

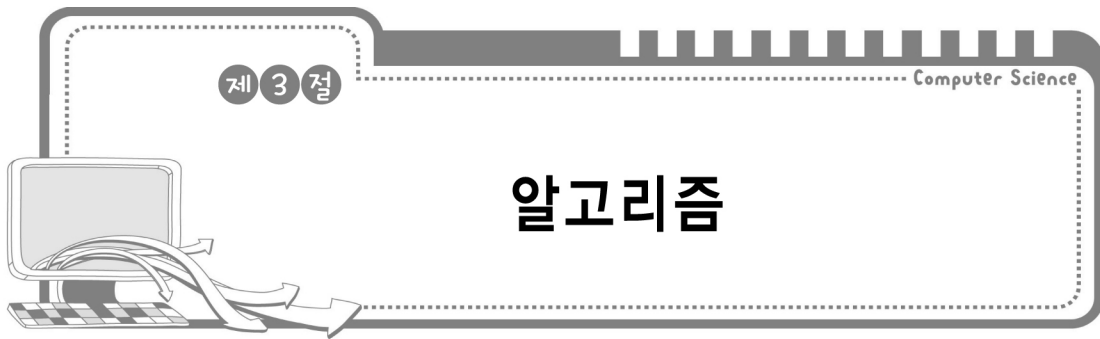
제1장

기 초

제 1 절 집 합

제 2 절 증명방법

제 3 절 알고리즘



알고리즘(algorithm)은 주어진 문제를 해결하기 위한 일련의 과정이나 절차를 순서대로 정확하게 나열한 것을 말한다. 알고리즘의 각 작업은 모호함이 없이 분명하게 표현되어야 한다. 컴퓨터 프로그램을 작성하려면 문제를 이해한 후 데이터를 어떻게 입력하고 처리하며 결과는 어떤 형식으로 출력할 것인지에 대해 기술해야 한다. 이때 알고리즘은 이러한 명령어들을 처리하는 일련의 절차라고 할 수 있으며 이는 프로그램 작성의 중요한 기초가 된다. 한 문제를 해결하기 위한 방법이나 과정은 여러 개 있을 수 있으므로 알고리즘도 다양하게 존재할 수 있다. 프로그램은 알고리즘을 기초로 작성되기 때문에 여러 알고리즘 중 가장 효율적이고 효과적인 알고리즘을 선택하는 것은 곧 좋은 프로그램의 작성을 의미한다.

알고리즘

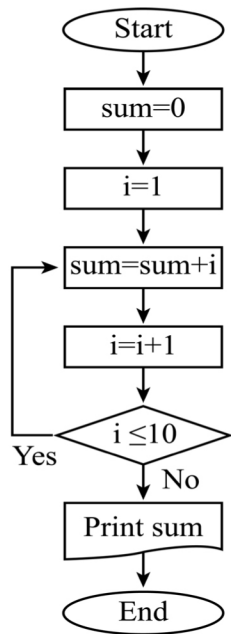
문제를 해결하기 위한 일련의 절차

알고리즘은 프로그램을 개발하기 위한 초기에 작성되는 것이며 전체 프로그램을 총괄하는 역할을 한다. 효율적인 알고리즘을 선택하는 방법은 여러 가지가 있으며 알고리즘을 선택하거나 개발할 때 다음 특성을 고려해야 한다.

- 입력(input) : 문제와 관련된 입력이 반드시 필요하다.
- 출력(output) : 처리한 결과인 출력이 반드시 존재해야 한다.
- 정확성(correctness) : 논리적으로 정확한 입력값에 대한 정확한 출력값을 만들어야 한다.
- 유한성(finiteness) : 입력은 제한된 수의 명령 단계를 거친 후 출력하고 반드시 종료해야 한다.
- 일반성(generality) : 같은 형태의 입력에 대해 항상 적용할 수 있어야 한다.

알고리즘은 순서도, 의사코드, 일반적인 언어로 작성할 수 있다. 순서도는 문제를 해결하는 방법과 절차에 따라 약속된 기호를 사용하여 명령들을 순서대로 표현하는 방식이다. 의사코

드는 프로그램 명령문 형식을 취하고 각 명령을 사람이 이해하기 쉽게 적당한 뜻을 가진 단어로 나타낸 것이다. 일반적인 언어로 작성한 알고리즘은 사람이 사용하는 문장으로 설명하고 있다.



