

### 2019 级弘毅班《编译原理》第三次练习

注意：2022 年 5 月 16 日交作业！

一、设有 C 语言说明语句文法  $D$  如下所示：

$$\begin{aligned} D &\rightarrow T L \\ T &\rightarrow \text{int} \mid \text{char} \\ L &\rightarrow *L \mid L[] \mid L() \mid (L) \mid \text{id} \end{aligned}$$

- (1) 由于该文法是二义文法，因此其 LR 分析表一定有移进/归约或归约/归约冲突，试用 JFLAP 指出那些状态集有怎样的冲突；
- (2) 请正确地对冲突项目选择移进/归约构造其 SLR 分析表，使得按照该分析表分析正表达式时，其运算的优先级别和结合次序与 C 语言说明语句中规定的优先级别和结合次序一致 (仅列出冲突项集合对应的 Action 表)；
- (3) 利用你的分析表分析输入正规表达式 “int \* id ( ) [ ]” 的分析过程。

二、设有文法  $G$  定义如下：

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow aAb \mid aA \mid a \\ B &\rightarrow bBa \mid bB \mid \varepsilon \end{aligned}$$

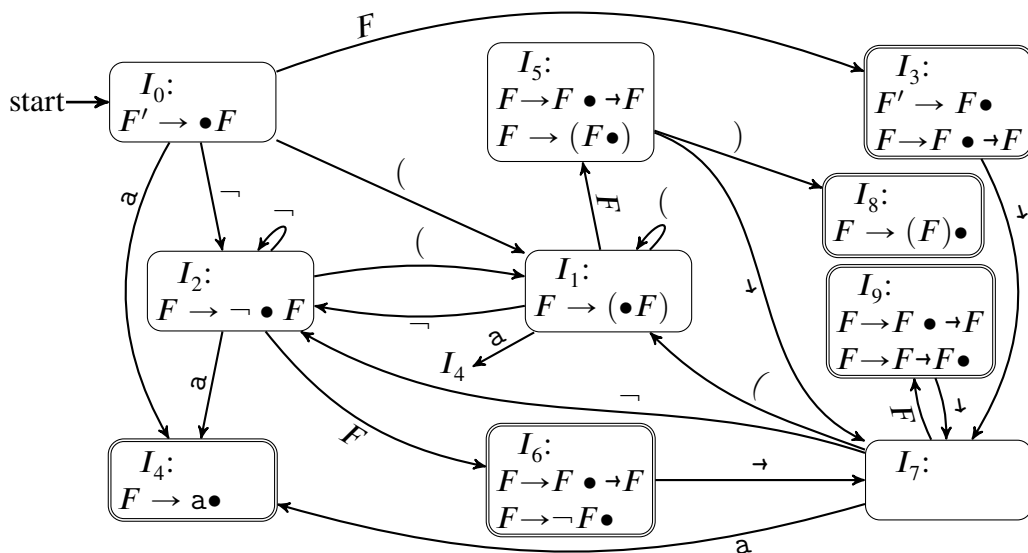
- (1) 试描述文法  $G$  所生成的语言；
- (2) 试不用构造分析表直接说明该文法不是 SLR 文法；
- (3) 试修改文法  $G$  使之成为 SLR 文法。

三、设  $G(F)$  的拓广文法  $G(F')$  如下所示：

(2020 年考题)

$$\begin{aligned} F' &\rightarrow F & (0) \\ F &\rightarrow F \rightarrow F & (1) \\ &\mid \neg F & (2) \\ &\mid (F) & (3) \\ &\mid a & (4) \end{aligned}$$

文法  $G(F')$  的识别活前缀 LR(0) 项目自动机  $M$  如下图所示 (注意每个状态仅列出了核心项目，状态  $I_7$  除外)：



- (1) 试求状态  $I_7$  所对应的 LR(0) 项目集;
  - (2) 试求仅由终结符号组成的活前缀对应的正则表达式;
  - (3) 试构造该文法的 SLR 分析表, 并对分析表中的移进/归约和归约/归约冲突选择正确的移进或归约动作, 使得文法  $G(F)$  的所有语句能被正确地分析且运算的优先级与结合次序与题三所规定的一致;
  - (4) 试利用你的分析表写出语句 “ $\neg a \rightarrow a$ ” 的分析过程.
- 四、 现需对题三文法  $G(F')$  所生成的命题公式转换为析取 ( $\vee$ )、合取 ( $\wedge$ ) 和仅有对原子取否的逻辑等价公式, 如:

