

# 编译原理 —— 期末复习

考试题型:

三道简答题 (5 分/题);

八到九题大题目(10 - 15 分/题)。

万海真情口述:

另外, 部分题是二选一, 因为计算机应用方向的老师教学方案略有不同, 故看到没有上过的知识点不必过于慌张!!

考试具体重点如下:

## 第一章: 引论

\*) 理解书本图1-6 中各个步骤在编译过程中的作用; (1.2 节详细看一下)

Eg1.

1. (9 points) Answer the following questions.

(1) How many phases does a compiler have? What are they?

编译程序分为: 词法分析、语法分析、语义分析、中间代码生成、代码优化、代码生成、符号表管理和出错处理八个阶段。

(2) Write three names of parsing methods.

非递归的预测分析(LL(1))技术, 算符优先分析技术, SLR 分析技术. (本题的答案不是唯一的.)

(3) What is a synthesized attribute? What is an inherited attribute?

在语法制导定义中, 每个文法有一组属性. 设对产生式  $A \rightarrow \alpha$  有形式为  $b := f(c_1, c_2, \dots, c_k)$  的语义规则, 其中  $f$  是函数,  $b$  和  $c_1, c_2, \dots, c_k$  是该产生式的文法符号的属性, 并且

(1) 如果  $b$  是  $A$  的属性,  $c_1, c_2, \dots, c_k$  是产生式右部文法符号的属性或  $A$  的其它属性, 那么,  $b$  叫做文法符号  $A$  的综合属性.

(2) 如果  $b$  是产生式右部某个文法符号  $X$  的属性,  $c_1, c_2, \dots, c_k$  是  $A$  的属性, 或右部文法符号的属性, 那么,  $b$  叫做文法符号  $X$  的继承属性.

Eg2.

7. 在编译系统中采用独立于机器的中间代码作为过渡可带来什么好处?

An2.

7. 便于编译系统的建立与移植; 便于进行独立于机器的代码优化工作。

\*) 1.3 1.4 1.5 节 不考

\*) 1.6 节建议看一下, 但是考试不一定会考(你懂的...)。

## 第二章：简单的语法制导分析器

\*) 不考

## 第三章：词法分析

\*) 3.1.2 词法单元、模式和词素认真看；

**词法单元：** 由一个词法单元名和一个可选的属性值组成，

**模式：**描述了一个词法单元的词素可能具有的形式（个人理解，我们写的正则表达式就是一个模式）

**词素：**一个字符序列，与词法单元的模式匹配（个人理解：一个字符串，看是否能跟上面模式中的正则表达式匹配）

\*) 3.2 不考

\*) 3.3 词法单元的规约 (写正则表达式) 必考！ 3.4、3.5 不考

Eg.

二、按以下自然语言描述，写出正则表达式

1. 在 $\{0, 1\}$ 上不以0开头的、以11结尾的字符串的集合。
2. 最多只含2个a的 $\{a, b\}$ 上的集合。

三、设字母表 $\Sigma=\{a,b\}$ ，用正则表达式（只使用 a, b,  $\epsilon$ , |, \*, +, ?）描述下列语言：

1. 不包含子串 ab 的所有由 a 和 b 组成的字符串
2. 不包含子串 abb 的所有由 a 和 b 组成的字符串
3. 不包含子序列 abb 的所有由 a 和 b 组成的字符串。

注意：关于子串（substring）和子序列（subsequence）的区别可以参考课本的内容。

An. 个人意见，欢迎指正

1.  $1(0|1)^*11 | 11$
2.  $b^*a?b^*a?b^*$

1.  $b^*a^*$
2.  $b^*(ab?)^*$
3.  $b^*a^*b?a^*$

Eg.

