

第 5 章 语法分析——自下而上分析

1. 文法:

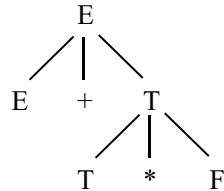
$E \rightarrow E+T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid i$

证明 $E+T * F$ 是它的一个句型, 并指出该句型所有短语、直接短语和句柄。

解: $E+T * F$ 是它的一个句型, 因为存在下面语法树:



短语: $T * F$, $E + T * F$

直接短语: $T * F$

句柄: $T * F$

2. 文法:

$S \rightarrow a \mid \wedge \mid (T)$

$T \rightarrow T, S \mid S$

(1) 给出 $(a, (a, a))$ 和 $((a, a), \wedge, (a)), a$ 的最左和最右推导;

(2) 指出 $((a, a), \wedge, (a)), a$ 的规范归约以及每一步的句柄, 根据这个规范归约, 给出“移进-归约”过程, 并给出它的语法树自下而上构造过程。

解:

(1) 略

(2) ① 规范句型及每一步的句柄 (用下划线标示):

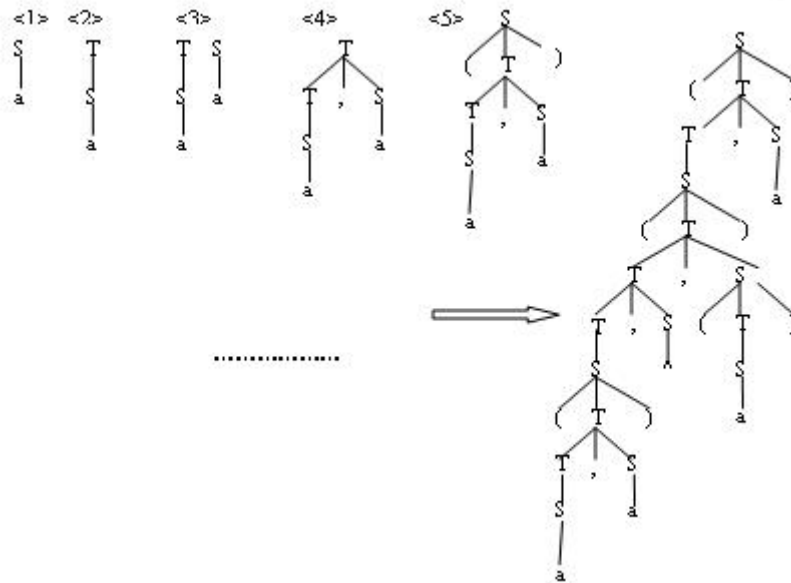
- | | |
|---|---------------------------------|
| 1) $((\underline{a}, a), \wedge, (a)), a$ | 10) $((T, (\underline{S})), a)$ |
| 2) $(((\underline{S}, a), \wedge, (a)), a)$ | 11) $((T, (\underline{T})), a)$ |
| 3) $(((\underline{T}, a), \wedge, (a)), a)$ | 12) $((\underline{T}, S), a)$ |
| 4) $(((\underline{T}, S), \wedge, (a)), a)$ | 13) $((\underline{T}), a)$ |
| 5) $(((\underline{T}), \wedge, (a)), a)$ | 14) $((\underline{S}), a)$ |
| 6) $((\underline{S}, \wedge, (a)), a)$ | 15) $((T, \underline{a}))$ |
| 7) $((T, \underline{\wedge}, (a)), a)$ | 16) $((\underline{T}, S))$ |
| 8) $((\underline{T}, S, (a)), a)$ | 17) $((\underline{T}))$ |
| 9) $((T, (\underline{a})), a)$ | 18) S |

② “移进-规约”过程:

步骤	分析栈	输入串	动作
(1)	#	$((a, a), \wedge, (a)), a \#$	预备
(2)	#($((a, a), \wedge, (a)), a \#$	移进
(3)	#(($(a, a), \wedge, (a)), a \#$	移进
(4)	#((($a, a), \wedge, (a)), a \#$	移进
(5)	#(((a	$, a), \wedge, (a)), a \#$	移进

