# Homework 4

PB17000297 罗晏宸

September 26 2019

## 1 Exercise 3.9

为习题 3.3 的文法构造预测分析器（递归下降分析程序）。

$$S \to S \textbf{ and } S \mid S \textbf{ or } S \mid \textbf{not } S \mid \textbf{true} \mid \textbf{false} \mid (S)$$

**解** 该文法是二义的，等价的非二义文法为

$$E \to E \textbf{ or } T \mid T$$
$$T \to T \textbf{ and } F \mid F$$
$$F \to \textbf{not } F \mid (E) \mid \textbf{true} \mid \textbf{false}$$

消除其中的左递归，得到

$$E \to TE'$$
$$E' \to \textbf{or } TE' \mid \varepsilon$$
$$T \to FT'$$
$$T' \to \textbf{and } FT' \mid \varepsilon$$
$$F \to \textbf{not } F \mid (E) \mid \textbf{true} \mid \textbf{false}$$

列出文法中各非终结符的开始符号和后继符号集合

$FIRST(E) = \{$ **not**, (, **true**, **false** $\}$ $\quad FOLLOW(E) = \{$ ), \$ $\}$

$FIRST(E') = \{$ **or**, $\varepsilon$ $\}$ $\quad FOLLOW(E') = \{$ ), \$ $\}$

$FIRST(T) = \{$ **not**, (, **true**, **false** $\}$ $\quad FOLLOW(T) = \{$ **or**, ), \$ $\}$

$FIRST(T') = \{$ **and**, $\varepsilon$ $\}$ $\quad FOLLOW(T') = \{$ **or**, ), \$ $\}$

$FIRST(F) = \{$ **not**, (, **true**, **false** $\}$ $\quad FOLLOW(F) = \{$ **and**, **or**, ), \$ $\}$

对于产生式 $E' \to \textbf{or}\ TE' \mid \varepsilon$

$$
\begin{aligned}
&FIRST(\textbf{or}\ TE') \cap FIRST(\varepsilon) \\
=&\{\ \textbf{or}\ \} \cap \{\ \varepsilon\ \} \\
=&\emptyset
\end{aligned}
$$

$$
\begin{aligned}
&FIRST(\textbf{or}\ TE') \cap FOLLOW(E') \\
=&\{\ \textbf{or}\ \} \cap \{\ \texttt{)}, \texttt{\$}\ \} \\
=&\emptyset
\end{aligned}
$$

对于产生式 $T' \to \textbf{and}\ FT' \mid \varepsilon$

$$
\begin{aligned}
&FIRST(\textbf{and}\ FT') \cap FIRST(\varepsilon) \\
=&\{\ \textbf{and}\ \} \cap \{\ \varepsilon\ \} \\
=&\emptyset
\end{aligned}
$$

$$
\begin{aligned}
&FIRST(\textbf{and}\ FT') \cap FOLLOW(T') \\
=&\{\ \textbf{and}\ \} \cap \{\ \textbf{or}, \texttt{)}, \texttt{\$}\ \} \\
=&\emptyset
\end{aligned}
$$

对于产生式 $F \rightarrow \textbf{not}\ F \mid (E) \mid \textbf{true} \mid \textbf{false}$

$$
\begin{aligned}
&FIRST(\textbf{not}\ F) \cap FIRST((E)) \cap FIRST(\textbf{true}) \cap FIRST(\textbf{false}) \\
=&\{\ \textbf{not}\ \} \cap \{\ )\ \} \cap \textbf{true} \cap \textbf{false} \\
=&\emptyset
\end{aligned}
$$

因此该文法是 LL(1) 文法。为其构造递归下降预测分析器如下

```
1   void match (terminal t)
2   {
3       if (lookahead == t)
4           lookahead = nextToken();
5       else
6           error();
7       return;
8   }
9
10  void E()
11  {
12      if ((lookahead == not) || (lookahead == '(')
            || (lookahead == true) || (lookahead ==
           false))
13      {
14          T();
15          E'();
16      }
17      else
18          error();
19      return;
20  }
21
22  void E'()
23  {
24      if (lookahead == or)
25      {
```

```
26              match(or);
27              T();
28              E'();
29          }
30          else if ((lookahead == ')') || (lookahead
                == '$'))
31              match(lookahead);
32          else
33              error();
34          return;
35      }
36
37      void T()
38      {
39          if ((lookahead == not) || (lookahead == '(')
                || (lookahead == true) || (lookahead ==
                false))
40          {
41              F();
42              T'();
43          }
44          else
45              error();
46          return;
47      }
48
49      void T'()
50      {
51          if (lookahead == and)
52          {
53              match(and);
54              F();
55              T'();
```

```
56          }
57          else if (lookahead == or)
58          {
59              match(or);
60              T();
61              E'();
62          }
63          else if ((lookahead == ')') || (lookahead
               == '$'))
64              match(lookahead);
65          else
66              error();
67          return;
68      }
69
70      void F()
71      {
72          if (lookahead == not)
73          {
74              match(not);
75              F();
76          }
77          else if (lookahead == '(')
78          {
79              match('(');
80              E();
81              match(')');
82          }
83          else if (lookahead == true)
84              match(true);
85          else if (lookahead == false)
86              match(false);
87          else
```

```
88              error();
89          return;
90      }
```

## 2 Exercise 3.11

构造下面文法的 LL(1) 分析表。

$$S \to aBS \mid bAS \mid \varepsilon$$
$$A \to bAA \mid a$$
$$B \to aBB \mid b$$

**解** 列出文法中各非终结符的开始符号和后继符号集合

$$FIRST(S) = \{\ a, b, \varepsilon\ \} \qquad FOLLOW(S) = \{\ \$\ \}$$
$$FIRST(A) = \{\ a, b\ \} \qquad FOLLOW(A) = \{\ a, b, \$\ \}$$
$$FIRST(B) = \{\ a, b\ \} \qquad FOLLOW(B) = \{\ a, b, \$\ \}$$

