

14. LALR 파싱테이블

충북대학교

이재성





학습내용

- SLR 파싱의 한계
- CLR파싱 테이블 구축 방법
- LALR 파싱 테이블 구축 방법



SLR, CLR, LALR 용어

■ 명칭

- SLR: Simple LR
- CLR: Canonical LR
- LALR: Look Ahead LR

■ 파싱시 look ahead symbol을 하나만 보는 SLR(1), CLR(1), LALR(1)은 일반적으로 1을 생략하여 SLR, CLR, LALR로 부름

■ 아이템 집합 구성시의 LR(0), LR(1)에서 숫자는 아이템에 포함시켜 나타낸 look ahead(예측기호)의 갯수를 나타냄



SLR 문법의 모호성 예(1)

LR(0) 아이템 집합군 구성

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow * R$

$L \rightarrow id$

$R \rightarrow L$

$\text{First}(S) = \text{First}(L)$
 $= \text{First}(R) = \{*, id\}$

$\text{Follow}(S) = \{\$\}$

$\text{Follow}(R) = \text{Follow}(L)$
 $= \{\$, =\}$

$I_0: S' \rightarrow \bullet S$
 $S \rightarrow \bullet L = R$
 $S \rightarrow \bullet R$
 $L \rightarrow \bullet * R$
 $L \rightarrow \bullet id$
 $R \rightarrow \bullet L$

$I_1: S' \rightarrow S \bullet$
 $I_2: S \rightarrow L \bullet = R$
 $R \rightarrow L \bullet$
 $I_3: S \rightarrow R \bullet$

$I_4: L \rightarrow * \bullet R$
 $R \rightarrow \bullet L$
 $L \rightarrow \bullet * R$
 $L \rightarrow \bullet id$
 $I_5: L \rightarrow id \bullet$

$I_6: S \rightarrow L = \bullet R$
 $R \rightarrow \bullet L$
 $L \rightarrow \bullet * R$
 $L \rightarrow \bullet id$

$I_7: L \rightarrow * R \bullet$
 $I_8: R \rightarrow L \bullet$
 $I_9: S \rightarrow L = R \bullet$



SLR 문법의 모호성 예(2)

$$\begin{aligned}I_0: S' &\rightarrow \bullet S \\S &\rightarrow \bullet L = R \\S &\rightarrow \bullet R \\L &\rightarrow \bullet * R \\L &\rightarrow \bullet id \\R &\rightarrow \bullet L\end{aligned}$$

$$I_1: S' \rightarrow S \bullet$$

$$\begin{aligned}I_2: S &\rightarrow L \bullet = R \\R &\rightarrow L \bullet\end{aligned}$$

$$I_3: S \rightarrow R \bullet$$

$$\begin{aligned}I_4: L &\rightarrow * \bullet R \\R &\rightarrow \bullet L \\L &\rightarrow \bullet * R \\L &\rightarrow \bullet id\end{aligned}$$

$$I_5: L \rightarrow id \bullet$$

$$\begin{aligned}I_6: S &\rightarrow L = \bullet R \\R &\rightarrow \bullet L \\L &\rightarrow \bullet * R \\L &\rightarrow \bullet id\end{aligned}$$

$$I_7: L \rightarrow * R \bullet$$

$$I_8: R \rightarrow L \bullet$$

$$I_9: S \rightarrow L = R \bullet$$

$$\begin{aligned}\text{Follow}(S) &= \{\$\} \\ \text{Follow}(R) &= \text{Follow}(L) \\ &= \{\$, =\}\end{aligned}$$

충돌 발생:

action[2,=] shift 6

action[2,=] reduce

(FOLLOW(R)의 '=' 포함)

R=...로 시작되는 우문장형식이
없으므로 reduce는 부적절

문법은 모호성이 없으나 충분한
문맥을 기억하지 못하기 때문



canonical LR 파싱 테이블 구성

■ 예측기호 포함

- 적합하지 않은 reduce를 위해 더 많은 정보 포함
- 두번째 요소로서 단말기호 포함
- $[A \rightarrow \alpha \bullet, a]$
- 입력 기호 a가 올 경우만 reduce 수행
- a는 FOLLOW(A)의 부분 집합



