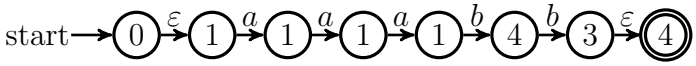
武汉大学计算机学院2013-2014学年第一学期
2011级《编译原理》参考答案

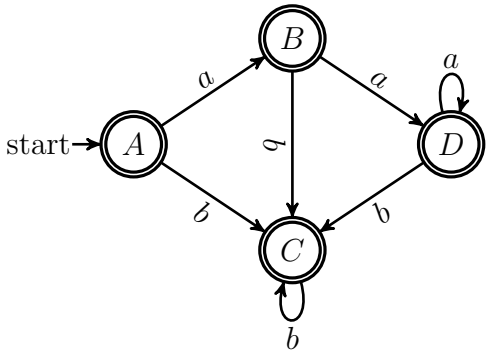
一、 (1)



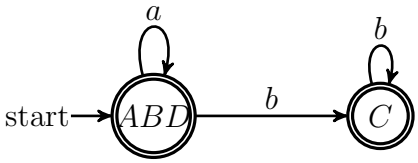
(2)

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

状态转换图为：



(3) 最小DFA如下所示：



(4) 零个或多个连续的a连接零个或多个连续的b.

(5) $r = a^*b^*$.

二、 (1) 最左推导如下：

$$\begin{aligned}
I &\underset{lm}{\Longrightarrow} \{L\} \\
&\underset{lm}{\Longrightarrow} \{L, L\} \\
&\underset{lm}{\Longrightarrow} \{I, L\} \\
&\underset{lm}{\Longrightarrow} \{\{L\}, L\} \\
&\underset{lm}{\Longrightarrow} \{\{I\}, L\} \\
&\underset{lm}{\Longrightarrow} \{\{n\}, L\} \\
&\underset{lm}{\Longrightarrow} \{\{n\}, I\} \\
&\underset{lm}{\Longrightarrow} \{\{n\}, n\}
\end{aligned}$$

(2) 消除左递归后的文法如下：

$$\begin{aligned}
I &\rightarrow \{L\} \mid n \\
L &\rightarrow IL' \\
L' &\rightarrow , \ LL' \mid \varepsilon
\end{aligned}$$

(3) First(I) = First(L) = { {, n }; First(L') == { ,, ε }.
Follow(I) = { }, ,, \$ }; Follow(L) = Follow(L') = { }, , }.

(4) LL(1)分析表如下所示：

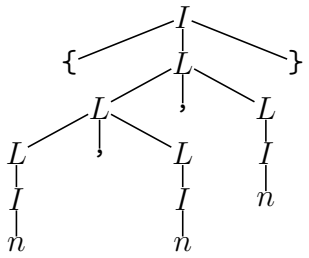
	,	n	{	}	\$
I		$I \to n$	$I \to \{L\}$		
L		$L \to IL'$	$L \to IL'$		
L'	$L' \to ,LL'$, $L \to \varepsilon$			$L' \to \varepsilon$	

(5) 语句“{{n},n}”的分析过程如下所示：

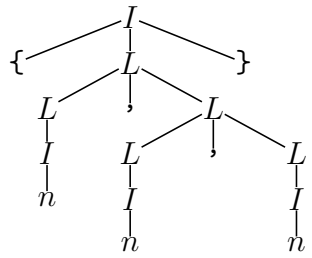
剩余串	分析栈	分析动作
{{n},n}\$	I\$	$I \to \{L\}$
{{n},n}\$	{L}\$	match-advance
{n},n}\$	L}\$	$L \to IL'$
{n},n}\$	IL'}\$	$I \to \{L\}$
{n},n}\$	{L}L'}\$	match-advance
n},n}\$	L}L'}\$	$L \to IL'$
n},n}\$	IL'}L'}\$	$I \to n$
n},n}\$	nL'}L'}\$	match-advance
},n}\$	L'}L'}\$	$L' \to \varepsilon$
},n}\$	}L'}\$	match-advance
,n}\$	L'}\$	$L' \to ,LL'$
,n}\$	,LL'}\$	match-advance
n}\$	LL'}\$	$L \to IL'$
n}\$	$IL'L'$}\$	$L \to IL'$
n}\$	$IL'L'$}\$	$I \to n$
n}\$	$nL'L'$}\$	match-advance
}\$	$L'L'$}\$	$L' \to \varepsilon$
}\$	L'}\$	$L' \to \varepsilon$
}\$	}\$	match-avdance
\$	\$	分析成功