(6)	#((S	,a),^(a),a)#	归约
(7)	#((T	,a),^(a),a)#	归约
(8)	#((T,	a),^(a),a)#	移进
(9)	#((T,a),^(a),a)#	移进
(10)	#((T,S),^(a),a)#	归约
(11)	#((T),^(a),a)#	归约
(12)	#((T)	,^(a),a)#	移进
(13)	#(S	,^(a),a)#	归约
(14)	#(T	,^(a),a)#	归约
(15)	#(T,	^(a),a)#	移进
(16)	#(T,^	,(a),a)#	移进
(17)	#(T,S	,(a),a)#	归约
(18)	#(T	,(a),a)#	归约
(19)	#(T,	(a),a)#	移进
(20)	#(T,(a),a)#	移进
(21)	#(T,(a)),a)#	移进
(22)	#(T,(S)),a)#	归约
(23)	#(T,(T)),a)#	归约
(24)	#(T,(T)),a)#	移进
(25)	#(T,S),a)#	归约
(26)	#(T),a)#	归约
(27)	#(T)	,a)#	移进
(28)	#(S	,a)#	归约
(29)	#(T	,a)#	归约
(30)	#(T,	a)#	移进
(31)	#(T,a)#	移进
(32)	#(T,S)#	归约
(33)	#(T)#	归约
(34)	#(T)	#	移进
(35)	#S	#	归约
(36)	#S	#	接受

③ 语法树的自下而上的构造过程:



3. (1) 计算练习 2 文法 G2 的 FIRSTVT 和 LASTVT;

(2) 计算 G2 的优先关系, G2 是一个算符优先文法吗?

(3) 给出输入串(a,(a,a))的算符优先分析过程。

解:

(1)

	FIRSTVT	LASTVT
S	{a, ^, (}	{a, ^,) }
T	{ , , a, ^, (}	{ , , a, ^,) }

(2) 优先关系表如下:

	a	^	()	,	#
a				>	>	>
^				>	>	>
(<	<	<	=	<	
)				>	>	>
,	<	<	<	>	>	
#	<	<	<			=

因为: 1) 该文法不含 ϵ 产生式;

2) 该文法是算符文法;

3) 由优先关系表可以看出, 任何终结符之间的优先关系至多满足一种优先关系; 所以该文法是算符优先文法。

(3)

步骤	分析栈	输入串	动作	原因
0	#	(a,(a,a))#	预备	
1	#(a,(a,a))#	移进	#<(
2	#(a	,(a,a))#	移进	(<a
3	#(S	,(a,a))#	归约	a>,
4	#(S,	(a,a))#	移进	(>,
5	#(S,(a,a))#	移进	,(<

6	#(S,(a	,a))#	移进	(<a
7	#(S,(S	,a))#	归约	a>,
8	#(S,(S,	a))#	移进	(<,
9	#(S,(S,a))#	移进	,<a
10	#(S,(S,S))#	归约	,>)
11	#(S,(T))#	归约	,>)
12	#(S,(T))#	移进	(=)
13	#(S,S)#	归约)>)
14	#(T)#	归约	,>)
15	#(T)	#	移进	(=)
16	#S	#	归约)>#
17	#S#		分析成功	

5. 考虑文法:

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

- (1) 列出这个文法所有的 LR(0)项目;
- (2) 构造这个文法的 LR(0)项目集规范族以及识别活前缀的 DFA;
- (3) 这个文法是 SLR 的吗? 若是, 构造分析表;
- (4) 这个文法是 LALR 或 LR(1)的吗?