Closure 연산

■ Closure(I), LR(1) 아이템 집합의 구성

- I에 있는 모든 아이템을 closure(I)에 넣는다.
- $[A \rightarrow \alpha \bullet B \beta, a]$ 가 closure(I)에 속해 있고, $B \rightarrow \gamma$ 및 FIRST(βa)에 포함되는 각 단말기호 b 에 대해 아이템 $[B \rightarrow \bullet \gamma, b]$ 를 I에 추가 (단, $[B \rightarrow \bullet \gamma, b]$ 가 closure(I)에 없을 경우)

■ Goto 연산

- goto(I, X): I는 아이템 집합, X는 문법기호
- goto(I, X)는 $[A \rightarrow \alpha \bullet X \beta, a]$ 에 대해 $[A \rightarrow \alpha X \bullet \beta, a]$ 의 집합 J



아이템 집합군 구성

■ canonical LR(1)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 C에 추가할 아이템 집합이 더 이상 없다.
end
```



아이템 집합구성

$$I_0: \begin{aligned} S' &\rightarrow \bullet S, \$ \\ S &\rightarrow \bullet CC, \$ \\ C &\rightarrow \bullet cC, c/d \\ C &\rightarrow \bullet d, c/d \end{aligned}$$

$$I_1: S' \rightarrow S \bullet, \$$$

$$I_2: \begin{aligned} S &\rightarrow C \bullet C, \$ \\ C &\rightarrow \bullet cC, \$ \\ C &\rightarrow \bullet d, \$ \end{aligned}$$

$$I_3: \begin{aligned} C &\rightarrow c \bullet C, c/d \\ C &\rightarrow \bullet cC, c/d \\ C &\rightarrow \bullet d, c/d \end{aligned}$$

초기

추가규칙 $S \rightarrow \bullet CC$
 $\text{FIRST}(\beta a) = \text{FIRST}(\epsilon \$) = \{\$\}$

추가규칙 $C \rightarrow \bullet cC, C \rightarrow \bullet d$
