확장된 PLR 문법의 테스팅
(Testing of Extended PLR Grammars)

이 경 옥 †
(Gyung-Ok Lee)

요 약 확장된 PLR 문법은 LR 문법의 넓은 범위의 부분 문법 클래스이면서 LL 문법의 장점을 갖는 문법 클래스이다. 한편 주어진 임의의 문법에 대한 확장된 PLR 문법의 테스팅 알고리즘은 아직까지 제시되지 않았다. 본 논문에서는 확장된 PLR 문법에 관한 테스팅 알고리즘을 제시한다.

키워드 : LR 문법, LL 문법, 확장된 PLR 문법

Abstract Extended PLR grammars are a large subclass of LR grammars and have good property of LL grammars. On the other hand, a testing algorithm of extended PLR grammars has not been given. The paper presents an algorithm for the testing of extended PLR grammars.

Key words : LR grammars, LL grammars, extended PLR grammars

1. 서 론

LR 문법[1]과 *LL* 문법[2]은 대표적인 파싱 방법인 LR 파싱과 LL 파싱이 가능한 문법 클래스이다. LR 문법은 길이 k의 미리보기(lookahead) 스트링을 이용한 결정적 파싱이 가능한 가장 큰 범위의 문법 클래스이며, LL 문법은 구현하기 쉽고 직관적인 LL 파싱이 가능한 문법 클래스이다. 이 두 클래스의 장단점을 서로 보완하는 확장된 *PLR* 문법[3]은 현재까지 알려진 *LL* 문법으로의 커버링 변환이 가능한 가장 큰 LR 문법 클래스이다. 한편 확장된 PLR 문법에 관한 테스팅 알고리즘은 아직 제시되지 않았다.

본 논문에서는 임의의 문법이 주어진 경우에 확장된 PLR 문법인지를 테스팅하는 방법을 제시한다. 기존 연구에서 확장된 PLR 문법의 정의가 문법의 제한적 우측 유도 형태로 표현이 되었기에 확장된 PLR 문법의 테스팅 방법이 직관적이지 않으며 별도의 방법이 필요하다. 본 논문에서는 테스팅 방법을 제공하기에 앞서서 기존의 확장된 PLR 문법의 정의를 간략한 형태로 표현할 수 있음을 보인다. 이 후에 간단한 관계식에 근거하여 확장된 PLR 문법의 테스팅 알고리즘을 제안한다.

본 논문의 구성으로는 2장에선 기본 정의와 표기법을 제시한다. 3장에서는 확장된 PLR 문법 정의의 간략한 형태를 제시하고, 4장에서는 테스팅 알고리즘을 제시하고 알고리즘의 정확성을 증명한다. 끝으로 5장에서 결론을 맺는다.

2. 기본 정의와 표기법

본 논문에서는 재언급 없이 [3-5]에서의 정의와 표기법을 따른다.

문법 $G = (N, \Sigma, P, S)$에 규칙 $S' \rightarrow S\$^k$을 추가한 문법을 대상으로 하며, G가 LR(k) 문법임을 가정한다. $A(\in N)$에 대해 G^A는 A를 시작 심볼로 하는 축소된(reduced) 문법[5]을 표기한다.

$\alpha \in V^*(V = N \cup \Sigma)$에 대해서 $k{:}\alpha$, $\alpha{:}k$는 각각 α의 길이 k의 전위 스트링(prefix)과 후위 스트링(suffix)을 표기한다. $L^G(\alpha) = \{x \mid S \Rightarrow^* x, x \in \Sigma^*\}$, $FIRST_k^G(\alpha) = \{k{:}x \mid \alpha \Rightarrow^* x, x \in \Sigma^*\}$, $FOLLOW_k^G(\alpha) = \{k{:}x \mid S \Rightarrow^* \beta\alpha x, x \in \Sigma^*\}$, $RC_k^G(\alpha) = \{k{:}xz \mid S \Rightarrow^*_{rm} \beta Bz \Rightarrow_{rm} \beta\gamma\delta z \Rightarrow^*_{rm} \beta\gamma xz, \alpha = \beta\gamma, xz \in \Sigma^*\}$이다. 이들의 표기방법에서 G가 문맥상 명확할 경우는 이를 생략한다.

