

第 6 章 自底向上优先分析

第 1 题

已知文法 $G[S]$ 为:

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

$T \rightarrow T, S$

(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) 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	\trianglelefteq	\triangleleft	
)				\triangleright	\triangleright	\triangleright
,	\triangleleft	\triangleleft	\triangleleft	\triangleright	\triangleright	
#	\triangleleft	\triangleleft	\triangleleft			\trianglelefteq

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# \in FIRSTVT(S)，LASTVT(S) \triangleright #。

(3)对输入串 (a,a) #的算符优先分析过程为

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

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	动作 (ACTION)
#	($a,(a,a)\#$	Move in
#(a	$, (a,a)$	Move in
#(a	,)#	Reduce:
#(N	,	(a,a)	$S \rightarrow a$ Move
#(N,	()#	in Move in
#(N,(a	(a,a)	Move in
#(N,(a	,)#	Reduce:
#(N,(N	,	a,a))	$S \rightarrow a$ Move
#(N,(N	a	#	in Move in
#(N,(N,a)	,a)	Reduce:
#(N,(N,N))#	$S \rightarrow a$
#(N,(N)	a)	Reduce:
#(N,(N)))#	$T \rightarrow T,S$ Move
#(N,N)	a)	in
#(N))#	Reduce:
#(N)	#)#	$S \rightarrow (T)$
#N	#)#	Reduce:

Success!

第 2 题

已知文法 $G[S]$ 为：

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

$T \rightarrow T,S \mid S$

(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	栈	输	动
1	#	(a, (a,	移
2	#(a, (a,	移
3	#(a	, (a,	归约, SÆa
4	#(S	, (a,	归约, LÆS
5	#(T	, (a,	移
6	#(T,	(a,	移
7	#(T, (a,	移
8	#(T,	,	归约, SÆa
9	#(T,	,	归约, TÆS
1	#(T,	,	移
1	#(T,	a))	移
1	#(T, (T,a))	归约, SÆa
1	#(T, (T, S))	归约, TÆT, S
1	#(T,))	移
1	#(T,)#	归约, SÆ(T)
1	#(T,)#	移
1	#(T,	#	归约, TÆT, S
1	#(T)	#	归约, SÆ(T)
1	#S	#	接

(a,a)的规范归约过程:

步	栈	输 入	动
1	#	(a,	移
2	# (a,	移
3	# (a	,	归约, SÆa
4	#(S	,	归约, TÆS
5	#(T	,	移
6	#(T,	a)	移
7	#(T,)	归约, SÆa
8	#(T,)	归约, TÆT, S
9	#(T)	移
1	#(T)	#	归约, SÆ(T)

1	# S	#	接
---	-----	---	---

(2)算符优先文法在归约过程中只考虑终结符之间的优先关系从而确定可归约串，而非终结符无关，只需知道把当前可归约串归约为某一个非终结符，不必知道该非终结符的名字是什么，因此去掉了单非终结符的归约。

规范归约的可归约串是句柄，并且必须准确写出可归约串归约为哪个非终结符。

第3题：

有文法 $G[S]$:

$S \rightarrow V$

$V \rightarrow T | ViT$

$T \rightarrow F | T+F$

$F \rightarrow)V*($

- (1) 给出 $(+i)$ 的规范推导。
- (2) 指出句型 $F+Fi$ 的短语，句柄，素短语。
- (3) $G[S]$ 是否为 OPG? 若是，给出(1)中句子的分析过程。

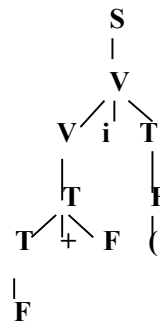
答案：

(1) $S \Rightarrow V \Rightarrow ViT \Rightarrow ViF \Rightarrow Vi(\Rightarrow T i(\Rightarrow T+F i(\Rightarrow T+(i(\Rightarrow F+(i(\Rightarrow +(i($

