13. SLR 파싱테이블

충북대학교

이 재 성

학습내용

- 아이템 집합군(Canonical LR(0)) 구성
- First와 Follow함수
- SLR 파싱테이블 작성 방법

SLR파싱 테이블의 구성

LR(0) 아이템

- 문법 G의 생성규칙 오른쪽 임의의 위치에 점이 찍혀 있는 규칙

  $A \rightarrow \bullet XYZ$

  $A \rightarrow X \bullet YZ$

  $A \rightarrow XY \bullet Z$

  $A \rightarrow XYZ \bullet$

SLR 방법의 중심 개념

- 바이어블 프리픽스를 인식하는 결정 유한 오토마톤 구성
- 아이템들은 SLR파서의 상태에 해당하는 집합
  - NFA의 상태에 해당
  - "부분집합구성 방법" 으로 아이템들을 함께 묶음

Closure 연산

Closure(I), 아이템 집합의 구성

- I에 있는 모든 아이템을 closure(I)에 넣는다.
- A → a • Bβ가 closure(I)에 속해 있고, B → γ 가 생성규칙이라면 아이템 B → • γ 를 I에 추가 (단, B → • γ 가 closure(I)에 없을 경우)

Goto 연산

- goto(I, X): I는 아이템 집합, X는 문법기호
- goto(I, X)는 A → α • Xβ 가 I에 있을 때 모든 아이템 A → α X • β 의 closure로 정의
- I가 바이어블 프리픽스 γ에 유효하다면, goto(I, X)는 바이어블 프리픽스 γX 에 대한 유효한 아이템 집합

I:		goto(I,X):
γ •Xβ	⇨	γ X •β
γ •Y		

아이템 집합군 구성

- canonical LR(0)의 아이템 집합군 구성 알고리즘

```
procedure items(G');
begin
        C:={closure({[S'->S]})};
        repeat
                for C의 각 아이템 집합 I와 문법기호 X에 대해
                    (단, goto(I, X)는 빈 것이 아니고 C에도 포함되지 않음)
                do
                        goto(I, X)를 C에 추가
        until C 에 추가할 아이템 집합이 더 이상 없다.
end
```

Canonical LR(0) 아이템 구성 예(1)

문법확장후, 아이템 집합군 구성

$E' \rightarrow E$
$E \rightarrow E + T$
$E \rightarrow T$
$T \rightarrow T * F$
$T \rightarrow F$
$F \rightarrow (E)$
$F \rightarrow id$

I_0: $E' \rightarrow \bullet E$
$E \rightarrow \bullet E + T$
$E \rightarrow \bullet T$
$T \rightarrow \bullet T * F$
$T \rightarrow \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

I_1: $E' \rightarrow E \bullet$
$E \rightarrow E \bullet + T$

I_2: $E \rightarrow T \bullet$
$T \rightarrow T \bullet * F$

I_3: $T \rightarrow F \bullet$

I_4: $F \rightarrow (\bullet E)$
$E \rightarrow \bullet E + T$
$E \rightarrow \bullet T$
$T \rightarrow \bullet T * F$
$T \rightarrow \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

I_5: $F \rightarrow id \bullet$

I_6: $E \rightarrow E + \bullet T$
$T \rightarrow \bullet T * F$
$T \rightarrow \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

I_7: $T \rightarrow T * \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

I_8: $F \rightarrow (E \bullet)$
$E \rightarrow E \bullet + T$

Canonical LR(0) 아이템 구성 예(2)

I_0: $E' \to \bullet E$
$E \to \bullet E + T$
$E \to \bullet T$
$T \to \bullet T * F$
$T \to \bullet F$
$F \to \bullet (E)$
$F \to \bullet id$

I_1: $E' \to E \bullet$
$E \to E \bullet + T$

I_2: $E \to T \bullet$
$T \to T \bullet * F$

I_3: $T \to F \bullet$

I_4: $F \to (\bullet E)$
$E \to \bullet E + T$
$E \to \bullet T$
$T \to \bullet T * F$
$T \to \bullet F$
$F \to \bullet (E)$
$F \to \bullet id$

I_5: $F \to id \bullet$

I_6: $E \to E + \bullet T$
$T \to \bullet T * F$
$T \to \bullet F$
$F \to \bullet (E)$
$F \to \bullet id$