 $\text{FIRST}(\beta a) = \text{FIRST}(C \$) = \{c, d\}$

추가규칙 $C \rightarrow \bullet cC, C \rightarrow \bullet d$
 $\text{FIRST}(\beta a) = \text{FIRST}(\epsilon \$) = \{\$\}$

기본규칙

$[A \rightarrow \alpha \bullet B\beta, a]$ 가 closure(I)에 속해 있고, $B \rightarrow \gamma$ 및 $\text{FIRST}(\beta a)$ 에 포함되는 각 단말기호 b 에 대해 아이템 $[B \rightarrow \bullet \gamma, b]$ 를 I에 추가

$$\begin{aligned} S' &\rightarrow S \\ S &\rightarrow CC \\ C &\rightarrow cC \mid d \end{aligned}$$



아이템 집합구성

$I_0:$ $S' \rightarrow \bullet S, \$$
 $S \rightarrow \bullet CC, \$$
 $C \rightarrow \bullet cC, c/d$
 $C \rightarrow \bullet d, c/d$

$I_1:$ $S' \rightarrow S \bullet, \$$

$I_2:$ $S \rightarrow C \bullet C, \$$
 $C \rightarrow \bullet cC, \$$
 $C \rightarrow \bullet d, \$$

$I_3:$ $C \rightarrow c \bullet C, c/d$
 $C \rightarrow \bullet cC, c/d$
 $C \rightarrow \bullet d, c/d$

$I_4:$ $C \rightarrow d \bullet, c/d$
 $I_5:$ $S \rightarrow CC \bullet, \$$

$I_6:$ $C \rightarrow c \bullet C, \$$
 $C \rightarrow \bullet cC, \$$
 $C \rightarrow \bullet d, \$$

$I_7:$ $C \rightarrow d \bullet, \$$

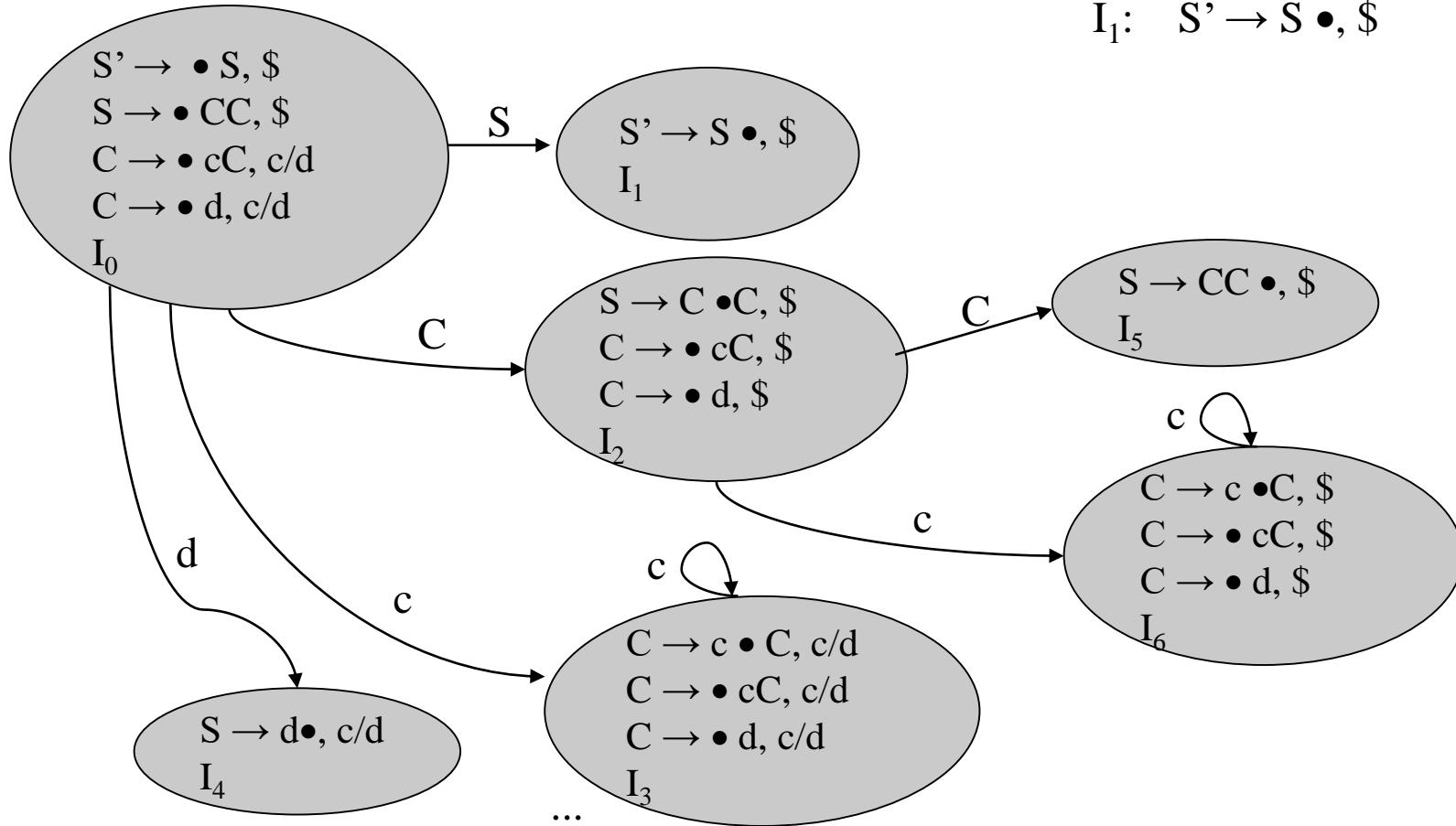
$I_8:$ $C \rightarrow cC \bullet, c/d$

$I_9:$ $C \rightarrow cC \bullet, \$$

$S' \rightarrow S$
 $S \rightarrow CC$
