단일 상태 파싱 오토마톤의 생성

(Constructing a of Single State Parsing Automaton)

이 경 옥 †

(Gyung-Ok Lee)

요 약 일반 오토마톤은 다중 입력 전이를 허용하기에 과거 전이 경로가 필요한 경우에는 별도 작업이 필요하다. LR 오토마톤의 경우는 스택을 이용하여 과거 전이 경로를 저장한 후에, 파싱 시에 이를 이용한다. 한편 과거 경로의 정보를 포함하도록 상태 구성이 가능한 경우에는 과거 정보 추적을 위한 오버헤드를 피할 수가 있다.

본 논문에서는 과거 전이에 의존하지 않는 단일 상태 파싱 오토마톤을 제안한다. 적용가능한 문법 클래스는 LR 문법보다 작으나, 오토마톤의 상태가 과거 경로의 정보를 포함하기에 LR 오토마톤과 달리 파싱 시에 과거 정보의 추적이 불필요하다.

키워드 : LR 오토마톤, 파싱 오토마톤, 단일 상태 파싱 오토마톤

Abstract A general automaton allows multiple input transitions, so a special treatment is required when the history of transitions is needed. An LR automaton keeps the past transitions in the stack to use them during parsing. On the other hand, when each state in an automaton contains in itself the past transition history, the trace overhead of past transitions is unnecessary.

The paper suggests a single state parsing automaton that does not depend on the past transitions. The applicable grammar class is less than LR grammars, but each state in a new automaton contains the past information, so the tracing of the history is not required compared to LR automaton.

Key words : LR automaton, parsing automaton, single state parsing automaton

1. 서 론

일반 오토마톤에서는 다중 방향 전이를 허용하기에 과거의 상태 전이 경로가 필요한 경우에는 별도의 처리가 필요하다. LR 오토마톤[1]의 경우에는 스택을 이용하여 과거의 경로를 저장한 후에 파싱 시에 이를 이용한다. 이와 달리 오토마톤의 행동이 지난 경로에 의존없이 현 상태로만 결정 가능한 경우에 이를 *단일 상태 파싱 오토마톤*이라 한다.

본 논문에서는 단일 상태 파싱 오토마톤의 생성과 이에 기반한 파싱 방법을 제시한다. 제안되는 단일 상태 파싱 오토마톤은 단순성으로 인하여 적용가능 문법 클래스는 LR 문법보다 작지만, 상태 자체 내에 지나온 경로 정보가 포함되어 있기에 별도의 과거 경로 정보를 위한 노력이 불필요하다는 특성을 가진다.

2장에서 본 논문의 전개를 위한 기본 사항을 정의하며, 3장에서 단일 상태 파싱 오토마톤의 생성을 위한 모델을 제시하고, 단일 상태 오토마톤에 기반한 파싱 방법을 제시한다. 이후 4장에서 단일 상태 오토마톤의 성질에 관해 논한 후, 5장에서 결론을 맺는다.

· 이 논문은 한신대학교 학술연구비 지원에 의하여 연구되었음

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

논문접수 : 2008년 4월 14일
심사완료 : 2008년 10월 9일

정보과학회논문지: 소프트웨어 및 응용 제35권 제11호(2008.11)

2. 기본 정의

본 논문에서는 [2-4]에서의 정의와 표기법을 따른다.

문법 G = (N, Σ, P, S)는 $S' \rightarrow S\$^k$로 확장된 LR(k) 문법임을 가정한다. $\alpha \in V^*$ ($V = N \cup \Sigma$)에 대해서 $k{:}\alpha$, $\alpha{:}k$는 각각 α의 길이 k의 전위 스트링과 후위 스트링을 표기한다. $FIRST_k^G(\alpha) = \{k{:}x \mid \alpha \Rightarrow^* x,\ x \in \Sigma^*\}$, $FOLLOW_k^G(\alpha) = \{k{:}x \mid S \Rightarrow^* \beta\alpha x,\ x \in \Sigma^*\}$, $RC_k^G(\alpha) = \{k{:}xz \mid S \Rightarrow^*_{rm} \beta Bz \Rightarrow_{rm} \beta\gamma\delta z \Rightarrow^*_{rm} \beta\gamma xz,\ \alpha = \beta\gamma,\ xz \in \Sigma^*\}$이다. 이들의 표기 방법에서 G가 문맥

