

Chapter 4 – HW02

2015K8009929049 冯吕

2018 年 7 月 9 日

1. 解：产生回溯的原因是，即使当非终结符用某个产生式匹配成功，但是这种成功可能只是暂时的，因为没有足够的信息来唯一地确定可能的产生式，所以分析过程就会产生回溯。不可以。例如对于产生式 $A \rightarrow \alpha \mid \beta$ ， $FIRST(\alpha)$ 与 $FIRST(\beta)$ 交集为空集，但 ϵ 是其中某个 $FIRST$ 集合的元素，不是一般性，假设 $\epsilon \in FIRST(\alpha)$ ，想要避免回溯，则还需要考虑 $FOLLOW(A)$ 与 $FIRST(\beta)$ 的情况。

2. 解： a 消除左递归后的文法如下：

$$lexp \rightarrow atom \mid list$$

$$atom \rightarrow number \mid identifier$$

$$list \rightarrow (lexp - seq)$$

$$lexp - seq \rightarrow lexp A'$$

$$A' \rightarrow lexp A' \mid \epsilon$$

b. $FIRST$ 集合：

$$FIRST(lexp) = \{number, identifier, (\}$$

$$FIRST(atom) = \{number, identifier\}$$

$$FIRST(list) = \{ (\}$$

$$FIRST(number) = \{number\}$$

$$FIRST(identifier) = \{identifier\}$$

$$FIRST(()) = \{ (\}$$

$$FIRST()) = \{) \}$$

$$FIRST(lexp - seq) = \{number, identifier, (\}$$

$$FIRST(A') = \{number, identifier, (, \epsilon\}$$

$$FIRST((lexp - seq)) = \{(\}$$

$$FIRST(lexp A') = \{number, identifier, (\}$$

$$FIRST((lexp - seq)) = \{(\}$$

$$FIRST(lexp - seq) = \{number, identifier, (\}$$

FOLLOW 集合：

$$FOLLOW(lexp) = \{\$,), number, identifier, (\}$$

$$FOLLOW(atom) = \{\$,), number, identifier, (\}$$

$$FOLLOW(list) = \{\$,), number, identifier, (\}$$

$$FOLLOW(lexp - seq) = \{) \}$$

$$FOLLOW(A') = \{) \}$$

c. 证明：

- 对于规则 $lexp \rightarrow atom \mid list$, $atom$ 推出的串的首字符为 `number` 或 `identifier`, $list$ 推出的串的首字符为 `(`, 满足条件；
- 对于规则 $atom \rightarrow number \mid identifier$, 显然满足条件；
- 对于规则 $A' \rightarrow lexp A' \mid \epsilon$, 也满足 $LL(1)$ 文法的条件；

因此, 该文法是 $LL(1)$ 文法。

d. $LL(1)$ 分析表如下：

非终结符号	输入符号				
	number	identifier	()	\$
$lexp$	$lexp \rightarrow atom$	$lexp \rightarrow atom$	$lexp \rightarrow list$		
$atom$	$atom \rightarrow number$	$atom \rightarrow identifier$			
$list$			$list \rightarrow (lexp - seq)$		
$lexp - seq$	$lexp - seq \rightarrow lexp A'$	$lexp - seq \rightarrow lexp A'$	$lexp - seq \rightarrow lexp A'$		
A'	$A' \rightarrow lexp A'$	$A' \rightarrow lexp A'$	$A' \rightarrow lexp A'$	$A' \rightarrow \epsilon$	

