

# 미리 결정된 리덕션 골들을 가진 LR 파서

## (An LR Parser with Pre-determined Reduction Goals)

이 경 옥 <sup>†</sup>      최 광 무 <sup>\*\*</sup>

(Gyung-Ok Lee) (Kwang-Moo Choe)

**요약** LR 파싱에서 리덕션 시간 전에 미리 결정될 수 있는 리덕션 골들을 예상하기 위한 정형식들이 주어진다. 이들에 기초해서 예상이 가능한 골들을 미리 아는 LR 파서가 생성된다. 제안된 LR 파서에 대해 구단위(phrase-level) 오류 복구(recovery)와 우문맥 계산시의 효율적인 적용 방법을 제시한다.

**Abstract** A prediction of reduction goals in LR parsing is formalized. Based on it, an LR parser with pre-determined goals is generated. In addition, we show that the proposed LR parser can be applied to phrase-level error recovery and the computation of right context.

### 1. 서론

LR 파싱에서 리덕션(reduction) 골은 리덕션 시간에 알려지도록 LR 파서가 설계되어 있다. 하나 때때로 그 시간 전에 미리 리덕션 골을 알 수 있는 경우도 있다. 이런 리덕션 골에 대한 예상은 제약된 범위 내의 LR 문법을 LL 문법으로 변환하기 위한 근본 아이디어로 Hammer에 의해 [1]에서 제안되었고, [2]에서 확장되었다. 한편 과거 작업들은 예상 아이디어를 직접 LR 파서에 적용하지 않았다.

본 논문에선 예상 아이디어를 LR 파서에 적용하기 위한 정형론들을 도입한 뒤에, 미리 결정될 수 있는 골들을 가지는 LR 파서의 생성 방법을 제시한다. 본 논문의 목적을 위해서 LR 아이템들에 기반을 둔 과거의 복잡한 정형론은 불필요하며, 대신에 문법 골에 기반을 둔 단순화된 정형론을 제시한다. 이에 기초하여 LR 파서내의 각 LR 상태는 예상된 리덕션 골과 스택의 후위 부분(suffix)의 쌍을 레벨로 갖는다. 여기서 스택의 후위 부분은 LR 파싱시에 남은 입력 스트링의 전위 부분(prefix)과 함께 예상되는 리덕션 골로 확실히 리듀스(reduce)가 일어날 스트링이다.

2장에선 본 논문에서 사용될 기본적인 용어를 정의하고, 3장에선 리덕션 골 예상을 위한 관계(relation)들을 정의하고, 이를 기반으로 레벨된 LR 오토마톤을 생성하는 알고리즘을 제시한다. 또한 제안된 오토마톤을 이용한 예상 리덕션 골을 가진 LR 파서의 생성 방법을 보인다. 4장에선 제안된 LR 파서의 응용에 대해 살펴보고, 5장에선 결론을 맺는다.

### 2. 기본 정의들

본 논문에선 독자가 [4]의 파싱 이론과 친숙함을 가정하고, [4]에서의 표기법과 정의들을 다시 언급함 없이 사용한다. 본 논문에선  $G$ 는 임의의 고정된 자유 문맥 문법  $G=(N, \Sigma, P, S)$  ( $V=N \cup \Sigma$ )임을 가정한다.  $G$ 내에  $N$ 에 속하지 않는 골  $S'$ 에 대해  $P$ 가  $S' \rightarrow S$ 를 포함한다면,  $G$ 를 확장된 문법이라고 가정한다.  $A \in N$ 일 때  $G^A$ 는 시작 골  $A$ 를 가진  $G$ 의 리듀스(reduced) [4]부분 문법이다.  $FIRST(a) = \{x \mid a \Rightarrow^* x \text{ in } G, x \in \Sigma^*\}$ 이다.

$A \rightarrow a\beta \in P$ 이고  $\Sigma$ 에 속하지 않는  $\$$ 가 입력 스트링의 마지막 표시자가 아니면,  $u \in FIRST(\Sigma^*\$)$ 일 때  $[A \rightarrow a.\beta, u]$ 는 LR 아이템이다.  $I^G$ 를 LR 아이템들의 집합이라고 하자. 비결정적인(nondeterministic) LR 유한 오토마톤은  $\widehat{M}^G = (I^G, V, \rightarrow, [S' \rightarrow .S, \$], \emptyset)$ 로 정의된다. 전이함수  $\rightarrow$ 는  $I^G \times (V \cup \{\epsilon\})$ 로부터  $2^{I^G}$ 로 정의된다. (i) 어떤  $X \in V$ 에 대해서  $[A \rightarrow a.X\beta, u] \rightarrow^X$

<sup>†</sup> 비회원 : 한국과학기술원 전산학과  
golee@pllab.kaist.ac.kr

<sup>\*\*</sup> 종신회원 : 한국과학기술원 전산학과 교수  
choe@pllab.kaist.ac.kr

논문접수 : 1999년 2월 9일

심사완료 : 1999년 5월 27일

$[A \rightarrow \alpha X, \beta, u]$  이고 (ii)  $v \in FIRST(\beta u)$  일 때  $[A \rightarrow \alpha, \beta \beta, u] \rightarrow [B \rightarrow \gamma, v]$  이다.  $I_0 \rightarrow^{X_1} I_1 \rightarrow^{X_2} \dots \rightarrow^{X_n} I_n (n \geq 0)$  이면  $I_0 \rightarrow^{*X_1 X_2 \dots X_n} I_n$  라고 쓴다.