 $C \rightarrow cC | d$



goto 그래프



$I_0:$ $S' \rightarrow \bullet S, \$$
 $S \rightarrow \bullet CC, \$$
 $C \rightarrow \bullet cC, c/d$
 $C \rightarrow \bullet d, c/d$

$I_2:$ $S \rightarrow C \bullet C, \$$
 $C \rightarrow \bullet cC, \$$
 $C \rightarrow \bullet d, \$$

$I_1:$ $S' \rightarrow S \bullet, \$$

$I_3:$ $C \rightarrow c \bullet C, c/d$
 $C \rightarrow \bullet cC, c/d$
 $C \rightarrow \bullet d, c/d$

$I_4:$ $C \rightarrow d \bullet, c/d$

$I_5:$ $S \rightarrow CC \bullet, \$$

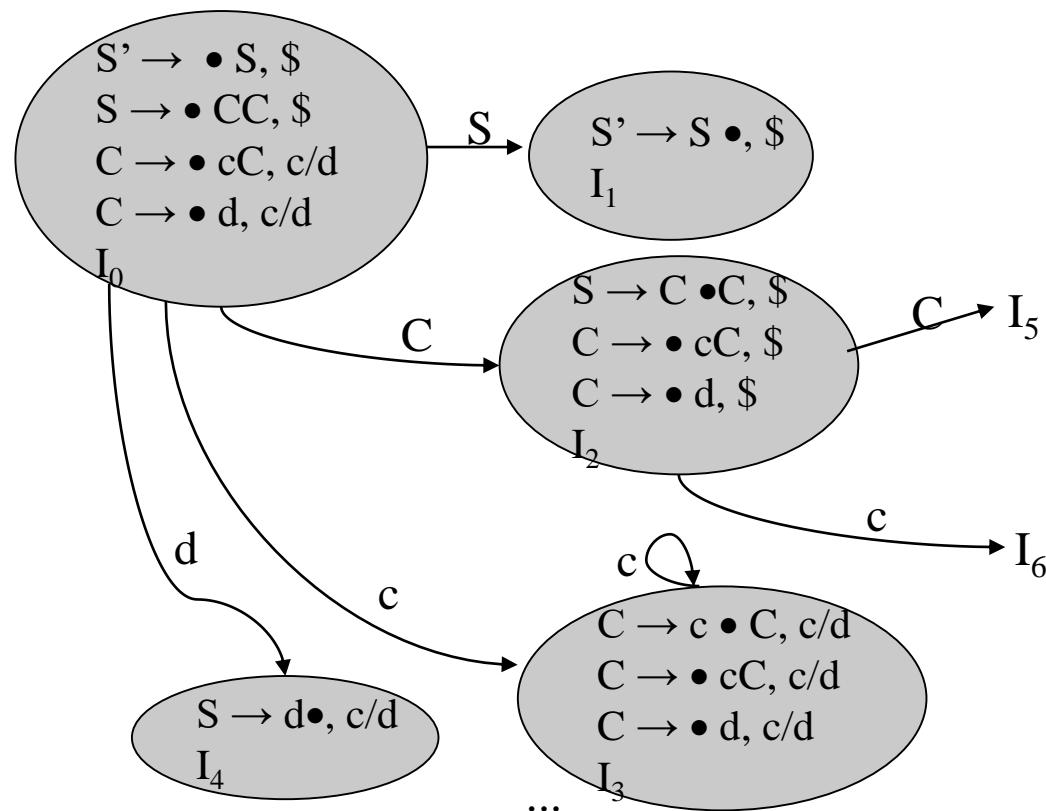
$I_6:$ $C \rightarrow c \bullet C, \$$
 $C \rightarrow \bullet cC, \$$
 $C \rightarrow \bullet d, \$$



canonical 파싱 테이블

1. $S \rightarrow CC$
2. $C \rightarrow cC$
3. $C \rightarrow d$

상태	action			goto	
	c	d	\$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		





LALR 파싱

■ 기본 원리

- 핵: 문법의 첫번째 요소(예측 기호를 제외한 부분)
- 같은 핵을 가진 문법을 합쳐 상태수를 줄임
- CLR보다 오류를 늦게 감지

$$\begin{aligned} S' &\rightarrow S \\ S &\rightarrow CC \\ C &\rightarrow cC \mid d \end{aligned}$$

$I_4: C \rightarrow d\bullet, c/d$

$I_7: C \rightarrow d\bullet, \$$

$I_{47}: C \rightarrow d\bullet, c/d/\$$



LALR 파싱 테이블 구성

- 입력: 확장 문법 G'
- 출력: G' 에 대한 LALR 파싱 테이블 함수
- 방법
 - LR(1) 아이템 집합군 $C=\{I_0, I_1, \dots, I_n\}$ 을 구성