상 명확할 경우는 이를 생략한다. 문법 유도를 나타내는 $\Rightarrow_{A,R}$은 다음과 같이 정의된다: $Ar \Rightarrow_{rm}^* \gamma Bzr \Rightarrow_{rm}^p \gamma X\delta zr$이 존재한다고 하자. $\gamma = \varepsilon$, $X = A$인 경우에는 $FIRST_k(\delta zr) \cap R = \varnothing$일 때만 $\gamma Bzr \Rightarrow_{A,R}^p \gamma X\delta zr$이고, $\gamma = \varepsilon$, $X = A$가 아닌 경우에는 항상 $\gamma Bzr \Rightarrow_{A,R}^p \gamma X\delta zr$이 성립한다. $RC_k^{A,R}(\alpha)$은 집합 $\{k{:}yzr \mid Ar \Rightarrow_{A,R}^* \beta Bzr \Rightarrow_{A,R} \beta\gamma\delta zr \Rightarrow_{A,R}^* \beta\gamma yzr,\ x \in \Sigma^*\}$을 표기한다.

$A \in N$, $r, z \in \Sigma^k$, $X \in N \cup \{\varepsilon\}$, $\alpha \in V^*$에 대해 $(A, r)\ d^{\alpha}\ (X, z)$는 $A \rightarrow \alpha X\beta \in P$, $r \in FOLLOW_k(A)$, $z \in FIRST_k(\beta r)$인 경우에 성립한다. d 관계에 대응하는 d-그래프의 경로는 $(A_0, r_0)\ d^{\alpha 1}\ (A_1, r_1)\ d^{\alpha 2}\ (A_2, r_2) \ldots (A_{n-1}, r_{n-1})\ d^{\alpha n}\ (A_n, r_n)$이다. $\langle A, r, \alpha, u\rangle$는 집합 $\{h \mid h = (A_0, r_0)\ d^{\alpha 1}\ (A_1, r_1)\ d^{\alpha 2}\ (A_2, r_2) \ldots (A_{n-1}, r_{n-1})\ d^{\alpha n}\ (A_n, r_n),\ A_0 = A,\ r_0 = r,\ \alpha = \alpha_1\alpha_2 \ldots \alpha_n,\ A_n r_n = u\}$를 나타낸다.

정의 2.1 $(A, R, \beta\gamma)\ \Pi_u\ (B, W, \gamma)$가 성립하기 위한 필요 충분 조건은 다음과 같다:

$\langle A, r, \beta\gamma, u\rangle$, $r \in R$내의 각 경로에 대한 $|\beta|$-세그먼트 [4]를 $(A_0, r_0)\ d^{\varepsilon}\ (A_1, r_1) \ldots (A_{n-1}, r_{n-1})\ d^{\varepsilon}\ (A_n, r_n)$라고 하면, $A_m = B$를 만족하는 m $(1 \le m \le n)$이 존재하고, $W = \cup_h \{r_m\}$(여기서 합집합은 $\langle A, r, \beta\gamma, u\rangle$내의 모든 경로 h에 대한 것이다)이 성립한다. □

3. 파싱 오토마톤

본 장에서는 단일 상태 파싱 오토마톤 모델을 정의하고, 생성되는 파싱 오토마톤에 기반한 파싱 방법을 제시한다.

3.1 파싱 오토마톤 모델

파싱 오토마톤 모델은 파싱 오토마톤과 Indivisible 집합으로 구성된다. Indivisible 집합은 상태의 무한 여부를 통해서 올바른 파싱 오토마톤의 생성을 판단하기 위해 사용된다.

정의 3.1 (파싱 오토마톤 모델)

고정된 문법 G와 Π의 부분 관계 Π′에 대한 파싱 오토마톤 모델 $P(G, \Pi') = (M(G, \Pi'), Indivisible(G, \Pi'))$이다. (여기서 Π′는 $(A, R, \alpha)\ \Pi'_u\ (B, W, \gamma)$, $(A, R, \alpha)\ \Pi'_u\ (C, X, \delta)$인 경우에 $(B, W, \gamma) = (C, X, \delta)$을 만족한다.)

