五.计算题（10分）

已知文法

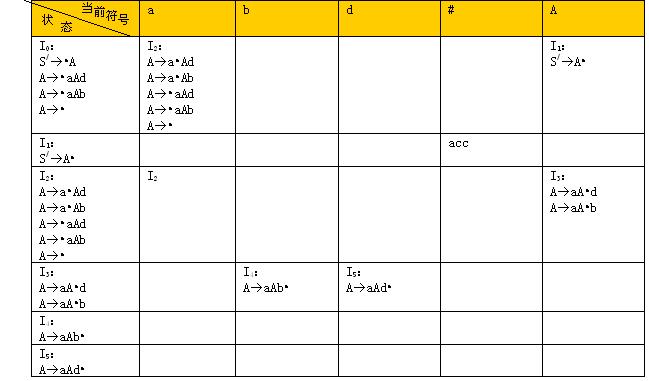
A->aAd|aAb| ε

判断该文法是否是 SLR(1) 文法，若是构造相应分析表，并对输入串 ab# 给出分析过程。

解：增加一个非终结符S/后，产生原文法的增广文法有：

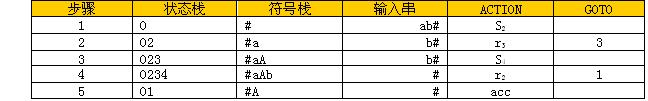
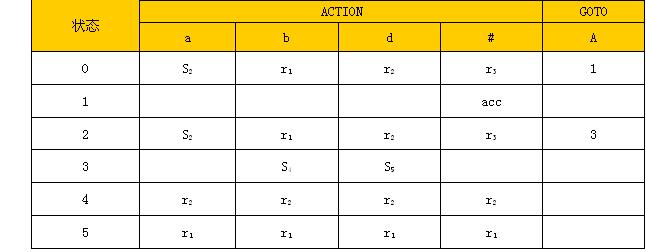
S'->A   
A->aAd|aAb|ε   
下面构造它的LR(0)项目集规范族为：

从上表可看出,状态I0和I2存在移进-归约冲突，该文法不是LR(0)文法。对于I0来说有：FOLLOW(A)∩{a}={b,d,#}∩{a}=Φ，所以在I0状态下面临输入符号为a时移进，为b,d,#时归约，为其他时报错。对于I2来说有也有与I0完全相同的结论。这就是说，以上的移进-归约冲突是可以解决的，因此该文法是SLR(1)文法。



其SLR(1)分析表为：

对输入串ab#给出分析过程为：



**----------------------分割-----------------------------------------------------**

1.设文法G(S):

S→^ | a | (T)

T→T,S | S

⑴ 消除左递归；

⑵ 构造相应的FIRST和FOLLOW集合；

⑶ 构造预测分析表

2.语句 if E then S

　　(1) 改写文法，使之适合语法制导翻译；

　　(2) 写出改写后产生式的语义动作。

3.设文法G（S）：

S→(T) | a

T→T+S | S

(1）计算FIRSTVT 和LASTVT；

（2）构造优先关系表。

4.设某语言的for语句的形式为

for i:＝E(1) to E(2) do S

其语义解释为

i:＝E(1)

LIMIT:＝E(2)

again: if i＜＝LIMIT then

Begin

S;

i:＝i＋1

goto again

End;

（1）写出适合语法制导翻译的产生式；

（2）写出每个产生式对应的语义动作。

5.把语句

while a<10 do

if c>0 then a:=a+1

else a:=a\*3-1;

翻译成四元式序列。

6.设有基本块

　　D:=A-C

　　E:=A\*C

　　F:=D\*E

　　S:=2

　　T:=A-C

Q:=A\*C

　　G:=2\*S

　　J:=T\*Q

K:=G\*5

　　L:=K+J

M:=L

　　假设基本块出口时只有M还被引用，请写出优化后的四元序列。

7.已知文法G(S)

S→a | ^ | (T)

T→T,S | S

(1) 给出句子(a,(a,a))的最左推导；

(2) 给出句型((T,S),a)的短语, 直接短语，句柄。

8.对于 C 语言do S while E语句

　　 (1)改写文法，使之适合语法制导翻译；

　　 (2)写出改写后产生式的语义动作。

9.已知文法G(S)

　　 S→aAcBe

　　 A→Ab| b

　　 B→d

　　(1)给出句子abbcde的最左推导及画出语法树；

　　(2)给出句型aAbcde的短语、素短语。

10.设文法G(S):

S→(T) | aS | a

T→T,S | S

⑴消除左递归和提公共左因子；

⑵构造相应的FIRST和FOLLOW集合；

⑶构造预测分析表。

11.把语句

if X>0 ∨ Y<0

then while X>0 do X:=A\*3

else Y:=B+3;

翻译成四元式序列。

12.已知文法G(S)

　 E→E+T | T

　 T→T\*F| F

　 F→(E)| i

　 (1) 给出句型 (i+i)\*i+i的最左推导及画出语法树；

　 (2) 给出句型 (E+T)\*i+F 的短语，素短语和最左素短语。

13.设文法G（S）：

S→T | S∨T

T→U |T∧U

U→i |-U

（1）计算FIRSTVT 和LASTVT；

（2）构造优先关系表。

**答案：**(1)消除左递，文法变为G’[S]：

S→^ | a | (T)'