$A \in N$, $R \subseteq FOLLOW_k(A)$라고 하자. 제한적 문법 유도를 나타내는 $\Rightarrow_{A,R}$은 다음과 같이 정의된다: $Ar \Rightarrow_{rm}^* \gamma Bzr \Rightarrow_{rm}^p \gamma X\delta zr (r \in R)$이 존재한다고 하자. $\gamma = \varepsilon$, X

· 이 논문은 한신대학교 학술연구비 지원에 의하여 연구되었음

† 종신회원 : 한신대학교 정보통신학과 교수
golee@hs.ac.kr

논문접수 : 2009년 8월 17일
심사완료 : 2009년 10월 14일

정보과학회논문지: 소프트웨어 및 응용 제37권 제1호(2010.1)

$= A$인 경우에는 $\mathrm{FIRST}_k(\delta zr) \cap R = \emptyset$의 조건을 만족 시에 $\gamma Bzr \Rightarrow_{A,R}^{p} \gamma X\delta zr$이고, $\gamma = \varepsilon$, $X = A$가 아닌 경우에는 앞서의 조건에 무관하게 $\gamma Bzr \Rightarrow_{A,R}^{p} \gamma X\delta zr$이다. $RC_k^{A,R}(\alpha) = \{k{:}yzr \mid Ar \Rightarrow_{A,R}^{*} \beta Bzr \Rightarrow_{A,R} \beta\gamma\delta zr \Rightarrow_{A,R}^{*} \beta\gamma yzr,\ \alpha = \beta\gamma,\ r \in R,\ yz \in \Sigma^{*}\}$로 정의한다.

관계 d는 다음과 같이 정의된다: (i) $A \rightarrow \alpha B\beta \in P$이면 $A\ d^{\alpha}\ B$이다; (ii) $A \rightarrow \alpha a\beta \in P$, $v \in \mathrm{FIRST}_{\max\{k-1,0\}}(\beta \mathrm{FOLLOW}_k(A))$이면 $A\ d^{\alpha}\ (a,v)$이다; (iii) $A \rightarrow \alpha \in P$, $v \in \mathrm{FOLLOW}_k(A)$이면 $Ad^{\alpha}(\varepsilon,v)$이다. d 관계에 대응하는 그래프는 d 그래프이며, 그래프상의 경로는 $A_0\ d^{\alpha_1}\ A_1\ d^{\alpha_2}\ A_2 \ldots A_{n-1}\ d^{\alpha_n}\ A_n$이거나 $A_0\ d^{\alpha_1}\ A_1\ d^{\alpha_2}\ A_2 \ldots A_{n-1}\ d^{\alpha_n}\ (A_n, v)$이다. $\langle A, \alpha, u\rangle$-경로는 $A_0\ d^{\alpha_1}\ A_1\ d^{\alpha_2}\ A_2 \ldots A_{n-1}\ d^{\alpha_n}\ (A_n,v)$, $A_0 = A$, $\alpha = \alpha_1\alpha_2 \ldots \alpha_n$, $A_n v = u$를 나타낸다.

3. 확장된 PLR 문법의 정의

본 절에서는 확장된 PLR 문법 정의의 간략한 형태를 제시한다. 다음은 확장된 PLR 문법의 정의이다.

정의 3.1 [3] G가 확장된 PLR 문법이기 위한 필요충분 조건으로는 다음의 (문장 1)이 성립한다.

(문장 1) G의 생존 가능 스트링(viable prefix)인 $\alpha(|\alpha| > n)$, $L^G(\alpha)$의 원소인 x, $RC_k^G(\alpha)$의 원소인 v에 대해서 아래의 (조건 1)을 만족하는 B, W, $\gamma(\gamma \neq \varepsilon,\ |\gamma| \leq n,\ \alpha = \beta\gamma)$가 존재하도록 하는 상수 n이 존재한다.

