

# Compiler 4-2

孔静 2014K8009929022

October 22, 2017

## 1

1. 引起回溯的原因是在文法中当关于某个非终结符的产生式有多个候选时，而面临当前的输入符无法确定选用唯一的产生式，从而引起回溯。
2. 不能。
3. 相同的左部的产生式的右部 FIRST 集交集不为空会引起回溯。

## 2

1.

$$\begin{aligned}lexp &\rightarrow atom | list \\atom &\rightarrow number | identifier \\list &\rightarrow (lexp - seq) \\lexp - seq &\rightarrow lexp A \\A &\rightarrow lexp A | \varepsilon\end{aligned}$$

2.

$$\begin{aligned}\text{FIRST}(lexp) &= \{number, identifier, ()\} \\ \text{FIRST}(atom) &= \{number, identifier\} \\ \text{FIRST}(list) &= \{()\} \\ \text{FIRST}(lexp - seq) &= \{number, identifier, ()\} \\ \text{FIRST}(A) &= \{\varepsilon, number, identifier, ()\} \\ \\ \text{FOLLOW}(lexp) &= \{\$, ), number, identifier, ()\} \\ \text{FOLLOW}(atom) &= \{\$, ), number, identifier, ()\} \\ \text{FOLLOW}(list) &= \{\$, ), number, identifier, ()\} \\ \text{FOLLOW}(lexp - seq) &= \{()\} \\ \text{FOLLOW}(A) &= \{()\}\end{aligned}$$

3.

$$\begin{aligned}\text{FIRST}(atom) \cap \text{FIRST}(list) &= \varphi \\ \text{因为 } \text{FIRST}(lexp A) \cap \text{FIRST}(\varepsilon) &= \text{FIRST}(lexp) \cap \text{FIRST}(\varepsilon) = \varphi \\ \text{FIRST}(A) \cap \text{FOLLOW}(A) &= \varphi\end{aligned}$$

所以该文法是 LL(1) 文法。

4.

非终结符	输入符号				
	number	identifier	(	)	\$
lexp	lexp→atom	lexp→atom	lexp→list		
atom	atom→number	atom→identifier			
list			list→(lexp-seq)		
lexp-seq	lexp-seq→lexp A	lexp-seq→lexp A	lexp-seq→lexp A		
A	A→lexp A	A→lexp A	A→lexp A	A→ε	

5.

栈	输入	动作
\$ lexp	(a(b(2))(c))\$	lexp→list
\$ list	(a(b(2))(c))\$	list→(lexp-seq)
\$ )lexp-seq(	(a(b(2))(c))\$	match (
\$ )lexp-seq	a(b(2))(c))\$	lexp-seq→lexp A
\$ )A lexp	a(b(2))(c))\$	lexp→atom
\$ )A atom	a(b(2))(c))\$	atom→identifier
\$ )A identifier	a(b(2))(c))\$	match a
\$ )A	(b(2))(c))\$	A→lexp A
\$ )A lexp	(b(2))(c))\$	lexp→list
\$ )A list	(b(2))(c))\$	list→(lexp-seq)
\$ )A )lexp-seq(	(b(2))(c))\$	match (
\$ )A )lexp-seq	b(2))(c))\$	lexp-seq→lexp A
\$ )A )A lexp	b(2))(c))\$	lexp→atom
\$ )A )A atom	b(2))(c))\$	atom→identifier
\$ )A )A identifier	b(2))(c))\$	match b
\$ )A )A	(2))(c))\$	A→lexp A
\$ )A )A lexp	(2))(c))\$	lexp→list
\$ )A )A list	(2))(c))\$	list→(lexp-seq)
\$ )A )A )lexp-seq(	(2))(c))\$	match (
\$ )A )A )lexp-seq	2))(c))\$	lexp-seq→lexp A
\$ )A )A )A lexp	2))(c))\$	lexp→atom
\$ )A )A )A atom	2))(c))\$	atom→number
\$ )A )A )A number	2))(c))\$	match 2

栈	输入	动作
\$ )A)A)A	))(c))\$	$A \rightarrow \varepsilon$
\$ )A)A)	))(c))\$	match )
\$ )A)A	)(c))\$	$A \rightarrow \varepsilon$
\$ )A)	)(c))\$	match )
\$ )A	(c))\$	$A \rightarrow \text{lexp } A$
\$ )A $\text{lexp}$	(c))\$	$\text{lexp} \rightarrow \text{list}$
\$ )A $\text{list}$	(c))\$	$\text{list} \rightarrow (\text{lexp-seq})$
\$ )A)lexp-seq(	(c))\$	match (
\$ )A)lexp-seq	c))\$	$\text{lexp-seq} \rightarrow \text{lexp } A$
\$ )A)A $\text{lexp}$	c))\$	$\text{lexp} \rightarrow \text{atom}$
\$ )A)A $\text{atom}$	c))\$	$\text{atom} \rightarrow \text{identifier}$
\$ )A)A $\text{identifier}$	c))\$	match c
\$ )A)A	))\$	$A \rightarrow \varepsilon$
\$ )A)	))\$	match )
\$ )A	)\$	$A \rightarrow \varepsilon$
\$ )	)\$	match )
\$	\$	accept