

第 6 章 自底向上优先分析

第 1 题

已知文法 $G[S]$ 为：

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

$T \rightarrow T, S | S$

(1) 计算 $G[S]$ 的 FIRSTVT 和 LASTVT。

(2) 构造 $G[S]$ 的算符优先关系表并说明 $G[S]$ 是否为算符优先文法。

(3) 计算 $G[S]$ 的优先函数。

(4) 给出输入串 $(a,a)\#$ 和 $(a,(a,a))\#$ 的算符优先分析过程。

答案：

文法展开为：

$S \rightarrow a$

$S \rightarrow \wedge$

$S \rightarrow (T)$

$T \rightarrow T, S$

$T \rightarrow S$

(1) FIRSTVT - LASTVT 表：

非终结符	FIRSTVT 集	LASTVT 集
S	$\{ a \wedge (\}$	$\{ a \wedge) \}$
T	$\{ a \wedge (, \}$	$\{ a \wedge) , \}$

(2) 算符优先关系表：

	a	^	()	,	#
a				\triangleright	\triangleright	\triangleright
^				\triangleright	\triangleright	\triangleright
(\triangleleft	\triangleleft	\triangleleft	\equiv	\triangleleft	
)				\triangleright	\triangleright	\triangleright
,	\triangleleft	\triangleleft	\triangleleft	\triangleright	\triangleright	
#	\triangleleft	\triangleleft	\triangleleft			\equiv

表中无多重入口所以是算符优先（OPG）文法。

友情提示：记得增加拓广文法 $S' \rightarrow \#S\#$ ，所以 $\# \triangleleft \text{FIRSTVT}(S), \text{LASTVT}(S) \triangleright \#$ 。

(3)对应的算符优先函数为：

	a	()	,	^	#
f	2	1	2	2	2	1
g	3	3	1	1	3	1

(4) 对输入串 (a,a) #的算符优先分析过程为

栈 (STACK)	当前输入字符 (CHAR)	剩余输入串 (INPUT_STRING)	动作 (ACTION)
#	(a,a)#	Move in
#(a	,a)#	Move in
#(a	,	a)#	Reduce: $S \rightarrow a$
#(N	,	a)#	Move in
#(N,	a)#	Move in
#(N,a)	#	Reduce: $S \rightarrow a$
#(N,N)	#	Reduce: $T \rightarrow T,S$
#(N)	#	Move in
#(N)	#		Reduce: $S \rightarrow (T)$
#N	#		

Success!

对输入串 $(a,(a,a))\#$ 的算符优先分析过程为：

栈 (STACK)	当前字符 (CHAR)	剩余输入串 (INPUT_STRING)	动作 (ACTION)
#	(a,(a,a)#	Move in
#(a	,(a,a)#	Move in
#(a	,	(a,a)#	Reduce: $S \rightarrow a$
#(N	,	(a,a)#	Move in
#(N,	(a,a)#	Move in
#(N,(a	,a)#	Move in
#(N,(a	,	a)#	Reduce: $S \rightarrow a$
#(N,(N	,	a)#	Move in
#(N,(N	a)#	Move in
#(N,(N,a))#	Reduce: $S \rightarrow a$
#(N,(N,N))#	Reduce: $T \rightarrow T,S$
#(N,(N))#	Move in
#(N,(N))	#	Reduce: $S \rightarrow (T)$
#(N,N)	#	Reduce: $T \rightarrow T,S$
#(N)	#	Move in
#(N)	#		Reduce: $S \rightarrow (T)$
#N	#		

Success!

第 2 题

已知文法 $G[S]$ 为：

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

$T \rightarrow T,S | S$

(1) 给出 $(a,(a,a))$ 和 (a,a) 的最右推导，和规范归约过程。

(2) 将(1)和题 1 中的(4)进行比较给出算符优先归约和规范归约的区别。

答案：

(1) (a,a) 的最右推导过程为：

$S \Rightarrow (T)$
 $\Rightarrow (T,S)$
 $\Rightarrow (T,a)$
 $\Rightarrow (S,a)$
 $\Rightarrow (a,a)$