(조건 1) $S' \Rightarrow^{\pi}_{rm} \beta\gamma z \Rightarrow_{rm}^{*} xz$, $k{:}z=v$가 G상에 존재할 때 마다 $S' \Rightarrow_{rm}^{\pi_1} \beta Bz''$, $Bw \Rightarrow^{\pi_2}_{B,W} \gamma z'w$, $w = k{:}z''$ $(w \in W)$, $z'z'' = z$인 π_1와 π_2가 G상에 존재한다. (이때 n은 G의 *<u>확장된 PLR(k) 인덱스</u>*이다.) □

정의 3.1에서의 (문장 1)은 다음과 같이 간략화시킬 수가 있다.

보조정리 3.1 G가 확장된 PLR 문법이기 위한 필요충분 조건으로는 다음의 (문장 2)가 성립한다.

(문장 2) G의 생존 가능 스트링(viable prefix)인 $\alpha(|\alpha| > n)$, $L^G(\alpha)$의 원소인 x, $RC_k^G(\alpha)$의 원소인 v에 대해서 아래의 (조건 2)를 만족하는 B, $\gamma(\gamma \neq \varepsilon,\ \alpha = \beta\gamma,\ |\gamma| \leq n)$가 존재하도록 하는 n이 존재한다.

(조건 2) $S' \Rightarrow^{\pi}_{rm} \beta\gamma z \Rightarrow_{rm}^{*} xz$, $k{:}z=v$가 G상에 존재할 때 마다 $S' \Rightarrow^{\pi_1}_{rm} \beta Bz''$, $B \Rightarrow^{\pi_2}_{rm} \gamma z'$, $z'z'' = z$인 π_1와 π_2가 G상에 존재한다.

(증명) <u>(필요 조건)</u> (문장 2)가 성립한다고 하자. $W = RC_k^G(\beta B)$로 할 경우에 정의 3.1의 (문장 1)이 성립한다. 따라서 G는 확장된 PLR 문법이다.

<u>(충분 조건)</u> G가 확장된 PLR 문법이라고 가정하면 정의 3.1의 (문장 1)이 성립한다. 한편 (문장 2)는 (문장 1)의 부분 조건이다. 따라서 (문장 2)가 성립한다. □

보조정리 3.1에서는 기존의 $\Rightarrow_{B,W}$ 유도 과정을 $\Rightarrow_{rm}$ 형태로 간략하게 만들었다.

4. 확장된 PLR 문법의 테스팅

본 절에서는 확장된 PLR 문법의 테스팅 알고리즘을 제시하고 알고리즘의 정확성을 증명한다.

다음은 d 그래프상의 일정한 형태의 노드를 관계로 정의한다.

정의 4.1 $(A, \alpha)\ \Pi_u\ (B, \gamma)$가 성립하기 위한 필요충분 조건은 다음과 같다:

각 $\langle A, \alpha, u\rangle$-경로, $A_0\ d^{\alpha_1}\ A_1\ d^{\alpha_2}\ A2 \ldots A_{n-1}\ d^{\alpha_n}\ (A_n,v)$, $A_0 = A$, $\alpha = \alpha_1\alpha_2 \ldots \alpha_n$, $A_n v = u$에 대해서 $A_m = B$, $\alpha_m\alpha_{m+1} \ldots \alpha_n = \gamma$를 만족하는 m $(1 \leq m \leq n)$이 존재한다. □

다음은 우측 문법 유도와 Π 관계 사이의 밀접한 성질을 설명한다.