1	algorithm sum1to10(){
2	sum = 0;
3	i = 1;
4	while(i ≤ 10)
5	sum = sum + i;
6	i = i + 1;
7	endwhile
8	print sum;
9	}

[그림 1-3] 순서도와 의사코드의 예(1부터 10까지의 합을 구하는 알고리즘)

1. 의사코드

알고리즘은 특정 언어를 사용하여 표현해야 하는데 그러한 언어 중에는 일상적으로 사용하는 자연어, 컴퓨터 프로그래밍 언어, 순서도(flow chart), 의사코드(pseudocode) 등이 있다. 의사코드는 명령이 실행되는 절차를 일반적인 언어와 프로그램 코드를 활용하여 작성하는 방식이다. 프로그램 코드와 비슷한 형태로 알고리즘을 작성하는 의사코드는 일반적으로 C 언어 형식으로 작성한다.

(1) 지정문

지정문은 많이 사용하는 의사 코드 중 하나이며 연산의 결과를 특정 변수에 지정해주는 역할을 한다. 예를 들어

$$i = i + 1;$$

과 같은 형식에서 등호(=)의 의미는 오른쪽의 $i + 1$ 을 계산한 후 그 결과를 왼쪽의 변수 i 에 지정하라는 의미이다.

일반적으로 지정문은

$$\text{변수명} = \text{계산식};$$

과 같은 형식을 갖게 되는데 하나의 문장의 끝에는 세미콜론(;)이 붙는다.

예제34

두 변수 x 와 y 의 값을 더하여 그 결과를 변수 z 에 저장하는 지정문을 작성하시오.

풀이

지정문은 등호(=)를 기준으로 오른쪽의 식을 계산하여 왼쪽의 변수에 지정하라는 의미이므로 두 변수 x 와 y 를 더하는 연산을 오른쪽에 쓰고 계산의 최종 결과를 저장하는 변수명 z 는 왼쪽에 쓰면 다음과 같다.

$$z = x + y;$$

(2) 제어문

어떤 명령을 처리하는 알고리즘에서 특정 조건에 의해 명령의 실행 순서가 변경되는 경우가 있고 어떤 명령을 반복적으로 수행해야 하는 경우가 있다. 이러한 구문을 제어문이라 하고 제어문에는 조건문, 반복문, 명령 중지문 등이 있다. 조건문은 특정 조건이 기준이 되어 해당 조건을 만족할 때 명령문이 실행하도록 하는 구문이다. 이때 조건이 하나인 경우도 있고 여러 개인 경우도 있다.

if-then 형식은 조건문이 참이면 then 다음의 명령문을 실행한다.

```
if 조건문 then  
명령문;
```

if-then-else 형식은 조건문이 참이면 명령문1을 실행하고 그렇지 않으면 명령문2를 실행한다.

```
if 조건문 then  
명령문1;  
else  
명령문2;
```

조건이 여러 개인 경우 각 조건에 따라 다른 명령문을 실행하게 하려면 if-then-else if문을 사용하면 된다. 이런 경우 조건문1을 만족하면 명령문1을 실행하고, 조건문2를 만족하면 명령문2를 실행하는 등의 과정을 반복한다.

```
if 조건문1 then  
명령문1;  
else if 조건문2 then  
명령문2;  
else if 조건문3 then  
명령문3;  
...  
  
else  
명령문n;
```

예제35

세 수를 입력받아 그 중 가장 큰 수를 출력하는 알고리즘을 의사코드로 작성하시오.

풀이

```
algorithm large(x, y, z)
{
    max = x;
    if max < y then
        max = y;
    if max < z then
        max = z;
    print max;
}
```

반복문은 명령문을 반복적으로 실행하게 하는 것을 말한다. 이때 반복 실행하는 것을 루프라고도 하고 일정한 횟수만큼 반복하거나 특정 조건을 만족하는 동안 반복하게 할 수 있다. 반복문에는 for문, while문, do~while문이 있다.

for문은 변수의 초깃값을 주고 명령문을 한 번 반복 실행할 때마다 변수의 값을 하나씩 증가시키면서 최종값이 될 때까지 반복 실행하는 것이다. 만약 변수가 최종값이 되면 해당 for문을 빠져나와 더 이상 반복을 실행하지 않는다.

```
for 변수 = 초깃값 to 최종값
    명령문;
```

while문은 조건을 만족하는 동안 명령문을 반복 실행한다. 따라서 조건을 더 이상 만족하지 않으면 반복을 종료한다.

```
while 조건문
    명령문;
```

do-while문은 조건을 만족하는 동안 반복 실행한다는 점에서 while문과 비슷하지만 조건을 만족하지 않는 경우에도 최소한 한 번 명령문을 실행한다는 점이 다르다. 즉 while문은 조건을 먼저 비교하여 조건이 거짓이면 명령문을 한 번도 실행하지 않지만 do-while문은 일

단 명령문을 실행한 후 조건을 나중에 비교하므로 조건이 거짓이라 하더라도 명령문을 최소한 한번은 실행한다.

```
do
  명령문;
while 조건문
```

예제36

1부터 n 까지의 정수의 합계를 계산하는 알고리즘을 의사코드로 작성하시오.

