

编译原理第四章作业

2154312 郑博远

1. 考虑下面文法 G_1 :

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

$$T \rightarrow T, S \mid S$$

(1) 消去 G_1 的左递归。然后, 对每个非终结符, 写出不带回溯的递归子程序。

(2) 经改写后的文法是否是 $LL(1)$ 的? 给出它的预测分析表。

答: (1)

将状态 S 、 T 按照 1、2 编序。

当 $i = 1, j = 0$ 时, S 不存在直接左递归;

当 $i = 2, j = 1$ 时, 无需消除间接左递归,

消除 T 的直接左递归:

$$T \rightarrow ST'$$

$$T' \rightarrow ,ST' \mid \varepsilon$$

因此消去左递归后 G_1 的文法为:

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

$$T \rightarrow ST'$$

$$T' \rightarrow ,ST' \mid \varepsilon$$

分析其 $FIRST$ 集与 $FOLLOW$ 集:

$$FIRST(S) = \{ a, \wedge, (\}$$

$$FIRST(T) = FIRST(S) = \{ a, \wedge, (\}$$

$$FIRST(T') = \{ ,, \varepsilon \}$$

$$FOLLOW(S) = (FIRST(T') - \{\varepsilon\}) \cup FOLLOW(T) \cup \{\#\} = \{ ,,), \#\}$$

$$FOLLOW(T) = \{) \}$$

$$FOLLOW(T') = FOLLOW(T) = \{) \}$$

因此, 不带回溯的递归子程序为:

<pre> P(S) BEGIN IF ch = 'a' OR ch = '^' THEN read(ch); ELSE IF ch = '(' THEN BEGIN read(ch); P(T); IF ch = ')' THEN read(ch); ELSE ERROR; END ELSE ERROR; END </pre>
<pre> P(T) BEGIN P(S); P(T'); END </pre>
<pre> P(T') BEGIN IF ch = ',' THEN read(ch); P(S); P(T'); ELSE IF ch = ')' THEN return; ELSE ERROR; END </pre>

(2) 首先，该文法不存在左递归（包括直接左递归和间接左递归）；

其次，文法中每个非终结符 A 的各个产生式的候选首符集两两不相交；

最后，对于 $\varepsilon \in FIRST(A)$ 的非终结符 A (即 T')，满足 $FIRST(A) \cap FOLLOW(A) = \emptyset$ 。

因此，该文法是 $LL(1)$ 的。

预测分析表如下：

	()	a	\wedge	,	#
S	$S \rightarrow (T)$		$S \rightarrow a$	$S \rightarrow \wedge$		
T	$T \rightarrow ST'$		$T \rightarrow ST'$	$T \rightarrow ST'$		
T'		$T' \rightarrow \varepsilon$			$T' \rightarrow ,ST'$	

2. 对下面的文法 G ：

$$E \rightarrow TE'$$

$$E' \rightarrow +E \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow T \mid \varepsilon$$

$$F \rightarrow PF'$$

$$F' \rightarrow *F' \mid \varepsilon$$

$$P \rightarrow (E) \mid a \mid b \mid \wedge$$

(1) 计算这个文法的每个非终结符的 $FIRST$ 和 $FOLLOW$ 。

(2) 证明这个文法是 $LL(1)$ 的。

(3) 构造它的预测分析表。

(4) 构造它的递归下降分析程序。

答：(1)

$$FIRST(E) = FIRST(T) = FIRST(F) = FIRST(P) = \{ (, a, b, \wedge \}$$

$$FIRST(E') = \{ +, \varepsilon \}$$

$$FIRST(T') = FIRST(F') = \{ \varepsilon \}$$

$$FIRST(T') = FIRST(F) \cup \{\varepsilon\} = \{ (, a, b, \wedge, \varepsilon \}$$

$$FIRST(F) = FIRST(P) = \{ (, a, b, \wedge \}$$

$$FIRST(F') = \{ *, \varepsilon \}$$

$$FIRST(P) = \{ (, a, b, \wedge \}$$

$$FOLLOW(E) = \{ \#,) \}$$

$$FOLLOW(E') = FOLLOW(E) = \{ \#,) \}$$

$$FOLLOW(T) = (FIRST(E') - \{\varepsilon\}) \cup FOLLOW(E) = \{ \#,), + \}$$

$$FOLLOW(T') = FOLLOW(T) = \{ \#,), + \}$$

$$FOLLOW(F) = (FIRST(T') - \{\varepsilon\}) \cup FOLLOW(T) = \{ (, a, b, \wedge, \#,), + \}$$

$$FOLLOW(F') = FOLLOW(F) = \{ (, a, b, \wedge, \#,), + \}$$

$$FOLLOW(P) = (FIRST(F') - \{\varepsilon\}) \cup FOLLOW(F) = \{ (, a, b, \wedge, \#,), +, * \}$$

(2) 首先，该文法不存在左递归（包括直接左递归和间接左递归）；

其次，文法中每个非终结符 A 的各个产生式的候选首符集两两不相交；

最后，对于 $\varepsilon \in FIRST(A)$ 的非终结符 A （即 E', T', F' ），均满足 $FIRST(A) \cap FOLLOW(A) = \emptyset$ 。

因此，该文法是 $LL(1)$ 的。

(3)

	+	*	\wedge	a	b	()	#
E			$E \rightarrow TE'$	$E \rightarrow TE'$	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'	$E' \rightarrow +E$						$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T			$T \rightarrow FT'$	$T \rightarrow FT'$	$T \rightarrow FT'$	$T \rightarrow FT'$		
T'	$T' \rightarrow \varepsilon$		$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$

F			$F \rightarrow PF'$	$F \rightarrow PF'$	$F \rightarrow PF'$	$F \rightarrow PF'$		
F'	$F' \rightarrow \varepsilon$	$F' \rightarrow * F'$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$
P			$P \rightarrow \wedge$	$P \rightarrow a$	$P \rightarrow b$	$P \rightarrow (E)$		

(4)

<pre> P(E) BEGIN P(T); P(E'); END </pre>	<pre> P(E') BEGIN IF ch = '+' THEN BEGIN read(ch); P(E); END ELSE IF ch = '#' OR ch = ')' THEN return; ELSE ERROR; END </pre>
<pre> P(T) BEGIN P(F); P(T'); END </pre>	<pre> P(T') BEGIN IF ch = '#' or ch = ')' or ch = '+' THEN return; ELSE BEGIN P(T); END END </pre>
<pre> P(F) BEGIN P(P); P(F'); END </pre>	<pre> P(F') BEGIN IF ch = '*' THEN read(ch); P(F'); ELSE IF ch = 'a' OR ch = 'b' OR ch = '(' OR ch = ')' OR ch = '^' OR ch = '+' OR ch = '#' THEN return; ELSE ERROR; END </pre>
<pre> P(P) BEGIN IF ch = '(' THEN read(ch); </pre>	

<pre>P(E); if ch = ')' THEN read(ch); ELSE ERROR; ELSE IF ch = 'a' OR ch = 'b' OR ch = '^' THEN read(ch); ELSE ERROR; END</pre>	
---	--