## 第四章作业

### 练习4.2.1

1) S=>SS\*=>SS+S\*=>aS+S\*=>aa+S\*=>aa+a\*

2) S=>SS\*=>Sa\*=>SS+a\*=>Sa+a\*=>aa+a\*

3)



4) 没有二义性。

（1）先证明一个该文法产生串的长度的结论：设串的推导过程中使用产生式S=>SS+和S=>SS\*的次数为m，则串的长度L=2\*m+1，且串中包含m个运算符(+或\*)和(m+1)个a。

1）当m=0时，仅有S=>a一种情况，L=1，串由1个a和0个运算符构成，结论成立；

2）设当m<k(k>=1)时结论成立，则当m=k时，第一步推导必然为

S=>S1S2*op*

op为+或\*，S下标仅用于区分S的多次出现。设S1α，S2β，α、β均为使用S=>S1S2*op*少于k次得到的串，设二者推导过程中分别使用该产生式k1和k2次，根据假设有：

L(α)=2\*k1+1，L(β)=2\*k2+1

串长度L= L(α)+ L(β)+1=2\*(k1+k2+1)+1=2\*k+1；且串中a的个数为(k1+1)+(k2+1)=k+1；运算符个数为k1+k2+1=k，故结论成立。

（2）下面证明该文法无二义性，对串的长度做归纳。由前述证明可知，该文法产生的串长L可为任意非负奇数。对由该文法得到的长度为L=2\*k+1串**ω**：

1）当k=0时，L=2\*0+1=1，只有S=>a一种情况，显然没有二义性。

2）设当k<n时结论成立。S**ω，**根据**ω**末尾运算符可确定第一步推导使用的产生式，不妨设为:

S=>S1S2+

从后向前处理串**ω**，除去末尾的运算符，找到可以由S推导出的最短的串α，设α长度为m1，由前述结论可知m1=2\*k1+1，且α包含k1个运算符与(k1+1)个a，由归纳假设可知α无二义性，存在唯一的最左推导S α；

设串**ω**剩余部分为β，设β长度为m2，同理可知m2=2\*k2+1，β包含k2个运算符与(k2+1)个a，存在唯一最左推导S β，且满足k=k1+k2。

此时串ω可表示成如下形式：

**ω=**βα+

故存在唯一的最左推导：

S SS+ βS+ βα+

此时仍不存在二义性。

综上所述，该文法不具有二义性。

5) 由字符a与运算符+、\*构成的后缀表达式。

### 练习4.2.3

S->01S | 1S | ε

### 练习4.3.1

1) 该文法无左公因子

2) 不能，因此有左递归存在

3) rexpr->rterm rexpr'

rexpr'->+rterm rexpr' | ε

rterm->rfactor rterm'

rterm'->rfactor rterm'| ε

rfactor->rprimary rfactor'

rfactor'->\* rfactor'| ε

rprimary->a | b

4) 适合

### 1.

即使当非终结符用某个产生式匹配成功，但是这种成功可能只是暂时的，因为没有足够的信息来唯一地确定可能的产生式，所以分析过程就会产生回溯。

不可以。例如对于产生式A=>α | β, FIRST(α)与FIRST（β）交集为空集，但ε是其中某个FIRST集合的元素，不是一般性，假设ε∈FIRST(α)，想要避免回溯，则还需要考虑FOLLOW(A)与FIRST(β)的情况

### 2.

**a)消除左递归**

lexp → atom | list

atom → number | identifier

list → ( lexp-seq )

lexp-seq → lexp lexp-seq'

lexp-seq' → lexp lexp-seq' | ε

**b) 求该文法的FIRST集合和FOLLOW集合**

FIRST(lexp)={ number, identifier, ( }

FIRST (atom)={ number, identifier }

FIRST(list)={ ( }

FIRST(lexp-seq)={ number, identifier, ( }

FIRST(lexp-seq’)={ ε, number, identifier, ( }

FOLLOW(lexp)={ $, ), number, identifier, ( }

FOLLOW (atom)={ $, ), number, identifier, ( }

FOLLOW (list)={ $, ), number, identifier, ( }

FOLLOW (lexp-seq)={ ) }

FOLLOW (lexp-seq’)={ ) }

**c) 说明所得的文法是LL(1)文法**

可以根据LL(1)文法的定义来证明

因为FIRST (atom) ∩FIRST (list) = φ，且FIRST (lexp lexpseq') ∩FIRST (ε) = FIRST (lexp) ∩FIRST (ε) = φ,且 FIRST (lexp-seq') ∩ FOLLOW(lexp-seq') = φ，所以该文法是LL(1)文法

**d) 为所得得文法构造LL(1)分析表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **非终结符** | **输入符号** | | | | |
| number | Identifier | ( | ) | $ |
| lexp | lexp → atom | lexp → atom | lexp → list |  |  |
| atom | atom → number | atom →  identifier |  |  |  |
| list |  |  | list → ( lexp-seq ) |  |  |
| lexp-seq | lexp-seq →  lexp lexp-seq' | lexp-seq →  lexp lexp-seq' | lexp-seq →  lexp lexp-seq' |  |  |
| lexp-seq’ | lexp-seq' →  lexp lexp-seq' | lexp-seq' →  lexp lexp-seq' | lexp-seq' →  lexp lexp-seq' | lexp-seq' → ε |  |