d. 对输入串 $(a (b (2)) (c))$, $LL(1)$ 分析程序的动作如下：

<i>STACK</i>	<i>INPUT</i>	<i>ACTION</i>
<i>lexp</i> \$	(<i>a</i> (<i>b</i> (2)) (<i>c</i>))\$	
<i>list</i> \$	(<i>a</i> (<i>b</i> (2)) (<i>c</i>))\$	output <i>lexp</i> \rightarrow <i>list</i>
(<i>lexp</i> – <i>seq</i>)\$	(<i>a</i> (<i>b</i> (2)) (<i>c</i>))\$	output <i>list</i> \rightarrow (<i>lexp</i> – <i>seq</i>)
<i>lexp</i> – <i>seq</i>)\$	<i>a</i> (<i>b</i> (2)) (<i>c</i>))\$	match (
<i>lexp</i> <i>A'</i>)\$	<i>a</i> (<i>b</i> (2)) (<i>c</i>))\$	output <i>lexp</i> – <i>seq</i> \rightarrow <i>lexp</i> <i>A'</i>
<i>atom</i> <i>A'</i>)\$	<i>a</i> (<i>b</i> (2)) (<i>c</i>))\$	output <i>lexp</i> \rightarrow <i>atom</i>
<i>a</i> <i>A'</i>)\$	<i>a</i> (<i>b</i> (2)) (<i>c</i>))\$	output <i>atom</i> \rightarrow <i>a</i>
<i>A'</i>)\$	(<i>b</i> (2)) (<i>c</i>))\$	match <i>a</i>
<i>lexp</i> <i>A'</i>)\$	(<i>b</i> (2)) (<i>c</i>))\$	output <i>A'</i> \rightarrow <i>lexp</i> <i>A'</i>
<i>list</i> <i>A'</i>)\$	(<i>b</i> (2)) (<i>c</i>))\$	output <i>lexp</i> \rightarrow <i>list</i>
(<i>lexp</i> – <i>seq</i>) <i>A'</i>)\$	(<i>b</i> (2)) (<i>c</i>))\$	output <i>list</i> \rightarrow (<i>lexp</i> – <i>seq</i>)
<i>lexp</i> – <i>seq</i>) <i>A'</i>)\$	<i>b</i> (2)) (<i>c</i>))\$	match (
<i>lexp</i> <i>A'</i>) <i>A'</i>)\$	<i>b</i> (2)) (<i>c</i>))\$	output <i>lexp</i> – <i>seq</i> \rightarrow <i>lexp</i> <i>A'</i>
<i>atom</i> <i>A'</i>) <i>A'</i>)\$	<i>b</i> (2)) (<i>c</i>))\$	output <i>lexp</i> \rightarrow <i>atom</i>
<i>b</i> <i>A'</i>) <i>A'</i>)\$	<i>b</i> (2)) (<i>c</i>))\$	output <i>atom</i> \rightarrow <i>b</i>
<i>A'</i>) <i>A'</i>)\$	(2)) (<i>c</i>))\$	match <i>b</i>
<i>lexp</i> <i>A'</i>) <i>A'</i>)\$	(2)) (<i>c</i>))\$	output <i>A'</i> \rightarrow <i>lexp</i> <i>A'</i>
<i>list</i> <i>A'</i>) <i>A'</i>)\$	(2)) (<i>c</i>))\$	output <i>lexp</i> \rightarrow <i>list</i>
(<i>lexp</i> – <i>seq</i>) <i>A'</i>) <i>A'</i>)\$	(2)) (<i>c</i>))\$	output <i>list</i> \rightarrow (<i>lexp</i> – <i>seq</i>)
<i>lexp</i> – <i>seq</i>) <i>A'</i>) <i>A'</i>)\$	2)) (<i>c</i>))\$	match (
<i>lexp</i> <i>A'</i>) <i>A'</i>) <i>A'</i>)\$	2)) (<i>c</i>))\$	output <i>lexp</i> – <i>seq</i> \rightarrow <i>lexp</i> <i>A'</i>
<i>atom</i> <i>A'</i>) <i>A'</i>) <i>A'</i>)\$	2)) (<i>c</i>))\$	output <i>lexp</i> \rightarrow <i>atom</i>
2 <i>A'</i>) <i>A'</i>) <i>A'</i>)\$	2)) (<i>c</i>))\$	output <i>atom</i> \rightarrow 2
<i>A'</i>) <i>A'</i>) <i>A'</i>)\$) (<i>c</i>))\$	match 2
) <i>A'</i>) <i>A'</i>)\$) (<i>c</i>))\$	output <i>A'</i> \rightarrow ϵ
<i>A'</i>) <i>A'</i>)\$) (<i>c</i>))\$	match (
) <i>A'</i>)\$) (<i>c</i>))\$	output <i>A'</i> \rightarrow ϵ
<i>A'</i>)\$	(<i>c</i>))\$	match (
<i>lexp</i> <i>A'</i>)\$	(<i>c</i>))\$	output <i>A'</i> \rightarrow <i>lexp</i> <i>A'</i>
<i>list</i> <i>A'</i>)\$	(<i>c</i>))\$	output <i>lexp</i> \rightarrow <i>list</i>
(<i>lexp</i> – <i>seq</i>) <i>A'</i>)\$	(<i>c</i>))\$	output <i>list</i> \rightarrow (<i>lexp</i> – <i>seq</i>)
<i>lexp</i> – <i>seq</i>) <i>A'</i>)\$	<i>c</i>))\$	match (
<i>lexp</i> <i>A'</i>) <i>A'</i>)\$	<i>c</i>))\$	output <i>lexp</i> – <i>seq</i> \rightarrow <i>lexp</i> <i>A'</i>
<i>atom</i> <i>A'</i>) <i>A'</i>)\$	<i>c</i>))\$	output <i>lexp</i> \rightarrow <i>atom</i>
<i>c</i> <i>A'</i>) <i>A'</i>)\$	<i>c</i>))\$	output <i>atom</i> \rightarrow <i>c</i>
<i>A'</i>) <i>A'</i>))\$	match <i>c</i>
) <i>A'</i>)\$)\$	output <i>A'</i> \rightarrow ϵ
<i>A'</i>)\$)\$	match)
)\$)\$	output <i>A'</i> \rightarrow ϵ
\$	\$	match)