱이다

$$|A \times B| = |A| \times |B|$$[3].

(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.

$$R \subseteq A \times B \tag{4}$$

이때 집합 $A$를 관계 $R$의 정의역(domain), $B$를 치역(range)이라 각각 부른다. $(a,b) \in R$일 때 $aRb$로 쓴다.

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

---

1) 수학에서는 $U \ni a$로 프로그램 언어 Java에서는 $U\ a$로 표시한다.
2) $(a,b) \neq (b,a)$이고 $\{a,b\} = \{b,a\}$인 집합과는 다르므로 **순서**쌍이라고 부른다.
3) **합집합** $A \cup B$의 크기는 $|A \cup B| \leq |A| + |B|$임에 유의하라.

$R \subseteq A \times A$ (5)

(정의) 집합 $A$와 관계 $E \subseteq A \times A$의 순서쌍 $G = (A, E)$을 집합 $A$의 그래프라고 정의한다.

(예) 자연수의 집합 N속에 있는 관계의 예(=, <, ≤)

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

reflexive, symmetric, transitive

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

irreflexive, asymmetric, antisymmetric, transitive

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

reflexive, antisymmetric, transitive

(정의) 집합 정의역과 치역이 같은 집합 $A$안에 있는 관계 $R$은 합성(composition 또는 곱(product))이 쉽게 정의되므로 관계 $R$이 여러($n$)번 반복적으로 곱하여지는 **반복곱** $R^n (n \geqq 0.)$이 아래와 같이 recursive하게 쉽게 정의된다.

$R^0 =_B id_A$

$R^n =_R R^{n-1} \circ R \qquad n \geqq 1.$ (6)

관계의 p-closure

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

(1) $R' \supseteq R$

(2) $R'$은 (1)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 작거나 같다고 한다.

(예) Reflexive closure of $R$, $R' = R \cup id_A$, 단 $id_A = \{(a,a) | a \in A\}$.

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

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

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

관계와 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) Reflexive한 경우(=, ≤)는 모든 노드에서 자기 자신으로 가는 에지(self edge)가 항상 있고, irreflexive한 경우(<)는 항상 self edge가 없는 경우이므로, reflexive인지 irreflexive인지를 미리 알고 있다면(reflexive/irreflexive closure), 그래프에서 self edge를 표시하지 않을 수도 있다,

(2) Symmetric(=)이나 antisymmetric(<, ≤)한 경우에 임의에 노드 $a$에서 노드 $b$로 가는

에지가 i) symmetric 경우 양방향으로 두 개 모두 있거나, ii) antisymmetric 경우 정해진 방향으로 반드시 한 개만 있는 것이므로, 에지에서 화살표를 뺀 방향 없는(undirected) 그래프로 간단하게 표현할 수도 있다. 방향이 정하여지지 않은 symmetric의 경우는 관계가 없으나, 방향이 정하여진 antisymmetric의 경우는 에지가 출발하는 노드를 위쪽에 그려서 비방향성그래프로 표현하기도 한다(예 뿌리 있는 나무).

Asymmetric($<$, $\le$)과 antisymmetric($\le$)은 방향이 정하여진다는 점에서 같으나, 같은 노드 $a$에서 $a$로 가는 에지를 antisymmetric 경우에는 제한이 없으나, asymmetric 경우는 항상 허용하지 않는다는 점이 다르다. 따라서 asymmetric 조건은 irreflexive나 antisymmetric이 되기 위한 **충분**(더 **강한**)조건이다. Symmetric이나 Antisymmetric은 reflexive나 irreflexive 조건과는 관계없다(incomparable).

(3) Transitive한 경우는 관계 그래프에서 중간 노드를 통하여 에지가 연결하면, 직접 에지가 있는 경우이다. 이는 도시간의 연결 그래프가 전형적인, 경우로 전체 도시 간에 연결 상황은 기본적인 도시간의 연결 관계$R$의 reflexive-transitive closure$R^*$[4])를 구하여 해결한다. 이 때 기본적인 도시 연결 상황만을 Hasse diagram이라 부른다.

<u>함수(function)</u>

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

$$f : A \to B$$

로 표시하며 특히 $afb$대신에 $f(a) = b$로 표시하기도 한다. 이 때 집합 $A$를 함수 $f$의 정의역(domain) 집합 $B$를 함수 $f$의 치역(range)라 부른다.

$f : A \to B$, if

$$(f \subseteq A \times B) \ \wedge \ (\forall a \in A, \exists_1 b \in B : afb)$$

위의 조건 (1)을 만족하지 않는 경우 관계 $f$를 부분함수(partial function)이라 부르고, 만족하는 경우 함수 또는 전함수(total function)이라 부른다.

<u>집합의 동등(대응; isomorphism[5])</u>

(정의) $A \cong_f B \overset{\text{def}}{=} (\exists f : A \to B \ \wedge \ f^{-1} : B \to A)$.

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

$A \cong_f B$일 때, 집합 $A$와 집합 $B$는 함수 $f$를 통하여(with respect to) 동등(대응; isomorphic)하다고 한다. 대응의 의미는 집합 $A$와 함수 $f$가 주어졌을 때, 집합 $B$를 $B = f(A)$로 집합 $B$와 함수 $f$가 주어졌을 때, 집합 $A$를 $A = f^{-1}(B)$로 나타낼 수 있기 때문이다.

