

문법의 동등변환의 한가지 방법

리은실, 한용환

문법의 변환문제는 프로그램체계와 응용프로그램개발에서 중요한 문제이므로 많이 연구되고있으나 특별히 어떤 구조의 문법이 편리한가에 대한 절대적기준은 없고 모두 개별적기교에 의하여 규칙삽입, 변경의 방법으로 진행하고있다.[1, 2, 4]

론문에서는 선행한 방법과는 달리 새규칙을 첨가하는 방법으로 문법을 변환하는 한가지 방법을 제안하였다. 그리고 론문에서 정의없는 개념은 [3, 4]에 준한다.

정의 1 문법 $G=(V, \Sigma, P, \sigma)$ 가 주어졌다고 하자.

이때 규칙 $U \rightarrow W (U, W \in V, \alpha \in (V \cup \Sigma)^*)$ 이 P 에 존재하면 $U < W$ 로 표시하고 문법 G 에 선형순서관계가 주어졌다고 말한다.

정의 2 자유문맥문법 $G=(V, \Sigma, P, \sigma)$ 가 주어졌다고 하자.

이때 $U \rightarrow \alpha (U \in V, \alpha \in (V \cup \Sigma)^*)$ 형태의 규칙을 U -규칙이라고 부른다.

우리는 이 개념에 기초하여 주어진 문법 $G=(V, \Sigma, P, \sigma)$ 를 규칙모임 P 에 빈규칙이 하나도 없으며 또 $\sigma \rightarrow e$ 와 다른 모든 규칙은 $U \rightarrow a\alpha$ 형태의 규칙을 가지는 문법으로 변환하는 알고리즘을 구성하였다.

변환알고리즘은 다음과 같다.

① 주어진 문법 G 의 비최종기호모임 V 에서 선형순서관계를 만들되 때 U -규칙은 최종기호로 시작되든가 혹은 $U \rightarrow W$ 로 되는 그러한 비최종기호 W 로 시작되도록 한다. 그리고 $U_1 < U_2 < U_3 < \dots < U_n$ 이 되도록 모임 $V=\{U_1, U_2, \dots, U_n\}$ 을 순서화한다.

② $i=n-1$ 로 놓는다.

③ 만일 $i=0$ 이면 결음 ⑤으로 이행한다. 반대 경우에는 $U_i \rightarrow U_j \alpha (j > i)$ 형태의 매 규칙을 규칙 $U_i \rightarrow \beta_1 \alpha | \beta_2 \alpha | \dots | \beta_m \alpha$ 와 같은 형태의 규칙들로 바꾼다.

여기서 $U_i \rightarrow \beta_1 | \beta_2 | \dots | \beta_m$ 은 모든 U -규칙이다. 그리고 매 사슬 $\beta_1, \beta_2, \dots, \beta_m$ 들은 최종기호로 시작된다.

④ $I:=i-1$ 로 놓고 ③으로 이행한다.

⑤ 매 규칙의 오른쪽부분은 최종기호로 시작된다. 그리고 매 규칙

$$U \rightarrow aX_1 aX_2 \dots aX_k$$

에서 X_j 를 새로운 비최종기호 X'_j 로 바꾼다.

⑥ 결음 ⑤에서 도입한 새로운 매 비최종기호 X'_j 에 대하여 규칙 $X'_j \rightarrow X_j$ 를 보충한다.

이 알고리즘으로부터 새로 구성된 문법 $G'=(V', \Sigma, P', \sigma)$ 는 빈규칙이 하나도 없고 $\sigma \rightarrow e$ 와 다른 모든 규칙들은 $U \rightarrow a\alpha$ 형태로 되었다는것을 알수 있다. 이때 언어모임은 변경되지 않는다.

이제 문법 $G=(V, \Sigma, P, \sigma)$ 가 주어지고 P 는 규칙 $U \rightarrow xWy$ 를 포함한다고 하자.

여기서 $W \in V$, $x, y \in (V \cup \Sigma)^*$ 이다. 그리고 $W \rightarrow \gamma_1 | \gamma_2 | \dots | \gamma_k$ 는 주어진 문법의 모든 W -규칙이라고 하자. 이때 새롭게 얻어지는 규칙모임

$$P' = (P - \{U \rightarrow xWy\}) \cup (U \rightarrow x\gamma_1 y | U \rightarrow x\gamma_2 y | U \rightarrow x\gamma_k y)$$

로서 구성되는 문법 $G' = (V, \Sigma, P', \sigma)$ 에 의하여 얻어지는 언어는 문법 G 에 의하여 얻어지는 언어와 같다. 즉 $L(G) = L(G')$ 로 된다.

이 사실에 기초하여 위에서 제기한 변환알고리즘이 정확하다는것을 증명하자.

정리 변환알고리즘에 의하여 주어진 문법을 빈규칙이 없고 $\sigma \rightarrow e$ 와 다른 모든 규칙들은 $U \rightarrow a\alpha (a \in \Sigma, \alpha \in V^*)$ 과 같은 형태를 가지는 문법으로 변환되며 이때 두 문법에 의하여 생성된 언어는 같다.

증명 증명은 $n-1$ 에 대한 귀납법 즉 반대로 i 가 $n-1$ 로부터 시작하여 $i=1$ 에 끝나도록 하는 방식으로 진행한다. I 에 대하여 변환알고리즘의 걸음 ③을 수행하면 모든 $U_i \rightarrow$ 규칙의 오른쪽부분은 최종기호로 시작되게 된다는것을 알수 있다. 여기서 기본은 정의 1에서 정의한 선형순서관계 $<$ 가 리용된다는것이다.

걸음 ⑤를 수행하면 주어진 문법은 결국 빈규칙이 없고 $\sigma \rightarrow e$ 와 다른 모든 규칙들은 $U \rightarrow a\alpha (a \in \Sigma, \alpha \in V^*)$ 와 같은 형태를 가지는 문법으로 변환된다는것을 알수 있다. 이때 변환된 문법에 의하여 생성된 언어는 주어진 문법에 의하여 생성된 언어와 같게 된다.

맺는 말

주어진 문법에서 불필요한 규칙들을 제거하고 모든 규칙들이 해석과 처리에 편리하도록 변환하는 한가지 방법을 제기하였다. 제기한 알고리즘들에 의하여 임의의 문법을 순수 $U \rightarrow a\alpha$ 형태의 규칙만을 가진 문법으로 변환할수 있다.

참고 문헌

- [1] Р. И. Подловченко; Программирование, 6, 43, 2010.
- [2] В. Е. Котов; Кибернетика, 3, 23, 2008.
- [3] Э. В. Трахтенгерц; Кибернетика, 4, 33, 2009.
- [4] N. J. Wirth; Communs. ACM, 9. 1., 13, 2004.

주체103(2014)년 3월 5일 원고접수

A Method for the Equal Transformation of Grammar

Ri Un Sil, Han Yong Hwan

This paper proposed a method to transform a given grammar to its equal one by modifying or adding the generating rules of that.

We introduced the concept of linear sequential relationship and U -rules, and proposed algorithm to translate the given grammar into the one that has no empty rules and has $U - \alpha$ style rules according to the introduced concept.

Key words: equal transformation, U -rule