序号	原命题公式	转换后的命题公式
1	$A \rightarrow B$	$\neg A \vee B$
2	$\neg \neg A$	$A$
3	$\neg (A \rightarrow B)$	$(A) \wedge (\neg B)$
4	$A \rightarrow (B \rightarrow C)$	$\neg A \vee \neg B \vee C$
5	$(A \rightarrow B) \rightarrow C$	$(A) \wedge (\neg B) \vee C$

为此设计继承属性  $F.is\_neg$ , 其取值为布尔量 **True** 和 **False**; 综合属性  $F.nnf$ , 其取值为  $F$  所表示的语法成分对应的转换后的命题公式 (字符串);  $a.lexeme$  取值为  $a$  所对应的字符串.  $F.is\_neg$  的语义规则如下所示: (2020 年考题)

产生式	语义规则
$F' \rightarrow F$	$F.is\_neg = \text{False}$
$F \rightarrow F_1 \rightarrow F_2$	$F_1.is\_neg = \neg F.is\_neg; F_2.is\_neg = F.is\_neg$
$F \rightarrow \neg F_1$	$F_1.is\_neg = \neg F.is\_neg$
$F \rightarrow (F_1)$	$F_1.is\_neg = F.is\_neg$

- (1) 试写出属性  $F.nnf$  的语法制导定义;

(2) 试求 “ $\neg(((A \rightarrow \neg B) \rightarrow C) \rightarrow \neg(D \rightarrow E))$ ” 转换后的命题公式.

五、设有正规表达式文法  $G$  定义如下:

$$\begin{aligned} T &\rightarrow AS \\ S &\rightarrow S + S \mid SS \mid S^* \mid (S) \mid \epsilon \mid c \\ A &\rightarrow \epsilon \quad (\text{空产生式}) \end{aligned}$$

其中: ‘+’ 表示并运算, 终结符 ‘ $c$ ’ 表示任意的字符, ‘ $\epsilon$ ’ 为正规表达式中的空串, 不是空产生式用到的空串, 其运算优先级与结合次序与正规表达式的规定一致.

现给出一个正规表达式直接到 DFA 的算法, 该算法需要对每个正规表达式计算如下属性 (参见龙书 §3.9 Optimization of DFA-Based Pattern Matchers):

- (a) 终结符  $c$  的属性  $lexval$  为  $c$  所对应的字符, 如:  $(a + b) * abb$  的第一个  $a$  在词法分析抽象为单词  $c$  后, 其对应单词  $c$  的属性  $lexval = a$ ;
- (b) 终结符  $c$  的属性  $pos$  为  $c$  所对应的字母在正规表达式中出现位置序号, 如:  $(a + b) * abb$  的序号为:  

$$\underbrace{(a)}_1 \mid \underbrace{b)}_2 * \underbrace{a}_3 \underbrace{b}_4 \underbrace{b}_5;$$
- (c) 非终结符  $S$  的属性  $firstpos$  为  $S$  所表示的正规表达式中出现的字符对应  $pos$  集合, 且该  $pos$  上的字符能作为  $S$  所表达的字符串的首字符, 如:  $(a + b) * abb$  的  $firstpos = \{1, 2, 3\}$ ,  $pos = 4$  上的字符  $b$  虽然能作为首字符出现在  $(a + b) * abb$  所生成的字符串的首字符, 但是该  $b$  是由  $pos$  为 2 上的  $b$  提供, 不是  $pos$  为 4 上的  $b$  提供, 所以该正规表达式的  $firstpos$  不包含 4;
- (d) 非终结符  $S$  的属性  $S.lastpos$  为  $S$  所表达的正规表达式中出现的字符对应  $pos$  集合, 且该字符能作为  $S$  所表达的字符串的尾字符, 如:  $(a + b) * abb$  的  $lastpos = \{5\}$ ,  $pos = 4$  上的字符  $b$  虽然能作为尾字符出现在  $(a + b) * abb$  所生成的字符串的首字符, 但是该  $b$  是由  $pos$  为 5 上的  $b$  提供, 不是  $pos$  为 4 上的  $b$  提供, 所以该正规表达式的  $lastpos$  不包含 4;
- (e) 非终结符  $S$  和终结符  $c$  的属性  $nullable$  为  $true$ , 当且仅当  $S$  所表达的字符串集合含有空串  $\epsilon$ , 否则为  $false$ , 如:  $(a + b) * abb$  的  $nullable = false$ ;
- (f) 非终结符  $S$  和终结符  $c$  的属性  $followpos$  为  $pos$  或输入结束标志 ‘\$’ 组成的集合, 某一个  $pos$  属于  $S.followpos$  或  $c.followpos$  当且仅当该  $pos$  对应的字符能作为  $S$  或  $c$  对应的子正规表达式的后随字符. 如  $S$  对应的是整个正规表达式, 则  $followpos$  为  $\{\$ \}$ ;  $pos = 1$  的  $a$  所对应单词  $c$  的  $followpos = \{1, 2, 3\}$ , 因为  $pos = 1$  的  $a$  出现在  $(a + b) * abb$  所生成的字符串之中时, 其后随符号可以是  $pos = 1$  提供的  $a$ , 或  $pos = 2$  提供的  $b$ , 或  $pos = 3$  上提供的  $a$ , 但不可能在该  $a$  之后紧随  $pos = 4$  提供的  $b$ .