I_7: $T \to T * \bullet F$
$F \to \bullet (E)$
$F \to \bullet id$

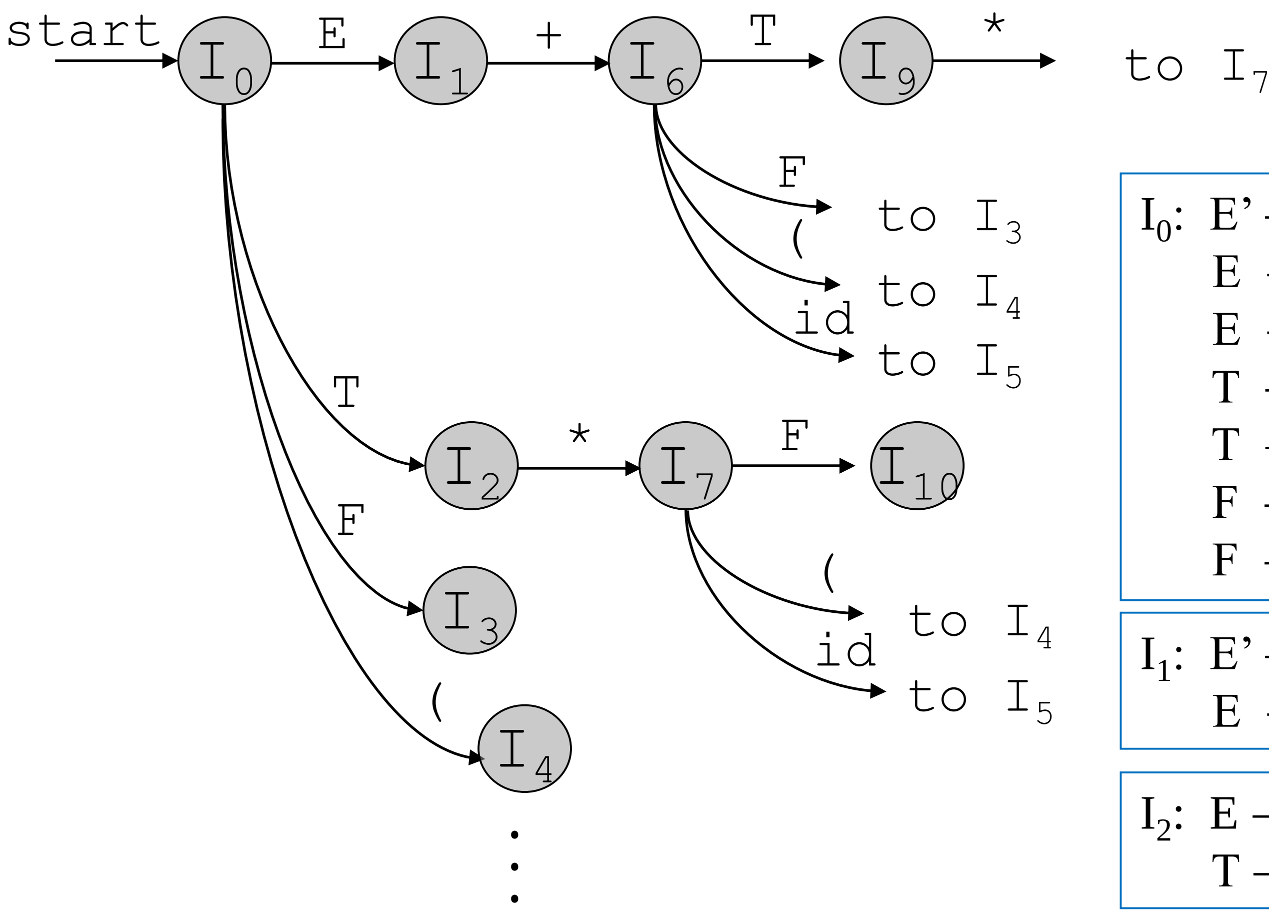
I_8: $F \to (E \bullet)$
$E \to E \bullet + T$

I_9: $E \to E + T \bullet$
$T \to T \bullet * F$

I_{10}: $T \to T * F \bullet$

I_{11}: $F \to (E) \bullet$

바이어블 프리픽스에 대한 DFA D의 전이도 예



I_0: $E' \rightarrow \bullet E$
$E \rightarrow \bullet E + T$
$E \rightarrow \bullet T$
$T \rightarrow \bullet T * F$
$T \rightarrow \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