填写分析表如下

表 1: LL(1) 分析表

| | $a$ | $b$ | $\$$ |
|---|---|---|---|
| $S$ | $S \to aBS$ | $S \to bSS$ | $S \to \varepsilon$ |
| $A$ | $A \to a$ | $A \to bAA$ | |
| $B$ | $B \to aBB$ | $B \to b$ | |

## 3 Exercise 3.16

**(a)** 用习题 3.1 的文法构造 $(a, (a, a))$ 的最右推导，说出每个右句型的句柄。

$$S \to (L) \mid a$$
$$L \to L, S \mid S$$

**解** 由此前构造分析树，可以得到句子 $(a, (a, a))$ 的最右推导

$$\begin{aligned} S &\Rightarrow_{\text{lm}} (L) \Rightarrow_{\text{lm}} (L, S) \Rightarrow_{\text{lm}} (L, (L)) \Rightarrow_{\text{lm}} (L, (L, S)) \Rightarrow_{\text{lm}} (L, (L, a)) \\ &\Rightarrow_{\text{lm}} (L, (S, a)) \Rightarrow_{\text{lm}} (L, (a, a)) \Rightarrow_{\text{lm}} (S, (a, a)) \\ &\Rightarrow_{\text{lm}} (a, (a, a)) \end{aligned}$$

给其中的 $a$ 以下标，并给每个右句型的句柄添加下划线

$$\begin{aligned} S &\Rightarrow_{\text{lm}} (\underline{L}) \\ &\Rightarrow_{\text{lm}} \underline{(L, S)} \\ &\Rightarrow_{\text{lm}} (L, \underline{(L)}) \\ &\Rightarrow_{\text{lm}} (L, \underline{(L, S)}) \\ &\Rightarrow_{\text{lm}} (L, (L, \underline{a_3})) \\ &\Rightarrow_{\text{lm}} (L, (\underline{S}, a_3)) \\ &\Rightarrow_{\text{lm}} (L, (\underline{a_2}, a_3)) \\ &\Rightarrow_{\text{lm}} (\underline{S}, (a_2, a_3)) \\ &\Rightarrow_{\text{lm}} (\underline{a_1}, (a_2, a_3)) \end{aligned}$$

## 4 Non-textbook Exercise

**(1)** 删除以下文法 G 中的左递归，并由此得到文法 G1。

| | 文法 G：$A$ 是开始符号 |
|---|---|
| 1 | $A \to Ba$ |
| 2 | $B \to dab$ |
| 3 | $B \to Cb$ |
| 4 | $C \to cB$ |
| 5 | $C \to Ac$ |

**(2)** G1 是否为 LL（1）的文法？如不是，适当修改该文法 G1，使之成为 LL(1) 的。

**解**

**(1)** 用 $A$ 的产生式 $A \to Ba$ 代换 $C \to Ac$ 中的 $A$，再用产生式 $B \to Cb$ 代换其中的 $B$，用 $B$ 的产生式 $B \to Cb$ 代换 $A \to Ba$ 中的 $B$，再用产生式 $C \to Ac$ 代换其中的 $C$ 得到如下文法

| | 文法 G：$A$ 是开始符号 |
|---|---|
| 1 | $A \to Ba$ |
| 2 | $B \to dab$ |
| 3 | $B \to Cb$ |
| 4 | $C \to Cbac \mid dabac \mid cB$ |

删除其中的直接左递归，得到如下的文法

表 2: 消除左递归的文法 G1

| | 文法 G：$A$ 是开始符号 |
|---|---|
| 1 | $A \to Ba$ |
| 2 | $B \to dab$ |
| 3 | $B \to Cb$ |
| 4 | $C \to dabacC'$ |
| 5 | $C \to cBC'$ |
| 6 | $C' \to bacC'$ |
| 7 | $C' \to \varepsilon$ |

**(2)** 列出文法中各非终结符的开始符号和后继符号集合

$$
\begin{aligned}
&FIRST(A) = \{\ c, d\ \} && FOLLOW(A) = \{\ \$\ \} \\
&FIRST(B) = \{\ c, d\ \} && FOLLOW(B) = \{\ a, b\ \} \\
&FIRST(C) = \{\ c, d\ \} && FOLLOW(C) = \{\ b\ \} \\
&FIRST(C') = \{\ b, \varepsilon\ \} && FOLLOW(C') = \{\ b, \$\ \}
\end{aligned}
$$

对于产生式 $B \to dab \mid Cb$

$$
\begin{aligned}
&FIRST(dab)\ \cap\ FIRST(Cb) \\
=&\{\ d\ \}\ \cap\ FIRST(C) \\
=&\{\ d\ \}\ \cap\ \{\ c, d\ \} \\
=&\{\ d\ \}
\end{aligned}
$$

因此 G1 不是 LL(1) 文法。将 $C$ 的产生式 $C \to dabacC' \mid C \to cBC'$ 代入 $B$ 的产生式 $B \to dab \mid Cb$，提左因子得到文法 G2

表 3: 消除左递归并提左因子的 LL(1) 文法 G2

| | 文法 G：$A$ 是开始符号 |
|---|---|
| 1 | $A \to Ba$ |
| 2 | $B \to cBC'b$ |
| 3 | $B \to dabB'$ |
| 4 | $B' \to acC'b$ |
| 5 | $B' \to \varepsilon$ |
| 6 | $C \to dabacC'$ |
| 7 | $C \to cBC'$ |
| 8 | $C' \to bacC'$ |
| 9 | $C' \to \varepsilon$ |