KAIST 전산학과 최광무

5.2 문맥자유문법의 문법나무와 유도순서

(**정의** 5.9) **문법나무(parse tree, syntactic tree)**

문맥자유문법 $G = (N, T, P, S)$에서 $X \in N \cup T$의 **문법나무** $T_G(X)$를 X가 뿌리인 **뿌리나무**(rooted tree) $T_G(X) = (V, E, X)$[1]로 아래와 같이 정의한다.

(basis) $a \in T$에 대하여 뿌리가 a인 **문법나무**$T(a) = (\{a\}, \varnothing, a)$다.

(recursion) $A \to X_1 X_2 \cdots X_n \in P$ (단 $A \in N$, $1 \leq {}^{\forall} i \leq n$: $X_i \in N \cup T$)에서,

$1 \leq {}^{\forall} i \leq n$: $T_G(X_i) = (V_i, E_i, X_i)$가 X_i를 뿌리로 하는 **문법나무**이면,

A를 뿌리로 하는 새로운 **문법나무** $T_G(A) = (V_A, E_A, A)$는

(1) $V_A = \{A\} \bigcup \cup_{i \in N_n} V_i$,

(2) $E_A = \{(A, X_i) | A \to X_1 X_2 \cdots X_n \in P\} \bigcup \cup_{i \in N_n} E_i$로 정의한다.

(**정의** 5.10) 쓴 문법규칙열(rule string)을 표기하도록 유도($\Rightarrow$)를 확장, 즉 $\Rightarrow^{\pi}$ 단 $\pi \in P^*$.

(basis) $\alpha \Rightarrow^{\epsilon} \alpha$, For $\alpha \in V^*$.

(recursion) If $\alpha \Rightarrow^{\pi} \beta \wedge \beta \Rightarrow^{r} \gamma$, then $\alpha \Rightarrow^{\pi \cdot r} \gamma$. 단 $\pi \in P^*$, $r \in P$.

(**정의** 5.11) 문법 $G = (N, \Sigma, P, S)$에서 $S \Rightarrow^{\pi} x$(단 $\pi \in P^*$)이면 **문장** $x \in T^*$에 대한 문법나무 $T_G(S)$는 S를 뿌리로 하는 문법나무 $T_G(S) = (V, E, S)$중 잎(leaves)이 문자열 x이고 이때 사용된 규칙열이 π인 경우이다.

$S \Rightarrow^{\pi} x \in T^*$이고 $\pi \in P^*$일 때, 사용된 문법규칙으로 subtree를 만들면 전체 문장 $x \in T^*$에 대한 트리를 만들 수 있고, 이를 파스트리라 부른다.

문맥자유문법의 **문장형태**는 일반적으로 여러 개에 넌 터미널을 가지는데, 어떤 넌 터미널이 유도에 먼저 참여할까 하는 것은 문맥자유-문법나무에 최종모양과는 관계가 **없다**[2]. 따라서 문장형태에서 가장 왼쪽에 있는 넌 터미널부터 다시 쓰기를 하는 왼쪽부터 고쳐 쓰기(leftmost derivation)과 오른쪽부터 고쳐 쓰기(rightmost derivation)에 두 가지 극단적인 경우를 생각할 수 있다. 이를 각각 $\Rightarrow_{lm}$와 $\Rightarrow_{rm}$로 표시한다.

(**정의** 5.12) 왼쪽부터 고쳐 쓰기(leftmost derivation): $\Rightarrow_{lm}^{\pi_L}$, $\pi_L \in P^*$.

(basis) $\alpha \Rightarrow_{lm}^{\epsilon} \alpha$, For $\alpha \in V^*$.

(recursion) If $\alpha \Rightarrow_{lm}^{\pi} xB\gamma$ and $xB\gamma \Rightarrow_{lm}^{B \to \beta} x\beta\gamma$, then $\alpha \Rightarrow_{lm}^{\pi \cdot B \to \beta} x\beta\gamma$.

단 $x \in T^*$, $B \in N$, $B \to \beta \in P$, $\pi \in P^*$.

