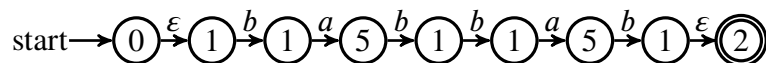


武汉大学计算机学院 2015-2016 学年第一学期
2013 级《编译原理》参考答案

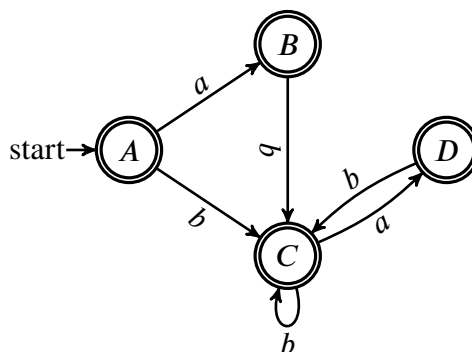
一、 (1)



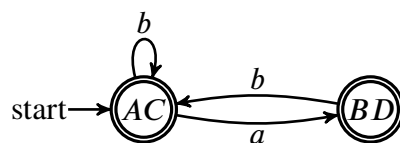
(2)

$$A = \{0, 1, 2, 4, 5\}, B = \{3, 4, 5\}, C = \{1, 2, 4, 5\}, D = \{4, 5\}.$$

状态转换图为：



(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.

$$(5) r = (ab \mid b)^*(a \mid \epsilon).$$

二、 (1) 最左推导如下：

$$\begin{array}{ll}
 S \xRightarrow{lm} \{I\} & \xRightarrow{lm} \{i : i, I\} \\
 \xRightarrow{lm} \{I, I\} & \xRightarrow{lm} \{i : i, C\} \\
 \xRightarrow{lm} \{C, I\} & \xRightarrow{lm} \{i : i, i : E\} \\
 \xRightarrow{lm} \{i : E, I\} & \xRightarrow{lm} \{i : i, i : i\}
 \end{array}$$

(2) 消除左递归后的文法如下：

$$\begin{array}{l}
 S \rightarrow \{I\} \\
 I \rightarrow CI' \\
 I' \rightarrow ,II' \mid \epsilon \\
 C \rightarrow i : E \\
 E \rightarrow i \mid S
 \end{array}$$

- (3) $\text{First}(S) = \{\{\}\}; \text{First}(I) = \text{First}(C) = \{i\};$
 $\text{First}(I') = \{\epsilon, ,\}; \text{First}(E) = \{i, \{\}\}.$
 $\text{Follow}(S) = \{\$, \}, _ \};$
 $\text{Follow}(I) = \text{Follow}(I') = \text{Follow}(C) = \text{Follow}(E) = \{\}, _, _ \}.$

(4) LL(1) 分析表如下所示:

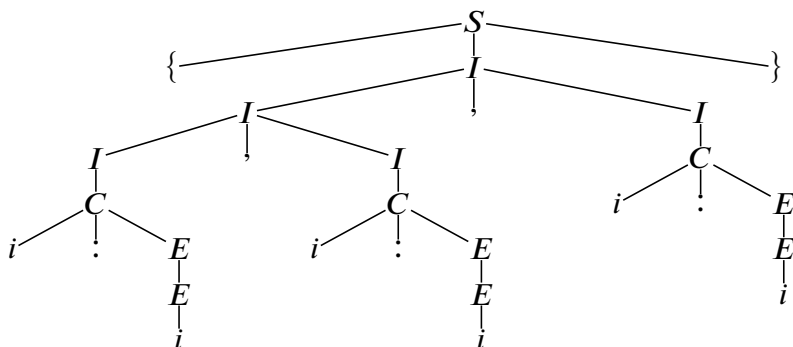
	i	$\{$	$\}$	$,$	$:$	$\$$
S		$S \rightarrow \{I\}$				
I	$I \rightarrow CI'$					
I'			$I' \rightarrow \epsilon$	$I' \rightarrow , II' \mid \epsilon$		
C	$C \rightarrow i : E$					
E	$E \rightarrow i$	$E \rightarrow S$				

