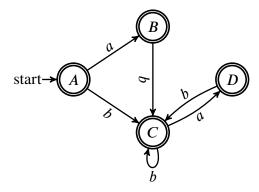
武汉大学计算机学院 2015-2016 学年第一学期 2013 级《编译原理》参考答案

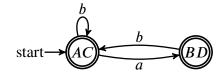
一、(1)

start
$$\longrightarrow 0$$
 $\stackrel{\varepsilon}{\longrightarrow} 1$ $\stackrel{b}{\longrightarrow} 1$ $\stackrel{a}{\longrightarrow} 5$ $\stackrel{b}{\longrightarrow} 1$ $\stackrel{\varepsilon}{\longrightarrow} 2$

(2)



(3) 最小 DFA 如下所示:



- (4) 由 a 和 b 组成且没有连续 a 的字符串.
- (5) $r = (ab \mid b)^*(a \mid \varepsilon)$.
- 二、(1)最左推导如下:

$$S \implies \{I\} \qquad \implies \{i:i,I\}$$

$$\implies \{I,I\} \qquad \implies \{i:i,C\}$$

$$\implies \{C,I\} \qquad \implies \{i:i,i:E\}$$

$$\implies \{i:E,I\} \qquad \implies \{i:i,i:i\}$$

(2) 消除左递归后的文法如下:

$$\begin{array}{ll} S & \rightarrow & \{I\} \\ I & \rightarrow & CI' \\ I' & \rightarrow & , II' \mid \varepsilon \\ C & \rightarrow & i : E \\ E & \rightarrow & i \mid S \end{array}$$

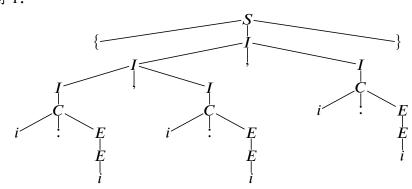
- (3) $\operatorname{First}(S) = \{\underline{\{}\}; \operatorname{First}(I) = \operatorname{First}(C) = \{i\}; \\ \operatorname{First}(I') = \{\underline{\epsilon}, \underline{,}\}; \operatorname{First}(E) = \{i, \underline{\{}\}. \\ \operatorname{Follow}(S) = \{\$, \underline{\}}, \underline{,}\}; \\ \operatorname{Follow}(I) = \operatorname{Follow}(I') = \operatorname{Follow}(C) = \operatorname{Follow}(E) = \{\underline{\}}, \underline{,}\}.$
- (4) LL(1) 分析表如下所示:

	i	{	}	,	:	\$
S		$S \rightarrow \{I\}$				
I	$I \rightarrow CI'$					
I'			$I' \to \varepsilon$	$I' \rightarrow , II' \mid \varepsilon$		
C	$C \rightarrow i : E$					
E	$E \rightarrow i$	$E \rightarrow S$				

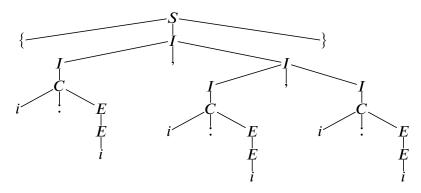
(5) 语句 " $\{i:i\}$ " 的分析过程如下所示:

剩余串	分析栈	分析动作
$\{i:i\}$ \$	<i>S</i> \$	$S \rightarrow \{I\}$
$\{i:i\}$ \$	$\{I\}$ \$	match-advance
i:i	I }\$	$I \rightarrow CI'$
i:i	CI' }\$	$C \rightarrow i : E$
i:i	i:EI'	match-advance
: <i>i</i> }\$	EI' }\$	match-advance
<i>i</i> }\$	EI' }\$	$E \rightarrow i$
<i>i</i> }\$	iI' }\$	match-advance
}\$	I' }\$	$I' \to \varepsilon$
}\$	}\$	match-advance
\$	\$	分析成功

三、 (1) 语句 " $\{a:a,a:a,a:a\}$ " 的两颗不同的语法树为: 语法树 1:



语法树 2:



(2) 无二义文法:

$$S \rightarrow \{I\}$$

$$I \rightarrow I, C \mid C$$

$$C \rightarrow i : E$$

$$E \rightarrow i \mid S$$

四、(1) 状态 I_1 的 LR(0) 项目集为

$$\overline{\{S \to \{\bullet I\}\}}$$

$$= \{S \to \{\bullet I\}, I \to \bullet I, I, I \to \bullet C, C \to \bullet i : E\}$$

- (2) 状态 I_1 所接受的终结符组成的活前缀即是从开始状态 I_0 到状态 I_1 所有可能的路经对应的边标记序列. 而经过终结符为状态转移边能到达状态 I_1 的只有状态 I_0 , I_1 , I_5 和 I_9 . 故对应的正则表达式为: $\{(i:\{)^*.$
- (3) Follow(*S*) = { \$, <u>}</u>, <u>,</u> }; Follow(*I*) = Follow(*C*) = Follow(*E*) = { <u>}</u>, <u>,</u> } SLR 分析表如下所示:

	action						goto			
状态	i	{	}	,	:	\$	S	I	C	\boldsymbol{E}
0		s1					4			
1	s5							3	2	
2			r3	r3						
3			s7	s6						
4						acc				
5					s9					
6	s5							8	2	
7			r1	r1		r1				
8			r2	r2						
9	s10	s1					12			11
10			r5	r5						
11			r4	r4						
12			r6	r6						

(4) 语句 " $\{i:i\}$ " 的分析过程如下所示:

剩余串	分析栈	分析动作
$\{i:i\}$ \$	0	shift
<i>i</i> : <i>i</i> }\$	0{1	shift
: <i>i</i> \$	0{1i5	shift
<i>i</i> }\$	0{1 <i>i</i> 5:9	shift
}\$	0{1 <i>i</i> 5:9 <i>i</i> <u>10</u>	reduce $E \rightarrow i$
}\$	0{1 <i>i</i> 5:9 <i>E</i> 11	reduce $C \rightarrow i : E$
}\$	0{1 <i>C</i> 2	reduce $I \rightarrow C$
}\$	0{1 <i>I</i> 3	shift
\$	0{1 <i>I</i> 3}7	reduce $S \to \{I\}$
\$	0.54	分析成功

五、(1)

产生式	语义规则
$S \to \{I\}$	S.xml = I.xml
$I \rightarrow I_1, I_2$	$I.xml = I_1.xml + I_2.xml$
$I \to C$	I.xml = C.xml
$C \rightarrow i:E$	C.xml = "<" + i.lexval + ">"+
	E.xml + " " + i.lexval + " "
$E \rightarrow i$	E.xml = i.lexval
$E \rightarrow S$	E.xml = S.xml

(2) 转换后的 XML 语言如下:

```
<glossary>
  <title>
    example glossary
  </title>
    <GlossDiv>
    <title>
    S
    </title>
    <GlossList>
        <GlossEntry>
        <ID>
            SGML
        </ID>
        <Abbrev>
            ISO 8879:1986
```

```
</Abbrev>
                 <GlossDef>
                   <para>
                     A meta-markup language.
                   </para>
                 </GlossDef>
                 <GlossSee>
                   markup
                 </GlossSee>
               </GlossEntry>
            </GlossList>
          </GlossDiv>
        </glossary>
六、
                                    x := t1
     L0: t0 := x + 1
                               x := t0
                                    ifnot (e>f) goto L0
                                 ifnot (g>h) goto L1 if (i>k) goto L0
         ifnot (a>b) goto L1 |
                                     if (i>k) goto L0
         if (c>d) goto L1
                               t1 := x + 2
                               | L1:
```

七、 声明为数组的形参 C 语言编译器会自动转换为指针,因此 sizeof(src) = 4. 所以 foo() 仅拷贝了 s 前 4 字符到 t,且没有拷贝,\0'.