



1

형식 언어

- ▶ 언어는 한글을 비롯하여 영어 등 우리가 일상 생활에서 자주 사용하는 자연어(Natural Language)와 오토마타를 이용하여 만들어지는 이론적인 언어인 형식 언어(Formal Language)의 2가지가 있음

1

## 형식 언어

2

## 자연어(Natural Language)

- ▶ 일상적으로 사용하는 한국어나 영어 등과 같은 언어
- ▶ 자음과 모음 또는 알파벳을 여러 방법으로 결합한 것

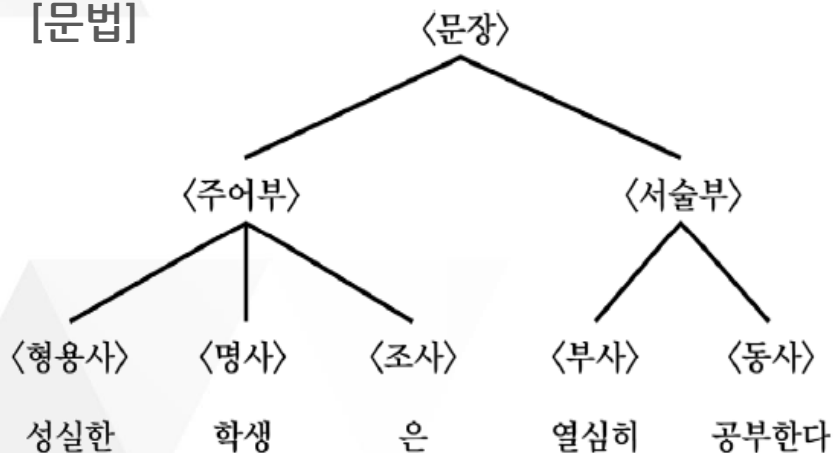
## 2 자연어(Natural Language)

- ▶ 문법은 언어가 따라야 할 규칙
- ▶ 문법이란 우리가 사용하는 자연어에서의 문법을 일컬음
- ▶ 우리말이나 영어의 경우에는 부정확하고 애매한 경우가 상당히 많음
- ▶ 컴퓨터에서 쓰이는 문법에는 애매성이 배제되어야 하며 일정한 규칙을 따라 엄밀하게 정의되어야 함
- ▶ 문법은 어떤 문장이 제대로 작성되었는지의 여부를 판정하는 기준이 됨

## 2 자연어(Natural Language)

▶ 예) 문장 ‘성실한 학생은 열심히 공부한다’의 구조

[문법]



- 여기서 하나의 <문장>은 <주어부>와 <서술부>로 나눌 수 있고 이들은 다시 <형용사>, <명사>, <조사>와 <부사>, <동사>로 나눌 수 있음
- 규칙에 따라 이들을 단어로 바꾸어 모아 놓으면 ‘성실한 학생은 열심히 공부한다.’라는 문장을 얻음

## 2 자연어(Natural Language)

▶ 예) 문장 ‘성실한 학생은 열심히 공부한다’의 구조

〈문장〉

〈주어부〉 〈서술부〉

〈형용사〉 〈명사〉 〈조사〉 〈부사〉 〈동사〉

성실한 〈명사〉 〈조사〉 〈부사〉 〈동사〉

성실한 학생 〈조사〉 〈부사〉 〈동사〉

성실한 학생 은 〈부사〉 〈동사〉

성실한 학생 은 열심히 〈동사〉

성실한 학생 은 열심히 공부한다

## 3 형식 언어(Formal Language)

- ▶ 유한한 기호들의 집합을 이용해 만들어지는 유한한 **문자열**의 집합
- ▶ 언어에 관한 이론을 체계적으로 전개하기 위해 잘 정의된 언어
- ▶ 구조, 범위 등이 명확히 규정되어 있는 언어이며 자연 언어의 문법 구조를 수학적 측면에서 형식화한 것으로서 자연 언어보다 훨씬 간단한 구조의 인공 언어
- ▶ 체계적으로 잘 정의된 규칙에 따라 결정되는 언어