1. 请给出单个正则表达式，它仅与一个完整的注释匹配，除此之外不匹配任何其他串。书写正则表达式时，要求仅使用最基本的正则表达式算子( $\epsilon$ ,  $|$ ,  $*$ ,  $+$ ,  $?$ )。

An. 略

一、(7分) 用正则表达式(只使用 0, 1,  $\epsilon$ , (, ),  $|$ ,  $*$ ,  $+$ ,  $?$ ) 描述下列语言:

1. (1分) 可以表示成  $2^k$  ( $k \geq 0$ ) 的所有二进制整数;
2. (3分) 可以表示成  $2k$  ( $k \geq 1$ , 且  $k$  为奇数) 的所有二进制整数;
3. (3分) 可以表示成  $2^m + 2^n$  ( $m \geq 0, n \geq 0$ ) 的所有二进制整数.

注意: 整数之前不要含多余的 0.

【参考答案】

1.  $10^*$
2.  $(1(0|1)^*)?10$
3.  $10+|10^*10^*$

\*) 3.6 有穷自动机 3.7 从正则表达式到自动机, 一道大题必考!

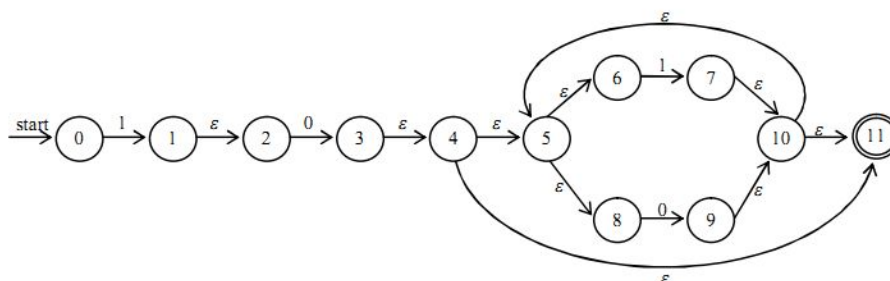
如: 从正则表达式构造出 NFA 并画出 NFA 并会 最小化状态数 (中文版 116 页例子 3.40 简明易懂)!

Eg.

1. 对正规表达式  $10(1|0)^*$  构造最小状态的 DFA。要求详细给出从正规表达式构造带  $\epsilon$  转移的 NFA、NFA 的确定化、以及 DFA 的极小化过程。

An. 注意:  $(a|b)^*$  的 NFA 一定要记住, 解题思路: 先求 move 再求  $\epsilon$ -closure

第一步 (5 分), 构造与正规表达式等价的具有  $\epsilon$  转移的 NFA:



第二步 (5 分), 将具有  $\epsilon$  转移的 NFA 确定化为 DFA:

令  $\epsilon$ -closure( $\{0\}$ ) =  $\{0\}$  = A, 以构造方式求各个新的状态如下

$$\delta(A, 1) = \epsilon\text{-closure}(\{1\}) = \{1, 2\} = B$$

$$\delta(B, 0) = \epsilon\text{-closure}(\{3\}) = \{3, 4, 5, 6, 8, 11\} = C$$

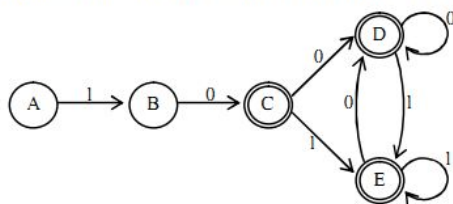
$$\delta(C, 0) = \epsilon\text{-closure}(\{9\}) = \{5, 6, 8, 9, 10, 11\} = D$$

$$\delta(C, 1) = \epsilon\text{-closure}(\{7\}) = \{5, 6, 7, 8, 10, 11\} = E$$

$$\delta(D, 0) = \epsilon\text{-closure}(\{9\}) = D$$

$$\delta(D, 1) = \epsilon\text{-closure}(\{7\}) = E$$

因 C、D 和 E 均含原终结状态 11，故均作为新终结状态。所求 DFA 为：

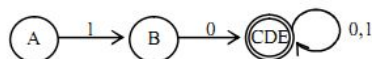


第三步（5 分），利用等价状态类化简 DFA。

B	$\times_2$			
C	$\times_1$	$\times_1$		
D	$\times_1$	$\times_1$	$\sqrt{3}$	
E	$\times_1$	$\times_1$	$\sqrt{4}$	$\sqrt{5}$
	A	B	C	D