 - 아이템 집합에서 같은 핵을 가진 아이템을 합친다.
 - 새로 합친 아이템 집합군에 대해 canonical LR 파싱 테이블 구성 알고리즘을 적용하여 파싱 테이블을 만든다. 오류면 LALR(1) 문법이 아님



아이템 집합구성

CLR용 LR(1) 아이템 집합

$$\begin{aligned} I_0: \quad & S' \rightarrow \bullet S, \$ \\ & S \rightarrow \bullet CC, \$ \\ & C \rightarrow \bullet cC, c/d \\ & C \rightarrow \bullet d, c/d \end{aligned}$$

$$I_1: \quad S' \rightarrow S \bullet, \$$$

$$\begin{aligned} I_2: \quad & S \rightarrow C \bullet C, \$ \\ & C \rightarrow \bullet cC, \$ \\ & C \rightarrow \bullet d, \$ \end{aligned}$$

$$\begin{aligned} I_3: \quad & C \rightarrow c \bullet C, c/d \\ & C \rightarrow \bullet cC, c/d \\ & C \rightarrow \bullet d, c/d \end{aligned}$$

$$I_4: \quad C \rightarrow d \bullet, c/d$$

$$I_5: \quad S \rightarrow CC \bullet, \$$$

$$\begin{aligned} I_6: \quad & C \rightarrow c \bullet C, \$ \\ & C \rightarrow \bullet cC, \$ \\ & C \rightarrow \bullet d, \$ \end{aligned}$$

$$I_7: \quad C \rightarrow d \bullet, \$$$

$$I_8: \quad C \rightarrow cC \bullet, c/d$$

$$I_9: \quad C \rightarrow cC \bullet, \$$$

$$S' \rightarrow S$$

$$S \rightarrow CC$$

$$C \rightarrow cC \mid d$$

LALR용 LR(1) 아이템 집합

$$\begin{aligned} I_0: \quad & S' \rightarrow \bullet S, \$ \\ & S \rightarrow \bullet CC, \$ \\ & C \rightarrow \bullet cC, c/d \\ & C \rightarrow \bullet d, c/d \end{aligned}$$