(5) 语句 “ $\{i : i\}$ ” 的分析过程如下所示:

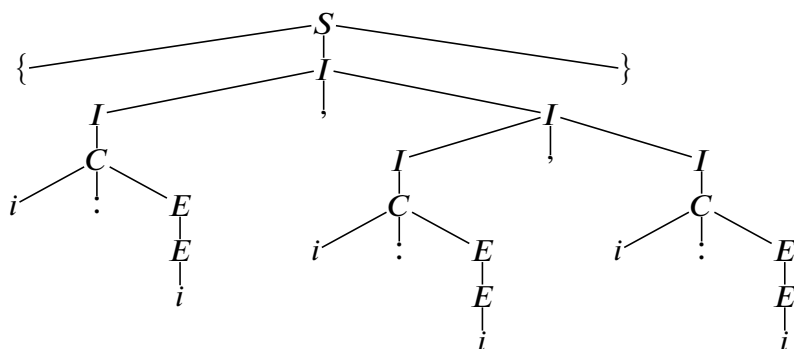
剩余串	分析栈	分析动作
$\{i : i\}\$$	$S\$$	$S \rightarrow \{I\}$
$\{i : i\}\$$	$\{I\}\$$	match-advance
$i : i\}\$$	$I\}\$$	$I \rightarrow CI'$
$i : i\}\$$	$CI'\}\$$	$C \rightarrow i : E$
$i : i\}\$$	$i : EI'\}\$$	match-advance
$: i\}\$$	$: EI'\}\$$	match-advance
$i\}\$$	$EI'\}\$$	$E \rightarrow i$
$i\}\$$	$iI'\}\$$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

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

语法树 1:



语法树 2:



(2) 无二义文法:

$$\begin{aligned}
 S &\rightarrow \{ I \} \\
 I &\rightarrow I, C \mid C \\
 C &\rightarrow i : E \\
 E &\rightarrow i \mid S
 \end{aligned}$$

四、 (1) 状态 I_1 的 LR(0) 项目集为

$$\begin{aligned}
 &\overline{\{ S \rightarrow \{ \bullet I \} \}} \\
 &= \{ S \rightarrow \{ \bullet I \}, I \rightarrow \bullet I, I, I \rightarrow \bullet C, C \rightarrow \bullet i : E \}
 \end{aligned}$$

(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 所有可能的路径对应的边标记序列. 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) $\text{Follow}(S) = \{ \$, \underline{\}, \underline{\} \}$;

$\text{Follow}(I) = \text{Follow}(C) = \text{Follow}(E) = \{ \underline{\}, \underline{\} \}$

SLR 分析表如下所示:

状态	action						goto			
	i	$\{$	$\}$	$,$	$:$	$\$$	S	I	C	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
$i:i\}\$$	0{1	shift
$:i\}\$$	0{1i5	shift
$i\}\$$	0{1i5:9	shift
$\}\$$	0{1i5:9i10	reduce $E \rightarrow i$
$\}\$$	0{1i5:9E11	reduce $C \rightarrow i:E$
$\}\$$	0{1C2	reduce $I \rightarrow C$
$\}\$$	0{1I3	shift
$\$$	0{1I3}7	reduce $S \rightarrow \{I\}$
$\$$	0S4	分析成功

五、 (1)

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

(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>
      </GlossEntry>
    </GlossList>
  </GlossDiv>
</glossary>

```

```

        </Abbrev>
        <GlossDef>
            <para>
                A meta-markup language.
            </para>
        </GlossDef>
        <GlossSee>
            markup
        </GlossSee>
    </GlossEntry>
</GlossList>
</GlossDiv>
</glossary>

```

六、

```

L0: t0 := x + 1      |      x := t1
    x := t0          |      ifnot (e>f) goto L0
    ifnot (a>b) goto L1 |      ifnot (g>h) goto L1
    if (c>d) goto L1  |      if (i>k) goto L0
    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'`.