풀이

```
algorithm sum(n)
{
  result = 0;
  for i = 1 to n
    result = result + i;
  next i
  print sum;
}
```

2. 알고리즘의 복잡도

올바른 알고리즘은 실행한 후 정확한 결과를 얻을 수 있어야 한다. 그러나 제대로 된 결과를 계산해 내는 알고리즘이라 할지라도 수행 시간과 메모리 공간을 많이 사용한다면 좋은 알고리즘이라 할 수 없다. 따라서 하나의 문제를 해결하는 여러 알고리즘 중 처리 시간이나 차지하는 메모리 용량이 적은 알고리즘을 선택하여 프로그램의 효율성을 높여야 한다. 알고리즘은 어떤 식으로 작성하느냐에 따라 실행되는 시간과 공간이 다르며 이것을 알고리즘의 복잡도(complexities of algorithm)라고 한다. 알고리즘의 수행에 필요한 시간과 공간의 양을 각각 그 알고리즘의 시간복잡도, 공간복잡도라고 한다. 알고리즘은 복잡도가 낮을수록 효율적이며 실제 프로그램을 실행시켜서 측정하는 것이 아니라 작성된 알고리즘을 보고 분석하여 판단한다. 알고리즘의 복잡도는 처리해야 하는 데이터의 양이나 표현방법, 컴파일러 등에 따

라 달라지기 때문에 성능을 측정하기가 매우 어렵다. 특히 입력되는 데이터가 커질수록 실행 속도가 느려지거나 성능이 저하되는 경우가 많으므로 입력 데이터가 무한히 많아질 때의 알고리즘의 성능을 주로 평가한다. 알고리즘 복잡도는 주로 대문자 O 를 사용하여 나타내며 이를 'Big-O 표기법'이라고 한다.

알고리즘의 복잡도

알고리즘이 수행될 때 필요한 시간이나 공간에 대한 비용

- 시간복잡도(time complexity) : 알고리즘이 수행되는 시간
- 공간복잡도(space complexity) : 알고리즘이 수행될 때 필요한 메모리 공간

예제37

함수 $f(n) = n \log_2 n$ 을 Big-O 표기법으로 나타내시오.

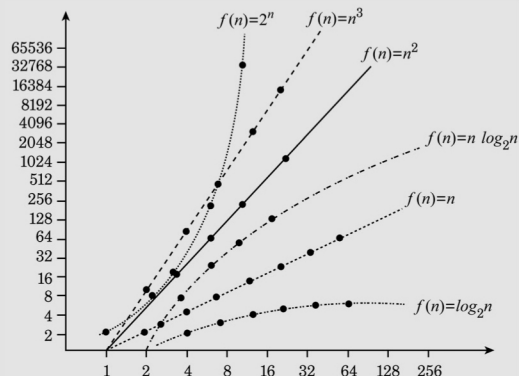
풀이

$$\begin{aligned} f(n) &= n \log_2 n \\ &\leq n \cdot n \\ &= n^2 \end{aligned}$$

따라서 $f(n) = n \log_2 n = O(n^2)$ 이다.

복잡도 간의 관계

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(2^n)$$



확인 학습 문제

01 다음 집합을 조건제시법으로 나타낸 것 중 옳지 않은 것은?

- ① $\{1, 3, 5, 7, 9\} \rightarrow \{a | 1 \leq a \leq 10, a \text{는 홀수}\}$
- ② $\{0, 6, 12, 18, 24\} \rightarrow \{a | 0 \leq a \leq 25, a \text{는 } 6 \text{의 배수}\}$
- ③ $\{-3, -2, -1, 0, 1, 2, 3\} \rightarrow \{a | -3 < a < 3, a \text{는 정수}\}$
- ④ $\{1, 2, 3, 6, 12, 18, 36\} \rightarrow \{a | a \text{는 } 36 \text{의 약수}\}$

㉠ ③에서 -3 과 3 이 포함되어야 하므로 조건제시법으로 나타내면 $\{a | -3 \leq a \leq 3, a \text{는 정수}\}$ 와 같다.

02 전체집합 $U = \{a, b, c, d, e, f, g, h, i, j, k\}$ 이고 집합 $A = \{a, b, c, d, e\}$, $B = \{a, c, e, g, i\}$, $C = \{a, b, c, h, i, j, k\}$ 일 때 다음 집합을 계산한 것 중 옳지 않은 것은?

- ① $A \cup B = \{a, b, c, d, e, g, i\}$
- ② $A \cap B \cap C = \{a, c\}$
- ③ $(A - B) \cap C = \{a, b\}$
- ④ $A \cap (B - C) = \{e\}$

㉠ $(A - B) \cap C = \{b\}$ 이다.