(2) 句型 $F+Fi$ 的语法树：

短语：F, $F+F$, $($, $F+Fi$ 句柄：F 素短语： $($

(3) FIRSTVT 和 LASTVT



	FIRSTVT	LASTVT
S	i,+,),(i,+,*,(
V	i,+,),(i,+,*,(
T	+,),(+,*,(
F),,(*,(

算符优先关系

	i	+	*	()	#
i	\triangleright	\triangleleft	\triangleright	\triangleleft	\triangleleft	\triangleright
+	\triangleright	\triangleright	\triangleright	\triangleleft	\triangleleft	\triangleright
*	\triangleright	\triangleright	\triangleright			\triangleright
(\triangleright	\triangleright	\triangleright			\triangleright
)	\triangleleft	\triangleleft		\triangleleft	\triangleleft	
#	\triangleleft	\triangleleft		\triangleleft	\triangleleft	\equiv

因为该文法是 OP，同时任意两个终结符的优先关系唯一，所以该文法为 OPG。

(+i)的分析过程

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。

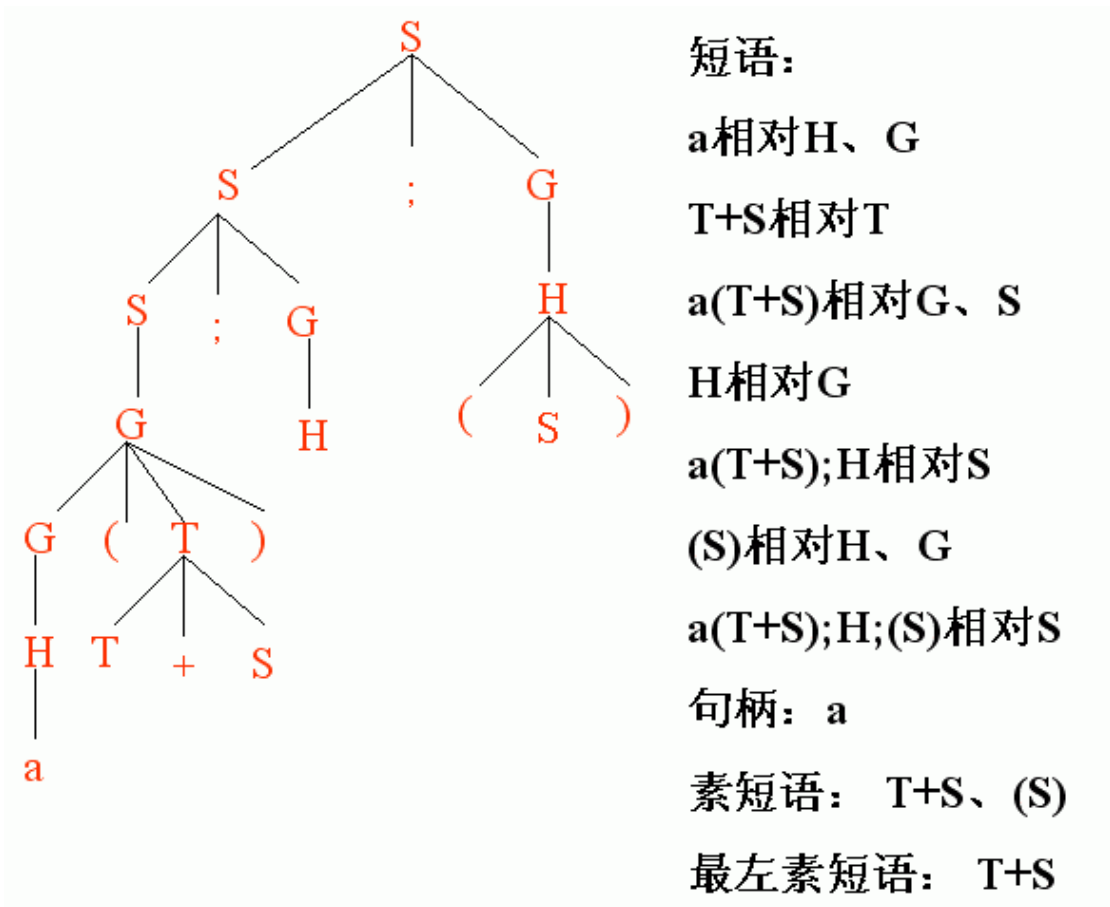
答案:

- (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

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。