## 3 형식 언어(Formal Language)

- ▶ 인공적으로 만들어진 언어
  - 예) 컴퓨터 프로그래밍 언어(C, JAVA 등)
- ▶ 구문 법칙을 가지며 이러한 구문 법칙에 따라 문자열들이 정확한 단어 또는 문장을 이루고 있는지를 판정
- ▶ 기호 : 언어에 있어서 가장 기본이 되는 요소
  - 예) 0, 1,  $\neg$ , A

## 3 형식 언어(Formal Language)

- ▶ 예) 기호 집합이  $V=\{a, b, c\}$ 일 때  
여기서  $a, b, c$ 는 기호이고  $a, b, c, aa, ab, ac,$   
 $ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, abb,$   
 $abc, \dots$  등은  $V$ 에 속하는 단어임

## 4 알파벳(Alphabet)과 문자열(String)



정의

- 공집합이 아닌 기호들의 유한집합  $\Sigma$ 를 **알파벳**이라고 함
- 알파벳에 포함된 기호들의 유한한 순서쌍을 **문자열**이라고 함
- 문자열 중에서 아무런 기호도 포함하지 않은 문자열을 **공 문자열**(Empty String)이라 하며  $\epsilon$ (Epsilon) 또는  $\lambda$ (Lambda)로 표기함 (문자열의 길이가 0)

## 4 알파벳(Alphabet)과 문자열(String)

- 알파벳  $\Sigma$ 는 기호들의 유한 집합이며  
언어는 문장들의 집합으로 정의되며  
알파벳은 문장을 이루는 기본적인 기호
- 알파벳의 예  
:  $T_1 = \{0, 1\}$ ,  
 $T_2 = \{ \neg, \wedge, \dots, \vee, \vdash, \dots, \dashv, \mid \}$
- 문자열(String)  
: 알파벳  $T$ 에 속하는 기호나  
 $T$ 에 속하는 하나 이상의 기호 연결

## 4 알파벳(Alphabet)과 문자열(String)

- ▶ 예) 알파벳  $\Sigma = \{0, 1\}$ 일 때  $\lambda, 0, 1, 10, 010, 0001, 10010$  등은 모두 알파벳  $\Sigma$ 를 통해서 만들어 낼 수 있는 문자열임
- ▶ 예)  $T = \{a, b\}$ 일 때,  $a, b, aa, ab, ba, bb, aaa, \dots$ 는 모두  $T$ 에서 만들 수 있는 문자열임

## 4 알파벳(Alphabet)과 문자열(String)

- ▶ 문자열의 길이
  - 문자열을 이루는 기호들의 개수
  - 어떤 문자열  $w$ 의 길이는  $|w|$ 로 표기함

▶ 예)  $x = 0110$        $|x| = 4$   
       $y = \text{dog}$          $|y| = 3$   
       $z = \text{house}$       $|z| = 5$

▶ 예) 문자열  $w = \text{abc}$ 의 길이를 구하시오  
풀이)  $|w| = 3$

5  $\Sigma^n, \Sigma^+, \Sigma^*$ , 언어(Language)

## ◇ 정의

- $\Sigma$ 를 유한 알파벳이라고 하자.  
양의 정수  $n$ 에 대하여  $\Sigma^n, \Sigma^+, \Sigma^*$ 는 다음과 같이 정의
  - $\Sigma^n$  : 길이가  $n$ 인  $\Sigma$ 상의 기호들의 결합으로부터 만들어지는 모든 문자열의 집합
  - $\Sigma^+$  : 길이가 적어도 1 이상인  $\Sigma$ 상의 기호들의 결합으로부터 만들어지는 모든 문자열의 집합( $\lambda$ 는 포함하지 않음)
  - $\Sigma^*$  : 길이가 0 이상인  $\Sigma$ 상의 기호들의 결합으로부터 만들어지는 모든 문자열의 집합
- $\Sigma^*$ 의 임의의 부분 집합을 언어라고 함

※ 언어는 기호들을  
원소로 갖는 집합

5  $\Sigma^n, \Sigma^+, \Sigma^*$ , 언어(Language)

