

Assignment6

ID	周鹏宇
name	2019K8009929039

练习4.2.1：使得文法的预测分析产生回溯的原因是什么？仅使用**FIRST**集合可以避免回溯吗？为什么？

- 如果没有足够的信息来唯一的确定可能的产生式，那么分析过程中就会产生回溯；回溯需要来回扫描，如果已经分析出了一段语义，则应当撤销，会极其低效；
- 并不可以。不妨以课本上的 $A \rightarrow \alpha | \beta$ 为例，即使 $\text{FIRST}(\alpha)$ 和 $\text{FIRST}(\beta)$ 的交集为空，若 $\epsilon \in \text{FIRST}(\alpha)$ ，若能构造出一个非空串 x ，使得 $x \in \text{FOLLOW}(A)$ 且 $x \in \text{FIRST}(\beta)$ ，则会出现信息不确定的状况，即可能产生回溯。

练习4.2.2：考虑文法：

$lexp \rightarrow atom \mid list$

$atom \rightarrow number \mid identifier$

$list \rightarrow (lexp-seq)$

$lexp-seq \rightarrow lexp-seq \ lexp \mid lexp$

- 消除左递归
- 求得该文法的**FIRST**集合和**FOLLOW**集合
- 说明所得的文法是**LL(1)**文法
- 为所得的文法构造**LL(1)**分析表
- 对输入串(a (b (2)) (c))给出相应得**LL(1)**分析程序的动作

a).

$$\begin{aligned}
lexp &\rightarrow atom \mid list \\
atom &\rightarrow number \mid identifier \\
list &\rightarrow (lexp-sep) \\
lexp-sep &\rightarrow lexp lexp-sep' \\
lexp-sep' &\rightarrow lexp lexp-sep' \mid \epsilon
\end{aligned}$$

b).

- $FIRST(lexp) = \{number, identifier, \{\}$
- $FIRST(atom) = \{number, identifier\}$
- $FIRST(list) = \{\{\}$
- $FIRST(lexp-sep) = \{number, identifier, \{\}$
- $FIRST(lexp-sep') = \{\epsilon, number, identifier, \{\}$
- $FOLLOW(lexp-sep) = \{ \}$
- $FOLLOW(lexp-sep') = \{ \}$
- $FOLLOW(lexp) = \{\$, number, identifier, (,)\}$
- $FOLLOW(atom) = \{\$, number, identifier, (,)\}$
- $FOLLOW(list) = \{\$, number, identifier, (,)\}$

c), 按照定义证明即可：

- 对于lexp, 有 $FIRST(atom) \cap FIRST(list) = \phi$
- 对于atom, 有 $FIRST(number) \cap FIRST(identifier) = \phi$
- 对于lexp-sep', 有 $FIRST(lexp) \cap FIRST(\epsilon) = \phi$, 且由于 ϵ 推导出空串, 而 $FIRST(lexp) \cap FOLLOW(lexp-sep') = \phi$

故其为LL1文法

d).

<u>Aa</u> LL(1) P- table	$\equiv ($	$\equiv)$	$\equiv number$	$\equiv identifier$	$\equiv \$$
<u>lexp</u>	$lexp \rightarrow list$		$lexp \rightarrow atom$	$lexp \rightarrow atom$	

<u>Aa</u> LL(1) P- table	$\equiv ($	$\equiv)$	$\equiv \text{number}$	$\equiv \text{identifiler}$	$\equiv \$$
<u>atom</u>			atom \rightarrow number	atom \rightarrow identifiler	
<u>list</u>	list \rightarrow (lexp- sep)				
<u>lexp- sep</u>	lexp- sep \rightarrow lexp lexp-sep'		lexp-sep \rightarrow lexp lexp-sep'	lexp-sep \rightarrow lexp lexp-sep'	
<u>lexp- sep'</u>	lexp- sep' \rightarrow lexp lexp-sep'	lexp \rightarrow epsilon	lexp- sep' \rightarrow lexp lexp-sep'	lexp-sep' \rightarrow lexp lexp-sep'	lexp- sep' \rightarrow epsilon

e). 由于标记过于麻烦，故简化如下：

- lexp \rightarrow E
- list \rightarrow L
- atom \rightarrow A
- lexp-sep \rightarrow S
- lexp-sep' \rightarrow S'

则有分析程序动作如下：

Stack	Input	Move
\$E	(a(c b(2))(c))\$	$E \rightarrow L$
\$L	(a(c b(2))(c))\$	$L \rightarrow (S)$
\$)SC	(a(c b(2))(c))\$	✓
\$)S	a(c b(2))(c)\$	$S \rightarrow ES'$
\$)S'E	a(c b(2))(c)\$	$E \rightarrow A$
\$)S'A	a(c b(2))(c)\$	$A \rightarrow id$
\$)S'id	a(c b(2))(c)\$	✓
\$)S'	(b(2))(c)\$	$S' \rightarrow ES'$
\$)S'E	(b(2))(c)\$	$E \rightarrow L$
\$)S'L	(b(2))(c)\$	$L \rightarrow (S)$
\$)S)SC	(b(2))(c)\$	✓
\$)S)S	b(2))(c)\$	$S \rightarrow ES'$
\$)S)S'E	b(2))(c)\$	$E \rightarrow A$
\$)S)S'A	b(2))(c)\$	$A \rightarrow id$
\$)S)S'id	b(2))(c)\$	✓
\$)S)S'	(2))(c)\$	$S' \rightarrow ES'$
\$)S)S'E	(2))(c)\$	$E \rightarrow L$
\$)S)S'L	(2))(c)\$	$L \rightarrow (S)$
\$)S)S)SC	(2))(c)\$	✓
\$)S)S)S	2))(c)\$	$S \rightarrow ES'$
\$)S)S)S'E	2))(c)\$	$E \rightarrow A$
\$)S)S)S'A	2))(c)\$	$A \rightarrow num$
\$)S)S)S'num	2))(c)\$	✓
\$)S)S)S') (c)\$	$S' \rightarrow \epsilon$
\$)S)S)) (c)\$	✓
\$)S)S') (c)\$	$S' \rightarrow \epsilon$
\$)S)) (c)\$	✓
\$)S'	(c)\$	$S' \rightarrow ES'$
\$)S'E	(c)\$	$E \rightarrow L$
\$)S'L	(c)\$	$L \rightarrow (S)$
\$)S)SC	(c)\$	✓
\$)S)S	(c)\$	$S \rightarrow ES'$
\$)S)S'E	(c)\$	$E \rightarrow A$
\$)S)S'A	(c)\$	$A \rightarrow id$
\$)S)S'id	(c)\$	✓
\$)S)S')\$	$S' \rightarrow \epsilon$
\$)S))\$	✓
\$)S')\$	$S' \rightarrow \epsilon$
\$))\$	✓
\$	\$	✓

