

## 06-07 级样卷答案 by42

一、（2007） by OnionYST

1.

scope	symbol
Global	Computer, Mac, Main
Computer.Class	cpu, Crash
Computer.Crash.Param	@this, numTimes
Computer.Crash.Local	i

2.

powerbook

offset	content
0	vtable@Mac
4	cpu
8	mouse

vtable@Mac

offset	content
0	vtable@Computer
4	@ClassName
8	Crash@Mac

3.

```
d[i] = *(base + 32 + 2 * i)
```

c 和 d 捆绑销售

编者按：按照课件的说法，第 2 小问 Mac 的 VTable 可以采用更详细的描述：

Offset	content
0	指向 class Computer 的 VTable 的指针
4	指向 class Mac 类名字串的指针
8	class Mac 的 Crash 函数代码入口指针

第 3 小问“c 和 d 捆绑销售”具体指，将图中的 c 也改成 d 的形式，且在最上面的动态数组区域采用 c[0]、d[0]、c[1]、d[1]……的分配方法。如下表所示：

栈中元素	Offset
栈顶	11+4N
...	...
d[1]	17
c[1]	15
d[0]	13
c[0]	11
e	9
指向 d 空间的指针	8
指向 c 空间的指针	7
c、d 的上界（N）	6
b	4
a	3
控制信息	0

一、（2006）by OnionYST

1.

a, b, p, q, q.x

2.

id	content
18	0
19	13
21	q.x
22	0
23	18

3.

静态作用域：1

动态作用域：0

二、by 栋哥

1.

句型	短语	直接短语	句柄
aAbc	aA、b、aAbc	aA、b	aA

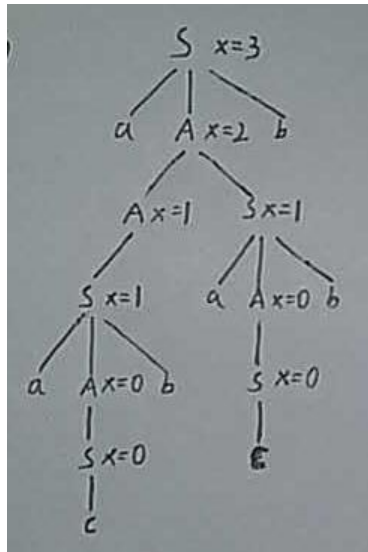
2.

G 中的规则 r	First (rhs (r))	Follow (lhs (r))	PS (r)
$S \rightarrow AB$	a,c	#	a,c
$A \rightarrow \varepsilon$	$\varepsilon$	b,c,#	b,c,#
$A \rightarrow aAbB$	a	b,c,#	a
$B \rightarrow cA$	c	b,c,#	c

$PS(A \rightarrow \varepsilon) \cap PS(A \rightarrow aAbB) = \emptyset$ , 故是 LL (1) 文法。

三、by 栋哥

1.(1)



- (2)  $S \rightarrow aAb \{v[top-2].x := v[top-1].x + 1\}$   
 $S \rightarrow c \quad \{v[top].x := 0\}$   
 $A \rightarrow A_1S \{v[top-1].x := v[top-1].x + v[top].x\}$   
 $A \rightarrow S \quad \{v[top].x := v[top].x\}$

2. Lecture 09 第 9 题

参考解答:

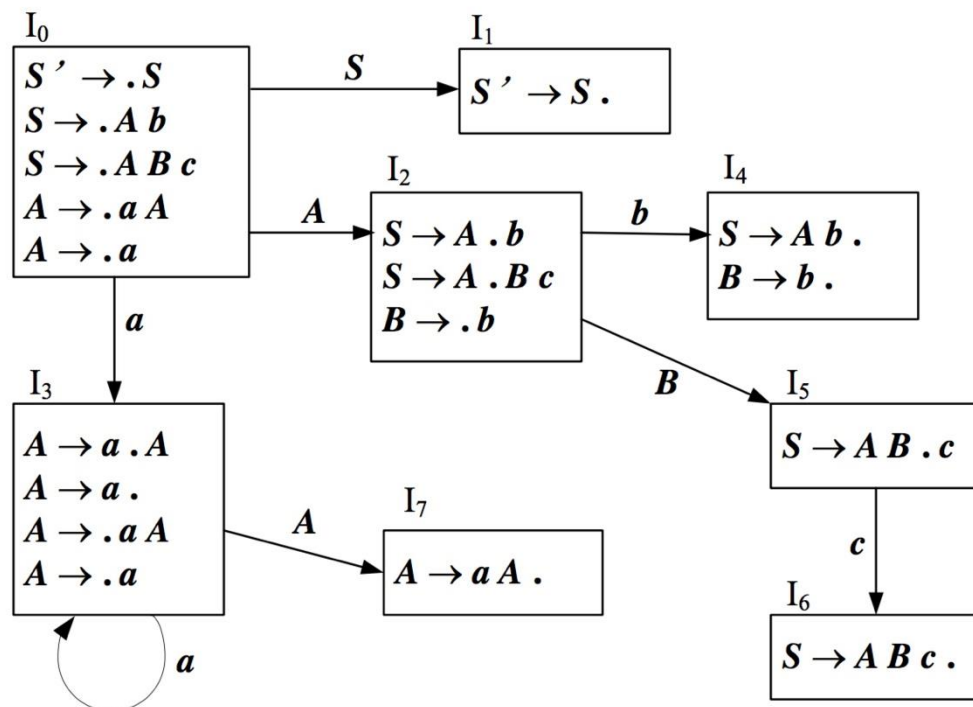
$D \rightarrow D_1 ; T \{ L.type := T.type; L.offset := D_1.width; L.width := T.width \} L$   
 $\{ D.width := D_1.width + L.num \times T.width \}$   
 $D \rightarrow MNT \{ L.type := T.type; L.offset := M.s; L.width := T.width \} L$   
 $\{ D.width := L.num \times T.width \}$   
 $T \rightarrow \underline{integer} \quad \{ T.type := int; T.width := 4 \}$   
 $T \rightarrow \underline{real} \quad \{ T.type := real; T.width := 8 \}$   
 $L \rightarrow \{ L_1.type := L.type; L_1.offset := L.offset; L_1.width := L.width; \} L_1, \underline{id}$   
 $\{ enter(\underline{id.name}, L.type, L.offset + L_1.num \times L.width); L.num := L_1.num + 1 \}$   
 $L \rightarrow \underline{id} \quad \{ enter(\underline{id.name}, L.type, L.offset); L.num := 1 \}$   
 $M \rightarrow \varepsilon \quad \{ M.s := 0 \}$   
 $N \rightarrow \varepsilon \quad \{ \}$

#### 四、by 鑫腿

```
1  void ParseS(){
2      F_val := ParseF();
3      P_i := F_val;
4      P_s := ParseP(P_i);
5      print(P_s);
6  }
7
8  int ParseP(int P_i){
9      switch (lookahead){
10         case '+':
11             MatchToken('+');
12             F_val = ParseF();
13             P1_i = P_i + F_val;
14             P1_s = ParseP(P1_i);
15             P_s = P1_s;
16             break;
17
18         case '#':
19             P_s = P_i;
20             break;
21
22         default:
23             printf("syntax error \n");
24             exit(0);
25     }
26     return P_s;
27 }
28
29 int ParseF(){
30     if(lookahead == 'a'){
31         MatchToken('a');
32         F_val = 1;
33         return F_val;
34     }else{
35         printf("syntax error \n");
36         exit(0);
37     }
38 }
```

五、by 鑫腿

1.



2. I<sub>3</sub> 存在移进-归约冲突

I<sub>4</sub> 存在归约-归约冲突

3. 使用 SLR(1) 分析方法, 发现该状态机的所有状态皆没有冲突, 因而可知该文法是 SLR(1) 文法 (可计算出 S、A、B 的 Follow 集合加以说明)。

## 六、Lecture06 例 11, 21 页。by 翔班

解 为文法  $G[E]$  增加产生式  $S \rightarrow E$ , 得到增广文法  $G'[S]$ 。根据 3.2.3 节介绍的方法, 我们得到该文法的 LR(0) 有限状态机, 如图19所示。

