武汉大学计算机学院 **2019 - 2020** 学年第一学期

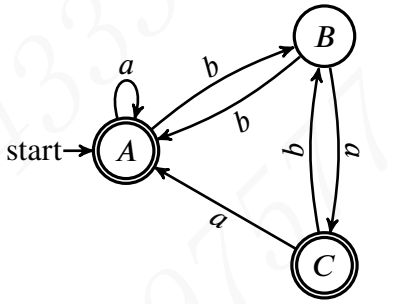
2017 级《编译原理》**(**期末考试参考答案 **A)**

一、 (1)

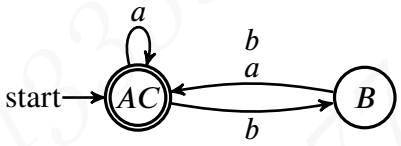
(2)

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

状态转换图为：



(3) 最小 DFA 如下所示：



(4) 不以奇数 b 结尾.

(5) $r = (bb \mid ba \mid a)^*$.

二、 (1) 语句“$\texttt{id}(\texttt{id},\texttt{id})$”的最左推导如下：

$$\begin{aligned} E &\underset{lm}{\Longrightarrow} E(A) &&\underset{lm}{\Longrightarrow} \texttt{id}(E,A)\\ &\underset{lm}{\Longrightarrow} \texttt{id}(A) &&\underset{lm}{\Longrightarrow} \texttt{id}(\texttt{id},A)\\ &\underset{lm}{\Longrightarrow} \texttt{id}(A,A) &&\underset{lm}{\Longrightarrow} \texttt{id}(\texttt{id},\texttt{id}) \end{aligned}$$

(2) 消除左递归后的文法如下：

$$\begin{aligned} E &\rightarrow (E)E' \mid \texttt{id}\,E'\\ E' &\rightarrow (A)E' \mid \varepsilon\\ A &\rightarrow E\,A'\\ A' &\rightarrow ,A\,A' \mid \varepsilon \end{aligned}$$

(3)

非终结符	First	Follow
E	id (	) , $
E'	ε (	) , $
A	id (	) ,
A'	ε ,	) ,

(4) LL(1) 分析表如下所示

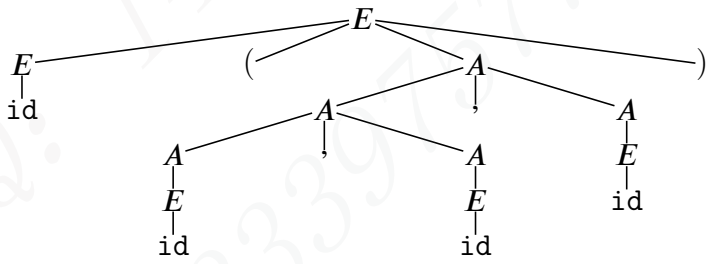
	id	(	)	,	$
E	$E \to \texttt{id}E'$	$E \to (E)E'$			
E'		$E' \to (A)E'$	$E' \to \varepsilon$	$E' \to \varepsilon$	$E' \to \varepsilon$
A	$A \to EA'$	$A \to EA'$			
A'			$A \to \varepsilon$	$A' \to \varepsilon \mid , AA'$	

(5) 语句“id(id)”的分析过程如下所示：

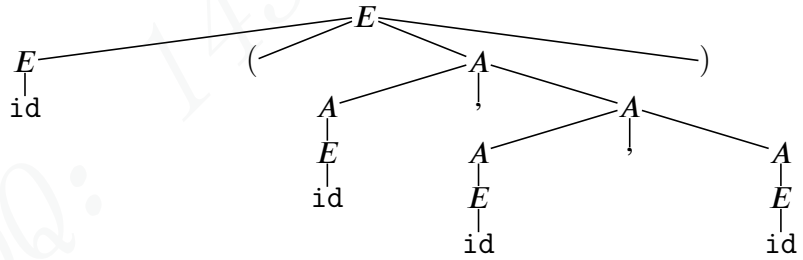
剩余串	分析栈	分析动作
id(id)$	$E$$	$E \to \texttt{id}E'$
id(id)$	id$E'$$	match-advance
(id)$	$E'$$	$E' \to (A)E'$
(id)$	$(A)E'$$	match-advance
id)$	$A)E'$$	$A \to EA'$
id)$	$EA')E'$$	$E \to \texttt{id}E'$
id)$	id$E'A')E'$$	match-advance
)$	$E'A')E'$$	$E' \to \varepsilon$
)$	$A')E'$$	$A' \to \varepsilon$
)$	$)E'$$	match-advance
$	$)E'$$	$E' \to \varepsilon$
$	$	分析成功