03 다음 중 무한집합인 것은?

- ① 함수의 집합 ② 요일의 집합
③ 영어 소문자의 집합 ④ 무지개 색의 집합

 ① 홀수의 집합은 무한집합이다.

② 요일의 집합을 A 라고 하면 $A = \{\text{월요일, 화요일, 수요일, 목요일, 금요일, 토요일, 일요일}\}$ 이고, $|A| = 7$ 이다.

③ 영어 소문자의 집합을 B 라고 하면 $B = \{a, b, c, \dots, x, y, z\}$ 이고, $|B| = 26$ 이다.

④ 무지개 색의 집합을 C 라 하면 $C=\{\text{빨, 주, 노, 초, 파, 남, 보}\}$ 이며 개수는 $|C|=7$ 이다.

04 전체집합 U 가 양의 정수의 집합이고 $A = \{n | n^2 \leq 25, n \text{은 양의 정수}\}$, $B = \{1, 2, 3\}$ 일 때 $A - B$ 는?

- ① {1, 2, 3, 4, 5}

③ {4, 5}

② {1, 2, 3}

④ {1, 3, 5}

 집합 $A = \{n | n^2 \leq 25, n \text{은 양의 정수}\} = \{1, 2, 3, 4, 5\}$ 이다. 여기서 $B = \{1, 2, 3\}$ 을 빼면 $\{4, 5\}$ 이다.

05 다음 집합의 연산에서 등식이 성립하지 않는 것은?

- $$\begin{array}{ll} \textcircled{1} \quad A \cup \emptyset = A & \textcircled{2} \quad (A \cup B) \cap B = A \cap B \\ \textcircled{3} \quad (A \cup A^c) \cap B = B & \textcircled{4} \quad (A \cup B) \cap (A \cup A^c) = A \cup B \end{array}$$

 $(A \cup B) \cap B = B$ □

06 다음 알고리즘의 복잡도를 구하면?

```
for i=1 to n
    a=3*n+i*i;
```

- | | |
|---------------|-----------------|
| ① $O(n)$ | ② $O(n^2)$ |
| ③ $O(\log n)$ | ④ $O(n \log n)$ |

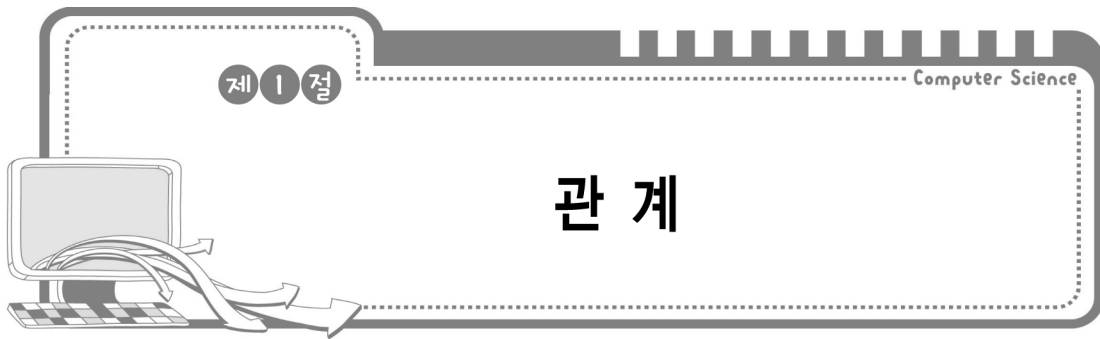
 i 가 1부터 n 까지 $a=3*n+i*i$ 문장이 수행된다. 즉 i 는 n 번 수행되므로 $O(n)$ 이다.

07 다음 알고리즘의 복잡도를 구하면?

```
int a=0;
int i, j;
for i=0 to n-1
    for j=3 to n+1
        a=a+j;
        print a;
    next j
next i
```

- | | |
|---------------|-----------------|
| ① $O(n)$ | ② $O(n^2)$ |
| ③ $O(\log n)$ | ④ $O(n \log n)$ |

 중첩 for문을 사용하여 $a=a+j$ 가 수행된다.
 첫 번째 for문에서는 i 가 0부터 $n-1$ 까지 n 번 수행되고, 두 번째 for문에서는 j 가 3부터 $n+1$ 까지 $n-1$ 번 수행된다. i 가 하나씩 증가할 때마다 j 가 반복수행되므로 $n(n-1)=n^2-n$ 이 되어 $O(n^2-n)=O(n^2)$ 이다.



관계는 서로 다른 집합에 있는 원소들 사이의 관련성을 의미한다. 집합에서 원소들 사이에는 아무런 순서가 없지만 관계에서는 원소들 간의 순서를 고려한 순서쌍의 형태로 표현한다. 관계는 서로 다른 집합에 속한 원소 간의 순서쌍으로 표현하고 이러한 순서쌍의 집합을 이항관계라고 한다.