$\widehat{M}^G$ 을 결정적인(deterministic) 유한 오토마톤으로 변환시킨 LR 오토마톤은  $M(G) = (Q, V, GOTO, q_0, \emptyset)$ 로 정의된다.  $Q$ 는  $\{q_0\} \cup \{GOTO(q, X) \mid q \in Q, X \in V\}$ 로 정의되는 LR 상태들의 집합이다.  $Q \times V$ 로부터  $Q$ 로의  $GOTO$ 는  $GOTO(q, X) = \{M \rightarrow^* J, I \in q\}$ 으로 정의된다.  $q_0$ 는  $\{I[S \rightarrow \cdot S, \$] \rightarrow^* I\}$ 에 의해 정의되는 시작 상태이다.  $GOTO(q_0, \theta)$ 는  $VALID(\theta)$  또는  $\{\theta\}$ 으로 나타낸다.  $q \in Q$ 일 때  $FIRST(q) = \{x \in FIRST(\beta u) \mid [A \rightarrow \alpha, \beta, u] \in q\}$ ,  $KERNEL(q) = \{[A \rightarrow \alpha, \beta, u] \in q \mid \alpha \neq \epsilon \text{ or } A = S'\}$ 으로 정의한다. LR 파서의 컨피규레이션(configuration)은  $[\epsilon] \dots [\theta]$ 가 스택 스트링이고  $x\$\$$ 가 남은 입력 스트링일 때  $[\epsilon] \dots [\theta]x\$\$$ 이다.  $x$ 에 대한 시작 컨피규레이션은  $[\epsilon]x\$\$$ 이고 최종 컨피규레이션은  $[\epsilon][S]x\$\$$ 이다.  $[\epsilon]x_1x_2\$\$ \Rightarrow^* [\epsilon] \dots [\theta]y\$\$ \Rightarrow^* [\epsilon][S]y\$\$$ 인 동작이  $M(G)$  내에 있으면,  $[\epsilon] \dots [\theta]y\$\$$ 는 타당한(valid) LR 피싱 컨피규레이션이다. LR 파서는 쉬프트, 리듀스 행동을 가지며, 이들에 대한 동작은 다음과 같다. (i) 쉬프트 행동:  $[A \rightarrow \alpha, \alpha \beta, u] \in VALID(\theta)$ 이면  $[\epsilon] \dots [\theta]x\$\$ \Rightarrow [\epsilon] \dots [\theta]x\alpha\$\$$ ; (ii) 리듀스 행동:  $A \neq S'$ ,  $u = 1 : x \ \$$ 이고  $[A \rightarrow \alpha, u] \in VALID(\eta a)$ 이면  $[\epsilon] \dots [\eta] \dots [\eta a]x \Rightarrow [\epsilon] \dots [\eta][\eta a]$ 이다. 본 논문에선 대상이 되는 문법은 LR(1) 문법임을 가정한다. 따라서 LR 파서는 결정적(deterministic)으로 동작한다.

**3. 레벨된 LR 오토마톤**

이 장에선 각 LR 상태에 대해 예상할 수 있는 리덕션 골로 레벨링하는 방법을 제안한다. 이를 위해 문법 골간의 의존 관계를 나타내는  $d$ -관계를 정의하고, 이 관계에 대응되는 그래프의 경로(path)들을 분석하여, 예상할 수 있는 리덕션 골을 찾는  $\Pi$ -관계를 정의한다. 이  $\Pi$ -관계를 이용하여 각 LR 상태에서 예상 골로 레벨된 오토마톤이 생성한다.

**3.1  $d$ -관계**

$A \in N, X \in V, a \in V^*$ 라 하자. 이 때 어떤  $\beta \in V^*$ 에 대해  $A \rightarrow \alpha X \beta \in P$ 이면  $Ad^*X$ 로  $d$ -관계는 정의된다.  $Ad^0X$ 는  $Ad^eX$ 을 나타낸다.  $Ad^nX, n > 0$ 는  $Ad^{n-1}B$ 과  $Bd^1X$ 의 합성(composition)이다.  $d$ -관계의 재귀적 전이 클로져(reflexive transitive closure)는  $d^*$ 로 표기되

며,  $\bigcup_{n \geq 0} d^n$ 로 정의된다.  $d$ -관계에 대응되는 그래프는  $d$ -그래프이다.  $d$ -그래프의 경로  $h$ 는  $X_0 d^{a_1} X_1 d^{a_2} X_2 \dots X_{n-1} d^{a_n} X_n$ 의 유한(finite)한 배열(sequence)이다. 표기법  $\langle A, \alpha, X \rangle$ 는  $(h \mid h = X_0 d^{a_1} X_1 d^{a_2} X_2 \dots X_{n-1} d^{a_n} X_n, X_0 = A, \alpha = \alpha_1 \alpha_2 \dots \alpha_n, X_n = X)$ 인 경로들의 집합을 나타낸다.  $d$ -그래프의 경로는 다음의 성질과 같이 LR 파서의 리덕션 골들의 배열을 알 수 있게 한다.

**성질 1**  $A_i \in V, 0 \leq i \leq n-1$ 라 하자. 이 때  $d$ -그래프 내의  $A_0 d^{a_1} A_1 d^{a_2} A_2 \dots d^{a_{n-1}} A_{n-1} d^{a_n} A_n$  경로가 존재한다면 어떤  $z_i \in \Sigma^*, \beta_i \in V^*, i = 1, \dots, n$ 에 대해  $G$  내에  $A_0 \Rightarrow^* \alpha_1 A_1 \beta_1 \Rightarrow^* \alpha_1 A_1 z_1 \Rightarrow^* \alpha_1 \alpha_2 A_2 \beta_2 z_1 \Rightarrow^* \alpha_1 \alpha_2 \dots \alpha_{n-1} A_{n-1} z_{n-1} z_{n-2} \dots z_1 \Rightarrow^* \alpha_1 \alpha_2 \dots \alpha_{n-1} \alpha_n A_n \beta_n z_{n-1} z_{n-2} \dots z_1 \Rightarrow^* \alpha_1 \alpha_2 \dots \alpha_n A_n z_{n-1} z_{n-2} \dots z_1$ 이 존재한다. 이 명제의 역도 성립한다.