下标为计算次序，依据：1、区别终态与非终态；2、 $\delta(A, 1) = B, \delta(B, 1) = \perp$ ；3、 $\delta(C, 0) = D = \delta(D, 0), \delta(C, 1) = E = \delta(D, 1)$ ；4、 $\delta(C, 0) = D = \delta(E, 0), \delta(C, 1) = E = \delta(E, 1)$ ；5、 $\delta(D, 0) = D = \delta(E, 0), \delta(D, 1) = E = \delta(E, 1)$ 。

结论，所求化简 DFA 为：



）还有一种是可以直接写出 DFA 的方法，但是万海没有强调，有时间的话还是看看的好

## 第四章：语法分析

\*) 4.1.3 语法错误的处理 详细看！书上 123 页!!!

- 词法错误
- 语法错误
- 语义错误
- 逻辑错误

\*) 4.1.4 错误恢复策略 详细看！

- ✚ 恐慌模式的的恢复
- ✚ 短语层次的回复
- ✚ 错误产生式
- ✚ 全局纠正

\*) 4.2 上下文无关文法(本次考试的重中之重)

要会做 4.2 节所有的例题...

3. (10 points) Are these following languages regular languages, context-free languages or non-context-free languages? Construct a regular expression for each regular language, and a context-free grammar for each context-free language.

(1)  $L_1 = \{w^2w \mid w \in \{0,1\}^*\}$ .

不是上下文无关语言, 是上下文有关语言.

(2)  $L_2 = \{w \mid w \in \{0,1\}^* \text{ and } w \text{ does not contain the substring } 101\}$ .

是正则语言. 正则表达式为:  $0^*(1+00+)^*1^*0^*$

(3)  $L_3 = \{w^2w^R \mid w \in \{0,1\}^*\}$ . Note that  $w^R$  denotes the reverse of  $w$ . For example,  $0121^R = 1210$ .

是上下文无关语言. 上下文无关文法为:  $S \rightarrow 0S0 \mid 1S1 \mid \epsilon$

\*) 4.2.7 上下文无关语法与正则表达式: 关系会考 130

考前看一看即可(包括书上例题)

\*) 4.3 其中每一个小节都很重要, 都要仔细看!!! 另外, 要会证明一个文法不是一个上下文无关的文法! 注意书上的原题那类的一定不会再考了, 所以例题只能作为参考, 千万!! 别指望着背例题!! 但是千万要看!!! 另外要注意如何消除二义性

一、用上下文无关文法描述下列定义在字母表  $\Sigma = \{a, b\}$  上的语言:

- 所有首字符和尾字符相同的非空字符串.
- $L = \{0^i1^j \mid i \leq j \leq 2i \text{ 且 } i \geq 0\}$ .
- 所有含有相同个数的 0 和 1 的字符串(包括空串).

二、考虑以下文法:

$S \rightarrow aABe$

$A \rightarrow Abc \mid b$

$B \rightarrow d$

- 用最左推导(leftmost derivation)推导出句子 abbcde.
- 用最右推导(rightmost derivation)推导出句子 abbcde.
- 画出句子 abbcde 对应的分析树(parse tree).

An.