두 집합 A, B 가 있을 때 집합 A 에서 B 로의 이항관계는 $A \times B$ 의 부분집합이다. 이때 $a \in A$ 이고 $b \in B$ 인 $(a, b) \in R$ 를 a 는 b 에 대해 R 의 관계가 있다고 하고 ${}_aR_b$ 로 나타내고, $(a, b) \notin R$ 일 경우에는 $\overline{{}_aR_b}$ 로 나타낸다.

관계는 데이터베이스나 알고리즘에서 많이 사용하는 개념인데 데이터베이스에서는 저장된 데이터들 사이의 연관성을 표현하거나 알고리즘에서는 변수들 간의 관계, 명령어와의 관계 등에서 많이 사용한다.

예제1

두 집합 $A = \{1, 2, 3, 4\}$, $B = \{1, 3, 5, 7\}$ 에 대한 이항관계 R 에서 ${}_aR_b$ 를 $a + 2 = b$ 라고 할 때 관계 R 를 순서쌍으로 나타내시오.

풀이

첫 번째 원소에 2를 더한 값이 두 번째 원소가 되는 순서쌍의 집합이다.
 $R = \{(1, 3), (3, 5)\}$

예제2

집합 $A = \{4, 5, 6\}$, $B = \{\text{이산수학}, \text{C언어}\}$ 일 때 이 두 집합 사이에 가능한 관계 R 를 구하시오.

풀이

$$A \times B = \{(4, \text{이산수학}), (4, \text{C언어}), (5, \text{이산수학}), (5, \text{C언어}), (6, \text{이산수학}), (6, \text{C언어})\}$$

정의역(domain), 공변역(codomain), 치역(range)

- 정의역: 집합 A 에서 집합 B 로 가는 이항관계 R 에 속한 순서쌍의 첫 번째 원소가 포함되어 있는 집합, 즉 집합 A

$$\text{dom}(R) = \{a \mid a \in A\}$$

- 공변역: 집합 A 에서 집합 B 로 가는 이항관계 R 에 속한 순서쌍의 두 번째 원소가 포함되어 있는 집합, 즉 집합 B

$$\text{codom}(R) = \{b \mid b \in B\}$$

- 치역: 집합 A 에서 집합 B 로 가는 이항관계 R 에 속한 순서쌍의 두 번째 원소들을 모아놓은 집합, 즉 공변역의 부분집합

$$\text{ran}(R) = \{b \mid (a, b) \in R\} \subseteq B$$

예제3

집합 $A = \{x \mid 1 \leq x \leq 3, x \text{는 정수}\}$, $B = \{x \mid 1 \leq x \leq 5, x \text{는 짝수}\}$ 일 때 A 에서 B 로 가는 관계 R 의 순서쌍을 나타내고 정의역, 공변역, 치역을 구하시오.

풀이

$$A = \{1, 2, 3\}, B = \{2, 4\} \text{이다.}$$

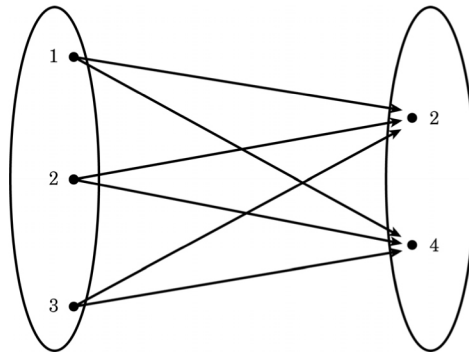
$$\text{따라서 관계 } R = \{(1, 2), (1, 4), (2, 2), (2, 4), (3, 2), (3, 4)\}$$

$$\text{정의역 } \text{dom}(R) = A = \{x \mid 1 \leq x \leq 3, x \text{는 정수}\}$$

$$\text{공변역 } \text{codom}(R) = B = \{x \mid 1 \leq x \leq 5, x \text{는 짝수}\}$$

$$\text{치역 } \text{ran}(R) = \{2, 4\} \text{이다.}$$

이 관계를 관계 그래프로 나타내면 다음과 같다. 집합 A 의 원소는 왼쪽에 나타내고 집합 B 의 원소는 오른쪽에 나타낸 후 순서쌍에 해당하는 원소들 사이를 화살표로 연결하면 된다.



[그림 2-1] 예제의 관계 그래프

예제 4

집합 $A = \{a, b, c, d\}$, $B = \{1, 2, 3, 4\}$ 일 때 A 에서 B 로의 관계가 $R = \{(a, 2), (a, 3), (b, 1), (b, 4), (c, 2), (c, 3), (d, 2)\}$ 일 때 관계 R 을 불 행렬로 나타내시오.