T→ST’ | S

T’→,ST’ |ε

此文法无左公共左因子。

(2)构造相应的FIRST和FOLLOW集合：

FIRST(S)={a, ^, (}， FOLLOW(S)={#, ,, )}

FIRST(T)={a, ^, (} ，FOLLOW(T)={}}

FIRST(T’)={,, ε} ，FOLLOW(F)={)}

(3)构造预测分析表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | ^ | ( | ) | , | # |
| S | S→a | S→^ | S→(T)' |  |  |  |
| T | T→ST’ | T→ST’ | T→ST’ |  |  |  |
| T’ |  |  |  | T’→ε | T’→,ST’ |  |

2. (1)

C→if E then

S→CS(1)

(2)

C→if E then {BACK(E.TC, NXQ); C.chain:=E.FC}

S→CS(1) {S.chain:=MERG(C.Chain, S(1). Chain)}

3. (1) FIRSTVT(S)={a, ( }

FIRSTVT(T)={+, aa, (}

LASTVT(S)={a, ) }

LASTVT(T)={+, a, )}

(2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | + | ( | ) |
| a |  | .> |  | .> |
| + | <. | .> | <. | .> |
| ( | <. | <. | <. | =. |
| ) |  | .> | .> | >. |

4. (1) F→for i:=E(1) to E(2)  do

S→FS(1)

1. F→for i:=E(1) to E(2)  do

{GEN(:=, E(1).place, \_, entry(i));

F.place:=entry(i);

LIMIT:=Newtemp;

GEN(:=, E(2).place, \_, LIMIT);

Q:=NXQ;

F.QUAD:=q;

GEN(j≤, entry(i), LIMIT, q+2)

F.chain:=NXQ;

GEN(j, \_, \_, 0)}

S→FS(1)

{BACKPATCH(S(1).chain, NXQ);

GEN(+, F.place, 1, F.place);

GEN(j, \_, \_, F.QUAD);

S.chain:=F.chain

}

1. (1) (j<, a, ‘10’, (3))
   1. (j, \_, \_, (12))
   2. (j>, c, ‘0’, (5))
   3. (j, \_, \_, (8))
   4. (+, a, ‘1’, T1))
   5. (:=, T1, \_, a)
   6. (j, \_, \_, (1))
   7. (\*, a, ‘13’, T2)
   8. (-, T2, ‘1’, T3)
   9. (:=, T3, \_, a)
   10. (j, \_, \_, (1))

6.优化后的四元序列

D:=A-C

E:=A\*C

F:=D\*E

M:=F+20

7. 最左推导

S=(T)=>(T,S)=>(S,S)=>(a,S)=>(a,(T))=>(a,(T,S))=>(a,(S,S))=>(a,(a,S))=>(a,(a,a))

短语

((T,S),a)

(T,S),a

(T,S)

T,S

a

直接短语

T,S

a

句柄

T,S

8.(1)

S→do M1 S1while M2 E

M→ε

(2)

M→ε {M.quad=nestquad;}

S→do M1 S1while M2 E {backpatch(s1.nextlist, M2.quad);

backpatch(E.truelist, M1.quad);

S.nextlist=E.falelist;

}

9.(1) S=>aAcBe=>AAbcBe=>abbcBe=>abbcde

(2) 短语: aAbcde, Ab, d

素短语: Ab, d

10.(1) S →(L) | aS’

S’→S |ε

L→SL’

L’→,SL’ |ε

(2) FIRST(S)={a, (} FIRST(S’)={a, (, ε}

FIRST(L)={a, (} FIRST(L’)={,, ε}

FOLLOW(S)={,, ), #} FOLLOW(S’)={,, ), #}

FOLLOW(L)={ )} FOLLOW(L’)={ )}

(3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ( | ) | a | , | # |
| S | S →(L) |  | S → aS’ |  |  |
| S’ | S’→S | S’→ε | S’→S | S’→ε | S’→ε |
| L | L→SL’ |  | L→SL’ | L’→,SL’ |  |
| L’ |  | L’→ε |  |  |  |

11.(1) (j>, X, 0, (5))

(2) (j, \_, \_, (3))

(3) (j<, Y, 0, (5))

(4) (j, \_, \_, (11))

(5) (j>0, X, 0, (7))

(6) (j, \_, \_, (7))

(7) (\*, A, 3, T1)

(8) (:=, T1, \_, N)

(9) (j, \_, \_, (5))

(10) (j, \_, \_, (13))

(11) (+, B, 3, T2)

(12) (:=, T2, \_, Y)

12.(1) E=>E+T=>T+T=>T\*F+T=>F\*F+T=>(E)\*F+T=>(E+T)\*F+T=>(T+T)\*F+T

=>(F+T)\*F+T=>(i+T)\*F+T=>(i+F)\*F+T=>(i+i)\*F+T=>(i+i)\*i+T

=>(i+i)\*i+F=>(i+i)\*i+i

(2) 短语 i, F, E+T, (E+T), (E+T)\*i, (E+T)\*i+F

素短语 i, E+T

最左素短语 E+T

13.(1) FIRSTVT(S)={∨, ∧, i, - }

FIRSTVT(T)={∧, i, -}

FIRSTVT(U)={i, -}

LASTVT(S)={∨, ∧, i, - }

LASTVT(T)={∧, i, -}

LASTVT(U)={i, -}

(2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | i | ∨ | ∧ | - |
| S |  | .> | .> |  |
| ∨ | <. | .> | <. | <. |
| ∧ | <. | .> | .> | <. |
| - | <. | .> | .> | <. |