



编译原理与技术

计算机科学与技术学院
李诚



3.16 用文法

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

(a) 构造(a, (a, a))的最右推导,

(b) 给出对应(a)的最左推导的移进归约分析器的步骤。



Assignment 6 — — 3.16b

栈	输入	动作
\$	(a,(a,a))\$	移进
\$(a,(a,a))\$	移进
\$(a	,(a,a))\$	归约
\$(s	,(a,a))\$	归约
\$(L	,(a,a))\$	移进
\$(L,	(a,a))\$	移进
\$(L,(a,a))\$	移进
...
...
\$(L)	\$	归约
\$\$	\$	



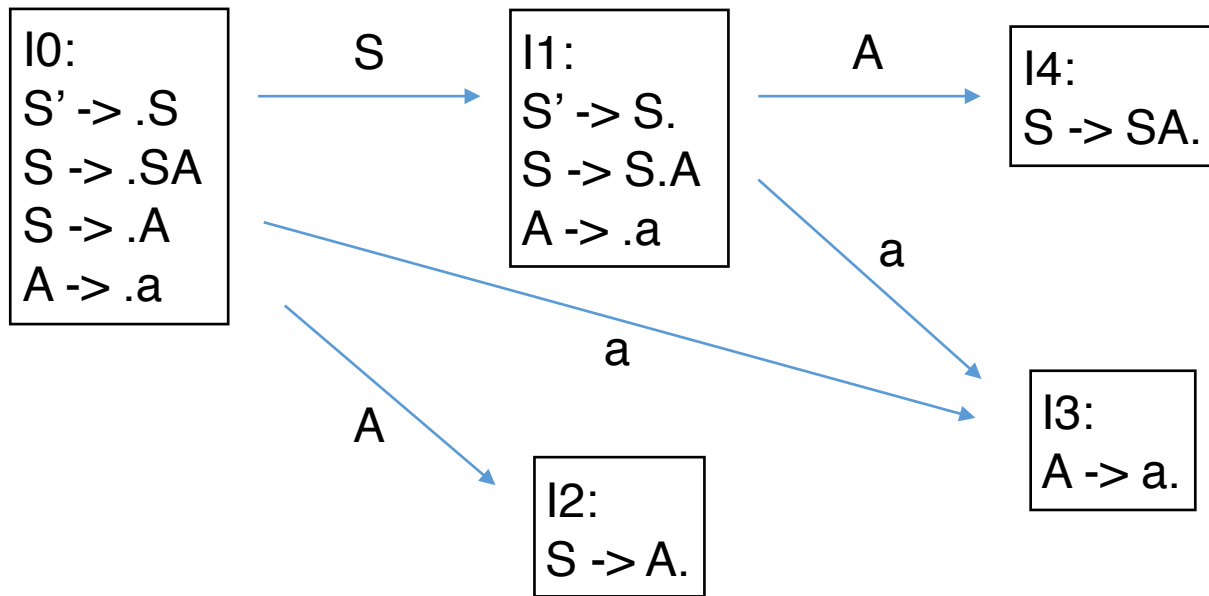
3.20 证明下面的文法

$$S \rightarrow S A I A$$
$$A \rightarrow a$$

是SLR(1)文法，但不是LL(1)文法。



活前綴DFA:





SLR分析表：

状态	动作		转移	
	a	\$	S	A
0	S3		1	2
1	S3	acc		4
2	r1	r1		
3	r2	r2		
4	r1	r1		

没有冲突，故该文法为SLR(1)文法



$\text{First}(SA) = \text{First}(A) = \{a\}$

故该文法不是LL(1)文法

3.27 文法G的产生式如下：

$S \rightarrow II R$ $I \rightarrow d I I d$ $R \rightarrow W p F$

$W \rightarrow W d \mid \epsilon$ $F \rightarrow F d I d$

(a) 令d表示任意数字，p表示十进制小数点，那么非终结符S, I, R, W和F在编程语言中分别表示什么？

(b) 该文法是LR(1)文法吗？为什么？



(a)

S: 整数或浮点数

I: 整数

R: 浮点数

W: 浮点整数部分

F: 浮点数小数部分



(b)

IO:
S' -> .S,\$
S -> .I,S
S -> R,S
I -> .d, d/\$
I -> .Id,d/\$
R -> .WpF.\$
W -> .Wd, d/p
W -> .,d/p

S →

I →

R →

d →

...

...



(b) SLR分析表:

状态	动作			转移				
	d	p	\$	S	I	R	F	W
0	s4/r8	r8		1	2	3		5
...								
...								