三、 (1) 语句“id(id, id, id)”的两棵不同的语法树为：

语法树 1：



语法树 2：



(2) 无二义文法：

$$
\begin{aligned}
E &\to E(A) \mid (E) \mid \texttt{id} \\
A &\to A, E \mid E
\end{aligned}
$$

四、 (1) 状态 I_5 的 LR(0) 项目集为

$$\begin{aligned}&\overline{\{E \to E(\bullet A)\}}\\ =&\{E \to E(\bullet A), A \to \bullet A, A, A \to \bullet E, E \to \bullet(E), E \to \bullet E(A), E \to \bullet \texttt{id}\}.\end{aligned}$$

(2) 识别活前缀的自动在吃进 $E(A,$ 进入状态 I_{10}, 状态 I_{10} 还能接受形成句柄的终结符号串为：$(^*\texttt{id}.$

(3) Follow(E) = {(,), , , \$}，Follow(A) = {), , }. 状态 I_{11} 面对“,”有移进/归约冲突，左结合选归约. 分析表如下所示：

	action					goto	
状态	id	(	)	,	$	E	A
0	s3	s1				2	
1	s3	s1				4	
2		s5			acc		
3		r3	r3	r3	r3		
4		s5	s6				
5	s3	s1				8	7
6		r2	r2	r2	r2		
7			s9	s10			
8		s5	r5	r5			
9		r1	r1	r1	r1		
10	s3	s1				8	11
11			r4	**r4**			

(4) 语句“id(id)”的分析过程如下所示：

剩余串	分析栈	分析动作
id(id)\$	0	shift
(id)\$	0id3	reduce $E \to$ id
(id)\$	0E2	shift
id)\$	0E2(5	shift
)\$	0E2(5id3	reduce $E \to$ id
)\$	0E2(5E8	reduce $A \to E$
)\$	0E2(5A7	shift
\$	0E2(5A7)9	reduce $E \to E(A)$
\$	0E2	reduce 分析成功

五、 (1)

产生式	语义规则
$E \to E_1(A)$	E.kind = OPEN lp = " "; rp = "" **if** (A.kind == OPEN) **then** lp = " ("; rp = ")" E.curry = E_1.curry + lp + A.curry + rp
$E \to (E_1)$	E.kind = E_1.kind E.curry = E_1.curry
$E \to$ id	E.kind = CLOSE E.curry = id.lexeme

$A \to A_1, A_2$	A.kind = CLOSE
	lp1 = ""; rp1 = " "
	lp2 = ""; rp2 = ""
	if (A_1.kind == OPEN) **then**
	  lp1 = "("; rp1 = ") "
	if (A_2.kind == OPEN) **then**
	  lp2 = "("; rp2 = ")"
	A.curry = lp1 + A_1.curry + rp1 + lp2 + A_2.curry + rp2

$A \to E$	A.kind = E.kind A.curry = E.curry

(2) `a (b (c d e) f) g (h i j)`

六、

```
L1: ifnot (a > b) goto L2 | L0: ifnot (i > k) goto L1
    if (c > d) goto L2    |     t0 := y + 2
    x := x + 1            |     x := to
    ifnot (e > f) goto L0 |     goto L1
    if (g > h) goto L1    | L2:
```

七、 `s` in the expression is in fact of the type pointer to array. so the value of `*s` is the address of the first element of the array. but if casting `s` as (`(char **)`), `*s` will take the value of the address of the first element of the array (`'H'`). so after applying second dereferencing (`**s`), it will cause memory access violation. if we change `s` to array of pointer (`char *s[]`), it will work correctly.

simply: diff of pointer to array and array of pointer.