(1) $M(G, \Pi') = (Q, \rightarrow, [S, \{\$^k\}, \varepsilon, FIRST_k(S\$^k)], [S, \{\$^k\}, S, \{\$^k\}])$

(i) 시작 상태는 $[S, \{\$^k\}, \varepsilon, FIRST_k(S\$^k)]$이다.

(ii) 최종 상태는 $[S, \{\$^k\}, S, \{\$^k\}]$이다.

(iii) 상태 집합 Q는 $N \times 2^{\Sigma^k} \times V^* \times 2^{\Sigma^k}$의 부분집합이다:

$Q =_s \{[S, \{\$^k\}, \varepsilon, FIRST_k(S\$^k)]\} \cup \{[S, \{\$^k\}, S, \{\$^k\}]\} \cup \{p \mid q \rightarrow p,\ q \in Q\} \cup \{p \mid q \rightarrow^a p,\ q \in Q,\ a \in \Sigma\}$

여기서 $=_s$는 우변에 기술된 조건을 만족하는 가장 작은 집합을 말한다.

(iv) →는 $Q \times (\Sigma \cup \{\varepsilon\})$에서 Q로의 함수로 다음과 같이 정의된다.

$[A, R, \alpha, U] \in Q$, $Z = \{u \in U \mid (A, R, \alpha)\ \Pi'_u\ (B, W, \gamma)\}$라 하자.

(a) $A\ d^{*\alpha}\ (a, z)$, $az \in U$, $V = RC_k^{A,R}(\alpha a)$, $FIRST_k(aV) \cap Z = \varnothing$인 조건하에

$[A, R, \alpha, U] \rightarrow^a [A, R, \alpha a, V]$이 생성된다.

(b) $(A, R, \alpha)\ \Pi'_u\ (B, W, \gamma)$ $(\alpha = \beta\gamma,\ u \in U)$가 존재하는 경우에,

$V = \{u \in U \mid (A, R, \alpha)\ \Pi'_u\ (B, W, \gamma)\}$인 조건하에

$[A, R, \alpha, U] \rightarrow [B, W, \gamma, V]$이 생성된다.

(2) $Indivisible(G, \Pi) = \{[A, R, \alpha, U] \in Q \mid [A, R, \alpha, U]$은 싸이클릭 상태 [4]이다$\}$ □

정의 3.2 $P(G, \Pi') = (M(G, \Pi'), Indivisible(G, \Pi'))$에 대해서 $Indivisible(G, \Pi') = \varnothing$이면 $M(G, \Pi')$는 *올바른 오토마톤*이고, 그렇지 않는 경우에는 $M(G, \Pi')$는 *올바르지 않는 오토마톤*이다. □

Π′ 집합의 선택에 따라서 주어진 문법에 대해 생성된 오토마톤이 올바른 오토마톤이거나 올바르지 않는 오토마톤이 된다. 또한 어떤 문법의 경우에는 올바른 오토마톤을 생성할 수 있는 Π′이 존재하지 않기도 한다. 그러므로 올바른 오토마톤을 생성할 수 있는 문법은 LR 문법의 부분집합이다. 논문의 원활한 진행을 위해서 앞으로 기술되는 G는 올바른 오토마톤을 생성하는 문법이며, $M(G, \Pi')$는 G의 올바른 오토마톤임을 가정한다.

정의 3.4 주어진 $M(G, \Pi')$상에 ACTION은 $Q \times \Sigma^k$에서 {터미널이동}∪{예상 $[B, W, \gamma, V] \mid [B, W, \gamma, V] \in Q$}∪{파스 $A \rightarrow \alpha$ 생성$\mid A \rightarrow \alpha \in P$}로의 함수이다: $[A, R, \alpha, U] \in Q$, $u \in U$라고 하자.