▶ 예)  $\Sigma = \{0, 1\}$ 일 때,  
다음과 같은 언어들을 정의할 수 있다  
각각의 언어에 대한 예를 두가지씩 들어보시오

- ①  $L = \{0, 1\}$
- ②  $L = \{0, 00, 1, 11, 01, 010, 0001\}$
- ③  $L = \{0^n 1^n \mid n \geq 1\}$
- ④  $L = \{0^n 110^n \mid n \geq 1\}$

5  $\Sigma^n, \Sigma^+, \Sigma^*$ , 언어(Language)

▶ 예)  $\Sigma = \{0, 1\}$ 일 때,  
 다음과 같은 언어들을 정의할 수 있다  
 각각의 언어에 대한 예를 두가지씩 들어보시오

- ①  $L = \{0, 1\}$
- ②  $L = \{0, 00, 1, 11, 01, 010, 0001\}$
- ③  $L = \{0^n 1^n \mid n \geq 1\}$
- ④  $L = \{0^n 110^n \mid n \geq 0\}$

풀이)

- ① 0, 1
- ② 0, 00
- ③ 01, 0011
- ④ 11, 0110

## 6 언어의 연산(Operations Of Languages)



정의

- 두 형식 언어  $L_1$ 과  $L_2$ 가 있을 때,  
다음과 같은 연산들을 정의한다.

①  $L_1$ 과  $L_2$ 의 **접속**(Concatenation)은  
 $L_1$ 의 문자열  $x$ 와  $L_2$ 의 문자열  $y$ 를 접속시켜 만든  
언어들의 집합

$$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$$

## 6 언어의 연산(Operations Of Languages)



정의

- 두 형식 언어  $L_1$ 과  $L_2$ 가 있을 때,  
다음과 같은 연산들을 정의한다.

- ②  $L_1$ 과  $L_2$ 의 **교집합**은  
 $L_1$ 과  $L_2$ 에 동시에 속해 있는 문자열의 집합

$$L_1 \cap L_2 = \{x \mid x \in L_1 \wedge x \in L_2\}$$

## 6 언어의 연산(Operations Of Languages)



정의

- 두 형식 언어  $L_1$ 과  $L_2$ 가 있을 때,  
다음과 같은 연산들을 정의한다.

③  $L_1$ 과  $L_2$ 의 **합집합**은  
 $L_1$  또는  $L_2$ 에 속해 있는 문자열의 집합

$$L_1 \cup L_2 = \{x \mid x \in L_1 \vee x \in L_2\}$$

## 6 언어의 연산(Operations Of Languages)

- ▶ 예) 두 형식 언어  $L_1$ 과  $L_2$ 가  
 $L_1 = \{0, 00\}$ ,  $L_2 = \{1, 11\}$ 이라고 할 때  
 $L_1 L_2$ 와  $L_2 L_1$ 를 구하시오.

풀이)

$L_1 L_2 = \{01, 011, 001, 0011\}$

$L_2 L_1 = \{11, 111, 1111\}$

➡  $L_1 L_2 \neq L_2 L_1$

## 6 언어의 연산(Operations Of Languages)

- ▶ 예) 두 형식 언어  $L_1$ 과  $L_2$ 가  
 $L_1 = \{0, 01\}$ ,  $L_2 = \{2, 12\}$ 이라고 할 때  
 $L_1L_2$ 와  $L_2L_1$ 를 구하시오

풀이)

$L_1L_2 = \{02, 012, 0112\}$

$L_2L_1 = \{20, 201, 120, 1201\}$

## 2 형식 문법

## 1 형식 문법(Formal Grammar)

- ▶ 형식 언어의 문자열들을 생성해 낼 수 있는 유한개의 **규칙**
- ▶ 프로그래밍 언어의 생성 규칙을 추상화한 개념
- ▶ 형식 언어를 정의하는 방법
- ▶ 유한개의 규칙을 이용해서 특정 언어에 해당하는 문자열들을 생성하거나 기존의 문자열이 특정 언어에 포함되는지를 판단함