I_1: $E' \rightarrow E \bullet$
$E \rightarrow E \bullet + T$

I_2: $E \rightarrow T \bullet$
$T \rightarrow T \bullet * F$

I_3: $T \rightarrow F \bullet$

I_4: $F \rightarrow (\bullet E)$
:

I_6: $E \rightarrow E + \bullet T$
$T \rightarrow \bullet T * F$
$T \rightarrow \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

I_9: $E \rightarrow E + T \bullet$
$T \rightarrow T \bullet * F$

I_7: $T \rightarrow T * \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

First

FIRST(X)

- 문법기호 X의 맨 처음에 나올 수 있는 단말기호
- 규칙
  1. X가 단말이면 FIRST(X) = {X}
  2. 만약 $X \rightarrow \varepsilon$ 가 있으면, FIRST(X)에 ε 를 추가
  3. X가 비단말이고, $X \rightarrow Y_1Y_2 \ldots Y_k$라는 생성식이 있을 경우
     - FIRST(Y_1)의 원소 중 ε 가 아닌 모든 기호를 추가
     - FIRST(Y_1)에 ε 원소가 있었을 경우,
       - FIRST(Y_2)의 ε 가 아닌 모든 원소 추가
     - FIRST(Y_1)과 FIRST(Y_2)에 모두 ε 가 있을 경우,
       - FIRST(Y_3)의 ε 가 아닌 모든 원소추가
     - 위의 과정 반복
     - 모든 i에 대해 FIRST(Y_i)가 ε 를 가지고 있다면 FIRST(X)에 ε 추가

Follow

FOLLOW(A)

1. FOLLOW(S)에 $를 넣음 (S: 시작기호, $: 입력의 끝)
2. A → αBβ라는 생성식이 있으면,
   - ε 를 제외한 FIRST(β)의 모든 원소들은 FOLLOW(B)에 속함

   RHS 에 중간에 나타난 ...
3. A → αBβ라는 생성식이 있고, A → αB 라는 생성식이 있거나 FIRST(β)가 ε 를 가질 경우,
   - FOLLOW(A)의 모든 원소는 FOLLOW(B)에 속함

   RHS 에 맨 끝에 있을 때 ...

FIRST 와 FOLLOW 예1

FIRST(E) = FIRST(T) =FIRST(F) = { (, id}

FOLLOW(E) = { +,), $}

FOLLOW(T) = FOLLOW(E) U {*} = {+,), $, *}

FOLLOW(F) = FOLLOW(T)

1. $
2. +,)
3. X

E → E + T
E → T
T → T * F
T → F
F → (E)
F → id

1. ∅
2. *
3. FOLLOW(E)

1. ∅
2. ∅
3. FOLLOW(T)

FIRST 와 FOLLOW 예2

FIRST(E) = FIRST(T) =FIRST(F) = { (, id}
FIRST(E’) ={+, ε}
FIRST(T’) ={*, ε}

1. \$ 2.) 3. ∅

FOLLOW(E) = {), \$}
FOLLOW(E) = FOLLOW(E’)

1. ∅ 2. ∅
3. FOLLOW(E)

FOLLOW(T) =FIRST’(E’) U FOLLOW(E) = {+,), \$} (ε 제외)
FOLLOW(F) = FIRST’(T’) U FOLLOW(T) ={+, *,), \$} (ε 제외)

1. ∅
2. +,
3.), \$

E → TE’
E’ → +TE’ | ε
T → FT’
T’ → *FT’ | ε
F → (E) | id

FOLLOW(T’) = {), \$}

1. ∅
2. FIRST(T’) = {*, ε}
3. FOLLOW(T’)