(**정의** 5.13) 오른쪽부터 고쳐 쓰기(rightmost derivation): $\Rightarrow_{rm}^{\pi}$, $\pi_R \in P^*$.

---

1) 뿌리나무 $T=(V, E, X)$에서 (1) V는 vertex의 집합, (2) $E \subseteq V \times V$는 edge의 집합, (3) $X \in V$는 뿌리라 불리는 특별한 vertex이다. 그래프 $G=(V, E)$와 뿌리나무 $T=(V, E, X)$에 관한 기초적인 정의는 CS204 이산수학 참조.

2) Context-free 문법의 특성이다.

(basis) $\alpha \Rightarrow_{rm}^{\epsilon} \alpha$, For $\alpha \in V^*$.

(recursion) If $\alpha \Rightarrow_{rm}^{\pi} \alpha B z$ and $\alpha B z \Rightarrow_{\mathrm{rm}}^{B\to\beta} \alpha\beta z$, then $\alpha \Rightarrow_{rm}^{\pi\cdot B\to\beta} \alpha\beta z$.

단 $z \in T^*$, $B \in N$, $B \to \beta \in P$, $\pi \in P^*$.

문장

(예 5.1) $G = (\{E, T, F\}, \{a, +, \times, (,)\}, P, E)$에서

P: $E \to E + T \mid T \times F \mid a \mid (E)$

$T \to T \times F \mid a \mid (E)$

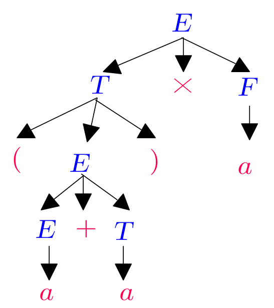
$F \to a \mid (E)$

$(a+a)\times a$에 관한 두 가지 유도과정

$E \Rightarrow_{lm}^{E\to T\times F} T\times F \Rightarrow_{lm}^{T\to(E)} (E)\times F \Rightarrow_{lm}^{E\to E+T} (E+T)\times F \Rightarrow_{lm}^{E\to a} (a+T)\times F$

$\Rightarrow_{lm}^{T\to a} (a+a)\times F \Rightarrow_{lm}^{F\to a} (a+a)\times a$.

$\pi_L = (E\to T\times F)\cdot(T\to(E))\cdot(E\to E+T)\cdot(E\to a)\cdot(T\to a)\cdot(F\to a) \in P^6$.



$E \Rightarrow_{\mathrm{rm}}^{E\to T\times F} T\times F \Rightarrow_{\mathrm{rm}}^{F\to a} T\times a \Rightarrow_{\mathrm{rm}}^{T\to(E)} (E)\times a \Rightarrow_{\mathrm{rm}}^{E\to E+T} (E+T)\times a$

$\Rightarrow_{\mathrm{rm}}^{T\to a} (E+a)\times a \Rightarrow_{\mathrm{rm}}^{E\to a} (a+a)\times a$.

$\pi_R = E\to T\times F \cdot F\to a \cdot T\to(E) \cdot E\to E+T \cdot T\to a \cdot E\to a \in P^6$.

π_R^R[3] $= E\to a \cdot T\to a \cdot E\to E+T \cdot T\to(E) \cdot F\to a \cdot E\to T\times F \in P^6$.

$S \Rightarrow_{lm}^{\pi_L^1} xA\gamma \Rightarrow_{lm}^{B\to\beta} x\beta\gamma \Rightarrow_{lm}^{\pi_L^2} xy\gamma \Rightarrow_{lm}^{\pi_L^3} xyz$, $\pi_L^1, \pi_L^2, \pi_L^3 \in P^*$,

단 $x \in T^*$, $\gamma \in (N\cup T)^*$, $B\to\beta \in P$, $y, z \in T^*$, $\beta \Rightarrow^* y$, $\gamma \Rightarrow^* z$.