**e) 对输入串(a (b (2)) (c))给出相应得LL(1)分析程序的动作**

记lexp为E，list为 L，atom为A，lexp-seq为S，lexp-seq’为S'， number为num，identifier为id，则分析过程如下：

|  |  |  |
| --- | --- | --- |
| 栈 | 输入 | 动作 |
| $ E | (a(b(2))(c))$ | E → L |
| $ L | (a(b(2))(c))$ | L → (S) |
| $ )S( | (a(b(2))(c))$ | match |
| $ )S | a(b(2))(c))$ | S → E S' |
| $ ) S'E | a(b(2))(c))$ | E → A |
| $ ) S'A | a(b(2))(c))$ | A → id |
| $ ) S'id | a(b(2))(c))$ | match |
| $ ) S' | (b(2))(c))$ | S'→ E S' |
| $ ) S'E | (b(2))(c))$ | E → L |
| $ ) S'L | (b(2))(c))$ | L → (S) |
| $ ) S') S ( | (b(2))(c))$ | match |
| $ ) S') S | b(2))(c))$ | S → E S' |
| $ ) S') S'E | b(2))(c))$ | E → A |
| $ ) S') S'A | b(2))(c))$ | A → id |
| $ ) S') S'id | b(2))(c))$ | match |
| $ ) S') S' | (2))(c))$ | S'→ E S' |
| $ ) S') S'E | (2))(c))$ | E → L |
| $ ) S') S'L | (2))(c))$ | L → (S) |
| $ ) S') S') S ( | (2))(c))$ | match |
| $ ) S') S') S | 2))(c))$ | S → E S' |
| $ ) S') S') S'E | 2))(c))$ | E → A |
| $ ) S') S') S'A | 2))(c))$ | A → num |
| $ ) S') S') S'num | 2))(c))$ | match |
| $ ) S') S') S' | ))(c))$ | S'→ ε |
| $ ) S') S') | ))(c))$ | match |
| $ ) S') S' | )(c))$ | S'→ ε |
| $ ) S') | )(c))$ | match |
| $ ) S' | (c))$ | S'→ E S |
| $ ) S'E | (c))$ | E → L |
| $ ) S'L | (c))$ | L → (S) |
| $ ) S')S( | (c))$ | match |
| $ ) S')S | c))$ | S → E S' |
| $ ) S')S'E | c))$ | E → A |
| $ )S ')S'A | c))$ | A → id |
| $ ) S')S'id | c))$ | match |
| $ ) S')S' | ))$ | S'→ ε |
| $ ) S') | ))$ | match |
| $ ) S' | )$ | S'→ ε |
| $ ) | )$ | Match |
| $ | $ | Accept |

### 练习4.5.2

1. SS+

2. SS+

3.a

### 练习4.6.2

增广文法如下：

1)S' -> S

2)S -> SS+

3)S -> SS\*

4)S -> a

GOTO函数见下图：



FOLLOW(S) = { a, $,+,\* }

语法分析表如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 状态 | ACTION | | | | GOTO |
| a | + | \* | $ | S |
| 0 | S1 |  |  |  | 2 |
| 1 | R4 | R4 | R4 | R4 |  |
| 2 | S1 |  |  | acc | 3 |
| 3 | S1 | S4 | S5 |  | 3 |
| 4 | R2 | R2 | R2 | R2 |  |
| 5 | R3 | R3 | R3 | R3 |  |

因为没有冲突，所以是SLR文法。

### 练习4.7.1

增广文法如下：

1)S' -> S

2)S -> SS+

3)S -> SS\*

4)S -> a

1. 正规LR项目集族如下：

**I0**

S’->.S,$

S->.SS+, $/a

S->.SS\*, $/a

S->.a, $/a

**I1**

S->a., $/a

**I2**

S’->S.,$

S->S.S+, $/a

S->S.S\*, $/a

S->.SS+, +/\*/a

S->.SS\*, +/\*/a

S->.a, +/\*/a

**I3**

S->a., +/\*/a

**I4**

S->SS.+, $/a

S->SS.\*, $/a

S->S.S+, +/\*/a

S->S.S\*, +/\*/a

S->.SS+, +/\*/a

S->.SS\*,+/\*/a

S->.a,+/\*/a

**I5**

S->SS+.,$/a

**I6**

S->SS\*.,$/a

**I7**

S->SS.+, +/\*/a

S->SS.\*, +/\*/a

S->S.S+, +/\*/a

S->S.S\*, +/\*/a

S->.SS+, +/\*/a

S->.SS\*, +/\*/a

S->.a, +/\*/a

**I8**

S->SS+.,+/\*/a

**I9**

S->SS\*.,+/\*/a

状态转换图如下：



语法分析表如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 状态 | ACTION | | | | GOTO |
| a | + | \* | $ | S |
| 0 | S1 |  |  |  | 2 |
| 1 | R4 |  |  | R4 |  |
| 2 | S3 |  |  | acc | 4 |
| 3 | R4 | R4 | R4 |  |  |
| 4 | S3 | S5 | S6 |  | 7 |
| 5 | R2 |  |  | R2 |  |
| 6 | R3 |  |  | R3 |  |
| 7 | S3 | S8 | S9 |  | 7 |
| 8 | R2 | R2 | R2 |  |  |
| 9 | R3 | R3 | R3 |  |  |

2. 归并得到LALR项目集族如下：

**I0**

S’->.S,$

S->.SS+, $/a

S->.SS\*, $/a

S->.a,$/ a

**I13**

S->a., $/+/\*/a

**I2**

S’->S.,$

S->S.S+, $/a

S->S.S\*,$/ a

S->.SS+, +/\*/a

S->.SS\*, +/\*/a

S->.a, +/\*/a

**I47**

S->SS.+, +/\*/a/$

S->SS.\*, +/\*/a/$

S->S.S+, +/\*/a

S->S.S\*, +/\*/a

S->.SS+, +/\*/a

S->.SS\*, +/\*/a

S->.a, +/\*/a

**I58**

S->SS+.,+/\*/a/$

**I69**

S->SS\*.,+/\*/a/$