(증명) 순방향의 증명은  $n$ 에 대한 인덕션에 의해 쉽게 증명될 수 있고, 역방향의 증명은  $d$  관계의 정의로부터 직접 얻어질 수 있다. □

$G$ 의 자랄 수 있는 스트링(viable prefix)은  $M(G)$ 의 자랄 수 있는 스택 스트링(viable stack string)이고, 그 역도 성립하기에 다음의 성질을 성질 1로부터 얻을 수 있다.

**성질 2**  $A_0 = S', A_i \in N (i = 1, \dots, n-1), A_n \in \Sigma$ 이라 하자. 이 때  $d$ -그래프 내에 경로  $A_0 d^{a_1} A_1 \dots A_{i-1} d^{a_i} A_i \dots A_{n-1} d^{a_n} A_n$ 가 존재한다면  $M(G)$ 내에 어떤  $y_n, \dots, y_1, y_0 \in \Sigma^*$ 에 대해  $[\epsilon] \dots [a_1] \dots [a_1 \alpha_1 \dots \alpha_n] \mid A_n y_n \$\$ \Rightarrow^* [\epsilon] \dots [a_1] \dots [a_1 \alpha_1 \alpha_2] \mid y_1 \$\$ \Rightarrow^* [\epsilon] \dots [a_1] [a_1 A_1] \mid y_1 \$\$ \Rightarrow^* [\epsilon][A_0] \mid y_0 \$\$$ 가 존재한다. 그 역도 성립한다. □

성질 2로부터 타당한 LR 피싱 컨피규레이션  $[\epsilon] \dots [a] \mid \alpha x \$, x \in \Sigma^*$ 로부터 모든 가능한 리덕션 골들의 배열을  $\langle S', \alpha, a \rangle$  내의 경로들로부터 찾을 수 있다. 따라서 그 컨피규레이션에서 리덕션이 반드시 일어날 골은 관련된 경로들의 형태를 조사함으로써 구할 수 있다.

**3.2  $\Pi$ -관계**

**정의 1 ( $\Pi$ -관계)**  $A, B \in N, a$ 를  $G^A$ 의 자랄 수 있는 스트링,  $\alpha = \beta \gamma, a \in \Sigma$ 는  $Ad^+ a$ 인 골이라고 하자.  $(A, \alpha) \Pi_2^G(B, \gamma)$ 가 성립하기 위해선  $\langle A, \alpha, a \rangle$  내의 모든 경로  $A_0 d^{a_1} A_1 d^{a_2} X_2 \dots A_l d^{a_{l+1}} A_{l+1} \dots A_{n-1} d^{a_n} A_n$  (여기서  $A_0 = A, \alpha = \alpha_1 \alpha_2 \dots \alpha_n, A_n = a$ )에 대해  $A_l = B, \alpha_{l+1} \dots \alpha_n = \gamma$ 인  $l(1 \leq l \leq n)$ 이 존재한다. □

$(A, a)\Pi_a(B, \gamma)$ 는 그림 1의 트리 (a)내의  $\alpha$ 의 후위(suffix) 부분  $\gamma$ 와  $x$ 의 어떤 전위(prefix) 부분이 트리 (b)에서 보는 것과 같이  $B$ 로부터 유도됨에 틀림없다는 것을 함축(imply)한다.

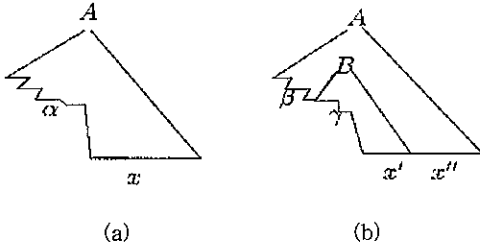


그림 1 유도 트리 ( $\alpha = \beta\gamma$ )

예제 1  $G_1 = (\{S, A, C, B, X, Y\}, \{a, b, c\}, P_1, S)$ 이고  $P_1 = \{S \rightarrow A, S \rightarrow C, A \rightarrow BX, A \rightarrow BY, C \rightarrow Ba, B \rightarrow b, X \rightarrow BA, Y \rightarrow BC, X \rightarrow c\}$ 이라 하자. 경로 집합  $\langle A, BBB, b \rangle$ ,  $\langle A, BBB, b \rangle$  내의 경로들이 그림 2에 나타나 있다. 그림에서  $(A, BBB)\Pi_b(A, B)$ 이고  $(A, BBB)\Pi_c(A, B)$ 임을 관측할 수 있다. 다음의 성질은  $\Pi$ -관계로부터 얻어지는 성질 2에 근거한 단순 결과이다.  $\square$

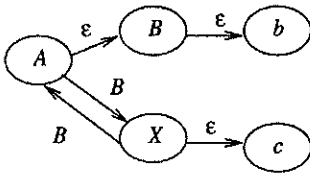


그림 2  $d$ -그래프의 부분

성질 3  $(A, a)\Pi_a(B, \gamma)$ ,  $\alpha = \beta\gamma$ 라 하자. 이때  $M(G)$ 내의  $[\epsilon]x\$\Rightarrow^*[\epsilon]\dots[\eta a]y\$\Rightarrow^*[\epsilon]\dots[\eta A]z\$\Rightarrow^*[\epsilon][S]y\$(1:  $y\$ = a$ )는 항상  $[\epsilon]x\$\Rightarrow^*[\epsilon]\dots[\eta a]y\$\Rightarrow^*[\epsilon]\dots[\eta\beta B]u\$\Rightarrow^*[\epsilon]\dots[\eta A]z\$\Rightarrow^*[\epsilon][S]y\$$  형태이다.  $\square$$

3.3 알고리즘

이 절에선 각 LR 상태  $q$ 에 대해  $(\hat{q}, A, a)$  형태의 레벨을 생성하는 알고리즘을 제시한다. 이 레벨은  $VALID(\eta) = \hat{q}$ ,  $VALID(\eta a) = q$ 인 모든 가능한 타당한 LR 파싱 컨피규레이션  $[\epsilon]\dots[\eta]\dots[\eta a]y\$$ 에 대해  $[\epsilon]\dots[\eta]\dots[\eta a]y\$\Rightarrow^*[\epsilon]\dots[\eta][\eta A]z\$$ 이 존재한다는 것을 의미한다.