$(a,(a,a))$ 的最右推导过程为：

$S \Rightarrow (T)$
 $\Rightarrow (T,S)$
 $\Rightarrow (T,(T))$

$\Rightarrow(T,(T,S))$
 $\Rightarrow(T,(T,a))$
 $\Rightarrow(T,(S,a))$
 $\Rightarrow(T,(a,a))$
 $\Rightarrow(S,(a,a))$
 $\Rightarrow(a,(a,a))$

(a,(a,a))的规范归约过程:

步骤	栈	输 入	动 作
1	#	(a, (a, a))#	移进
2	#(a, (a, a))#	移进
3	#(a	, (a, a))#	归约, $S \rightarrow a$
4	#(S	, (a, a))#	归约, $L \rightarrow S$
5	#(T	, (a, a))#	移进
6	#(T,	(a, a))#	移进
7	#(T, (a, a))#	移进
8	#(T, (a	, a))#	归约, $S \rightarrow a$
9	#(T, (S	, a))#	归约, $T \rightarrow S$
10	#(T, (T	, a))#	移进
11	#(T, (T,	a))#	移进
12	#(T, (T,a)#	归约, $S \rightarrow a$
13	#(T, (T, S)#	归约, $T \rightarrow T, S$
14	#(T, (T)#	移进
15	#(T, (T))#	归约, $S \rightarrow (T)$
16	#(T, S)#	移进
17	#(T, S)	#	归约, $T \rightarrow T, S$
18	#(T)	#	归约, $S \rightarrow (T)$
19	#S	#	接受

(a,a)的规范归约过程:

步骤	栈	输 入	动 作
1	#	(a, a)#	移进
2	# (a, a)#	移进
3	# (a	, a)#	归约, $S \rightarrow a$
4	#(S	, a)#	归约, $T \rightarrow S$
5	#(T	, a)#	移进
6	#(T,	a)#	移进
7	#(T, a)#	归约, $S \rightarrow a$
8	#(T, S)#	归约, $T \rightarrow T, S$
9	#(T)#	移进
10	#(T)	#	归约, $S \rightarrow (T)$
11	# S	#	接受

步骤	栈	优先关系	当前符号	剩余输入串	移进或归约
1	#	# < ((+(i(#	移进
2	#((> +	+	(i(#	归约
3	#F	# < +	+	(i(#	移进
4	#F+	+ < ((i(#	移进
5	#F+((> i	i	(#	归约
6	#F+F	+ > i	i	(#	归约
7	#F	# < i	i	(#	移进
8	#Fi	i < ((#	移进
9	#Fi((> #	#		归约
10	#FiF	i > #	#		归约
11	#F	# == #	#		接受

第4题

文法G [S] 为:

$$\begin{aligned} S &\rightarrow S;G \mid G \\ G &\rightarrow G(T) \mid H \\ H &\rightarrow a \mid (S) \\ T &\rightarrow T+S \mid S \end{aligned}$$

- (1) 构造G [S] 的算符优先关系表, 并判断G [S] 是否为算符优先文法。
- (2) 给出句型a(T+S);H;(S)的短语、句柄、素短语和最左素短语。
- (3) 给出a;(a+a)和(a+a)的分析过程, 说明它们是否为G [S] 的句子。
- (4) 给出(3)中输入串的最右推导, 分别说明两输入串是否为G[S]的句子。
- (5) 由(3)和(4)说明了算符优先分析的哪些缺点。
- (6) 算符优先分析过程和规范归约过程都是最右推导的逆过程吗?