풀이

A 에서 B 로의 관계이므로 행에는 A 집합의 원소를, 열에는 B 집합의 원소를 대응시키면 된다. 따라서 1행에는 a , 2행에는 b , 3행에는 c , 4행에는 d 를 대응시키고 1열에는 1, 2열에는 2, 3열에는 3, 4열에는 4를 대응시킨다. 그런 다음 관계의 순서쌍이 존재하는 행렬 원소에는 1로 표현하고 관계의 순서쌍이 존재하지 않으면 0으로 표현하면 된다.

$$\begin{matrix} & 1 & 2 & 3 & 4 \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

어떠한 관계가 주어졌을 때 이 관계를 이용하여 다른 관계를 만들 수 있는데 이러한 방법에는 역관계(inverse relation)와 합성관계(composition relation)가 있다.

역관계

집합 A 에서 B 로의 관계 R 에 대한 역관계는 B 에서 A 로의 관계를 말한다. 즉 관계 R 의 순서쌍의 앞, 뒤 원소를 바꾸면 역관계가 되며 R^{-1} 로 나타내고 다음과 같이 정의한다.

$$R^{-1} = \{(b, a) \mid (a, b) \in R\}$$

역관계 R^{-1} 은 관계 R 가 존재할 때만 만들 수 있다. 또한 관계 R 의 정의역은 R^{-1} 의 치역이 되고 관계 R 의 치역은 R^{-1} 의 정의역이 된다.

예제5

두 집합 $A = \{1, 3, 5\}$, $B = \{3, 7, 9\}$ 일 때 관계 $R = \{(1, 3), (3, 7), (3, 9), (5, 1), (5, 9)\}$ 이면 관계 R 의 역관계 R^{-1} 은 무엇인지 구하시오.

풀이

역관계는 원래 관계에서 순서쌍의 앞, 뒤 순서를 바꾸면 된다.

$$R^{-1} = \{(3, 1), (7, 3), (9, 3), (1, 5), (9, 5)\}$$

예제6

집합 $A = \{1, 2, 3, 4, 5\}$ 일 때 $A \times A$ 에 대하여 $(a, b) \in R$ 인 관계 R 가 성립하려면 $a + b$ 가 5로 나누어떨어져야 한다.

- ① R , R 의 정의역, R 의 치역을 구하시오.
- ② R^{-1} , R^{-1} 의 정의역, R^{-1} 의 치역을 구하시오.

풀이

- ① $R = \{(1, 4), (2, 3), (3, 2), (4, 1)\}$
 R 의 정의역 $\text{dom}(R) = \{1, 2, 3, 4\}$
 R 의 치역 $\text{ran}(R) = \{1, 2, 3, 4\}$
- ② $R^{-1} = \{(4, 1), (3, 2), (2, 3), (1, 4)\}$
 R^{-1} 의 정의역 $\text{dom}(R) = \{1, 2, 3, 4\}$
 R^{-1} 의 치역 $\text{ran}(R) = \{1, 2, 3, 4\}$

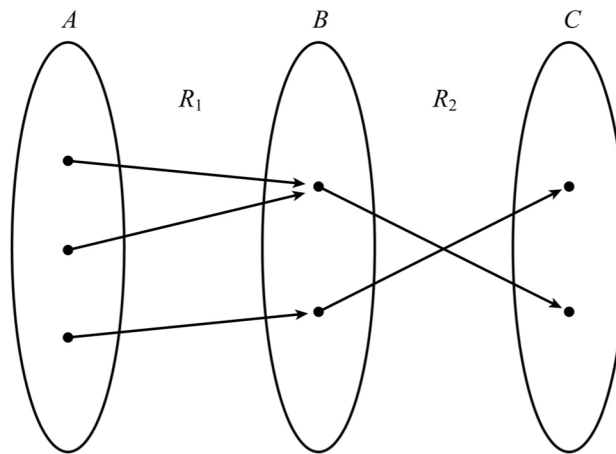
합성관계는 집합 A 에서 B 로의 관계 R_1 이 있고 집합 B 에서 C 로의 관계 R_2 가 있다면 R_1 을 적용한 후 R_2 를 적용하여 A 에서 C 로의 관계를 만드는 것이다.

합성관계

R_1 은 A 에서 B 로의 관계이고 R_2 은 B 에서 C 로의 관계일 때 합성관계 $R_2 \cdot R_1$ 은 R_1 을 적용한 후 R_2 를 적용하여 A 에서 C 로의 관계를 만든 것이다.

$$R_2 \cdot R_1 = \{(a, c) \mid \text{어떤 } b \text{는 } (a, b) \in R_1 \text{이고 } (b, c) \in R_2 \text{이다}\}$$

합성관계 $R_2 \cdot R_1$ 은 A 에서 B 로의 관계 R_1 과 B 에서 C 로의 관계 R_2 가 있을 때 이를 이용하여 A 에서 C 로의 관계를 만드는 것이다.