$$I_1: \quad S' \rightarrow S \bullet, \$$$

$$\begin{aligned} I_2: \quad & S \rightarrow C \bullet C, \$ \\ & C \rightarrow \bullet cC, \$ \\ & C \rightarrow \bullet d, \$ \end{aligned}$$

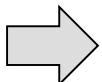
$$I_36: \quad C \rightarrow c \bullet C, c/d/\$$$

$$\begin{aligned} C \rightarrow \bullet cC, c/d/\$ \\ C \rightarrow \bullet d, c/d/\$ \end{aligned}$$

$$I_{47}: \quad C \rightarrow d \bullet, c/d/\$$$

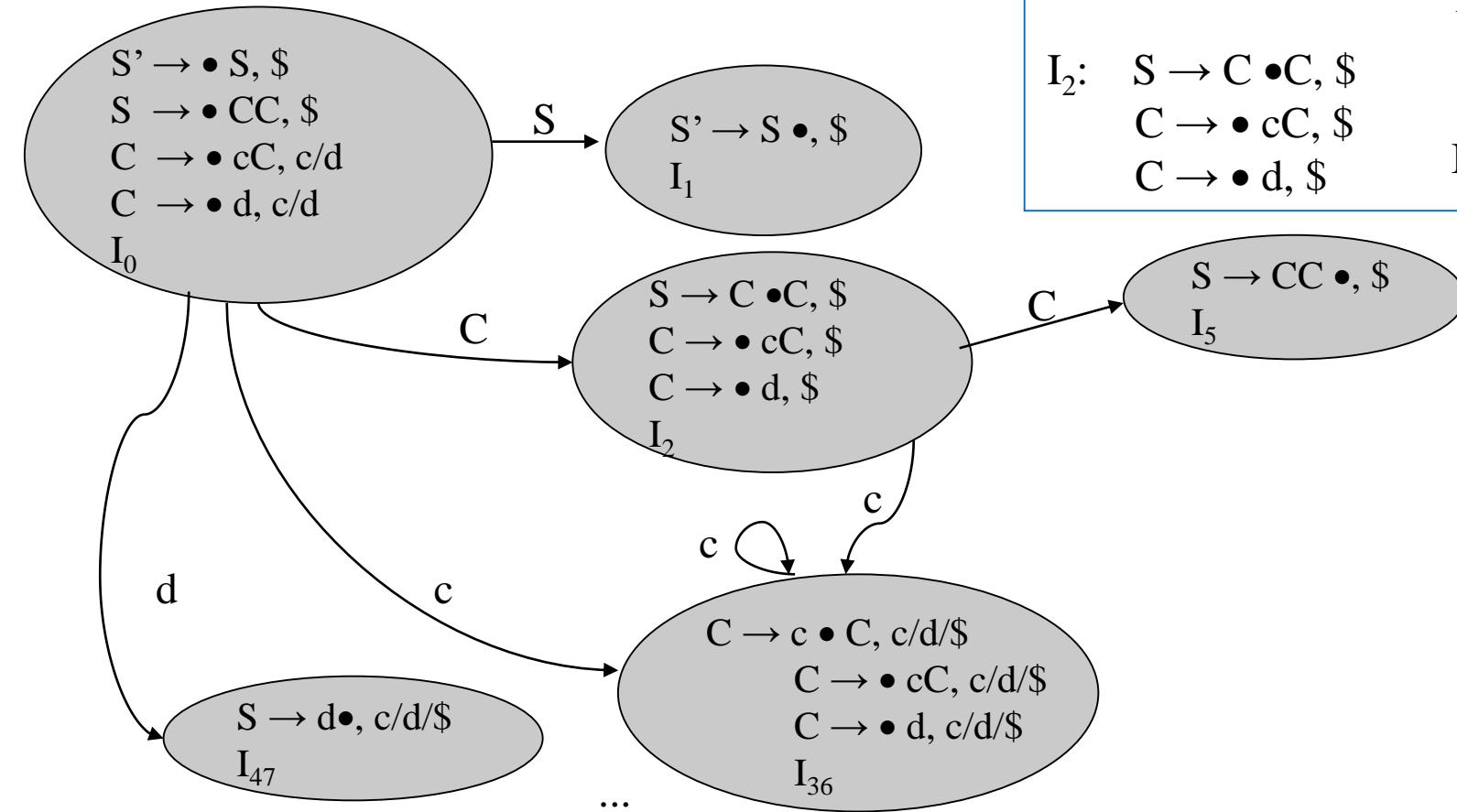
$$\begin{aligned} I_5: \quad & S \rightarrow CC \bullet, \$ \\ & C \rightarrow \bullet d, \$ \end{aligned}$$

$$I_{89}: \quad C \rightarrow cC \bullet, c/d/\$$$





goto 그래프



14. LALR 파싱 테이블

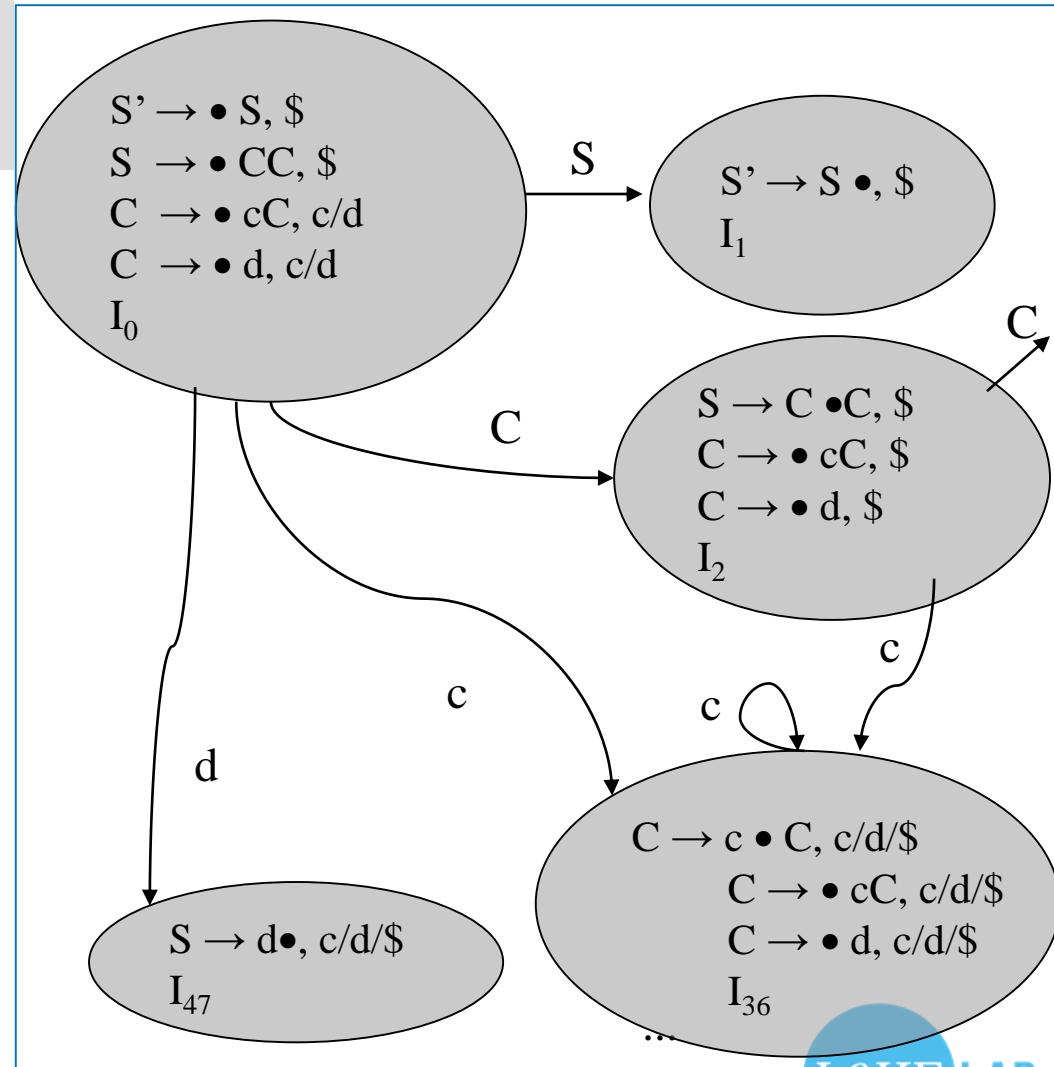
I ₀ :	$S' \rightarrow \bullet S, \$$	I ₃₆ :	$C \rightarrow c \bullet C, c/d/\$$
	$S \rightarrow \bullet CC, \$$		$C \rightarrow \bullet cC, c/d/\$$
	$C \rightarrow \bullet cC, c/d$		$C \rightarrow \bullet d, c/d/\$$
	$C \rightarrow \bullet d, c/d$		
		I ₄₇ :	$C \rightarrow d \bullet, c/d/\$$
I ₁ :	$S' \rightarrow S \bullet, \$$	I ₅ :	$S \rightarrow CC \bullet, \$$
I ₂ :	$S \rightarrow C \bullet C, \$$	I ₈₉ :	$C \rightarrow cC \bullet, c/d/\$$
