서론

컴퓨터의 사용이 일반화되고, 이로 인해 컴퓨터에 의한 계산이 어디에서든지 이루어지고 있다. 이러한 컴퓨터의 발전은 컴퓨터의 기초학문이라고 할 수 있는 컴퓨터 이론에 그 바탕을 두고 있다. 본서에서는 다음의 근본적인 질문을 고려한다.

- 컴퓨터로 무엇을 계산할 수 있고 무엇을 계산할 수 없는가? 놀랍게도 우리가 자연스럽게 물을 수 있는 문제 중에서 컴퓨터로 풀 수 없는 문제가 많이 있다.

위의 질문에 답하기 위해서는 먼저 컴퓨터로 계산한다는 것이 무엇인가를 규정해야 한다. 이를 위해 컴퓨터(계산)의 이론적인 모델들을 고려한다. 계산능력이 커지는 순서로 유한 오토마타, 내리누름 오토마타, 튜링기계 등을 배우게 된다. 또한 이러한 모델들과 밀접한 관련을 가진 문법(grammar)을 다루게 된다.

1 관계

집합은 원소들을 모아놓은 것이다. 자연수의 집합 N은 $\{0, 1, 2, \ldots\}$이다. 집합 A의 멱집합(power set) 2^A은 A의 모든 부분집합들의 집합이다. 따라서 $|2^A| = 2^{|A|}$ 이다. 예를 들어, $A = \{0, 1\}$이면 $2^A = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$이다.

순서쌍 (a, b)는 두 원소 a, b에 순서를 부여한 쌍이다. 여기에서 a, b는 같은 원소일 수 있다. 집합 A, B의 Cartesian(카티젼) 곱 $A \times B$는 $a \in A, b \in B$인 모든 순서쌍 (a, b)의 집합이다. 원소가 3개 이상일 때, 순서열 $(a_1, \ldots, a_n)$이라고 부른다.

- 두 집합 A, B에 대한 이원관계(binary relation)는 $A \times B$의 부분집합이다.
- 집합 A에서 집합 B로의 함수(function)는 A, B에 대한 이원관계 R로서 각 원소 $a \in A$에 대해 순서쌍 $(a, b) \in R$이 하나만 존재하는 성질을 가진다. A에서 B로의 함수를 일반적으로

$f : A \to B$로 적는다.

- 부분함수(partial function)는 각 원소 $a \in A$에 대해 순서쌍 $(a, b) \in R$이 하나 있거나 없는 경우이다.

예제 1 $A = \{a, b, c\}$*이고* $B = \{0, 1\}$*이라고 하자.* $\{(a, 0), (b, 0), (b, 1)\}$*은 이원관계이고,* $\{(a, 0), (b, 1), (c, 0)\}$*은 함수이고,* $\{(a, 1), (c, 0)\}$*은 부분함수이다.*

$R \subseteq A \times A$인 관계 R을 집합 A에서의 관계라고 부른다. 집합 A에서의 관계 R에 대하여 다음과 같은 성질을 고려하자.

- 모든 $a \in A$에 대하여 $(a, a) \in R$이면 R은 반사적(reflexive)이다.
- $(a, b) \in R$이면 $(b, a) \in R$을 만족하면 R은 대칭적(symmetric)이다.
- $(a, b) \in R$이고 $(b, c) \in R$이면 $(a, c) \in R$을 만족하면 R은 이행적(transitive)이다.

반사적이고 대칭적이고 이행적인 관계를 동치관계(equivalence relation)라고 부른다. R을 집합 A에서의 동치관계라고 하자. R에

의해 원소 $a \in A$와 동치인 원소들의 집합을 동치류(equivalence class)라고 부르고 $[a]$로 표시한다. 즉 $[a] = \{b : (a,b) \in R\}$. 동치관계 R은 집합 A를 동치류들로 분할한다. 즉 $A_1, A_2, \ldots$을 R의 동치류라고 하면

- $A_i \neq \emptyset$
- $i \neq j$에 대해 $A_i \cap A_j = \emptyset$
- $\bigcup_i A_i = A$

를 만족한다.