레벨  $(\hat{q}, A, a)$ 에 대해  $a$ 로 인한 레벨링의 무한 가능성이 있다. 알고리즘 1에서  $\Pi$ 관계는 이런 무한한 레벨링 과정을 제한하는 데에 기여한다. 그러나 여전히 무한한 가능성은 남아 있다.  $d$ -그래프상의 순환적(cyclic) 경로 집합이란 개념이 레벨링에 제한을 가하기 위해 정의된다. 경로 집합  $\langle A, a, a \rangle$ 에 대해  $|a| = n, a = X_1 X_2 \dots X_n$ 이라 하자. 이때  $Y_i = A_{i,0} d^\epsilon A_{i,1} d^\epsilon \dots d^\epsilon A_{i,m}$ ,  $(i = 0, 1, \dots, n, m, m \geq 0), A_{0,0} = A, A_{n,m} = a$ 인  $Y_0 d^{X_1} Y_1 d^{X_2} \dots d^{X_n} Y_n$ 가  $\langle A, a, a \rangle$ 내에 있다고 하자. 각  $i = 0, 1, \dots, n$ 에 대해  $Y_i$ 는 세그먼트  $i$ 라고 한다. (그림 3을 참조하라.)

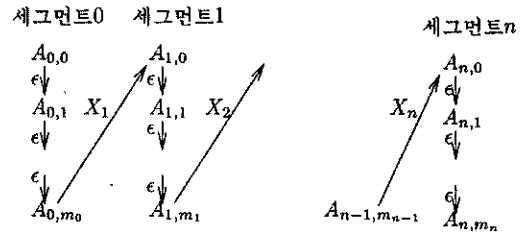


그림 3  $\langle A, a, a \rangle$ 내의 세그먼트  $i$

또한  $SEG_i \langle A, a, a \rangle = \{A_{i,0}, A_{i,1}, \dots, A_{i,m} | A_{i,0} d^\epsilon A_{i,1} d^\epsilon \dots d^\epsilon A_{i,m}\}$ 는  $\langle A, a, a \rangle$ 내의 세그먼트  $i$ 인 경로이다)로 정의하자.  $SEG_0, SEG_1, \dots, SEG_n$ 에 대해 만일  $SEG_i = SEG_j, (0 \leq i < j \leq n)$ 이고  $SEG_i, SEG_{i+1}, \dots, SEG_n$ 내의 어떤 다른 쌍도 같지 않다면,  $SEG_i, SEG_{i+1}, \dots, SEG_n$ 는 루프(loop)라고 한다.  $SEG_i, SEG_{i+1}, \dots, SEG_n$ 이 루프인 세 개 이상의 다른  $i, 1 \leq i \leq n$ 가 동일 루프에 대해 존재하면,  $\langle A, a, a \rangle$ 는 순환적이라 한다. 경로 집합  $\langle A, a, a \rangle$ 는 어떤  $B, \gamma$ 에 대해  $(A, a)\Pi_a(B, \gamma)$ 가 존재한다면 나눌 수 있다(divisible)라고 한다. 만일  $\langle A, a, a \rangle$ 가 나눌 수 없는 순환적 경로 집합이라면,  $\langle A, a, a \rangle$ 내의 루프를 포함하는 무한한 순환적 경로 집합을 레벨링으로부터 제외하는 것이 합리적이다. 이런 취지에서 다음의 알고리즘에선 만일  $\langle A, a', a' \rangle (a = a'a)$ 이 나눌 수 없는 순환적 경로 집합이면 레벨  $(\hat{q}, A, a)$ 는 레벨 집합에서 제외된다. 알고리즘 1에서 LABEL는 각 LR 상태들에 대해 레벨들의 집합을 가지는  $Q$ 로부터  $2^{Q \times N \times V}$ 로의 테이블이다.

```

입력:  $M(G), \Pi^{non-cyclic} (\Pi^{non-cyclic} = ((A, a)\Pi_a(B, \gamma) | \langle A, a, a \rangle \text{는 순환적이지 않다}))$ 
출력: LABEL 테이블
방법:
1 LABEL( $q_0$ ) =  $\{(q_0, S, \epsilon)\}$ 
2 for 시작 상태가 아닌 모든 LR 상태  $q$  do LABEL( $q$ ) =  $\emptyset$  endfor
3 repeat
  for  $q \in Q$  do
    (a) for  $(\hat{q}, A, a) \in LABEL(q)$  과  $a \in FIRST(q)$  do
      -for  $\langle A, a \rangle \Pi_a^{non-cyclic}(B, \gamma)$  가 성립하는  $B, \gamma$  do  $\hat{p}$  를
        GOTOX( $\hat{q}, \beta$ ) ( $\alpha = \beta\gamma$ ) 라 하자.
      (i)  $\hat{p} = GOTOX(\hat{q}, B)$  일때 LABEL( $\hat{p}$ ) = LABEL( $\hat{p}$ )  $\cup$ 
         $\{(\hat{q}, A, \beta B)\}$ 
      (ii) if  $[C \rightarrow \delta, a\epsilon, w] \in q$  가 존재한다 then
         $\hat{p} = GOTOX(\hat{q}, a)$  일때 LABEL( $\hat{p}$ ) = LABEL( $\hat{p}$ )  $\cup$   $\{(\hat{q}, B, \gamma a)\}$ 
        endif
      (iii) if  $[C \rightarrow \delta, a] \in q$  가 존재한다 then
         $\hat{p} = GOTOX(\hat{q}, \rho C)$ ,  $\gamma = \rho\delta$  일때
        LABEL( $\hat{p}$ ) = LABEL( $\hat{p}$ )  $\cup$   $\{(\hat{q}, B, \rho C)\}$  endif
      endfor
    - if  $\langle A, a \rangle \Pi_a^{non-cyclic}(B, \gamma)$  인  $B$  와  $\gamma$  가 존재하지 않는다 then
      (iv) if  $[C \rightarrow \delta, a\epsilon, w] \in q$  가 존재한다 then
         $\hat{p} = GOTOX(\hat{q}, a)$  일때
        LABEL( $\hat{p}$ ) = LABEL( $\hat{p}$ )  $\cup$   $\{(\hat{q}, A, a\alpha)\}$  endif
      (v) if  $[C \rightarrow \delta, a] \in q$  가 존재한다 then
         $\hat{p} = GOTOX(\hat{q}, \beta C)$ ,  $\alpha = \beta\delta$  일때 LABEL( $\hat{p}$ ) = LABEL( $\hat{p}$ )  $\cup$ 
         $\{(\hat{q}, A, \beta C)\}$  endif
      endif
    endfor
  (b) if  $\langle A, a', a \rangle$  이 순환적인 조건하에  $(\hat{q}, A, a) (a = a'a)$  가 새로이
    LABEL( $q$ ) 에 추가되었다 then LABEL( $q$ ) 로 부터  $(\hat{q}, A, a)$  를
    제거한다 endif
  endfor
until LABEL 테이블이 변하지 않는다
  