$S \Rightarrow_{\mathrm{rm}}^{\pi_R^1} \alpha Az \Rightarrow_{rm}^{B\to\beta} \alpha\beta z \Rightarrow_{\mathrm{rm}}^{\pi_R^2} \alpha yz \Rightarrow_{\mathrm{rm}}^{\pi_R^3} xyz$, $\pi_R^1, \pi_R^2, \pi_R^3 \in P^*$,

단 $z \in T^*$, $\alpha \in (N\cup T)^*$, $B\to\beta \in P$, $y, x \in T^*$, $\beta \Rightarrow^* y$, $\alpha \Rightarrow^* x$.

이 때 $\pi_L = \pi_L^1(A\to\beta)\pi_L^2\pi_L^3 \in P^*$라 하고 $\pi_R = \pi_R^1(A\to\beta)\pi_R^2\pi_R^3 \in P^*$라 하면 π_L과 π_R를 각각 문장 x에 대한 left parse(좌 분석)와 right parse(우 분석)라고 부른다. 문장 x에 대한 좌 분석이나 우 분석이 정해지면 해당하는 문법트리도 **하나로**(unique) 정해진다.

5.2 문맥자유문법과 문법나무 그리고 유도순서

3) π_R^R의 위에 ^{R}은 오른 파스 π_R을 뒤집은 것(Reversal)을 뜻한다.

문맥자유문법의 **문장형태**는 일반적으로 여러 개에 넌 터미널을 가지는데, 어떤 넌 터미널이 유도에 먼저 참여할까 하는 것은 파스트리에 최종모양과는 관계가 **없다**. 따라서 문장형태에서 가장 왼쪽에 있는 넌 터미널부터 다시 쓰기를 하는 왼쪽부터 고쳐 쓰기(leftmost derivation)과 오른쪽부터 고쳐 쓰기(rightmost derivation)에 두 가지 극단적인 경우를 생각할 수 있다. 이를 각각 $\Rightarrow_{lm}$와 $\Rightarrow_{\mathrm{rm}}$[4)]로 표시한다.

$$S \overset{\pi_1}{\Rightarrow}_{lm} xA\gamma \overset{B\to\beta}{\Rightarrow}_{lm} x\beta\gamma \overset{\pi_2}{\Rightarrow}_{lm} xy\gamma \overset{\pi_3}{\Rightarrow}_{lm} xyz,\ \ \pi_1, \pi_2, \pi_3 \in P^*,$$

$$\text{단 } x \in T^*,\ \gamma \in (N \cup T)^*,\ B\to\beta \in P,\ y, z \in T^*,\ \beta \overset{\pi_2}{\Rightarrow}_{lm} y,\ \gamma \overset{\pi_3}{\Rightarrow}_{lm} z.$$

$$S \overset{\pi_1^R}{\Rightarrow}_{\mathrm{rm}} \alpha Az \overset{B\to\beta}{\Rightarrow}_{\mathrm{rm}} \alpha\beta z \overset{\pi_2^R}{\Rightarrow}_{\mathrm{rm}} \alpha yz \overset{\pi_3^R}{\Rightarrow}_{\mathrm{rm}} xyz,\ \ \pi_1^R, \pi_2^R, \pi_3^R \in P^*,$$

$$\text{단 } z \in T^*,\ \alpha \in (N \cup T)^*,\ B\to\beta \in P,\ y, x \in T^*,\ \beta \overset{\pi_2}{\Rightarrow}_{lm} y,\ \alpha \overset{*}{\Rightarrow} x.$$

이 때 $\pi_L = \pi_L^1(A\to\beta)\pi_L^2\pi_L^3 \in P^\dagger$라 하고 $\pi_R = (\pi_R^1(A\to\beta)\pi_R^2\pi_R^3)^R \in P^\dagger$라 하면 π_L과 π_R를 각각 문장 x에 대한 left parse(좌 분석)와 right parse(우 분석)라고 부른다. 문장 x에 대한 좌 분석이나 우 분석이 정해지면 해당하는 문법트리도 정해진다.

문맥자유문법의 파싱(parsing)은 임의의 터미널 문자열 $x \in T^*$가 주어진 문법 G에 문장이면 해당하는 파스트리를 만들어주고, 문장이 아니면 이를 알려주는 과정이다.

4) $\Rightarrow_{lm}, \Rightarrow_{\mathrm{rm}} \subseteq P^*$이다.