## 2 구구조 문법(Phrase-structure Grammar)

### 정의

- 다음과 같이 4개의 원소의 순서쌍으로 정의되는  $G=(V, T, P, S)$ 를 **구구조 문법**이라고 함
  - $V$  : 비단말 기호들의 유한집합(대문자)
  - $T$  : 단말(Terminal) 기호들의 유한집합(소문자)
  - $P$  : 생성 규칙(Production)
  - $\alpha \rightarrow \beta, \alpha, \beta \in (V \cup T)^*$ ,  
 $\alpha$ 는 적어도 하나의 비단말 기호를 포함
  - $S$  :  $V$ 에 속하는 변수로써 시작 기호( $S \in V$ )

## 2 구구조 문법(Phrase-structure Grammar)

- ▶ 비단말 기호  
: 문법에 의해 생성되는 언어를 정의하기  
위한 중간 단계를 나타내는데 사용되는 문자열
- ▶ 단말 기호 : 언어를 구성하는 가장 기본적인 기호
- ▶ 시작기호  
: 문법에서 문장을 생성할 때  
생성의 시작 지점을 나타내는 비단말 기호

## 3 유도(Derivation)

▶ 정의

- 한 문자열에서 생성 규칙을 한번 적용해서 다른 문자열로 바꾸는 것
- 생성 규칙이  $\alpha \rightarrow \beta$ 이고  $x, y \in (V \cup T)^*$ 라면,  
 $x\alpha y$ 에  $\alpha \rightarrow \beta$ 를 적용해서  
 $x\alpha y$ 에  $x\beta y$ 로 바꾸어 쓸 수 있음  
 이때  $x\alpha y$ 에서  $x\beta y$ 로 **유도**되었다고 하며  
 $x\alpha y \Rightarrow x\beta y$ 로 표기

[참고]  $a_1 \Rightarrow a_2 \Rightarrow \dots \Rightarrow a_n$  이라면

$a_1 \xRightarrow{*} a_n$ 로 표기함.

## 3 유도(Derivation)

- ▶ 예)  $G=(V, T, P, S)$ 이고  $V=\{S\}$ ,  $T=\{1, 2\}$ ,  
 $P=\{S \rightarrow 1S2, S \rightarrow 12\}$ 에서 단어 111222를  
유도하시오

풀이)

$$S \Rightarrow 1S2 \Rightarrow 11S22 \Rightarrow 111222$$

## 4 형식 문법에 의해 생성된 언어

- ▶ 형식 문법에서 생성 규칙은 하나의 문자열을 다른 문자열로 바꾸어 주는 역할을 함
- ▶ 문법에서 정의된 생성 규칙을 이용하여 생성되는 문자열의 집합을 해당 문법에 의해 생성된 언어라고 부름

## 4 형식 문법에 의해 생성된 언어

◇ 문법  $G$ 에 의해 생성된 언어  $L(G)$

◇ 정의

- $G=(V, T, P, S)$ 를 문법이라 할 때,  
 $G$ 에 의한 생성되는 언어  $L(G)$ 는  
시작 기호  $S$ 로부터 유도될 수 있는  
모든 단말들로 구성된 문자열 집합으로서  
다음과 같이 정의함

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

## 4 형식 문법에 의해 생성된 언어

- ▶ 예)  $G = (\{S, A\}, \{a, b\}, \{S \rightarrow Ab, S \rightarrow Aa, A \rightarrow a\}, S)$   
일 때  $L(G)$ 는 무엇인가?

풀이)

$S \Rightarrow Ab \Rightarrow ab$

$S \Rightarrow Aa \Rightarrow aa$ 이므로

$L(G) = \{aa, ab\}$

## 4 형식 문법에 의해 생성된 언어

▶ 예) 언어  $\{a^m b^n \mid m, n \text{은 음이 아닌 정수}\}$ 를 생성하는 문법은?