```

알고리즘 1 LR 상태들의 레벨링

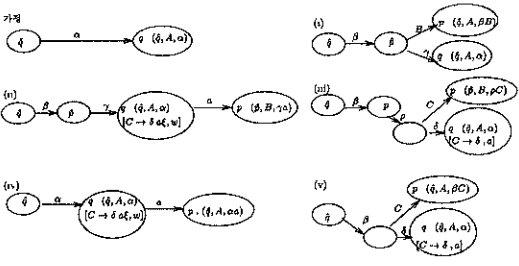


그림 4 레벨된 LR 상태들

그림 4는 알고리즘 1내의 단계 (i), (ii), (iii), (iv), (v)에 따라 레벨된 LR 상태들을 보인다.  $\Pi^{non-cyclic}$  를 대신한  $\hat{\Pi} \in \Pi^{non-cyclic}$  는 알고리즘 1내에 적용될 수 있

다. 이렇게  $\hat{\Pi}$  로써 레벨된 LR 오토마톤은  $M(G, \hat{\Pi})$  로 나타낸다.

예제 2 예제 1의  $G_1$ 에서  $\hat{\Pi}$  는  $(S, B)\Pi_b(A, B)$ ,  $(S, B)\Pi_c(A, B)$ ,  $(A, BBB)\Pi_b(A, B)$ ,  $(A, BBB)\Pi_c(A, B)$  으로 구성된다고 하자. 그림 5는 레벨된 LR 오토마톤을 보인다.

보조 정리 1  $(\hat{q}, A, a) \in LABEL(q)$  이라 하자. VALID( $\eta$ ) =  $\hat{q}$ , VALID( $\eta a$ ) =  $q$  인 모든 타당한 LR 파싱 컨피규레이션  $[e] \dots [\eta] \dots [\eta a] \dots$  에 대해  $M(G)$  내에  $[e] \dots [\eta] \dots [\eta a] \dots \Rightarrow^* [e] \dots [\eta] \dots [\eta A] \dots$  가 존재한다.

(증명) 이 보조 정리는 알고리즘 1의 (a) 블록의 반복의 수에 대한 인덕션에 의해 증명될 수 있다. 즉 (a) 블록내에  $(\hat{q}, A, a) \in LABEL(q)$  에 대해 본 보조 정리가 성립한다고 가정하자. 이 때 성질 3을 이용하여 (i), (ii), (iii)에 의한 LABEL 테이블에 추가된 레벨들은 본 보조 정리를 만족함을 보일 수 있다. 또한 (iv), (v)에 의해 추가된 레벨들 역시 레벨링의 원리에 의해 본 보조 정리가 성립함을 쉽게 알 수 있다.

미리 결정된 꼴을 가진 LR 파서는 레벨된 LR 오토마톤에 근거해 다음의 알고리즘에 의해 생성될 수 있다. CURRENT\_LABEL은 주어진 스택 스트링에 대한 미리 결정될 수 있는 꼴들의 집합을 갖는다.

```

입력 스택 스트링  $[e] \dots [\theta]$ 
출력  $[e] \dots [\theta]$  에 대한 CURRENT_LABEL
방법:
1. CURRENT_LABEL =  $\{(A, a) | \theta = \eta a \text{ 일 때 } (VALID(\eta), A, a) \in LABEL(VALID(\theta))\}$ 
2. if CURRENT_LABEL =  $\emptyset$  then  $[e] \dots [\theta]$  의 전위 스트링 중에서  $[e] \dots [\eta]$  에 대한 CURRENT_LABEL이 공집합이 아닌 가장 긴 길이의  $[e] \dots [\eta]$  를 찾는다.  $\theta = \eta a$  일때 CURRENT_LABEL =  $\{(B, \beta\gamma) | [e] \dots [\eta]$  에 대한 CURRENT_LABEL내에  $(\hat{q}, B, \beta)$  가 존재한다  $\}$  endif
  
```

알고리즘 2 미리 결정되는 꼴의 생성

실제 구현에선 응용 대상에 따라 상태  $q$  의 LABEL( $q$ ) 의 원소를 관심있는 레벨들만으로 국한하여 LABEL 공간의 크기를 줄일 수 있다. 가령 가장 긴 길이의 레벨 또는 가장 짧은 길이의 레벨과 같이 특별한 관심 대상을 선택할 수 있다. 이런 LABEL 공간의 축소는 CURRENT\_LABEL의 계산 시간을 줄임에 알 수 있다. 더불어 구현시에  $(q, A, a)$  를 LABEL 집합에 넣는 대신에  $(q, A, |a|)$  를 저장하는 방법도 생각할 수 있다.