[그림 2-2] 합성관계 $R_2 \cdot R_1$

예제7

집합 $A = \{a, b, c\}$, $B = \{1, 2\}$, $C = \{c, d\}$ 일 때 $R_1 = \{(a, 1), (b, 2), (c, 1)\}$, $R_2 = \{(1, d), (2, c)\}$ 일 때 합성관계 $R_2 \cdot R_1$ 를 구하시오.

풀이

$(a, 1)$ 과 $(1, d)$ 에서 (a, d) , $(b, 2)$ 와 $(2, c)$ 에서 (b, c) , $(c, 1)$ 과 $(1, d)$ 에서 (c, d) 를 구할 수 있다. 따라서 $R_2 \cdot R_1 = \{(a, d), (b, c), (c, d)\}$ 이다.

확인 학습 문제

01 다음 관계를 순서쌍의 집합으로 올바르게 표현한 것은?

집합 $\{1, 3, 5\}$ 에서 관계 R 는 $x > y$ 이다. 이때 이 관계를 만족하는 $(x, y) \in R$ 이다.

- ① $\{(3, 1), (3, 3), (3, 5)\}$ ② $\{(3, 1), (5, 1), (5, 3)\}$
 ③ $\{(1, 1), (3, 3), (5, 5)\}$ ④ $\{(5, 1), (5, 3), (5, 5)\}$

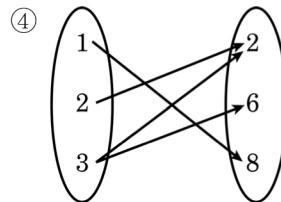
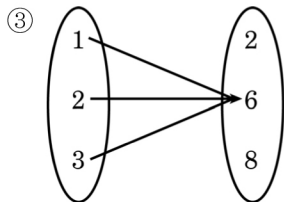
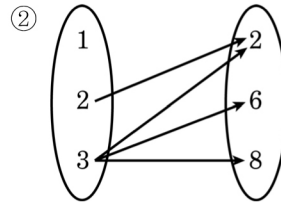
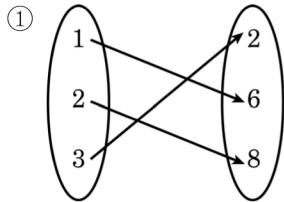
 x 와 y 가 R 의 원소이므로 x 와 y 는 각각 $\{1, 2, 3\}$ 에 해당하는 원소를 갖는다. 그 중에서 $x > y$ 라는 조건을 만족해야 하므로 이를 만족하는 순서쌍의 집합은 $\{(3, 1), (5, 1), (5, 3)\}$ 이다.

02 집합 $X = \{2, 4, 6\}$, $Y = \{2, 4, 6, 8\}$ 일 때 $x + y = 8$ 이라는 조건을 만족하는 순서쌍 (x, y) 를 올바르게 나열한 것은? (단, $x \in X$, $y \in Y$)

- ① $\{(2, 6), (4, 4), (6, 2)\}$ ② $\{(2, 4), (4, 2), (8, 4)\}$
 ③ $\{(2, 6), (6, 2), (6, 4)\}$ ④ $\{(2, 2), (2, 4), (2, 6), (4, 2), (4, 4)\}$

 집합 X 와 Y 의 원소 중 $x + y = 8$ 이라는 조건을 만족하는 순서쌍은 $\{(2, 6), (4, 4), (6, 2)\}$ 뿐이다.

03 집합 $X = \{1, 2, 3\}$, $Y = \{2, 6, 8\}$ 일 때 $x^2 \geq y$ 라는 조건을 만족하는 순서쌍 (x, y) 의 관계를 그래프로 나타낸 것은? (단, $x \in X$, $y \in Y$)



 $x^2 \geq y$ 라는 조건을 만족하는 순서쌍의 집합은 $\{(2, 2), (3, 2), (3, 6), (3, 8)\}$ 이다. 따라서 이러한 순서쌍을 그래프로 표현한 것은 ②이다.