- $S \rightarrow aTa \mid bTb$   
 $T \rightarrow a \mid b \mid \epsilon$
- $S \rightarrow S1 \mid 0S1 \mid 01 \mid \epsilon$
-



11、（5分）考察语言  $L = \{a^n b^m c^n d^m \mid n \geq 0 \wedge m \geq 0\}$ ，该语言是正则表达式  $a^* b^* c^* d^*$  定义的语言中满足  $a$  和  $c$  的个数相等、且  $b$  和  $d$  的个数相等的那些串。有学者证明了该语言不是一个上下文无关语言。

(1) （3分）语言  $L$  是程序设计语言中什么问题的抽象？

(2) （2分）该语言不是上下文无关的，这对于我们使用 BNF 定义程序设计语言的语法规则有什么启示？

### 【参考答案】

(1) 该语言是关于子程序（过程、函数、方法等）在声明时的形式参数与在调用时的实际参数之间个数保持一致性问题的抽象， $a^n b^m$  代表两个子程序声明的形式参数表中分别有  $n$  个和  $m$  个参数， $c^n d^m$  分别代表这两个子程序调用时的实际参数表。

(2) 由于 BNF 等价于上下文无关文法，这意味着我们使用 BNF 定义一门语言的语法规则时，无法表达形式参数与实际参数的一致性约束，故在语法定义中没有必要涉及参数的个数，无论是子程序的声明还是子程序的调用。因而，形式参数与实际参数的个数一致问题（以及类型匹配问题）是放在语义分析阶段考虑的。

### 【评分标准】

(1) 能回答参数传递个数一致性规则则加 2 分；正确地解释了理由则加 1 分。

(2) 能回答语法规则中无法表达上述规则即加 2 分。

Eg.

4. (9 points) Consider the following grammar:

$S \rightarrow SAS \mid (S) \mid a \mid b$

$A \rightarrow + \mid -$

(1) Demonstrate that this grammar is ambiguous.

对于句子  $a + a - b$ ，有两棵不同的分析树



因而文法是二义文法。

(2) Create an unambiguous grammar that generates the same language as the grammar above.

$S \rightarrow SAT \mid T$

$T \rightarrow (S) \mid a \mid b$

$A \rightarrow + \mid -$

Eg.

三、考虑以下文法：

$$S \rightarrow aSb$$
$$S \rightarrow aS$$
$$S \rightarrow \varepsilon$$

1. 这一文法产生什么语言（用自然语言描述）？
2. 证明这一文法是二义的.
3. 写出一个新的文法，要求新文法无二义且和上述文法产生相同的语言.

An.个人答案：欢迎指正

3,

$$S \rightarrow aTb$$
$$T \rightarrow aT$$
$$T \rightarrow \text{空}$$

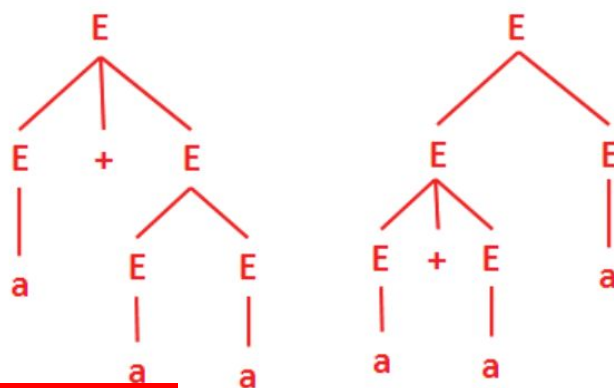
五、（7分）以下文法描述了由所有只含字符  $a$ ，且不含  $\varepsilon$  的正则表达式组成的语言（注意为了避免混淆，我们用 '+' 代替了 '|'）：

$$E \rightarrow E+E \mid EE \mid E^* \mid (E) \mid a$$

1. （3分）给出一个具有两棵不同分析树的字符串，并画出这两棵分析树.
2. （4分）写出一个新的文法，要求新文法无二义且和上述文法产生相同的语言.  
提示：你可以根据正则表达式运算符的优先级（依次为括号、闭包、连接、或）以及二元运算符的结合性质（均为左结合）来构造无二义文法.