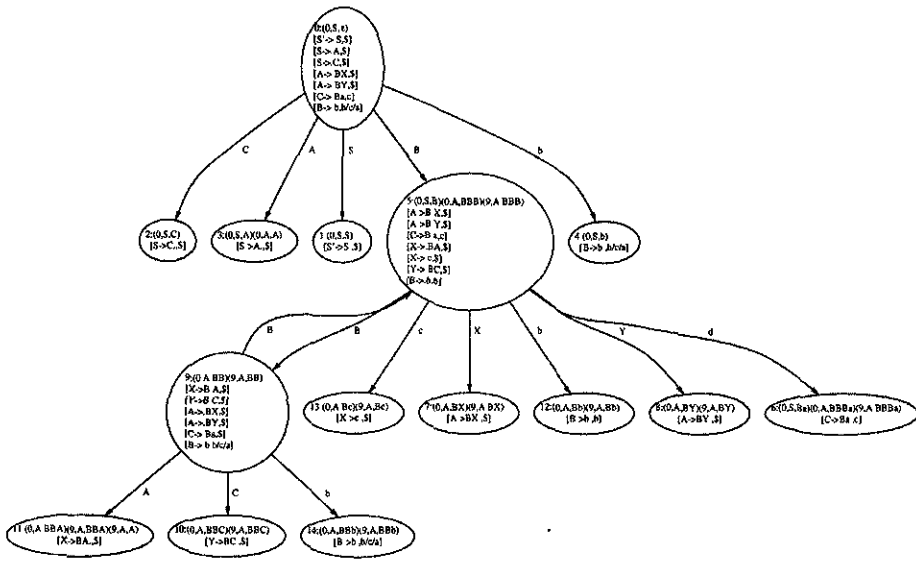


그림 5 레벨된  $M(G_1, \hat{T})$

그리고 알고리즘 2의 단계 2에서 가장 긴 길이의 전위 스트링을 찾는 과정은 파싱 시간의 복잡도를 높일 수 있으나, 실용적으로 사용되는 문법에 대해 CURRENT\_LABEL이 공집합이 되는 경우는 매우 드물다.

예제 3 예제 2의  $M(G_1, \hat{T})$ 내의 입력 스트링 "bbbc"이 주어졌다고 하자. 초기 컨피규레이션에서 최종 컨피규레이션까지의 CURRENT\_LABEL의 변화를 다음에서 보인다.

컨피규레이션 CURRENT_LABEL	0) bbbcs (S, ε)	1) bbcs (S, b)	2) bcs (S, B)	3) cs (A, BB)
1) bbcs (A, BB)	1) bbcs (A, BB)	2) bcs (A, BB)	3) cs (A, Bc)	4) cs (A, BX)
1) cs (A, BA)	1) cs (A, BX)	2) cs (S, A), (A, A)	1) cs (S, S)	

다음의 정리는 보조 정리 1로부터 얻어진다.

정리 1 [ε]...[θ]의 CURRENT\_LABEL내에 (A, a)가 있다고 하자. 모든 타당한 LR 파싱 컨피규레이션 [ε]...[θ]에 대해  $M(G_1, \hat{T})$ 내에  $\theta = \eta a$ 인 [ε]...[θ]에 대해 [ε]...[η][A]가 존재한다. □

### 4. 응용

3장에서 제안된 예상할 수 있는 리덕션 골을 가진

LR 파서의 두 가지 적용 가능 예를 보인다.

#### 4.1 가능한 리덕션 골(feasible reduction goal)들의 질제

가능한 리덕션 골(feasible reduction goal) [5]은 구단위(phrase-level) 오류 복구에서 유용한 개념이며, 미리 결정된 리덕션 골은 가능한 리덕션 골의 유용한 정제(refine)된 개념이다.

유일하게 결정될 수 있는 리덕션 골을 찾기 위한 몇 가지 경험적인 전략들이 [5]에서 제안되었다. 그러나 비록 가능한 리덕션 골이 유일하게 결정되더라도, 남은 입력 스트링들이 그 골로 리덕션이 될지는 불확실하다.

$q_0 a_1 \dots q_m a_1 \dots a_n$ 를 오류 컨피규레이션 [4]이라고 하자. 그림 6에서 보듯이 [4, Vol. II, page 305]에서 정의된 F에 대해  $F(q_{i+1} \dots q_m) = \{A\}$ 이나 아이템  $[C \rightarrow \gamma.X_{i+1} \dots X_m]$ 가  $q_i$  내에 존재하기에 A로의 리덕션을 보장할 수 없다.

이와 비교하여 오류가 발생한 컨피규레이션의 스택 스트링에 대해 알고리즘 2에 의해 예상된 골은 리덕션이 보장된 골이다. 즉, 스택 스트링  $q_0 a_1 \dots a_n$ 에 대해  $(A, a) \in \text{CURRENT\_LABEL}$ 이면, 정리 1에 의해 오류 컨피규레이션이 타당한 LR 파싱 컨피규레이션이라고 가정할 때  $q_0 a_1 \dots a_n$ 에서 A로의 리덕션이 존재한다. 더욱이 가능한 리덕션 골들의 이런 정제(refinement)는

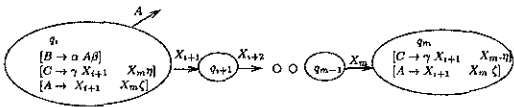


그림 6  $\gamma \neq \varepsilon$

경험에 의존하지 않고 각 상태의 레벨로부터 직접 얻어질 수 있다.

4.2 우문맥의 계산

우문맥 [3]은 LR에 기반을 둔 파서의 오류 보정 (repair)에 유용한 개념이다. 여기선 우문맥의 계산이 미리 결정된 끝들을 이용하여 효율적으로 인수분해될 수 있음을 보인다.  $\alpha \in V^*, R \in 2^{V^*}$ 에 대해  $\alpha \cdot R$ 는  $\{\alpha\beta \mid \beta \in R\}$ 을 의미하고,  $Q, R \in 2^{V^*}$ 에 대해  $Q \cdot R$ 는  $\{\alpha\beta \mid \alpha \in Q, \beta \in R\}$ 을 의미한다. 독자의 편의를 위해서 다음에서 우문맥의 정의를 보인다.