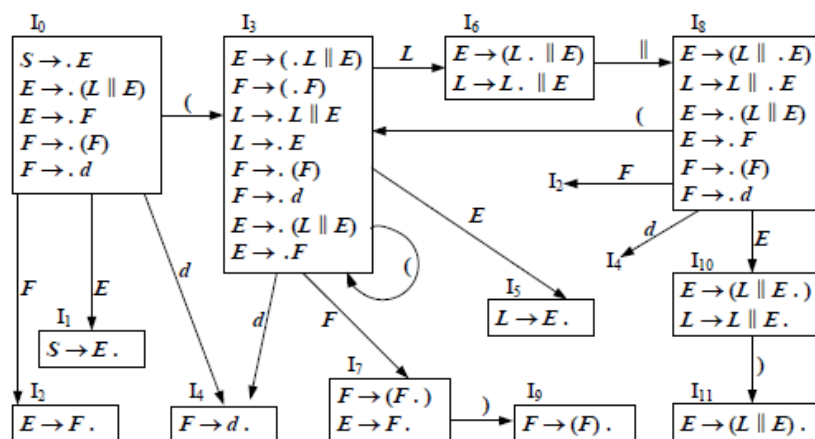


图 19 一个非 SLR(1) 文法的 LR(0) 有限状态机

可以验证, 图 19 中的 LR(0) 有限状态机中, 状态  $I_7$  和  $I_{10}$  存在移进-归约冲突, 因此文法  $G[E]$  不是 LR(0) 文法。

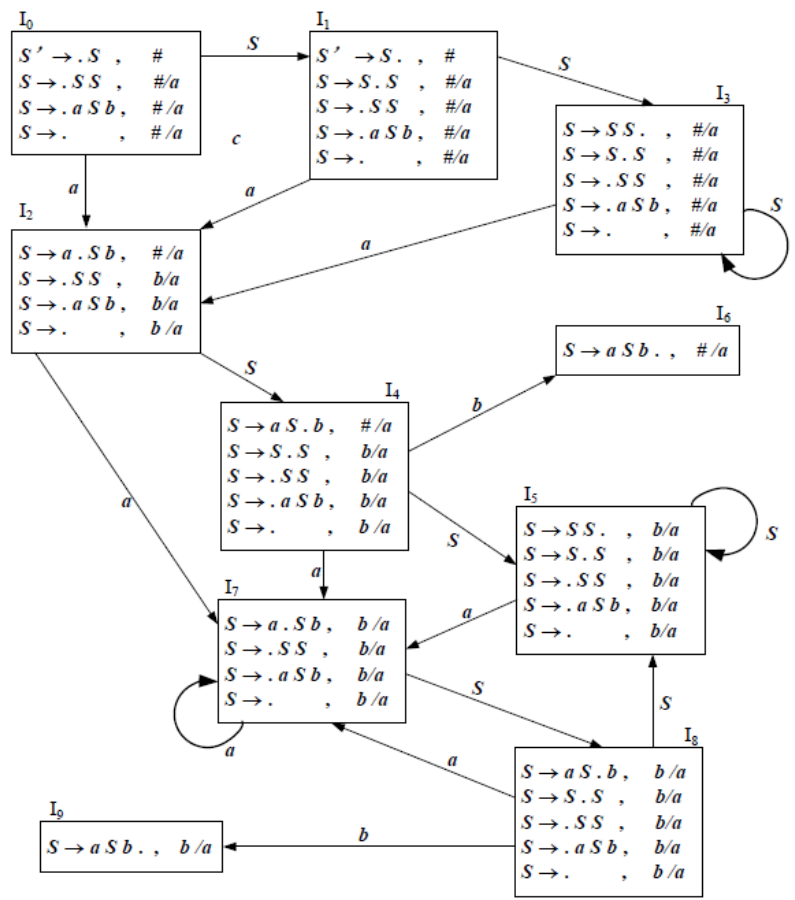
计算  $G[E]$  中非终结符的 Follow 集合, 我们得到:  $\text{Follow}(E) = \{ \}, \parallel, \#$ ,  $\text{Follow}(L) = \{ \parallel \}$ 。可以看出,  $) \notin \text{Follow}(L)$ , 所以状态  $I_{10}$  的移进-归约冲突可以通过 SLR(1) 方法解决。但, 因为有  $) \in \text{Follow}(E)$ , 所以状态  $I_7$  的移进-归约冲突不可用这种方法解决。因此, 我们得知, 文法  $G[E]$  不是 SLR(1) 文法, SLR(1) 分析方法不再适用。

我们来分析一下, 当到达状态  $I_7$  时, 分析栈顶部是 ' $(F$ ' (栈顶为 ' $F$ '). 如果将句柄  $F$  归约为  $E$ , 那么根据分析过程, 下一状态会进入状态  $I_5$ ; 接着将句柄  $E$  归约为  $L$ , 进入状态  $I_6$ ; 此时就会看到, 期望的下一个输入符号将是  $\parallel$ 。这样, 我们就得到一个解决状态  $I_7$  中移进-归约冲突的方案: 遇  $)$  时, 移进; 遇  $\parallel$  时, 归约; 遇其它符号时, 出错处理。