有冲突，故该文法不是SLR(1)文法



3.30 描述文法

$S \rightarrow aSbS \mid aS \mid \epsilon$

产生的语言，并为此语言写一个LR (1) 文法。



b的数量小于a的数量的ab组成的串。

$S \rightarrow aLbL$

$L \rightarrow aT$

$T \rightarrow S \mid \epsilon$



3.24 证明下面文法

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

是LR(1)文法，但不是LALR(1)文法。



Assignment 8 — — 3.24

I0:
S' -> .S,\$
S -> .Aa,\$
S -> .bAc,\$
S -> .Bc,\$
S -> .bBa,\$
A -> .d,a
B -> .d,c

...

b

d

....
...

d

I9:
A -> d.,c
B -> d.,a

I5:
A -> d.,a
B -> d.,c

状态	动作					转移		
	a	b	c	d	\$	S	A	B
0		s3		s5		1	2	4
1					acc			
...								
5	r5		r6					
...								
9	r9		r5					
...								

没有冲突，故该文法为LR(1)文法

合并状态5，9时有冲突，故该文法不是LALR(1)文法

3.37 下面是一个二义文法

$$S \rightarrow AS \mid b$$
$$A \rightarrow SA \mid a$$

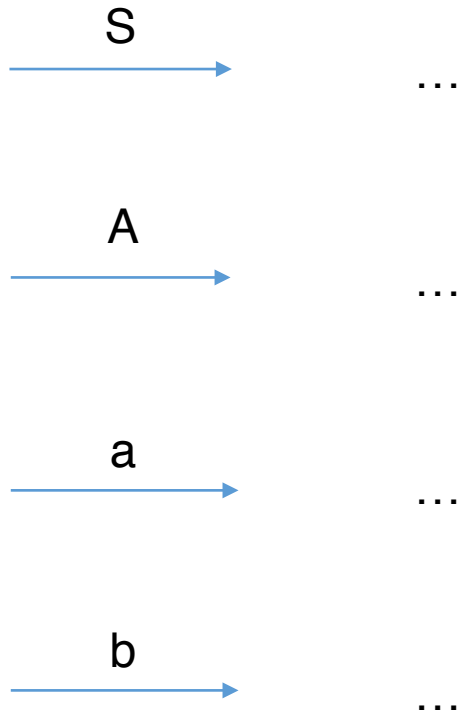
如果为该文法构造LR分析表，则一定存在某些有分析动作冲突的条目，它们是哪些？

假定分析表这样来使用：出现冲突时，不确定地选择一个可能的动作。给出对于输入abab所有可能的动作序列。



Assignment 8 — — 3.37

I0:
S' -> .S
S -> .AS
S -> .b
A -> .SA
A -> ,a





状态	动作			转移	
	a	b	\$	S	A
0	s4	s3		1	2
1	s4	s3	acc	6	5
2	s4	s3		7	2
3	r2	r2	r2		
4	r4	r4			
5	s3,r3	s3,r3		7	2
6	s4	s3		6	5
7	s4,r1	s3,r1	r1	6	5



abab:

(1) r4 -> s3 -> r2 -> s4 -> r4 -> s3 -> r2 -> r1

(2) r4 -> s3 -> r2 -> s4 -> r4 -> r3 -> s3 -> r2 -> r1

(3) r4 -> s3 -> r2 -> r1 -> s4 -> s3 -> r2 -> r1

(4) r4 -> s3 -> r2 -> r1 -> s4 -> r3 -> s3 -> t2 -> r1 -> acc



4.3 为文法

$$S \rightarrow (L) \mid a$$
$$L \rightarrow L, S \mid S$$

- (a) 写一个语法制导定义，它输出括号的对数。
- (b) 写一个语法制导定义，它输出括号嵌套的最大深度。



(a)

$S' \rightarrow S$

print S.num

$S \rightarrow (L)$

S.num = L.num + 1

$S \rightarrow a$

S.num = 0

$L \rightarrow L', S$

L.num = L'.num + S.num

$L \rightarrow S$

L.num = S.num



(b)

$S' \rightarrow S$ print S.depth

$S \rightarrow (L)$ S.depth = L.depth + 1

$S \rightarrow a$ S.depth = 0

$L \rightarrow L', S$ L.depth = max(L.depth + S.depth)

$L \rightarrow S$ L.depth = L.depth



4.14 程序的文法如下：

$P \rightarrow D$

$D \rightarrow D;D \mid id : T \mid \text{proc } id;D;S$

- (a) 写一个语法制导定义，打印该程序一共声明了多少个id。
- (b) 写一个翻译方案，打印该程序每个变量id的嵌套深度。



(a)

$P' \rightarrow P$

`print(P.num)`

$P \rightarrow D$

`P.num = D.num`

$D \rightarrow D1;D2$

`D.num = D1.num + D2.num`

$D \rightarrow id:T$

`D.num = 1 + T.num`

$D \rightarrow \text{proc } id;D';S$

`D.num = 1 + D'.num + S.num`



(b)

$P \rightarrow \{D.depth = 1\} D$

$D \rightarrow \{D1.depth = D.depth\} D1; \{D2.depth = D.depth\} D2$

$D \rightarrow id:T \{print\ id, D.depth\}$

$D \rightarrow proc\ id; D'\{D'.depth = D.depth + 1\}; S$



Q&A