	$C \rightarrow \bullet cC, \$$		
	$C \rightarrow \bullet d, \$$		



LALR 파싱 테이블

1. $S \rightarrow CC$
2. $C \rightarrow cC$
3. $C \rightarrow d$

상태	action			goto	
	c	d	\$	S	C
0	s36	s47		1	2
1			acc		
2	s36	s47			5
36	s36	s47			89
47	r3	r3	r3		
5			r1		
89	r2	r2	r2		





오류감지

늦은 오류 감지

- 이 문법은 c^*dc^*d 생성
- 첫번째 c들을 이동하고 다음의 d를 스택에 넣고 상태 4로 감
- CLR의 경우, 다음 입력 기호가 c나 d일 경우만 reduce C $\rightarrow d$ 를 수행
- 합친 상태에서 다음 입력 기호 \$에 대해 reduce를 수행하면 ccd와 같은 입력을 받아들임으로 오류가 됨
- => LALR파서는 일단 수행되다가 나중에 오류로 처리됨

$S' \rightarrow S$
 $S \rightarrow CC$
 $C \rightarrow cC \mid d$

$I_4: C \rightarrow d\bullet, c/d$

$I_7: C \rightarrow d\bullet, \$$

$I_{47}: C \rightarrow d\bullet, c/d/\$$



오류 감지 예 (ccd\$ 입력)

1. S → CC
2. C → cC
3. C → d

CLR

상태	action			goto	
	c	d	\$	S	C
0	s3	s4		1	2
1		acc			
2	s6	s7		5	
3	s3	s4		8	
4	r3	r3			
5		r1			
6	s6	s7		9	
7		r3			
8	r2	r2			
9		r2			

스택: 0c3c3d4

상태4에서 입력\$는 오류

14. LALR 파싱 테이블

LALR

상태	action			goto	
	c	d	\$	S	C
0	s36	s47		1	2
1		acc			
2	s36	s47		5	
36	s36	s47		89	
47	r3	r3	r3		
5		r1			
89	r2	r2	r2		

스택: 0c36c36d47

(47,\$)감소 수행(r3)

스택: 0c36c36C89

(89,\$)감소 수행(r2)

스택: 0c36C89

(89,\$)감소 수행(r2)

스택: 0C2

(2,\$)에서 오류



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”, 정의사, 2004.