由此可知, 当到达状态  $I_7$  时, 句柄  $F$  所期望的下一个输入符号实际上是  $\parallel$ , 而不是  $)$ 。这两个符号都属于  $\text{Follow}(E)$ , 所以 SLR(1) 分析方法失去了作用。在这一小节里, 我们介绍 LR(1) 分析方法, 可以弥补这一不足。

由橙色标记部分的分析, 需要观察的状态为  $I_3$ 、 $I_5$ 、 $I_6$ 。

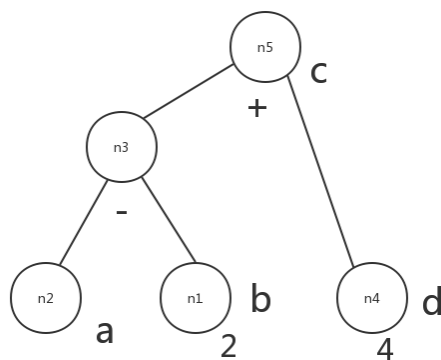
七、Lecture06 作业第 6 题。by 翔班



(b) 该文法不是 LR(1) 文法。存在“移进/归约冲突”的状态： $I_0, I_1, I_2, I_3, I_4, I_5, I_7, I_8$

存在“归约/归约冲突”的状态： $I_3, I_5$

八、by 姚远



1. (n2 下面添一个  $a_0$  比较好~)

2.

基本块	LiveUse	Def	LiveOut	LiveIn
B1	$\Phi$	$\{a\}$	$\{a\}$	$\Phi$
B2	$\{a\}$	$\{b, c, d, e\}$	$\{a, c, d\}$	$\{a\}$
B3	$\{a, d\}$	$\Phi$	$\{a, c, d\}$	$\{a, c, d\}$

B4	{a,c}	{e}	{a}	{a,c}
----	-------	-----	-----	-------

3. a, b, d

4. {(6), (7)}

九、by 姚远

Lecture10

6

$$\begin{aligned}
 E &\rightarrow \Delta ( E_1, M_1 E_2, M_2 E_3 ) \\
 &\{ \text{backpatch}(E_1.\text{falselist}, M_1.\text{gotostm}) ; \\
 &\quad \text{backpatch}(E_2.\text{truelist}, M_2.\text{gotostm}) ; \\
 &\quad E.\text{truelist} := E_3.\text{falselist} ; \\
 &\quad E.\text{falselist} := \text{merge}(\text{merge}(E_1.\text{truelist}, E_2.\text{falselist}), E_3.\text{truelist}) \}
 \end{aligned}$$

9(b) 好像最后一个应该是  $G.\text{beginlist} := \text{merge}(S.\text{nextlist}, N.\text{nextlist}) ;$

$$\begin{aligned}
 S &\rightarrow \text{do } M \text{ G od} \\
 &\{ \\
 &\quad \text{backpatch}(G.\text{beginlist}, M.\text{gotostm}) ; \\
 &\quad S.\text{nextlist} := G.\text{nextlist}; \\
 &\}
 \end{aligned}$$

$$\begin{aligned}
 G &\rightarrow E : M_1 S N \square M_2 G_1 \\
 &\{ \\
 &\quad \text{backpatch}(E.\text{truelist}, M_1.\text{gotostm}) ; \\
 &\quad \text{backpatch}(E.\text{falselist}, M_2.\text{gotostm}) ; \\
 &\quad G.\text{beginlist} := \text{merge}(S.\text{nextlist}, \text{merge}(N.\text{nextlist}, G_1.\text{beginlist})) \\
 &\quad G.\text{nextlist} := G_1.\text{nextlist}; \\
 &\}
 \end{aligned}$$

$$\begin{aligned}
 G &\rightarrow E : M S N \\
 &\{ \\
 &\quad \text{backpatch}(E.\text{truelist}, M.\text{gotostm}) ; \\
 &\quad G.\text{beginlist} := N.\text{nextlist} ; \\
 &\quad G.\text{nextlist} := E.\text{falselist} ; \\
 &\}
 \end{aligned}$$