计算上述属性的语法制导定义如下, 其中终结符  $c$  的属性  $pos$  置为  $count$ ;  $lexval$ ,  $firstpos$  和  $lastpos$  均置为  $c$  所对应的字符:

编号	产生式	语义规则
0	$T \rightarrow AS$	$S.followpos = \{\$ \}$
1	$A \rightarrow \epsilon$	$count = 1$ (初始化全局变量 $count$ )
2	$S \rightarrow S_1 + S_2$	$S.firstpos = S_1.firstpos \cup S_2.firstpos$ $S.lastpos = S_1.lastpos \cup S_2.lastpos$ $S.nullable = S_1.nullable \vee S_2.nullable$ $S_1.followpos = S.followpos$ $S_2.followpos = S.followpos$
3	$S \rightarrow S_1 S_2$	$S.firstpos = \text{if } (S_1.nullable) \text{ then}$ $\quad S_1.firstpos \cup S_2.firstpos$ $\text{else } S_1.firstpos$ $S.lastpos = \text{if } (S_2.nullable) \text{ then}$ $\quad S_1.lastpos \cup S_2.lastpos$ $\text{else } S_2.lastpos$ $S.nullable = S_1.nullable \wedge S_2.nullable$ $S_2.followpos = S.followpos$ $S_1.followpos = \text{if } (S_2.nullable) \text{ then}$ $\quad S_2.firstpos \cup S.followpos$ $\text{else } S_2.firstpos$
4	$S \rightarrow S_1^*$	$S.firstpos = S_1.firstpos$ $S.lastpos = S_1.lastpos$ $S.nullable = true$ $S_1.followpos = S.followpos \cup S_1.firstpos$
5	$S \rightarrow (S_1)$	$S.firstpos = S_1.firstpos$ $S.lastpos = S_1.lastpos$ $S.nullable = S_1.nullable$ $S_1.followpos = S.followpos$
6	$S \rightarrow c$	$S.firstpos = c.firstpos$ $S.lastpos = c.firstpos$ $S.nullable = false$ $c.followpos = S.followpos$ $count = count + 1$
7	$S \rightarrow \epsilon$	$S.firstpos = \emptyset$ $S.lastpos = \emptyset$ $S.nullable = true$

- (1) 分别指出属性  $pos$ ,  $lexval$ ,  $firstpos$ ,  $lastpos$ ,  $count$  和  $followpos$  是否为综合属性, L 属性和继承属性;
- (2) 试画出  $(a + b) * abb$  对应的附注语法树;
- (3) 利用每个单词  $c$  的  $followpos$  可以从正规表达式直接构造 DFA, 首先 DFA 的初始状态  $S_0$  为语法树树根  $T$  的儿子  $S$  的  $firstpos$  值, 即整个表达式对应的  $firstpos$ , 如果整个表达式的  $nulllable$  为  $true$ , 表示自动机接受空串,

即  $S_0$  此时需要加上元素 ‘\$’; 设  $S_i$  为 DFA 的一个状态, 则状态转换函数:

$$Dtrans(S_i, a) = \bigcup_{\mathbf{c}.pos \in S_i \wedge \mathbf{c}.lexval = a} \mathbf{c}.followpos.$$

$S_i$  是接受状态, 当且仅当  $\$ \in S_i$ ; 如  $(a + b) * abb$  的初始状态为  $\{1, 2, 3\}$ ,  $Dtrans(\{1, 2, 3\}, a) = \mathbf{c}_1.followpos \cup \mathbf{c}_3.followpos$ , 其中  $\mathbf{c}_1.pos = 1 \wedge \mathbf{c}_1.lexval = a$ ;  $\mathbf{c}_3.pos = 3 \wedge \mathbf{c}_3.lexval = a$ . 试用上述方法求  $(a + b) * abb$  对应的 DFA.

(4) 对正规表达式  $b * (ab * ab*) *$  做第 (2) 和第 (3) 问.