三、 (1) 语句“{n,n,n}”的两颗不同的语法树为：

语法树1：



语法树2：



(2) 无二义文法：

$$\begin{aligned} I &\rightarrow \{L\} \mid n \\ L &\rightarrow L,I \mid I \end{aligned}$$

四、 (1) 状态I_3的LR(0)项目集为

$$\begin{aligned} &\overline{\{\, I \rightarrow \{ \bullet L\}\,\}} \\ =&\{I \rightarrow \{ \bullet L\}, L \rightarrow \bullet L,L, L \rightarrow \bullet I, I \rightarrow \bullet\{L\}, I \rightarrow \bullet n\} \end{aligned}$$

(2) $\{^*(n|\varepsilon)$.

(3) SLR分析表如下所示：

	action					goto	
状态	n	,	{	}	$	I	L
0	s2		s3			1	
1					acc		
2		r2		r2	r2		
3	s2		s3			4	5
4		r4		r4			
5		s6		s7			
6	s2		s3			4	8
7		r1		r1	r1		
8		**r3**		r3			

(4) 语句“{n,n,n}”的分析过程如下所示：

剩余串	分析栈	分析动作
{n,n,n}$	0	shift
n,n,n}$	0{3	shift
,n,n}$	0{3n2	reduce $I \rightarrow n$
,n,n}$	0{3I4	reduce $L \rightarrow I$
,n,n}$	0{3L5	shift
n,n}$	0{3L5,6	shift
,n}$	0{3L5,6n2	reduce $I \rightarrow n$
,n}$	0{3L5,6I4	reduce $L \rightarrow L$
,n}$	0{3L5,6L8	reduce $L \rightarrow L,L$
,n}$	0{3L5	shift
n}$	0{3L5,6	shift
}$	0{3L5,6n2	reduce $I \rightarrow n$
}$	0{3L5,6I4	reduce $L \rightarrow I$
}$	0{3L5,6L8	reduce $L \rightarrow L,L$

}$	0{3L5	shift
$	0{3L5}7	reduce $I \to \{L\}$
$	0I1	分析成功

五、 (1)

产生式	语义规则
$I' \to MI$	I.level = 0, I.limit = 1
$M \to \varepsilon$	count = 0

$I \to \{LN\}$	**if** count % `getsize`(I.level) $\neq$ 0 **then**
	`error("the left brace is not in right pos")`
	if `getsize`(I.level) == 1 **then**
	`error("the brace level is more than array dim")`
	L.level = I.level + 1
	L.limit = `getsize`(I.level) + count
	N.limit = `getsize`(I.level) + count

$I \to n$	**if** I.level == 0 **then**
	`error("initializer is not in braces")`
	if count $\geqslant$ I.limit **then**
	`error("the array up bound exceeds")`
	`emit(getname() + "[" +` 4 $*$ count `+ "]=" +` n.val`)`
	count = count + 1

$N \to \varepsilon$	**while** count < N.limit `emit(getname() + "[" +` 4 $*$ count `+ "]=" +` 0`)` count = count + 1
$L \to L_1, L_2$	L_1.level = L.level, L_2.level = L.level L_1.limit = L.limit, L_2.limit = L.limit
$L \to I$	I.level = L.level, I.limit = L.limit

(2) `a[0]= 1; a[32]= 2; [36]= 3; a[64]= 4; a[80]= 5; a[84]= 6;`

六、

```
L1: if (a > b) goto L2     |     ifnot (m > n) goto L2
    ifnot (c > d) goto L2  |     t1 := y + 2
    ifnot (e > f) goto L0  |     y := t1
    if (i > j) goto L1     |     goto L1
L0: t0 := x + 1            | L2:
    x := t0
```

七、 实参反向压栈，即函数`main()`中调用`printf`时，首先将最后一个实参`count`压栈，这时其值为`0`. 再调用`fac(5)`, 并将函数调用的结果`120`压栈. 这时虽然`count`修改为`6`, 但是栈中所传的实参还是`0`.