解: 拓广文法如下:

$S' \rightarrow S$
 $S \rightarrow AS|b$
 $A \rightarrow SA|a$

(1) 所有 LR(0)项目如下:

- 1) $S' \rightarrow \cdot S$ 2) $S' \rightarrow S \cdot$ 3) $S \rightarrow \cdot AS$ 4) $S \rightarrow A \cdot S$ 5) $S \rightarrow AS \cdot$ 6) $S \rightarrow \cdot b$
7) $S \rightarrow b \cdot$ 8) $A \rightarrow \cdot SA$ 9) $A \rightarrow S \cdot A$ 10) $A \rightarrow SA \cdot$ 11) $A \rightarrow \cdot a$ 12) $A \rightarrow a \cdot$

(2) 该文法的 LR(0)项目集规范族如下:

I0: $S' \rightarrow \cdot S$	I2=Go(I0, A):	I5=Go(I1, A):	I7=Go(I2, S):
$S \rightarrow \cdot AS$	$S \rightarrow A \cdot S$	$A \rightarrow SA \cdot$	$S \rightarrow AS \cdot$
$S \rightarrow \cdot b$	$S \rightarrow \cdot AS$	$S \rightarrow A \cdot S$	$A \rightarrow S \cdot A$
$A \rightarrow \cdot SA$	$S \rightarrow \cdot b$	$S \rightarrow \cdot AS$	$A \rightarrow S \cdot A$
$A \rightarrow \cdot a$	$A \rightarrow \cdot SA$	$S \rightarrow \cdot b$	$A \rightarrow \cdot a$
I1=Go(I0, S):	$A \rightarrow \cdot a$	$A \rightarrow \cdot SA$	$S \rightarrow \cdot AS$
$S' \rightarrow S \cdot$	I3=Go(I0, b):	$A \rightarrow \cdot a$	$S \rightarrow \cdot b$
$A \rightarrow S \cdot A$	$S \rightarrow b \cdot$	I6=Go(I1, S):	
$A \rightarrow \cdot SA$	I4=Go(I0, a):	$A \rightarrow S \cdot A$	
$A \rightarrow \cdot a$	$A \rightarrow a \cdot$	$A \rightarrow \cdot SA$	
$S \rightarrow \cdot AS$		$A \rightarrow \cdot a$	
$S \rightarrow \cdot b$		$S \rightarrow \cdot AS$	
		$S \rightarrow \cdot b$	

DFA 略。

(3) $FIRST(S) = \{b, a\}$ $FIRST(A) = \{a, b\}$

$FOLLOW(S) = \{\#, a, b\}$ $FOLLOW(A) = \{a, b\}$

在 LR(0)项目集规范族中，同时存在“移进-归约”和“归约-归约”项目的项集有 I1, I5 和 I7；其中 I1 中“归约”项目是“接受”项目，面临#时接受，移进项目要求面临 a 和 b 时移进，不存在冲突；I5 中归约项目面临 FOLLOW(A)中元素 a,b 时归约，“移进”项目面临 a,b 时移进，存在冲突；同理，I7 也存在冲突。所以该文法不是 SLR 的。

(或者：构造出 SLR 分析表，指出存在多重入口)

(4) 构造 LR(1)项目集规范族：

I0:

$S' \rightarrow \cdot S, \#$
 $S \rightarrow \cdot AS, \#/a/b$
 $S \rightarrow \cdot b, \#/a/b$
 $A \rightarrow \cdot SA, a/b$
 $A \rightarrow \cdot a, a/b$

I1=Go(I0, S):

$S' \rightarrow S \cdot, \#$
 $A \rightarrow S \cdot A, a/b$
 $A \rightarrow \cdot SA, a/b$
 $A \rightarrow \cdot a, a/b$
 $S \rightarrow \cdot AS, a/b$
 $S \rightarrow \cdot b, a/b$

I8=Go(I2, S):

$S \rightarrow AS \cdot, \#/a/b$
 $A \rightarrow S \cdot A, a/b$
 $A \rightarrow \cdot SA, a/b$
 $A \rightarrow \cdot a, a/b$
 $S \rightarrow \cdot AS, a/b$
 $S \rightarrow \cdot b, a/b$

I2=Go(I0, A):

$S \rightarrow A \cdot S, \#/a/b$
 $S \rightarrow \cdot AS, \#/a/b$
 $S \rightarrow \cdot b, \#/a/b$
 $A \rightarrow \cdot SA, a/b$
 $A \rightarrow \cdot a, a/b$

I3=Go(I0, b):

$S \rightarrow b \cdot, \#/a/b$
I4=G(I0, a):
 $A \rightarrow a \cdot, a/b$

I9=Go(I5, S):