풀이)

$G = (\{S\}, \{a, b\}, \{S \rightarrow aS, S \rightarrow Sb, S \rightarrow \epsilon\}, S)$

언어에서 나타나는 문자  $a$ 가  $m$ 번 반복되고 난 후에  $b$ 가  $n$ 번 반복됨.  $m$ 과  $n$ 은 다른 정수일 수 있으므로  $a$ 가  $m$ 번 반복되는 것을 표현할 수 있는 생성규칙은  $S \rightarrow aS, S \rightarrow \epsilon$  임, 다음 뒤에  $b$ 가  $n$ 번 반복되는 것을 나타내기 위해서는 생성규칙  $S \rightarrow Sb$ 를 두면 됨

유도과정의 예는 다음과 같음

$$\begin{aligned} S &\Rightarrow aS \Rightarrow aaS \Rightarrow aaSb \\ &\Rightarrow aaSbb \Rightarrow aaSbbb \Rightarrow aabbbb \end{aligned}$$

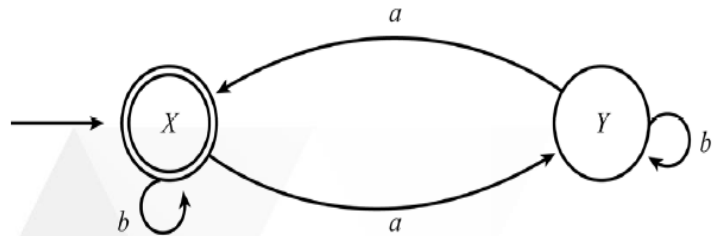
# 3 언어와 자동 장치

- ▶ 이론적인 계산 모델인 오토마타 중에서 유한 오토마타는 컴파일러의 어휘분석을 수행하는데 있어서 결정적인 역할을 함
- ▶ 오토마타, 형식 언어, 문법은 상호 밀접한 관계에 있는데 각종 컴퓨터 프로그램 언어들이 정해진 문법에 따른 형식 언어에 기반을 두고 만들어졌기 때문임

## 1 언어와 자동 장치

- ▶ 언어 L이 정규 언어가 되려면 L의 문자열을 수락하는 결정적 유한 상태 자동 장치가 있어야 함

DFA 상태도가 있을 때 상태 X에서 x가 입력되면 상태 X'로 가는 간선이 있다면  $X \rightarrow xX'$ 의 생성을 만들, 또 상태 X에서 x가 입력되면 수락 상태로 가는 간선이 있다면  $X \rightarrow x$ 의 생성을 만들, 문법  $G=(N, T, P, \sigma)$ 는 정규 문법이라 하면 수락하는 문자열의 집합은  $L(G)$ 와 동일함 ( $\sigma$ =초기 상태)

단말 기호 :  $a, b$ 비단말 기호 :  $X, Y$ 상태 :  $X, Y$ 초기 상태 :  $X$ (시작기호)
$$\begin{aligned}
 &X \rightarrow bX \\
 &X \rightarrow aY \\
 &Y \rightarrow aX \\
 &Y \rightarrow bY \\
 &X \rightarrow \lambda \\
 &Y \rightarrow a
 \end{aligned}$$


[DFA 상태도]

상태  $X$ 에서  $b$ 가 입력되면  
 상태  $X$ 로 전이되므로  $X \rightarrow bX$ 와 같이 표현  
 또한 상태  $X$ 에서  $a$ 가 입력되면  
 상태  $Y$ 로 전이되므로  $X \rightarrow aY$ 와 같이 표현할 수 있음  
 같은 방법으로  $Y \rightarrow aX, Y \rightarrow bY$ 를 정의할 수 있음

## 1 언어와 자동 장치

- ▶ 예) 다음 정규 문법  
 $G = \{N, T, P, S\}$ 에 해당하는 자동 장치를 구하시오

$T = \{a, b\}$

$N = \{X, Y, Z\}$

$P = \{X \rightarrow aY, X \rightarrow bZ, Y \rightarrow aX, Y \rightarrow bZ, X \rightarrow b, Y \rightarrow b, Z \rightarrow \lambda\}$

풀이)

단말 기호는  $a, b$ 이고 비단말기호는  $X, Y, Z$ 임

자동 장치를 구하려면 생성  $P$ 의 규칙에 따라 작성하면 됨

초기 상태를  $X$ 로 두면 자동 장치는 다음과 같음