예제 2 *자연수의 집합 N에서의 관계 $R = \{(a,b) : a \equiv b \pmod 4)\}$는 동치관계이고 다음 네 개의 동치류가 존재한다.*

$$\begin{aligned}
[0] &= \{0, 4, 8, 12, \ldots\} \\
[1] &= \{1, 5, 9, 13, \ldots\} \\
[2] &= \{2, 6, 10, 14, \ldots\} \\
[3] &= \{3, 7, 11, 15, \ldots\}
\end{aligned}$$

2 그래프

그래프 $G = (V, E)$는 정점(vertex)의 집합 V와 간선(edge)의 집합 E로 구성된다. 방향그래프(directed graph)에서 간선은 $a, b \in V$인 순서쌍 (a, b)이다. 무방향그래프(undirected graph)에서 간선은 $a, b \in V$인 집합 $\{a, b\}$이다. 무방향그래프에서 간선을 (a, b)로 표시하기도 한다.

그래프 $G = (V, E)$ 상의 경로(path)는 $(v_{i-1}, v_i) \in E$ $(1 \leq i \leq k)$를 만족하는 정점의 순서열 $(v_0, \ldots, v_k)$이다 (또는 간선의 순서열). 이 경로의 길이는 k이다. 방향그래프에서 경로 $(v_0, \ldots, v_k)$가 $k > 0$과 $v_0 = v_k$를 만족하면 사이클(cycle)이라 부른다. 또한 $v_1, \ldots, v_k$가 서로 다르면 단순 사이클이다. 무방향그래프에서 경로 $(v_0, \ldots, v_k)$가 $k > 0$과 $v_0 = v_k$를 만족하고 $v_1, \ldots, v_k$가 서로 다르면 사이클이라 부른다.

사이클이 없는 연결된 무방향그래프를 트리(tree)라고 부른다. 루트 노드(root node)를 가지고 각 노드의 자식 노드(child node)가 2개

이하인 트리가 이진 트리(binary tree)이다. 자식이 없는 노드를 단말 노드라고 부른다. 이진트리에서 각 노드의 두 자식을 왼쪽 자식과 오른쪽 자식이라고 부른다. 루트 노드에서 임의의 노드 x까지 경로의 길이를 x의 깊이라고 한다. 트리에서 단말 노드의 깊이의 최대 값을 트리의 깊이(또는 높이)라고 부른다.

3 언어

언어를 정의하기 위하여 먼저 알파벳과 스트링을 정의한다.

- 알파벳은 글자들의 유한 집합이다. 일반적으로 알파벳을 Σ로 표시하고 $\{0, 1\}$, $\{a, b, c, \ldots, z\}$ 등이 그 예이다.
- 알파벳 Σ상의 스트링은 Σ에 있는 글자들의 유한 열이다. 예를 들면, 1010과 00111은 $\Sigma = \{0, 1\}$상의 스트링들이고, abcab와 automata는 $\Sigma = \{a, b, c, \ldots, z\}$상의 스트링들이다.

- 스트링 w의 길이는 w에 있는 글자들의 개수이고 $|w|$로 표시된다. 즉 $|abcab| = 5$. 길이가 0인 스트링을 공스트링이라 부르고 ϵ로 표시한다.
- 공스트링을 포함하여 알파벳 Σ상의 모든 스트링의 집합을 Σ^*로 표시한다. 즉, $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, \ldots\}$.

스트링에 대한 두 가지 연산을 정의한다.

- 두 스트링 x, y의 접합(concatenation) $x \cdot y$ (또는 xy)는 x 뒤에 y를 붙인 것이다. 즉 $x = 011$이고 $y = 01$이면, $xy = 01101$이고 $yx = 01011$이다. 스트링 w에 대해서 $w^0 = \epsilon$이고, $w^k = w^{k-1}w$ $(k \geq 1)$이다. 따라서 $w = 011$이면, $w^2 = 011011$이다.
- 스트링 w의 역(reverse) w^R은 w를 역순으로 나열한 스트링이다. 즉 $w = 011$이면, $w^R = 110$이다. 두 스트링 u, v에 대하여 $(uv)^R = v^R u^R$ 이다.

스트링에 대한 몇 가지 용어를 정의한다.

- v가 스트링 w의 일부분일 때(즉 $w = xvy$), v는 w의 부분스트링이다. 즉 011은 00111의 부분스트링이다.
- v가 w의 앞부분에 있는 부분스트링일 때(즉 $w = vy$), v는 w의 어두(prefix)이다. 011의 어두는 0, 01, 011이다.
- v가 w의 뒷부분에 있는 부분스트링일 때(즉 $w = xv$), v는 w의 어미(suffix)이다. 011의 어미는 1, 11, 011이다.