$S \rightarrow AS \cdot, a/b$
 $A \rightarrow S \cdot A, a/b$
 $A \rightarrow \cdot SA, a/b$
 $A \rightarrow \cdot a, a/b$
 $S \rightarrow \cdot AS, a/b$
 $S \rightarrow \cdot b, a/b$

I5=Go(I1, A):

$A \rightarrow SA \cdot, a/b$
 $S \rightarrow A \cdot S, a/b$
 $S \rightarrow \cdot AS, a/b$
 $S \rightarrow \cdot b, a/b$
 $A \rightarrow \cdot SA, a/b$
 $A \rightarrow \cdot a, a/b$

I10=Go(I5, A):

$S \rightarrow A \cdot S, a/b$
 $S \rightarrow \cdot AS, a/b$
 $S \rightarrow \cdot b, a/b$
 $A \rightarrow \cdot SA, a/b$
 $A \rightarrow \cdot a, a/b$

I6=Go(I1, S):

$A \rightarrow S \cdot A, a/b$
 $A \rightarrow \cdot SA, a/b$
 $A \rightarrow \cdot a, a/b$
 $S \rightarrow \cdot AS, a/b$
 $S \rightarrow \cdot b, a/b$

I7=Go(I1, b):

$S \rightarrow b \cdot, a/b$

因为：

I5, I8, I9 中存在“移进-归约”冲突，所以该文法不是 LR(1)的，更不是 LALR 的。

(!此题另解：该文法是二义文法，故不是任何 LR 文法。)

8. 证明下面的文法是 LL(1)的但不是 SLR(1)的。

$$\begin{aligned} S &\rightarrow AaAb \mid BbBa \\ A &\rightarrow \varepsilon \\ B &\rightarrow \varepsilon \end{aligned}$$

【解答】

A. 因为 1) 该文法不存在左递归;

$$2) \because \text{First}(AaAb)=\{a\}, \text{First}(BbBa)=\{b\}$$

\therefore 产生式 $S \rightarrow AaAb \mid BbBa$ 的两个候选交集为空

所以该文法是 LL(1)文法。

B. 构造已知文法的 LR(0)项目集规范族如下:

$$I_0 = \{ S \rightarrow \bullet AaAb, S \rightarrow \bullet BbBa, A \rightarrow \bullet, B \rightarrow \bullet \}$$

$$I_1 = \text{GO}(I_0, A) = \{ S \rightarrow A \bullet aAb \}$$

$$I_2 = \text{GO}(I_0, B) = \{ S \rightarrow B \bullet bBa \}$$

$$I_3 = \text{GO}(I_1, a) = \{ S \rightarrow Aa \bullet Ab, A \rightarrow \bullet \}$$

$$I_4 = \text{GO}(I_2, b) = \{ S \rightarrow Bb \bullet Ba, B \rightarrow \bullet \}$$

$$I_5 = \text{GO}(I_3, A) = \{ S \rightarrow AaA \bullet b \}$$

$$I_6 = \text{GO}(I_4, B) = \{ S \rightarrow BbB \bullet a \}$$

$$I_7 = \text{GO}(I_5, b) = \{ S \rightarrow AaAb \bullet \}$$

$$I_8 = \text{GO}(I_6, a) = \{ S \rightarrow BbBa \bullet \}$$

考察: \because 项目集 I_0 中存在两个归约项目 $A \rightarrow \bullet$ 和 $B \rightarrow \bullet$,

$$\text{Follow}(A)=\{a, b\}, \text{Follow}(B)=\{a, b\},$$

\therefore 归约-归约冲突不能解决, 该文法不是 SLR(1)文法。

7. 证明下面文法是 SLR(1)但不是 LR(0)的。

$$\begin{aligned} S &\rightarrow A \\ A &\rightarrow Ab \mid bBa \\ B &\rightarrow aAc \mid a \mid aAb \end{aligned}$$

9. 证明下面文法是 LALR(1)但不是 SLR(1)的。

$$\begin{aligned} S &\rightarrow Aa \mid bAc \mid Bc \mid bBa \\ A &\rightarrow d \end{aligned}$$