용어정리

- 가중 그래프(weighted graph) : 그래프의 정점을 연결하는 간선마다 일정한 값을 할당할 수 있으며 이러한 그래프를 가중 그래프라 한다.
- 간접증명법 : 직접증명법 외의 다른 증명 방법들을 모두 간접증명법이라고 하는데, 간접증명법은 증명하고자 하는 명제를 논리에 어긋나지 않는 범위에서 증명하기 쉬운 명제로 변화시켜 증명하는 것이다.
- 결정적 유한상태 자동장치 : 각 상태와 입력에 따라 전이함수에 의해 다음 상태를 유일하게 결정하는 것을 말한다.
- 계승(factorial) : 1부터 n 개의 양의 정수를 모두 곱한 것으로 n 계승이라 하고 $n!$ 으로 나타낸다.
- 공변역(codomain) : 집합 A 에서 집합 B 로 가는 이항관계 R 에 속한 순서쌍의 두 번째 원소가 포함되어 있는 집합, 즉 집합 B
- 공집합(empty set) : 원소가 하나도 없는 집합을 말하고, $\{ \}$ 또는 $\emptyset$ 으로 나타낸다.
- 교집합 : 집합 A , B 가 있을 때 집합 A 와 B 에 모두 속하는 원소로 구성되는 집합이다.
- 나무(tree) : 나무는 사이클이 없는 연결 그래프이다. 또 나무를 구성하는 두 정점 사이에는 두 정점을 연결하는 단순 경로가 있다.
- 대우증명법 : 증명하고자 하는 명제의 대우명제를 이용하는 방법이다. 명제의 함축 $p \Rightarrow q$ 가 참이면 그 대우인 $\sim q \Rightarrow \sim p$ 도 참이며 두 명제는 서로 동치라는 점을 이용한다.
- 동형 그래프(isomorphic graph) : 모양은 다르지만 두 그래프가 똑같은 정점과 똑같은 간선으로 구성되어 있는 그래프를 말한다.
- 멱집합(power set) : 임의의 집합 A 의 모든 부분집합들을 원소로 갖는 집합을 의미하며 $P(A)$ 와 같이 표현한다.
- 명제 : 참이나 거짓을 명확히 판단할 수 있는 문장을 말한다.
- Mod 함수 : 나눗셈을 한 결과에서 나머지 값만을 구하는 함수이다.
- 모순증명법 : 주어진 명제를 부정한 뒤 그 식을 전개할 때 결론이 모순임을 보여 명제가 참임을 증명하는 방법이다.
- 무한집합(infinite set) : 집합 A 의 원소 개수가 무한할 경우를 말한다.
- 반대칭성 : 어떤 $a, b \in A$ 에 대해 $(a, b) \in R$ 일 때 $(b, a) \in R$ 이면서 $a = b$ 인 관계

- 반례증명법 : 주어진 명제에 모순이 되는 예를 하나 보임으로써 명제가 거짓임을 증명하는 방법이다. 즉 명제를 거짓이 되게 하는 단 하나의 예라도 존재하면 그 명제는 거짓이 된다.
- 반복법 : 점화식에서 가장 많이 이용하는 방법이고 해를 구하기 위해 앞의 항을 반복적으로 나타내는 방법이다.
- 반순서관계 : 집합 A 에 대한 관계 R 가 반사성, 반대칭성, 추이성이 성립하면 관계 R 를 반순서관계라고 한다.
- 벤 다이어그램(venn diagram) : 집합 사이의 연산이나 포함 관계를 보여주기 위한 그림으로 부분집합, 합집합, 교집합 등의 연산을 그림으로 표현하여 효율적이고 쉽게 설명할 수 있다.
- 부분 그래프(subgraph) : 어떤 그래프 G 를 구성하는 일부 정점과 간선으로만 구성된 그래프를 말한다.
- 부분신장 그래프(spanning graph) : 부분 그래프 중에서 그래프 G 의 정점을 모두 포함한 부분 그래프이다.
- 불 대수 : 논리 게이트의 동작을 수학적 표시법으로 표현한 것으로 0과 1을 입력값으로 갖는 논리계산을 형식화한 것이다.
- 비결정적 유한상태 자동장치(NFA ; Nondeterministic Finite state Automata) : 상태와 입력에 대하여 다음 상태가 여러 개이거나 없을 수도 있어서 비결정적이라고 부른다.
- 생성규칙(production rule) : 문장을 생성할 수 있는 일정한 규칙을 말하며 $x \rightarrow y$ 의 형태로 나타낸다.
- 수학적 귀납법 : 자연수 n 에 관한 명제 $p(n)$ 이 모든 자연수 n 에 대해 만족하는 것을 다음 과정으로 증명하는 방법이다.
- 순열(permutation) : 서로 다른 원소의 집합에서 순열은 그 원소들의 순차적인 배열이다. n 개의 서로 다른 원소로 이루어진 집합에서의 r -순열의 개수 $P(n, r)$ 는 $P(n, r) = \frac{n!}{(n-r)!}$ 이다.
- 순차회로 : 입력값뿐만 아니라 시스템의 상태에 따라서도 출력이 달라지는 회로를 말한다.
- 술어 : 술어는 유한개의 변수들을 가진 문장이며 특정 값을 변수에 대입하면 명제가 된다. 개체 변수의 정의역은 그 변수 자리에 대입할 수 있는 모든 값들의 집합이다.