(1) $[A, R, \alpha, U] \rightarrow^a [A, R, \alpha a, V]$가 존재하는 경우에

ACTION([A, R, α, U], u) = 터미널 이동

(2) $[A, R, \alpha, U] \rightarrow [B, W, \gamma, V]$가 존재하는 경우에

ACTION([A, R, α, U], u) = 예상 [B, W, γ, V]

(3) $A \rightarrow \alpha \in P$가 존재하는 경우에

ACTION([A, R, α, U], u) = 파스 $A \rightarrow \alpha$ 생성 □

정의된 파싱 오토마톤을 LR 오토마톤과 비교 시에 기존의 넌터미널 GOTO 테이블이 불필요함을 알 수 있다.

3.2 파싱

파서의 *상황*(configuration)은 상태 나열과 남은 입력

스트링으로 구성된다: 가령, $I_i \in Q, i = 0, 1, \cdots, n$, $y \in \Sigma^*$에 대해서 $I_0 I_1 \cdots I_{n-1} I_n | y$이다.

정의 3.4 (파서의 행동에 따른 상태의 변이)

주어진 $M(G, \Pi')$에 대한 현재 파서 상황은 $I_0 I_1 \cdots I_{n-1} I_n | y$, 현재 상태 I_n은 $[A, R, \alpha, U]$라고 하자. 각 행동에 대한 파서의 이동은 다음과 같다.

(1) $ACTION([A, R, \alpha, U], k{:}y)$ = 터미널 이동
$V = RC_k^{A,R}(\alpha a)$, $y = az$이라 하자.
$I_0\ I_1 \cdots I_{n-1}[A, R, \alpha, U] | y \rightarrow I_0\ I_1 \cdots I_{n-1}[A, R, \alpha a, V] \mid z$

(2) $ACTION([A, R, \alpha, U], k{:}y)$ = 예상 $[B, W, \gamma, V]$
$I_0 I_1 \cdots I_{n-1}[A, R, \alpha, U] | y \rightarrow I_0 I_1 \cdots I_{n-1}[A, R, \beta B, W][B, W, \gamma, V] \mid y$

(3) $ACTION([A, R, \alpha, U], k{:}y)$ = 파스 $A \rightarrow \alpha$ 생성
$I_0 I_1 \cdots I_{n-1}[A, R, \alpha, U] | y \rightarrow I_0 I_1 \cdots I_{n-1} | y$

□

4. 단일 상태 오토마톤의 성질

4.1 G와 M(G, Π')와의 관계

본 절에서는 G가 생성하는 언어와 $M(G, \Pi')$가 인식하는 언어가 일치함을 증명한다.

함수 τ는 다음과 같이 정의된다: $t = I_0 I_1 \cdots I_{n-1}[A, R, \alpha, U] | y \rightarrow I_0 I_1 \cdots I_{n-1} | y$이면 $\tau(t) = A \rightarrow \alpha$이다. 그 밖의 경우는 $\tau(t) = \varepsilon$이다.

보조 정리 4.1 $M(G, \Pi')$상에 $[A, R, \alpha, U] | y \rightarrow^{\pi} \varepsilon | \varepsilon$이 존재하면 G상에 $A \Rightarrow^{\tau(\pi)^R}_{rm} \alpha y$이 존재한다.

증명 $|\pi|$에 관한 귀납법으로 증명한다. $|\pi|$가 1인 경우는 $A \rightarrow \alpha$가 생성 규칙인 경우이기에 명백하게 본 성질이 참이다. $|\pi|$가 n보다 작은 경우에 본 성질이 성립한다고 가정하고 $|\pi|$가 n인 경우를 생각한다. $\pi = r\pi'$이고 r의 타입에 따라서 나누어 생각한다.

(경우1) $r = [A, R, \alpha, U] | y \rightarrow [A, R, \alpha a, V] | z$ $(y=az)$

$M(G, \Pi')$상의 $[A, R, \alpha a, V] | z \rightarrow^{\pi'} \varepsilon | \varepsilon$에 대하여 귀납 가정을 적용 시에 G상에 $A \Rightarrow^{\tau(\pi')^R}_{rm} \alpha az$이 존재한다. $\tau(r) = \varepsilon$이기에 $\tau(\pi)^R = \tau(\pi')^R$이고 본 성질이 성립한다.