알파벳 Σ상의 스트링의 집합, 즉 Σ^*의 부분집합을 언어(language)라고 부른다. 이것은 언어에 대한 가장 추상적인 정의인데, 구체적인 예로 프로그래밍 언어를 들 수 있다. C 프로그램이란 C 언어의 문법에 맞는 스트링이다. 따라서 C 언어는 C 프로그램(즉 스트링)들의 집합이 된다. 언어들의 집합을 언어 종류(language class)라고 부른다.

언어는 집합이므로 언어들에 대하여 합집합, 교집합, 여집합 등의 연산을 적용할 수 있다. 이에 더하여 다음 두 가지 연산을 정의한다.

- 언어 L_1, L_2의 접합 $L_1 \cdot L_2$ (또는 $L_1 L_2$)는

$$\{uv : u \in L_1, v \in L_2\}$$

이다. 즉 $L_1 = \{0, 00, 000\}$이고 $L_2 = \{1, 11\}$이면,
$L_1 L_2 = \{01, 001, 0001, 011, 0011, 00011\}$이다. $L^0 = \{\epsilon\}$이고,
$L^k = L^{k-1}L\ (k \geq 1)$이다. 즉 $L = \{1, 11\}$이면,
$L^2 = \{11, 111, 1111\}$이다.

- 언어 L의 Kleene(클리니) 곱 L^*는 L을 0번 이상 접합하여 만들어지는 모든 스트링의 집합이다. 즉,

$$L^* = L^0 \cup L^1 \cup L^2 \cup \cdots.$$

$L = \{0\}$이면, $L^* = \{\epsilon, 0, 00, 000, \ldots\}$이다. 따라서 Σ^*는 Σ의 Kleene 곱인 것을 알 수 있다. LL^*를 L^+로 표기하기도 한다 (즉 L^+는 L을 1번 이상 접합하여 만들어지는 모든 스트링의 집합).

4 증명 기법

증명 기법으로 수학적 귀납법과 비둘기집 원리를 많이 사용하게 된다.

수학적 귀납법(mathematical induction): 모든 자연수 n에 대하여 A가 성립함을 다음과 같이 증명할 수 있다.

1. 귀납 기초(basis): $n = 0$일 때, A가 성립함을 보인다.
2. 귀납 가설(induction hypothesis): 임의의 $n \geq 0$에 대하여 (또는 0부터 n까지 모든 수에 대하여) A가 성립한다고 가정한다.
3. 귀납 과정(induction step): $n + 1$에 대하여 A가 성립함을 보인다.

예제 3 *높이가 n인 이진 트리에서 단말 노드의 개수는 2^n 이하임을 수학적 귀납법으로 증명하라.*

- *귀납 기초: 높이가 0인 이진 트리의 단말 노드는 1 개이다.*
- *귀납 가설: 높이가 $n \geq 0$ 이하인 이진 트리의 단말 노드 개수가 2^n 이하라고 가정한다.*
- *귀납 과정: T는 높이가 $n + 1$인 이진 트리라고 하자. T의 루트 노드의 왼쪽 서브트리와 오른쪽 서브트리는 높이가 n 이하이다. 귀납 가설에 의해 왼쪽 서브트리의 단말 노드의 개수는 2^n*

이하이고, 오른쪽 서브트리도 마찬가지이다. 따라서 이진 트리 T 의 단말 노드 개수는 2^{n+1} 이하이다.

비둘기집 원리(pigeonhole principle): A와 B가 유한집합이고 $|A| > |B|$이면, A에서 B로의 일대일 함수가 존재하지 않는다. 즉, 비둘기가 $|A|$ 개이고, 비둘기집이 $|B|$ 개이면, 어떤 집에는 비둘기를 2 개 이상 넣어야 한다.

비둘기집 원리의 일반적인 형태는 다음과 같다. n 개의 비둘기를 c 개의 비둘기집에 넣으려면, 어떤 집에는 $\lceil n/c \rceil$ 개 이상의 비둘기를 넣어야 한다.

예제 4 *100 명의 사람이 있으면, 그 중 같은 달에 태어난 사람이 적어도* $\lceil 100/12 \rceil = 9$ *명 있다.*

5 연습 문제

1. 유한집합 A에 대하여 $|2^A| = 2^{|A|}$임을 수학적 귀납법으로 증명하라.

2. 두 명 이상 모인 모임에서 아는 사람의 수가 같은 사람이 적어도 2 명 있음을 증명하라. 여기에서 '안다'는 것은 '서로 안다'를 의미한다. (비둘기집 원리 사용)