성질 4.1 $A \Rightarrow_{rm}^{\pi} \beta\gamma x$, $k{:}xr = u(r \in \mathrm{FOLLOW}_k(A))$인 π가 존재할 때마다 $A \Rightarrow_{rm}^{\pi_1} \beta Bz$, $B \Rightarrow_{rm}^{\pi_2} \gamma y$, $yz = x$, $\pi_1\pi_2 = \pi$인 π_1, π_2가 존재한다면 $(A, \alpha)\ \Pi_u\ (B, \gamma)$가 성립한다. 이 역도 성립한다. □

$A \in N$, $\alpha \in V^{*}$, $\alpha = X_1 \cdots X_n$에 대해서 $q_i = \{[B \rightarrow \gamma.\delta, u] \mid A\ d^{*\beta}\ B,\ |\beta| + |\gamma| = i,\ \alpha = \beta\gamma\alpha',\ B \rightarrow \gamma\delta \in P,\ u \in RC_k^{G\ A}(\beta B)\}$, $i = 0, 1, \cdots, n$으로 정의하자. $q_i, q_{i+1}, \cdots, q_{i+j}$ $(0 \leq i < i+j \leq n)$에 대해서 $q_i = q_{i+j}$이고 $q_i, q_{i+1}, \cdots, q_{i+j}$내의 동일한 다른 쌍이 존재하지 않는다면 $q_i, q_{i+1}, \cdots, q_{i+j}$는 루프이다. $q_0, q_1, \cdots, q_n$상에 루프가 존재하는 경우에 (A, α)는 *<u>싸이클릭</u>*이다. (A, α)가 싸이클릭이고 어떤 $u \in RC_k^{G\ A}(\alpha)$에 대해서 $(A, \alpha)\ \Pi_u\ (B, \gamma)$가 성립하는 B, γ가 존재하지 않는 경우에 (A, α)는 *<u>나눌 수 없는 싸이클릭</u>*이다.

다음은 문법 G를 입력으로 받아서 확장된 PLR 문법인지를 판정하는 알고리즘이다.

알고리즘 1.

입력: 문법 G

출력: G가 확장된 PLR 문법인 경우에는 yes, 그렇지 않는 경우에는 no를 생성한다.

과정:

<u>단계1</u>. $Q = \{(S', \varepsilon)\}$; *found* = false

<u>단계2</u>. **repeat**

for 각 $(A, \alpha) \in Q$와 $u \in RC_k^{G\ A}(\alpha)$

if $A\ d^{*\alpha}\ a$ **then** $Q = Q \cup \{(A, \alpha a)\}$

if $(A, \alpha)\ \Pi_u\ (B, \gamma)$ **then** $Q = Q \cup \{(B, \gamma)\} \cup \{(A, \beta B) \mid \alpha = \beta\gamma\}$

if Q에 추가된 원소가 나눌 수 없는 싸이클릭이다 **then** *found* = true

endfor

until (*found* == true이거나 Q 집합에 새로운

원소가 추가되지 않는다)

단계3. **if** found == false **then** yes를 생성한다 **else** no을 생성한다. □

보조정리 4.1 주어진 문법 G에 대한 알고리즘 1의 결과가 yes이면 G는 확장된 PLR 문법이다.

(증명) 알고리즘 1의 결과가 yes이나 G가 확장된 PLR 문법이 아니라고 가정하자. 그러면 (조건 2)를 만족하는 B, γ가 존재하지 않는 생존 가능 스트링 α, $x \in L^G(\alpha)$, $v \in RC^G{}_k(\alpha)$가 존재한다. 따라서 $S' \Rightarrow^*_{rm} \alpha y \Rightarrow^*_{rm} xy$, $k{:}y = v$가 존재한다. 알고리즘 1의 결과가 yes이기에 (S', α)는 나눌 수 없는 싸이클릭이 아니다. 따라서 $(S', \alpha)\ \Pi_v\ (B, \gamma)$를 만족하는 예상 관계가 존재한다. 그러므로 성질 4.1에 의해서 $S' \Rightarrow^*_{rm} \alpha y$가 존재할 때마다 $S' \Rightarrow^*_{rm} \beta B y_2$, $B \Rightarrow^*_{rm} \gamma y_1$, $y = y_1 y_2$이 존재한다. 이때 $|\gamma|$을 n이라고 하면, 보조정리 3.1의 조건을 만족하기에 G는 확장된 PLR 문법이다. 이러한 모순은 본 보조정리가 성립함을 의미한다. □

보조정리 4.2 G가 확장된 PLR 문법이면 G에 대한 알고리즘 1의 결과가 yes이다.