(경우2) $r = [A, R, \alpha, U] | y \rightarrow [A, R, \beta B, W][B, W, \gamma, V] | y$

π'이 $[A, R, \beta B, W] | y_1 \rightarrow^{\pi_1} \varepsilon | \varepsilon$, $[B, W, \gamma, V] | y_2 \rightarrow^{\pi_2} \varepsilon | \varepsilon$ $(\pi' = \pi_1 \pi_2,\ y = y_1 y_2)$으로 구성된다고 하면, 귀납 가정에 의해서 G상에 $B \Rightarrow^{\tau(\pi_2)^R}_{rm} \gamma y_2$, $A \Rightarrow^{\tau(\pi_1)^R}_{rm} \beta B y_1$이 존재한다. $\tau(\pi_1)^R \tau(\pi_2)^R = \tau(\pi_1 \pi_2)^R = \tau(\pi)^R$이기에 본 성질이 성립한다. □

보조 정리 4.2 $[A, R, \alpha, U]$의 상태가 $M(G, \Pi')$에 존재한다고 하자. G상에 $A \Rightarrow^{\pi}_{A,R} \alpha x$이 존재한다면 $M(G, \Pi')$상에 $[A, R, \alpha, U] | x \rightarrow^{\pi'} \varepsilon | \varepsilon$, $\tau(\pi')^R = \pi$이 존재한다.

증명 $|\pi|$에 관한 귀납법으로 증명한다. $|\pi|$가 1인 경우에 $p = A \rightarrow \alpha x \in P$이고 연속적인 터미널 이동으로 $[A, R, \alpha, U] | x \rightarrow^{\pi_1'} [A, R, \alpha x, U'] | \varepsilon$이 존재하며, 그 다음에는 $[A, R, \alpha x, U'] | \varepsilon \rightarrow^{r} \varepsilon | \varepsilon$이 존재한다. $\tau(\pi_1' r)^R = \tau(\pi_1')^R \tau(r)^R = \varepsilon p = p$이기에 본 성질이 성립한다. 다음으론 $|\pi|$가 $t(t>0)$보다 작은 경우에 본 성질이 성립한다고 가정하고, $|\pi|$가 t인 경우를 생각한다.

(경우1) $(A, R, \alpha y)\ \Pi'_u\ (B, W, \gamma)$, $yz = x$, $u = k{:}zr$, $r \in R$인 y, B, W, γ가 존재한다.

파서 정의에 의해서 연속적인 터미널 이동으로 $[A, R, \alpha, U] | yz \rightarrow^{\pi'} [A, R, \alpha y, U'] | z$가 존재하며, 이후에는 $[A, R, \alpha y, U'] | z \rightarrow^{r} [A, R, \beta B, W][B, W, \gamma, V] | z$, $u \in V$, $\alpha y = \beta\gamma$가 존재한다. 한편 $A \Rightarrow^{\pi}_{A,R} \alpha x$는 $A \Rightarrow^{\pi_1}_{A,R} \beta B z_2$, $B \Rightarrow^{\pi_2}_{B,W} \gamma z_1$, $z_1 z_2 = z$, $\pi = \pi_1 \pi_2$로 나눌 수가 있다. 이때 각 유도 과정에 대해서 귀납 가정을 적용 시에 $[A, R, \beta B, W] | z_2 \rightarrow^{\pi_1^R} \varepsilon | \varepsilon$, $[B, W, \gamma, V] | z_1 \rightarrow^{\pi_2^R} \varepsilon | \varepsilon$이 존재한다. $\tau(\pi' r \pi_2^R \pi_1^R) = \tau(\pi_1 \pi_2)^R = \tau(\pi)^R$이기에 본 성질이 성립한다.

(경우2) 그 외의 경우

(경우2-1) π가 $A \Rightarrow^{\pi_1}_{A,R} \beta B z \Rightarrow^{r}_{A,R} \beta\gamma y z$, $\beta\gamma = \alpha$, $yz = x$인 π_1과 r로 구성된다.

