作业3

1、假设文法如右所示

S -> a | ^ | (T)

T -> T, S | S

（1）给出(a, (a, a))的最左推导

（2）给出(((a, a), ^, (a)), a)的最左推导

作业3

2、假设文法如右所示

(1) 是否是LL(1)文法？如果不是改为LL(1)文法

(2) 计算First集

(3) 计算Follow集

(4) 计算First_S集合

(5) LL(1)分析表

```
E -> E + T
   | T
T  -> T * F
   | F
F -> i
   | (E)
```

作业3

3、假设文法如右所示

(1) 计算First集

(2) 计算Follow集

(3) 计算First_S集合

(4) LL(1)分析表

(5) 写出对输入串aaabd进行语法分析时，分析栈的主要变化过程

S -> aH

H -> aMd

| d

M -> Ab

|

A -> aM

| e

作业3

4、假设文法如右所示

（1）改写该文法，成为LL(1)文法

（2）改写该文法，改写后的文法仍不是LL(1)文法

S -> Aa
| b
A -> SB
B -> ab

作业3

5、假设文法如右所示

（1）构造递归下降分析程序

E -> [F]G

G -> E | ε

F -> aH

H -> aH | ε

1. (1) $S \to (T)$
$\to (T, S)$
$\to (S, S)$
$\to (a, S)$
$\to (a, (T))$
$\to (a, (T, S))$
$\to (a, (S, S))$
$\to (a, (a, S))$
$\to (a, (a, a))$.

(2) $S \to (T)$
$\to (T, S)$
$\to (S, S)$
$\to ((T), S)$
$\to ((T, S), S)$
$\to ((T, S, S), S)$
$\to ((S, S, S), S)$
$\to (((T, S), S, S), S)$
$\to (((S, S), S, S), S)$
$\to (((a, S), S, S), S)$
$\to (((a, a), S, S), S)$
$\to (((a, a), \wedge, S), S)$
$\to (((a, a), \wedge, (T)), S)$
$\to (((a, a), \wedge, (S)), S)$
$\to (((a, a), \wedge, (a)), S)$
$\to (((a, a), \wedge, (a)), a)$.

2. (1) 得LL(1)文法，存在左递归
改为LL(1)文法：

$E \to TE'$
$E' \to +TE' \mid \varepsilon$
$T \to FT'$
$T' \to *FT' \mid \varepsilon$
$F \to i \mid (E)$.

(2) First集：

	0	1	2	3
E	{}	{}	{}	{i, (}
E'	{}	{+}	{+}	{+}
T	{}	{}	{i, (}	{i, (}
T'	{}	{*}	{*}	{*}
F	{}	{i, (}	{i, (}	{i, (}.

(3) Follow集

	0	1	2
E	{$}	{), $}	{), $}
E'	{}	{$}	{), $}
T	{}	{+, $}	{+,), $}
T'	{}	{+, $}	{+,), $}
F	{}	{*, +, $}	{*, +,), $}

(4) First_S:

1. E→TE'	{i, (}
2. E'→+TE'	{+}
3. E'→ε	{), $}
4. T→FT'	{i, (}
5. T'→*FT'	{*}
6. T'→ε	{+,), $}
7. F→i	{i}
8. F→(E)	{(}

(5) LL(1)分析表：

	i	(	+	)	*	$
E	1	1				
E'			2	3		3
T	4	4				
T'			6	6	5	6
F	7	8				

3. (1) First:

	0	1	2
S	{}	{a}	{a}
H	{}	{a,d}	{a,d}
M	{}	{}	{a,e}
A	{}	{a,e}	{a,e}

(2) Follow:

	0	1
S	{$}	{$}
H	{}	{$}
M	{}	{d,b}
A	{}	{b}

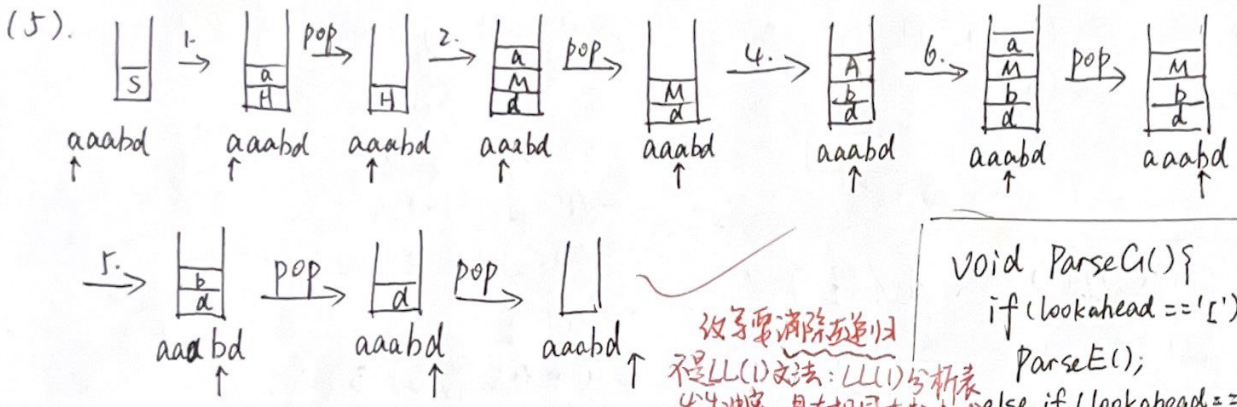
(3) First_S:

1. S→aH {a}
2. H→aMd {a}
3. H→d {d}
4. M→Ab {a,e}
5. M→{} {d,b}
6. A→M {a}
7. A→e {e}

(4) LL(1)分析表.

	a	b	c	d	e
S	1				
H	2			3	
M	4	5		5	4
A	6				7

(5).



4. (1) 消除间接左递归:

① S→SBa | b
B→ab

② S→bS'
S'→BaS' | ε
B→ab

改写消除左递归

不是LL(1)文法：LL(1)分析表发生冲突，具有相同左部为式

(2). 的First_S集交集不为空.

S→Aa|b S→SBa | b
A→bA'
A'→aBA'|ε B→ab
B→ab

5. First Sets:

	0	1
E	{}	{[}
G	{}	{[}
F	{}	{a}
H	{}	{a}

Follow Sets:

	0	1
E	{$}	{$}
G	{}	{$}
F	{}	{]}
H	{}	{}

Select Sets:

1. E→[F]G {[}
2. G→E {I}
3. G→ε {$}
4. F→aH {a}
5. H→aH {a}
6. H→ε {]}

应定义MatchToken

递归下降分析程序:

```
void ParseE(){
  switch(lookahead) {
    case '[':
      ParseF();
      MatchToken(']');
      ParseG();
      break;
    default:
      printf("Syntax Error\n");
      exit(0);
  }
}
```

A+

```
void ParseG(){
  if(lookahead=='[')
    ParseE();
  else if(lookahead==$);
  else{
    printf("SyntaxError\n");
    exit(0);
  }
}
```

```
void ParseF(){
  if(lookahead=='a')
    ParseH();
  else{
    printf("SyntaxError\n");
    exit(0);
  }
}
```

```
void ParseH(){
  if(lookahead=='a') ParseH();
  else if(lookahead==']') ;
  else{
    printf("SyntaxError\n");
    exit(0);
  }
}
```