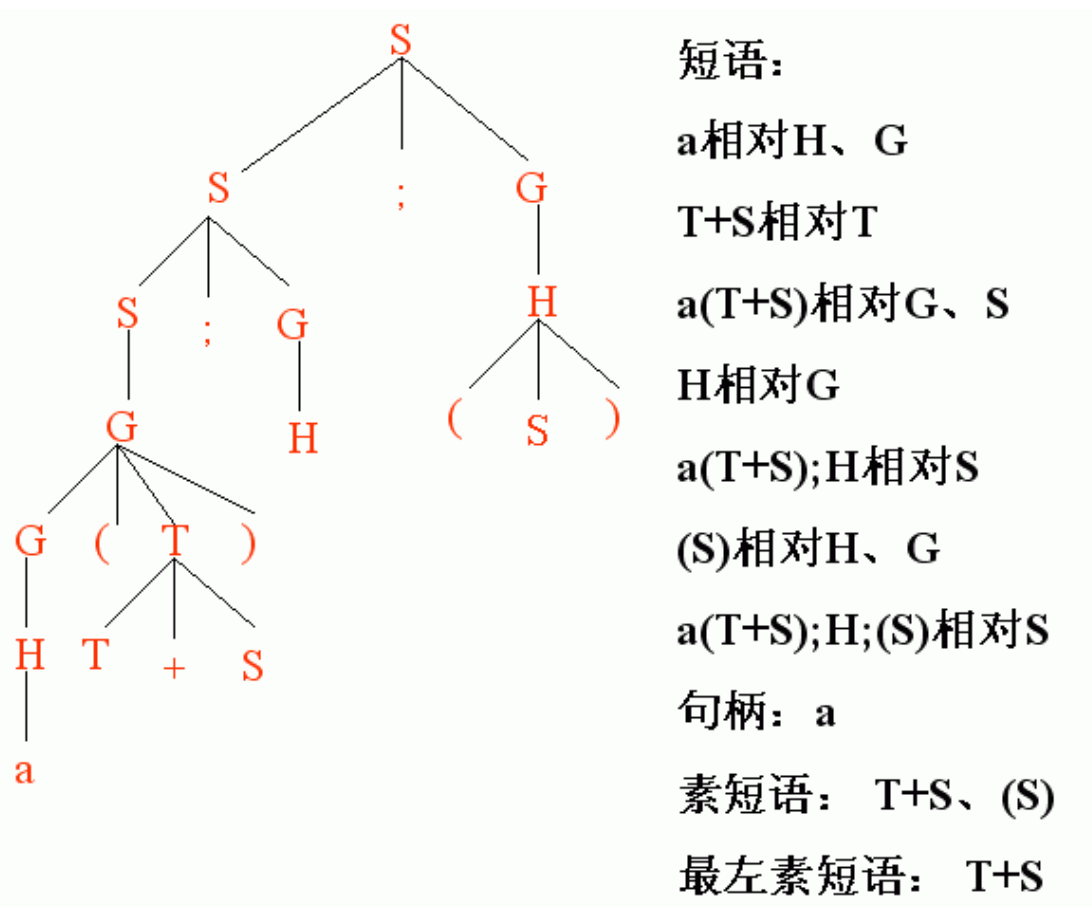
答案:

- (1) 构造文法G [S] 的算符优先关系矩阵:

	;	()	a	+	#
;	•>	<•	•>	<•	•>	•>
(<•	<•	=•	<•	<•	
)	•>	•>	•>		•>	•>
a	•>	•>	•>		•>	•>
+	<•	<•	•>	<•	•>	
#	<•	<•		<•		=•

在上表中可看出终结符之间的优先关系是唯一的, 或称G [S] 的算符优先关系矩阵不含多重入口, 因此, G [S] 是一个算符优先文法。

(2)



(3) 对输入串 (a+a) # 的分析过程如下:

步骤	栈	当前符号	剩余输入串	移进或归约
(1)	#	(a+a) #	移进
(2)	#(a	+a) #	移进
(3)	#(a	+	a) #	归约
(4)	#(N	+	a) #	移进
(5)	#(N+	a) #	移进
(6)	#(N+a)	#	归约
(7)	#(N+N)	#	归约
(8)	#(N)	#	移进
(9)	#(N)	#		归约
(10)	#N	#		分析成功

说明是它的句子。

(4) 试用规范推导:

$$S \Rightarrow G \Rightarrow H \Rightarrow (S)$$

由此往下S 不可能推导出a+a, 所以 (a+a)不是G[S] 的句子。

(5) 结果说明: 由于算符优先分析法去掉了单非终结符之间的归约, 尽管在分析过程中,

当决定是否为句柄时采取一些检查措施，但仍难完全避免把错误的句子得到正确的归约。

（6）算符优先分析过程不是最右推导的逆过程。规范归约过程是最右推导的逆过程。

附加题

问题 1:

对于文法 $S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

(1) 给出句子 $(a, ((a, a), (a, a)))$ 的一个最右推导，并指出右句型的句柄；

(2) 按照(1)的最右推导，说明移进-归约分析器的工作步骤。

答案:

(1) $S \Rightarrow \underline{(L)} \Rightarrow (L, S) \Rightarrow (L, \underline{(L)}) \Rightarrow (L, (L, S)) \Rightarrow (L, (L, \underline{(L)})) \Rightarrow (L, (L, (L, S))) \Rightarrow (L, (L, (L, \underline{a})))$

$\Rightarrow (L, (L, (\underline{S}, a))) \Rightarrow (L, (L, (\underline{a}, a))) \Rightarrow (L, (\underline{S}, (a, a))) \Rightarrow (L, ((\underline{L}), (a, a))) \Rightarrow (L, ((\underline{L}, S), (a, a)))$

$\Rightarrow (L, ((L, \underline{a}), (a, a))) \Rightarrow (L, ((\underline{S}, a), (a, a))) \Rightarrow (L, ((\underline{a}, a), (a, a))) \Rightarrow (\underline{S}, ((a, a), (a, a))) \Rightarrow (\underline{a}, ((a, a), (a, a)))$

(注：句柄下面有下划线)

(2)

步骤	栈	输入	动作
1	#	(a, ((a, a), (a, a)))#	移进
2	#(a, ((a, a), (a, a)))#	移进
3	#(a	, ((a, a), (a, a)))#	归约, $S \rightarrow a$
4	#(S	, ((a, a), (a, a)))#	归约, $L \rightarrow S$
5	#(L	, ((a, a), (a, a)))#	移进
6	#(L,	((a, a), (a, a)))#	移进
7	#(L, ((a, a), (a, a)))#	移进
8	#(L, ((a, a), (a, a)))#	移进
9	#(L, ((a	, a), (a, a)))#	归约, $S \rightarrow a$
10	#(L, ((S	, a), (a, a)))#	归约, $L \rightarrow S$
11	#(L, ((L	, a), (a, a)))#	移进
12	#(L, ((L,	a), (a, a)))#	移进
13	#(L, ((L, a), (a, a)))#	归约, $S \rightarrow a$
14	#(L, ((L, S), (a, a)))#	归约, $L \rightarrow L, S$
15	#(L, ((L), (a, a)))#	移进
16	#(L, ((L)	, (a, a)))#	归约, $S \rightarrow (L)$
17	#(L, (S	, (a, a)))#	归约, $L \rightarrow S$
18	#(L, (L	, (a, a)))#	移进
19	#(L, (L,	(a, a)))#	移进
20	#(L, (L, (a, a)))#	移进
21	#(L, (L, (a	, a)))#	归约, $S \rightarrow a$
22	#(L, (L, (S	, a)))#	归约, $L \rightarrow S$
23	#(L, (L, (L	, a)))#	移进
24	#(L, (L, (L,	a)))#	移进
25	#(L, (L, (L, a)))#	归约, $S \rightarrow a$
26	#(L, (L, (L, S)))#	归约, $L \rightarrow L, S$
27	#(L, (L, (L)))#	移进
28	#(L, (L, (L))))#	归约, $S \rightarrow (L)$
29	#(L, (L, S)))#	归约, $L \rightarrow L, S$
30	#(L, (L)))#	移进
31	#(L, (L)))#	归约, $S \rightarrow (L)$
32	#(L, S)#	归约, $L \rightarrow L, S$
33	#(L)#	移进
34	#(L)	#	归约, $S \rightarrow (L)$
35	#S	#	接受

问题 2:

试为下列各文法建立算符优先关系表。

$E \rightarrow E \text{ and } T | T$

$T \rightarrow T \text{ or } F | F$

$F \rightarrow \text{not } F | N$

$N \rightarrow (E) | \text{true} | \text{false}$

答案:

算符优先关系表如下:

	true	false	not	and	or	()	#
true				\triangleright	\triangleright		\triangleright	\triangleright
false				\triangleright	\triangleright		\triangleright	\triangleright
not	\triangleleft	\triangleleft	\triangleleft	\triangleright	\triangleright	\triangleleft	\triangleright	\triangleright
and	\triangleleft	\triangleleft	\triangleleft	\triangleright	\triangleright	\triangleleft	\triangleright	\triangleright
or	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleright	\triangleleft	\triangleright	\triangleright
(\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\equiv	
)				\triangleright	\triangleright		\triangleright	\triangleright
#	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft		