

## 编译原理第四章(二)

李鹏辉

2018 年 10 月 25 日

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？

因为根据输入串和非终结符的某个推导的选择过程中，字符是按照顺序读入，然后进行匹配和选择，所得到的关于字符串的信息是局部的。而我们产生某个推导选择的过程中，只能依靠当前所扫描到的字符串，因此当当前匹配不成功时，需要回溯到上一层，重新选择，进行匹配。

不能。FIRST集合仅仅是当文法满足LL(1)时，才可以根据FIRST集合做出没有回溯的唯一选择。还需要考虑FOLLOW集合等

2. 考虑文法：

为防止歧义，下表中使用下划线\_代替原有的—

$$\begin{array}{l} \textit{lexp} \rightarrow \textit{atom} | \textit{list} \\ \textit{atom} \rightarrow \textit{number} | \textit{identifiler} \\ \textit{list} \rightarrow (\textit{lexp\_seq}) \\ \textit{lexp\_seq} \rightarrow \textit{lexp\_seq} \textit{lexp} | \textit{lexp} \end{array}$$

a. 消除左递归。

$$\begin{array}{l} \textit{lexp} \rightarrow \textit{atom} | \textit{list} \\ \textit{atom} \rightarrow \textit{number} | \textit{identifiler} \\ \textit{list} \rightarrow (\textit{lexp\_seq}) \\ \textit{lexp\_seq} \rightarrow \textit{lexp} \textit{lexp\_seq}' \\ \textit{lexp\_seq}' \rightarrow \varepsilon | \textit{lexp} \textit{lexp\_seq}' \end{array}$$

b. 求得该文法的FIRST集合和FOLLOW集合

| Table 1: FIRST集合  |
|---|
| $FIRST(\textit{number}) = \{\textit{number}\}$  |
| $FIRST(\textit{identifiler}) = \{\textit{identifiler}\}$                                  |
| $FIRST(( )) = \{ ( \}$  |
| $FIRST( ) ) = \{ ) \}$  |
| $FIRST(\textit{lexp}) = \{\textit{number}, \textit{identifiler}, ( \}$                    |
| $FIRST(\textit{atom}) = \{\textit{number}, \textit{identifiler}\}$                        |
| $FIRST(\textit{list}) = \{ ( \}$  |
| $FIRST(\textit{lexp\_seq}) = \{\textit{number}, \textit{identifiler}, ( \}$               |
| $FIRST(\textit{lexp\_seq}') = \{\varepsilon, \textit{number}, \textit{identifiler}, ( \}$ |

首先需要判断该文法的开始符号为 $\textit{lexp}$

Table 2: FOLLOW集合

|   |
|---|
| $FOLLOW(lexp) = \{\$, number, identifier, (, )\}$ |
| $FOLLOW(lexp\_seq) = \{ \}$                       |
| $FOLLOW(atom) = \{\$, number, identifier, (, )\}$ |
| $FOLLOW(list) = \{\$, number, identifier, (, )\}$ |
| $FOLLOW(lexp\_seq') = \{ \}$                      |

c. 说明所得的文法是LL(1)文法

1.  $lexp \rightarrow atom|list$

$FIRST(atom) \cap FIRST(list) = \emptyset \Rightarrow$  满足条件

2.  $atom \rightarrow number|identifier$

$FIRST(number) \cap FIRST(identifier) = \emptyset \Rightarrow$  满足条件

3.  $lexp\_seq' \rightarrow \varepsilon|lexp\ lexp\_seq'$

$FIRST(lexp\ lexp\_seq') = FIRST(lexp)$

$FIRST(\varepsilon) = FOLLOW(lexp\_seq') = \{ \}$

$FIRST(lexp\ lexp\_seq') \cap FIRST(\varepsilon) = \emptyset \Rightarrow$  满足条件

d. 构造LL(1)文法分析表

Table 3: 文法分析表

| 非终结符         | $number$                                  | $identifier$                              | $($                                       | $)$                                  | $\$$ |
|--------------|---|---|---|--------------------------------------|------|
| $lexp$       | $lexp \rightarrow atom$                   | $lexp \rightarrow atom$                   | $lexp \rightarrow list$                   |                                      |      |
| $atom$       | $atom \rightarrow number$                 | $atom \rightarrow identifier$             |   |                                      |      |
| $list$       |   |   | $list \rightarrow (lexp\_seq)$            |                                      |      |
| $lexp\_seq$  | $lexp\_seq \rightarrow lexp\ lexp\_seq'$  | $lexp\_seq \rightarrow lexp\ lexp\_seq'$  | $lexp\_seq \rightarrow lexp\ lexp\_seq'$  |                                      |      |
| $lexp\_seq'$ | $lexp\_seq' \rightarrow lexp\ lexp\_seq'$ | $lexp\_seq' \rightarrow lexp\ lexp\_seq'$ | $lexp\_seq' \rightarrow lexp\ lexp\_seq'$ | $lexp\_seq' \rightarrow \varepsilon$ |      |

e. 对输入串 $(a(b(2))(c))$ 给出相应得到LL(1)分析程序的动作

Table 4:  $(a(b(2))(c))$ 

| 已匹配        | 栈  | 输入               | 动作   |
|------------|--|------------------|--|
|            | $lexp\$$   | $(a(b(2))(c))\$$ |  |
|            | $list\$$   | $(a(b(2))(c))\$$ | 输出 $lexp \rightarrow list$                   |
|            | $(lexp\_seq)\$$                                  | $(a(b(2))(c))\$$ | 输出 $list \rightarrow (lexp\_seq)$            |
| (          | $lexp\_seq)\$$                                   | $a(b(2))(c))\$$  | 匹配(  |
| (          | $lexp\ lexp\_seq'\$$                             | $a(b(2))(c))\$$  | 输出 $lexp\_seq \rightarrow lexp\ lexp\_seq'$  |
| (          | $atom\ lexp\_seq'\$$                             | $a(b(2))(c))\$$  | 输出 $lexp \rightarrow atom$                   |
| (          | $identifilier\ lexp\_seq'\$$                     | $a(b(2))(c))\$$  | 输出 $atom \rightarrow identifilier$           |
| (          | $a\ lexp\_seq'\$$                                | $a(b(2))(c))\$$  | 输出 $identifilier \rightarrow a$              |
| (          | $a\ lexp\_seq'\$$                                | $a(b(2))(c))\$$  | 匹配a  |
| (a         | $lexp\ lexp\_seq'\$$                             | $(b(2))(c))\$$   | 输出 $lexp\_seq' \rightarrow lexp\ lexp\_seq'$ |
| (a         | $list\ lexp\_seq'\$$                             | $(b(2))(c))\$$   | 输出 $lexp \rightarrow list$                   |
| (a         | $(lexp\_seq)\ lexp\_seq'\$$                      | $(b(2))(c))\$$   | 输出 $list \rightarrow (lexp\_seq)$            |
| (a(        | $lexp\_seq)\ lexp\_seq'\$$                       | $b(2))(c))\$$    | 匹配(  |
| (a(        | $lexp\ lexp\_seq')\ lexp\_seq'\$$                | $b(2))(c))\$$    | 输出 $lexp\_seq' \rightarrow lexp\ lexp\_seq'$ |
| (a(        | $atom\ lexp\_seq')\ lexp\_seq'\$$                | $b(2))(c))\$$    | 输出 $lexp\_seq' \rightarrow lexp\ lexp\_seq'$ |
| (a(        | $identifilier\ lexp\_seq')\ lexp\_seq'\$$        | $b(2))(c))\$$    | 输出 $atom \rightarrow identifilier$           |
| (a(        | $b\ lexp\_seq')\ lexp\_seq'\$$                   | $b(2))(c))\$$    | 输出 $identifilier \rightarrow b$              |
| (a(b       | $lexp\_seq')\ lexp\_seq'\$$                      | $(2))(c))\$$     | 匹配b  |
| (a(b       | $lexp\ lexp\_seq')\ lexp\_seq'\$$                | $(2))(c))\$$     | 输出 $lexp\_seq' \rightarrow lexp\ lexp\_seq'$ |
| (a(b       | $list\ lexp\_seq')\ lexp\_seq'\$$                | $(2))(c))\$$     | 输出 $lexp \rightarrow list$                   |
| (a(b       | $(lexp\_seq)\ lexp\_seq')\ lexp\_seq'\$$         | $(2))(c))\$$     | 输出 $list \rightarrow (lexp\_seq)$            |
| (a(b(      | $lexp\_seq)\ lexp\_seq')\ lexp\_seq'\$$          | $2))(c))\$$      | 匹配(  |
| (a(b(      | $lexp\ lexp\_seq')\ lexp\_seq')\ lexp\_seq'\$$   | $2))(c))\$$      | 输出 $lexp\_seq \rightarrow lexp\ lexp\_seq'$  |
| (a(b(      | $atom\ lexp\_seq')\ lexp\_seq')\ lexp\_seq'\$$   | $2))(c))\$$      | 输出 $lexp \rightarrow atom$                   |
| (a(b(      | $number\ lexp\_seq')\ lexp\_seq')\ lexp\_seq'\$$ | $2))(c))\$$      | 输出 $atom \rightarrow number$                 |
| (a(b(      | $2\ lexp\_seq')\ lexp\_seq')\ lexp\_seq'\$$      | $2))(c))\$$      | 输出 $number \rightarrow 2$                    |
| (a(b(2     | $lexp\_seq')\ lexp\_seq')\ lexp\_seq'\$$         | $)\))(c))\$$     | 匹配2  |
| (a(b(2     | $)\ lexp\_seq')\ lexp\_seq'\$$                   | $)\))(c))\$$     | 弹出 $lexp\_seq'$                              |
| (a(b(2     | $lexp\_seq')\ lexp\_seq'\$$                      | $)\)(c))\$$      | 匹配)  |
| (a(b(2     | $)\ lexp\_seq'\$$                                | $)\)(c))\$$      | 弹出 $lexp\_seq'$                              |
| (a(b(2     | $lexp\_seq'\$$                                   | $(c))\$$         | 匹配)  |
| (a(b(2))   | $lexp\ lexp\_seq'\$$                             | $(c))\$$         | 输出 $lexp\_seq' \rightarrow lexp\ lexp\_seq'$ |
| (a(b(2))   | $list\ lexp\_seq'\$$                             | $(c))\$$         | 输出 $lexp \rightarrow list$                   |
| (a(b(2))   | $(lexp\_seq)\ lexp\_seq'\$$                      | $(c))\$$         | 输出 $list \rightarrow (lexp\_seq)$            |
| (a(b(2))(  | $lexp\_seq)\ lexp\_seq'\$$                       | $c))\$$          | 匹配(  |
| (a(b(2))   | $lexp\ lexp\_seq')\ lexp\_seq'\$$                | $c))\$$          | 输出 $lexp\_seq \rightarrow lexp\ lexp\_seq'$  |
| (a(b(2))(  | $atom\ lexp\_seq')\ lexp\_seq'\$$                | $c))\$$          | 输出 $lexp \rightarrow atom$                   |
| (a(b(2))(  | $identifilier\ lexp\_seq')\ lexp\_seq'\$$        | $c))\$$          | 输出 $atom \rightarrow identifilier$           |
| (a(b(2))(  | $c\ lexp\_seq')\ lexp\_seq'\$$                   | $c))\$$          | 输出 $identifilier \rightarrow c$              |
| (a(b(2))(c | $lexp\_seq')\ lexp\_seq'\$$                      | $)\)$$           | 匹配c  |
| (a(b(2))(c | $)\ lexp\_seq'\$$                                | $)\)$$           | 弹出 $lexp\_seq'$                              |
| (a(b(2))(c | $lexp\_seq'\$$                                   | $\$$             | 匹配)  |
| (a(b(2))(c | $lexp\_seq'\$$                                   | $\$$             | 弹出 $lexp\_seq'$                              |