(증명) G가 확장된 PLR 문법이고 n을 G의 확장된 PLR(k) 인덱스라고 하자. G에 대한 알고리즘 1의 결과가 no라고 가정하면, 나눌 수 없는 싸이클릭인 (A, α), $|\alpha| > n$가 존재한다. 가정하기를 $S' \Rightarrow^*_{rm} \theta A w \Rightarrow^*_{rm} \theta \alpha y w \Rightarrow^*_{rm} xyw$가 G상에 있다고 하자. 또한 $u = k{:}yw$라고 하자. G가 확장된 PLR(k)이기에 생존 가능 스트링 $\theta\alpha$, $x \in L^G(\theta\alpha)$, u에 대해서 $S' \Rightarrow_{rm}^{\pi} \theta\alpha z \Rightarrow^*_{rm} xz$인 π가 존재하는 경우에 $S' \Rightarrow_{rm}^{\pi_1} \beta B z_2$, $B \Rightarrow_{rm}^{\pi_2} \gamma z_1$, $\beta\gamma = \theta\alpha$, $z_1 z_2 = z$, $\pi_1\pi_2 = \pi$가 G상에 존재하게 하는 B, γ가 존재한다. 한편 π는 $S' \Rightarrow_{rm}^{\pi'} \theta A z'' \Rightarrow_{rm}^{\pi''} \theta\alpha z' z''$, $z' z'' = z$, $\pi = \pi'\pi''$로 구성된다. 따라서 $A \Rightarrow^*_{rm} \alpha z''$가 존재할 때 마다 $A \Rightarrow^*_{rm} \beta' B z''' \Rightarrow^*_{rm} \beta' \gamma z''$, $\beta'\gamma = \alpha$인 B, γ가 존재하여, 성질 4.1에 의해서 $(A, \alpha)\ \Pi_u\ (B, \gamma)$가 성립한다. 그러므로 (A, α)는 나눌 수 없는 싸이클릭이 아니다. 이러한 모순은 본 보조정리가 성립함을 의미한다. □

보조정리 4.1과 4.2에 의해서 정리 4.1이 성립한다.

정리 4.1 주어진 문법 G에 대한 알고리즘 1의 결과가 yes이기 위한 필요충분 조건은 G는 확장된 PLR 문법이다. □

5. 결 론

기존의 확장된 PLR 문법의 정의는 제약적인 문법 유도 과정을 도입하였기에 문법 정의로부터 직관적인 테스팅 방법을 유추하기가 쉽지 않다. 이에 본 논문에서는 기존 확장된 PLR 문법의 정의를 간략하게 만들었으며, d 관계와 Π 관계에 의해서 문법 테스팅이 가능한 알고리즘을 제시하였다. 본 연구 결과로서 주어진 임의의 문법에 대한 확장된 PLR 문법인지의 테스팅이 수월해졌다.

참 고 문 헌

[1] D. E. Knuth, "On the translation of languages from left to right," *Information and Control*, vol.8, pp.607-623, 1965.

[2] D. E. Knuth, "Top down Syntax," *Acta Informatica*, vol.1, pp.79-110, 1971.

[3] G.-O. Lee and K.-M. Choe, "A powerful LL(k) covering transformation," *SIAM J. Computing*, vol.35, no.2, pp.359-377, 2006.

[4] A. V. Aho and J. D. Ullman, *The Theory of Parsing, Translation and Compiling*, vols.1 2. p.1002, Prentice-Hall, Englewood Cliffs, NJ, 1972, 1973.

[5] S. Sippu and E. Soisalon-Soininen. *Parsing theory*, vols. I, II. Springer, Berlin, 1990.

이 경 옥

1990년 2월 서강대학교 전자계산학과 졸업(학사). 1992년 8월 한국과학기술원 전산학과 졸업(석사). 2000년 2월 한국과학기술원 전산학과 졸업(박사). 2000년 8월부터 한신대학교 정보통신학과 근무. 현재 한신대학교 정보통신학과 부교수. 관심분야는 프로그래밍 언어와 컴파일러 등