파서 정의에 의해서 $[A, R, \alpha, U] | yz \rightarrow^{\pi''} [A, R, \alpha y, M] | z$, $[A, R, \alpha y, M] | z \rightarrow r_1\ [A, R, \beta B, W][B, W, \gamma y, W] | z \rightarrow r_2\ [A, R, \beta B, W] | z$이 존재하고, π_1에 대해서 귀납 가정을 적용 시에 $[A, R, \beta B, W] | z \rightarrow^{\pi_1'} \varepsilon | \varepsilon$이 존재한다. $\tau(\pi'' r_1 r_2 \pi_1')^R = \varepsilon\varepsilon\tau(r_2)^R \tau(\pi_1')^R = \tau(r_2 \pi_1')^R = \tau(\pi)^R$이기에 본 성질이 성립한다.

(경우2-2) π가 $A \Rightarrow^{\pi_1}_{A,R} \alpha y_1 B z \Rightarrow^{r}_{A,R} \alpha y_1 y_2 z$인 π_1과 r로 구성된다.

파서 정의에 의해서 연속적인 터미널 이동인 $[A, R, \alpha, U] | y_1 y_2 z \rightarrow^{\pi''} [A, R, \alpha y_1 y_2, M] | z$가 존재하고, $[A, R, \alpha y_1 y_2, M] | z \rightarrow r_1\ [A, R, \alpha y_1 B, W][B, W, y_2, T] \rightarrow r_2\ [A, R, \alpha y_1 B, W] | z$가 존재한다. 한편 π_1에 대해서 귀납 가정을 적용 시에 $[A, R, \alpha y_1 B, W] | z \rightarrow^{\pi_1'} \varepsilon | \varepsilon$, $\tau(\pi_1)^R = \pi_1'$이 존재한다. $\tau(\pi'' r_1 r_2 \pi_1')^R = \varepsilon\varepsilon\tau(r_2)^R \tau(\pi_1')^R = \tau(r_2 \pi_1')^R = \tau(\pi)^R$이기에 본 성질이 성립한다. □

보조 정리 4.1과 4.2로부터 정리 4.1을 얻을 수가 있다.

정리 4.1 $M(G, \Pi')$상에 $[S, \{\$^k\}, \varepsilon, FIRST_k(S\$^k)] | y \rightarrow^{\pi} [S, \{\$^k\}, S, \{\$^k\}] | \varepsilon$이 있는 경우에 G상에 $S \Rightarrow^{\tau(\pi)^R}_{rm} y$이 존재한다. 이 역도 성립한다. □

4.2 LR 파서와의 비교

제안된 파서의 파싱 시간을 LR 파서와 비교 시에 두 방법 모두 입력 스트링 길이에 비례하나, 제안된 파서는 넌터미널 GOTO 테이블의 참조 작업이 없다. 넌터미널

GOTO 테이블의 참조는 파스 수에 비례하며 해당 시간만큼 절약됨을 알 수 있다. 한편 파서 상태 수를 비교 시에 LR 파서의 상태 수는 $O\left(2^{|G|\times(|\Sigma|+1)^k}\right)$이며 제안된 파서의 상태 수는 $O\left(|N|\times 2^{(|\Sigma|+1)^k}\times|V|^C\right)$ (여기서 c는 바운드된 상수이다)이기에 제안된 파서의 상태 수가 현저하게 적음을 알 수 있다. 또한 LR 파싱 시에 필요한 넌터미널 GOTO 테이블의 크기는 LR 파서의 상태 수에 비례하며, 제안된 파서는 이 테이블이 필요하지 않기에 해당 공간만큼 절약됨을 알 수 있다.

5. 맺는 말

본 논문에서는 단일 상태 파싱 오토마톤을 정의하고, 이에 기반한 파싱 방법을 제안하였다. 제안된 단일 상태 파싱 오토마톤의 생성 가능 문법 클래스는 LR 문법 클래스보다 작지만, 이에 기반한 파서는 LR 파서와 비교 시에 공간과 시간적인 면에서 효율적임을 보였다.

참 고 문 헌

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

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

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

[4] G.-O. Lee and K.-M. Choe, A powerful LL(k) covering transformation, SIAM J. Computing, 35(2), pp. 359-377, 2006.

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. 관심 분야는 프로그래밍 언어와 컴파일러 등