정의 2 (우문맥) 스택 스트링  $q_0 q_1 \dots q_n$ 에 대해  $X_i$ 를  $q_i, i(1 \leq i \leq n)$ 의 엔트리 끝이라고 하고,  $q_{i,A} = GOTO(q_i, A)$ 라고 하자. 이 때  $q_0 q_1 \dots q_n$ 의 우문맥은  $RC(q_0 q_1 \dots q_n) = \bigcup_{\{A \rightarrow X_{i+1}, X_n, \beta, w \mid \in KERNEL(q_0)\}}$

$\{RC(X_{i+1} \dots X_n, \beta, w)\}, 1 \leq i \leq n$ 으로 정의된다. 여기서  $[A \rightarrow X_{i+1} \dots X_n, \beta, w] \neq [S \rightarrow S, \emptyset]$ 이면  $RC(q_0 q_1 \dots q_n, [A \rightarrow X_{i+1} \dots X_n, \beta, w]) = \beta \cdot RC(q_0 q_1 \dots q_n, \beta \cdot RC(q_0 q_1 \dots q_n, q_{i,A}))$ , 그 밖의 경우엔  $RC(q_0 q_1 \dots q_n, [A \rightarrow X_{i+1} \dots X_n, \beta, w]) = \varepsilon$ 이다. □

본 절에선 위의  $RC$ 는  $d$ -그래프의 경로 집합의 관점에서 표현될 수 있음을 보일 것이다. 이를 위해  $d$  관계는 다음과 같이 정제(refine)되고 확장된다.  $A d_\beta^\alpha X$  if  $A \rightarrow \alpha X \beta \in P, X \in V; A d_\varepsilon^\alpha \varepsilon$  if  $A \rightarrow \alpha \in P$ . 또한  $A d_\varepsilon^\alpha A$ 는  $A d_\varepsilon^\alpha A$ 을 나타내고, 이 관계의  $n(n > 0)$  번째 승(power)은 그림 7에 정의된다.  $A d_\beta^{X^*}$ 로 나타내는 재귀적 전이 클로저는  $\bigcup_{n \geq 0} A d_\beta^n X$ 으로 정의된다.

$A \in N, \alpha \in V^*, X \in \mathcal{W} \cup \{\varepsilon\}$ 에 대해  $\langle A, \alpha, X \rangle$ 에 대한 경로 집합 우문맥은  $RCP\langle A, \alpha, X \rangle = \{\beta \mid A d_\beta^{X^*}\}$ 으로 정의된다. 이 때 다음의 식이 성립함을 관측할 수 있다.

$$RCP\langle S', X_1 \dots X_{n-1}, X_n \rangle = \{\beta \theta S' d_\beta^{X_1^* X_2^* \dots X_{n-1}^* X_n}\} = \bigcup_{\{A \rightarrow X_{i+1}, X_n, \beta, w \mid \in KERNEL(q_0)\}} \beta \cdot RCP\langle S', X_1 \dots X_i, A \rangle$$

(여기서  $q_n = GOTO(q_0, X_1 \dots X_n)$ 이다).

이 식에서 보듯이  $RCP$ 의 계산은  $RC$ 의 계산과 같은

구조를 갖기에,  $RC$ 는  $RCP$ 에 의해 다음 성질에서 보는 것처럼 표현할 수 있다.

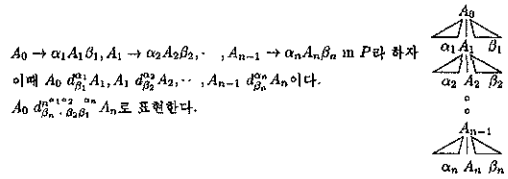


그림 7 정제된  $d$ 관계와 유도 트리

성질 4  $G$ 에 대해  $X_i$ 를  $i(1 \leq i \leq n)$ 에 대해  $q_i$ 의 엔트리 끝이라 하고,  $n > 0$ 이라 하자. 이 때  $RC(q_0 q_1 \dots q_n) = RCP\langle S', X_1 \dots X_{n-1}, X_n \rangle$ 이다. (여기서  $n=1$ 일 땐  $X_1 \dots X_{n-1} = \varepsilon$ 이다.) □

한편  $RCP$ 의 계산은 관련된 경로들의 형태에 따라 다음과 같이 표현될 수 있다.

성질 5 모든 경로  $A d^{X^*} X$ 가  $\alpha = \beta \gamma$ 인 조건 하에 경로  $A d^{X^*} B$ 와 경로  $B d^{X^*} X$ 의 합성으로 표현된다고 하자. 이 때  $RCP\langle A, \alpha, X \rangle = RCP\langle B, \gamma, X \rangle \cdot RCP\langle A, \beta, B \rangle$ 이다. □

성질 4, 성질 5, 정리 1을 이용하여 다음의 정리를 얻을 수 있다.

정리 2  $G$ 에 대해  $q_i$ 의 엔트리 끝을  $X_i(1 \leq i \leq n)$ 이고  $0 < m \leq n$ 이라 하자. 스택 스트링  $q_0 q_1 \dots q_n$ 에 대해  $(A, X_{m+1} \dots X_n) \in \text{CURRENT\_LABEL}$ 이라고 하면  $RC(q_0 q_1 \dots q_n) = RCP\langle A, X_{m+1} \dots X_{n-1}, X_n \rangle \cdot RC(q_0 q_1 \dots q_m q_{m,A})$ 이다.

