

一、

1. 四元式之间的联系是通过符号表联系的 ()
2. 文法 $S \rightarrow aA, A \rightarrow Ab, A \rightarrow b$ 是 LR(0)文法 (S 为文法开始符号) ()
3. 在编译前端的各个阶段都要进行查填表符号表的操作, 其最终目的是要解决符号空间到地址空间的映射问题 ()
4. 语法分析必须先消除文法中左递归 ()
5. 文法 G 的一个句子对应多个推导, 则 G 是二义的 ()
6. 综合属性适用于“自上而下”传递信息 ()
7. 每个过程活动记录的体积在编译时静态确定 ()
8. 采用三元式实现三地址代码时, 不利于对中间代码进行优化 ()
9. 一个语义子程序描述了一个文法所对应的翻译工作 ()
10. 目标代码生成时应考虑寄存器 ()

二、

1. 句型直接短语, 句柄
2. 功能上, 语句分为 () 语句、() 语句、分析表 M
3. 不含多重定义, 则为 () 文法

三、

1. 上机运行前经过 ()
 - a. 编辑 b. 编译 c. 连接 d. 运行
2. 目标代码生成阶段, 符号表用 ()
 - A. 目标代码生成 B. 语义检查 C. 语法检查 D. 地址分配
3. 在 LR 分析法中, 分析栈中存放的状态是识别规范句型 () DFA
 - A. 句柄 B. 前缀 C. 活前缀 D. LR(0)项目

$$Q \rightarrow Qa | a, \bar{a} \rightarrow \bar{a}$$

四、 $S \rightarrow aSb | P$

$P \rightarrow bPc | bQc$

$a \rightarrow Qa | a$

1. 改成 LL(1), 求 FIRST, FOLLOW.
2. LL(1)分析表 略.
3. 写 abacbLL(1)分析过程 略.

$$\begin{aligned} 1. S &\rightarrow aSb | P \\ P &\rightarrow bP' \\ P' &\rightarrow PC | QC \\ Q &\rightarrow aQ' \\ Q' &\rightarrow aQ' | \epsilon \end{aligned}$$

	first	follow
S	a, b	$\$, b$
P	b	$\$, b, c$
P'	b, a	$\$, b, c$
Q	a	c
Q'	a, ϵ	c

五、 $S \rightarrow A$
 $A \rightarrow aAd$

$A \rightarrow aAb$

$A \rightarrow \epsilon$

采用 LR(1)分析 1. 构造识别活前缀的 DFA

2. 根据 DFA 构造 LR(1)分析表

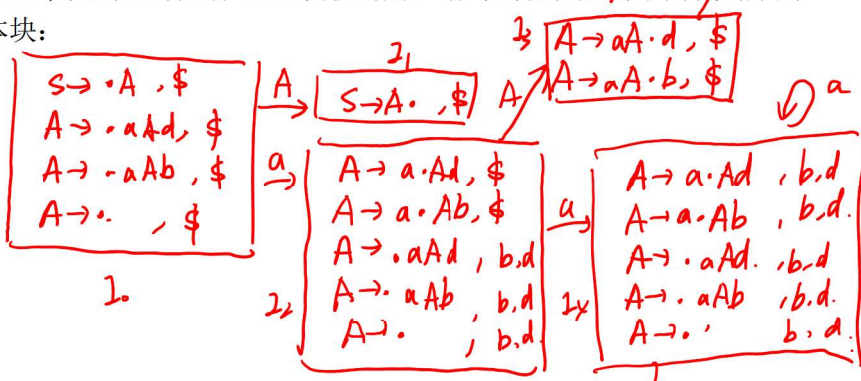


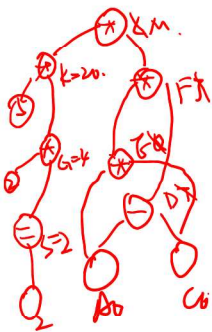
六、假设基本块出口时只有 M 还没被引用, 请写出优化后的中间代码序列
设有基本块:

$D = A - C$

$E = A * C$

$F = D * E$





S=2

T=A-C

Q=A*C

G=2*S

J=T*Q

K=G*5

M=K*J

M=L

假设基本块出口时只有 M 还没被引用

1. 画出上面代码段的 DAG 图

2. 根据 DAG 图, 写出优化后的中间代码序列

序列已在图中修改

七、1. 将下面语句翻译成三地址码序列 (注: 不进行优化; 指令编号从 1 开始)

if X>0 or Y<0

Then while X>0 do X=A*3

else Y=B+3

2. 画出 1 问中的三地址码序列的程序流程图

八、1. 访问链? 控制链? 简述它们在存储分配中的作用

2. 写出带有访问链的活动记录的数据项

九、下面文法产生代表正二进制数的 0 和 1 的串集

$B \rightarrow B0 \mid B1 \mid 1$

下面翻译方案计算这种正二进制数的十进制值

$B \rightarrow B_1 0 \{B.val = B_1.val * 2\}$

$B \rightarrow B_1 1 \{B.val = B_1.val * 2 + 1\}$

$B \rightarrow 1 \{B.val = 1\}$

请消除该基础文法左递归, 再重写一个翻译方案, 它仍然计算这种正二进制数的十进制值。

消除左递归

$B \rightarrow B_1 0 \mid B_1 1 \mid 1$

$B \rightarrow 1B'$

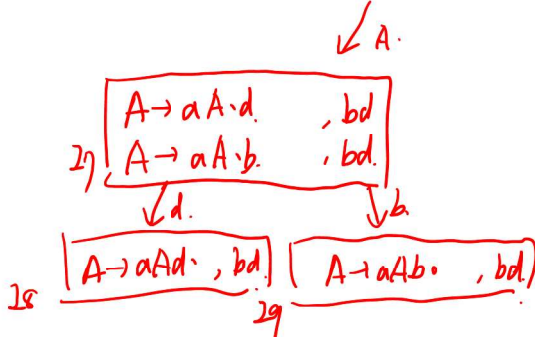
$B' \rightarrow 0B' \mid 1B' \mid \epsilon$

$B' \rightarrow \epsilon \{B'.val = 0 \quad B'.len = 0\}$

$B' \rightarrow 0B' \{B'.val = B_1.val \quad B'.len = B_1.len + 1\}$

$B' \rightarrow 1B' \{B'.val = B_1.val + 2^{B_1.len} \quad B'.len = B_1.len + 1\}$

$B \rightarrow 1B' \{B.val = B'.val + 2^{B'.len}\}$



	a	b	c	d	A	S
0	52			r4	1	
1				ac		
2	54	r4	r4		3	
3		56	55			
4	54	r4	r4		7	
5				r2		
6				r3		
7				r2		
8		r2	r2			
9		r3	r3			

1 if x>0 goto 5
 2 goto 3
 3 if y<0 goto 5
 4 goto 11
 5 if x>0 goto 7
 6 goto -
 7 t1 = A * 3
 8 x = t1
 9 goto 5
 10 goto -
 11 t2 = B + 3
 12 Y = t2