---

4) Algorithm 시간에 배우는 reachable vertex찾는 문제가 바로 이 문제이다.
5) Equivalent(같은 값)와 구분하기위하여 isomorphism이라는 좀 더 일반적인 용어를 사용하였다.

$$(\forall a \in A : \exists_1 f(a) \in B) \ \wedge \ (\forall b \in B : \exists_1 f^{-1}(b) \in A).$$

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

집합의 동등함($\cong$)은 집합의 같음($=$)을 포함하는 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.

무한집합의 동치

(예 1) 0부터 시작하는 자연수 집합 $N_0 = \{0,1,2,\cdots\}$과 1부터 시작하는 자연수 집합 $N_1 = \{1,2,3,\cdots\}$을 생각하자. $N_1 \subset N_0$이지만 $N_1 \cong N_0$[6]이다.

(예 2) 자연수 집합 $N = \{0,1,2,\cdots\}$과 짝수 집합 $E = \{0,2,4,\cdots\}$을 생각하자. $E \subset N$ 이지만 $E \cong N$이다.

(예 3) 자연수 집합 $N = \{0,1,2,\cdots\}$과 자연수의 순서쌍 집합 $N \times N = \{(i,j) | i,j \in N\}$을 생각하자. $N \cong N \times N$[7]이다.

(정의) $|A| = |B| \stackrel{\text{def}}{=} A \cong B$.
두 집합이 동등할 때($\cong$) 두 집합이 크기가 같다고 정의한다.

(정의) 자연수의 부분집합과 동등한 집합을 **셀 수 있는**(countable) 집합이라 정의하고, 그렇지 않으면 **셀 수 없**는(uncountable) 집합 이라고 정의한다.

(사실) **셀 수 있는** 집합은 **유한**집합과 **셀 수 있는 무한**집합 두 가지 종류가 있다.

(정리) 모든 **셀 수 있는** 무한집합은 **자연수**집합과 **동등**하다.

(사실) 모든 **셀 수 있는** 무한집합은 **자연수로 번호** 매길 수(**enumerable**) 있다.

$$A = \{a_0, a_1, \cdots, a_n, \cdots\}$$

(의문) **셀 수 없는 무한**집합은 어떤 것이 있는가?

(정리) 자연수의 **부분집합의 집합**은 **셀 수 없는** 무한이다.

(사실) 자연수의 **부분집합의 집합**은 **무한이진수(infinite binary string) 집합**과 동등하다.

(가정) 집합 $F = 2^N \cong \{f | f : N \rightarrow \{0,1\}\}$는

(결론) **셀 수 없는 무한**이다.

(증명) 집합 $F$가 **셀 수 있는 무한**이라고 **가정**하자.

그러면 집합 $F$의 원소 들은 자연수도 번호 매길 수 있을 것이다.

$$F = \{f_0, f_1, \cdots, f_n, \cdots\}$$

$$f_0 = (a_{00}, a_{01}, \cdots, a_{0n} \cdots), \ \forall i \in N, \ f_0(i) = a_{0i} \in \{0,1\} \leftrightarrow \{i \in N | f_0(i) = a_{0i} = 1\}.$$

$$f_1 = (a_{10}, a_{11}, \cdots, a_{1n} \cdots), \ \forall i \in N, \ f_1(i) = a_{1i} \in \{0,1\} \leftrightarrow \{i \in N | f_1(i) = a_{1i} = 1\}.$$

$$\cdots$$

$$f_n = (a_{n0}, a_{n1}, \cdots, a_{nn} \cdots), \ \forall i \in N, \ f_n(i) = a_{ni} \in \{0,1\} \leftrightarrow \{i \in N | f_n(i) = a_{ni} = 1\}.$$

---

6) 중학교에서는 1부터 자연수를 대학교에서는 0부터 자연수를 정의해도 되는 이유이다.
7) 이 때 짝짓기 함수 $f$는 무엇인가? $N \subset N \times N$인가? $|N| < |N \times N|$인가 $|N| = |N \times N|$인가?

$\cdots$

여기서 대각선화(diagonalization) 함수 $f = (a_{00,} a_{11,} \cdots, a_{nn} \cdots)$와 그것의 **부정** $\overline{f}$를 생각하자.

$\overline{f} = (\overline{a_{00,}}\,\overline{a_{11,}} \cdots, \overline{a_{nn}} \cdots)$, 단 $\forall i \in N,\ \overline{a_{ii}} = 0,\ \text{if}\ a_{ii} = 1;\ \overline{a_{ii}} = 1,\ \text{if}\ a_{ii} = 0.$

$\forall i \in N,\ \overline{f} \neq f_i$이어서 $\overline{f} \not\in F$이지만, $\overline{f}$는 집합 $F$의 원래 정의에 맞는 함수 이므로 $\overline{f} \in F$이어야 한다.

따라서 집합 $F$가 셀 수 있는 무한이라는 증명 초기의 **가정**이 **거짓**이다.

따라서 집합 $F$는 셀 수 있는 무한이 아니다. 즉 집합 $F$는 **셀 수 없는 무한**이다.