(증명) 정리 1과  $M(G)$ 가 결정적이라는 성질을 이용하여,  $M(G)$  내의  $[\varepsilon] \vdash \Rightarrow^* [\varepsilon] \dots [X_1 \dots X_n] \vdash \Rightarrow^* [\varepsilon] [S] \vdash \Rightarrow^* [\varepsilon] \dots [X_1 \dots X_n] \vdash \Rightarrow^* [\varepsilon] \dots [X_1 \dots X_m] \vdash \Rightarrow^* [\varepsilon] [S] \vdash \Rightarrow^* [\varepsilon] [S] \vdash$  형태가 된다. 성질 2에 의해  $S d^{X^* X_{n-1}^*} X_n$  경로는 항상  $S d^{X_1^* X_2^* \dots X_n^*} X_n$ 의 형태이다. 성질 5에 의해  $RCP\langle S, X_1 \dots X_{n-1}, X_n \rangle = RCP\langle A, X_{m+1} \dots X_{n-1}, X_n \rangle RCP\langle S, X_1 \dots X_m, A \rangle$ 이다. 성질 4에 의해  $RC(q_0 q_1 \dots q_n) = RCP\langle A, X_{m+1} \dots X_{n-1}, X_n \rangle RC(q_0 q_1 \dots q_m q_{m,A})$ 이다. □

$RC(q_0 q_1 \dots q_n)$ 의 계산은  $RC(q_0 q_1 \dots q_m q_{m,A})$ 의 관점에서 인수분해되었다. 한편  $RC(q_0 q_1 \dots q_m q_{m,A})$ 는 정리 2를 순환적으로 적용함에 의해서  $RCF$ 로 표현될 수 있다. 여기서 구현시에 LR 상태의 레벨에 따른  $RCP$ 의



계산은 미리 수행할 수 있음을 알 수 있다.

**예제 4** 예제 2의  $M(G_1, \hat{M})$ 에 대해  $\sigma \in Q^*$ 인 스택 스트링  $\sigma q_9 q_5 q_9$ 과  $\sigma q_9 q_5 q_9 q_5$ 에 대한 우문맥을 정리 2를 적용하여 계산해 보자. (i)  $\sigma q_9 q_5 q_9$ 에 대해  $(A, BB) \in \text{CURRENT\_LABEL}$ 이다.  $A d_A^* B, A d_C^* B$ 이기에  $RCP(A, B, B) = \{A, C\}$ 이다.  $GOTO(q_9, A) = q_{11}$ 이기에  $RC(\sigma q_9 q_5 q_9) = \{A, C\} \cdot RC(\sigma q_{11})$ 이다. (ii)  $\sigma q_9 q_5 q_9 q_5$ 에 대해  $(A, BBB) \in \text{CURRENT\_LABEL}$ 이다.  $A d_X^{*ab} B, A d_Y^{*ab} B, A d_a^{*ab} B$ 이기에  $RCP(A, BB, B) = \{X, Y, a\}$ 이다. 또한  $GOTO(q_9, A) = q_{11}$ 이기에  $RC(\sigma q_9 q_5 q_9 q_5) = \{X, Y, a\} \cdot RC(\sigma q_{11})$ 이다. 정의 2에 의하면 경우 (i)에 대해  $RC(\sigma q_9 q_5 q_9) = RCI(\sigma q_9 q_5 q_9, [X \rightarrow B, A, \$]) \cup RCI(\sigma q_9 q_5 q_9, [Y \rightarrow B, C, \$]) = A \cdot RC(\sigma q_{11}) \cup C \cdot RC(\sigma q_{11})$ 이기에  $RC(\sigma q_{11})$ 는 두 번 계산된다. 유사한 이유로 경우 (ii)에 대해선  $RC(\sigma q_{11})$ 는 세 번 계산된다.  $\square$

[3]에서 제안된 RC 계산을 위한 방법은 LR 상태를 간단히 조사함으로써 쉽게 리덕션 골을 예상할 수 있는 경우를 몇 가지 경우로 나누어 정형화한 것으로 생각될 수 있다. 한편 그 방법에선 예제 4의 중복 계산을 해결하지 못한다.

### 5. 결론

본 논문에선 리덕션 시간 전에 예상할 수 있는 리덕션 골들을 가진 LR 파서를 제안하였고, 이 파서에 대한 구단위 오류 복구와 우문맥 계산의 응용을 논하였다. 제안된 LR 파서에 대한 알고리즘은 단지 기본적인 방법을 제시하는 것이며, 실제의 응용을 위해서는 LABEL집합에 대한 선택을 신중히 해야 할 것이다.

### 참고 문헌

[1] Hammer, M., A New Grammatical Transformation into deterministic top down form., p.301, MIT, Mass., Project MAC Technical Report TR-119, 1974.  
 [2] Lee, Gyung-Ok and Choe, Kwang-Moo, An Extension of LL(k) Covering Grammars, KAIST, Technical Report, CS-TR-98-133, 1998.  
 [3] Jung, Min-Soo, Choe, Kwang-Moo, and Han, Taisook, "An Efficient Computation of Right Context for LR-based Error Repair," pp.63-72, Information Processing Letters, Vol 49, No 2, 1994.  
 [4] Sippu, S. and Soisalon-Soininen, E., Parsing Theory, vol I, II, p.228, p.426, Springer-Verlag

Berlin Heidelberg, 1990.

[5] Sippu, S. and Soisalon-Soininen, E., "A Syntax-Error-Handling Technique and Its Experimental Analysis," pp.656-679, ACM Tran. on Prog. Lang. and Sys., Vol 5, No 4, 1983.



**이 경 옥**  
 1990년 2월 서강대학교 컴퓨터학과 학사.  
 1992년 8월 한국과학기술원 전산학과 석사.  
 1992년 9월 ~ 현재 한국과학기술원 전산학과 박사과정. 관심분야는 프로그래밍 언어론과 컴파일러 구성론 등



**최 광 무**  
 1976년 서울대학교 전자공학과 졸업(학사).  
 1978년 한국과학기술원 전산학과 졸업(석사).  
 1984년 한국과학기술원 전산학과 졸업(박사). 현재 한국과학기술원 전산학과 교수. 관심분야는 프로그래밍 언어론, 논리 프로그램의 병렬 수행 및 컴파일러 구성임