SLR파싱 테이블의 구성 알고리즘

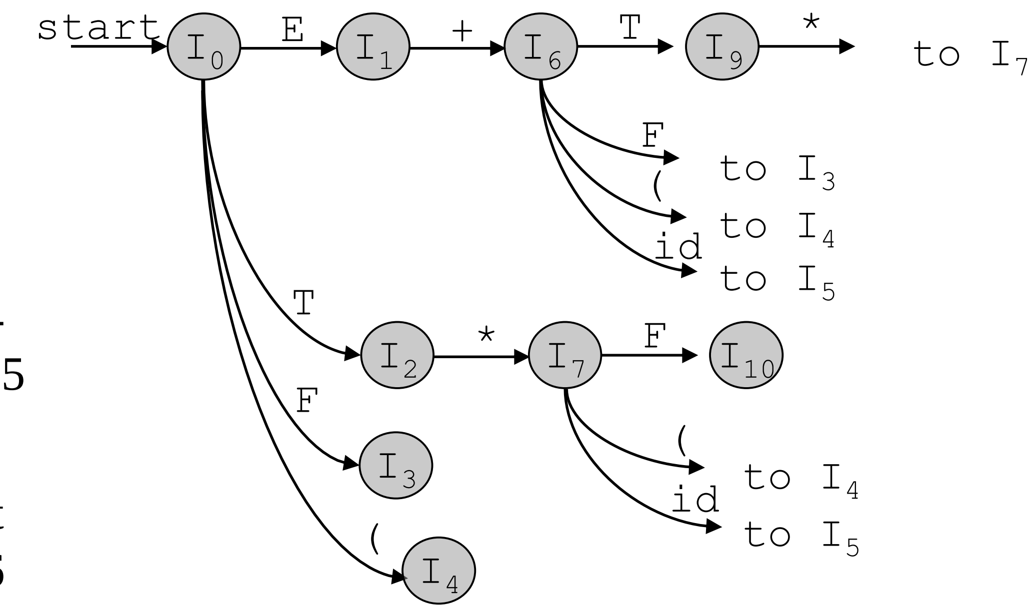
- 입력: 확장문법 G’
- 출력: G’에 대한 SLR 파싱 테이블 함수 action과 goto
- 방법
  1. G’에 대한 LR(0) 아이템 집합군 $C=\{I_0, I_1, ..., I_n)$구성
  2. I_i에서 상태i 를 구성
     - action[i, a]=shift j
       - $[A \rightarrow \alpha \bullet a\beta]$가 I_i안에 있고 $goto(I_i, a)=I_j$ 일 경우
     - action[i, a]= reduce $A \rightarrow \alpha$ (FOLLOW(A)에 있는 모든 a)
       - $[A \rightarrow \alpha \bullet]$이 I_i안에 있을 경우
     - action[i, $] = accept
       - $[S' \rightarrow S \bullet]$이 I_i에 있을 경우
  3. 상태 i에서 goto전이
     - $goto(I_i, A) = I_j$라면 goto[i, A]=j
  4. 기타 항목은 에러 (2,3에서 정의되지 않은 것)
  5. 초기 상태는 $[S' \rightarrow S]$를 담고 있는 아이템 집합

SLR파싱 테이블 작성예

I_0: $E' \rightarrow \bullet E$
$E \rightarrow \bullet E + T$
$E \rightarrow \bullet T$
$T \rightarrow \bullet T * F$
$T \rightarrow \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

goto(0, E)=1
goto(0, T)=2
goto(0, F)=3

action[0,(]=shift 4
action[0,id]=shift 5



I_1: $E' \rightarrow E \bullet$
$E \rightarrow E \bullet + T$

action[1,$]=accept
action[1,+]=shift 6

I_2: $E \rightarrow T \bullet$
$T \rightarrow T \bullet * F$

FOLLOW(E) = {$, +,)} 이므로
action[2,$]=action[2,+]=action[2,)]=reduce $E \rightarrow T$
action[2,*]=shift 7

I_3: $T \rightarrow F \bullet$

FOLLOW(T) = {$, +,), *} 이므로
action[3,$]=action[3,+]=action[3,)]=action[3,*]=reduce $T \rightarrow F$

SLR 파싱테이블 예

	action						goto		
상태	id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

문법

1. $E \rightarrow E + T$
2. $E \rightarrow T$
3. $T \rightarrow T * F$
4. $T \rightarrow F$
5. $F \rightarrow (E)$
6. $F \rightarrow id$

참고 문헌

- [1] Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, “Compilers – Principles, Techniques, and Tools,” Bell Telephone Laboratories, Incorporated, 1986.
- [2] 오세만, “컴파일러 입문”, 정익사, 2004.