【参考答案】

1. 对于输入串  $a+aa$ ，有如下两棵不同分析树：


$$\begin{aligned} 2. \quad & E \rightarrow E + T \mid T \\ & T \rightarrow TF \mid F \\ & F \rightarrow F^* \mid (E) \mid a \end{aligned}$$

4、(5 分) 考虑如下文法：

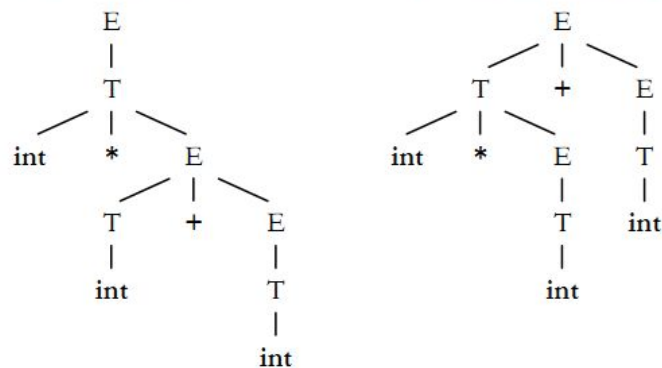
$E \rightarrow T \mid T + E$   
 $T \rightarrow \text{int} \mid \text{int} * E$

注意该文法是二义的。

(1) (3 分) 试找出一个引起分析时产生二义性的输入串，并画出该输入串的两棵分析树。

(2) (2 分) 写出一个与上述文法等价的无二义性文法，即两个文法产生的语言是相同的。

(1) 可有多种答案。例如，输入串  $\text{int} * \text{int} + \text{int}$  可画出两棵不同的分析树：



(2) 等价的无二义性文法如下：

$E \rightarrow T \mid T + E$   
 $T \rightarrow \text{int} \mid \text{int} * T$

5. (8 points) Consider the following grammar over the alphabet  $\{a, b, c\}$ :

$S \rightarrow Xa$

$X \rightarrow bX$

$X \rightarrow Y$

$Y \rightarrow Zc$

$Z \rightarrow bZ$

$Z \rightarrow \varepsilon$

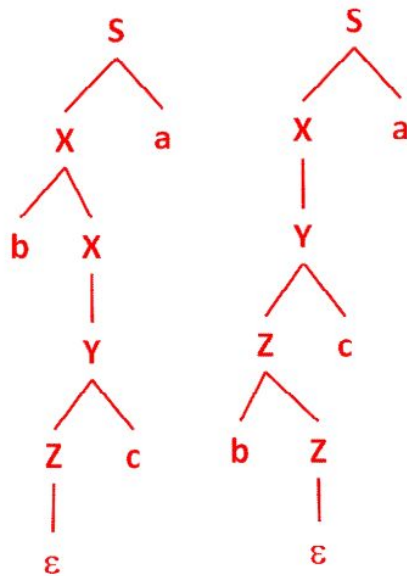
(1) (5 points) Demonstrate that this grammar is ambiguous.

(2) (3 points) Please remove exactly one production from this grammar to obtain an unambiguous grammar generating the same language.



【多写百来】

(1)



(2) 删除  $X \rightarrow bX$  或  $Z \rightarrow bZ$  均可.

个人观点：综上所述，此类题先认清优先级，然后每一级创建一个新的符号，最高优先级的部分才有可能（不一定）回溯到开头，其他只能往下，细节自己拿捏

\*) 4.4 一定会考! FIRST 集 & FOLLOW 集的求解与使用方法...

### ● FIRST 集

定义：FIRST( $\alpha$ )定义为可从 $\alpha$ 推导得到的串的首符号的集合。

算法：

- ①如果 $\alpha$ 是一个终结符，那么  $\text{FIRST}(\alpha) = \alpha$
- ②如果 $\alpha$ 是一个非终结符，且 $\alpha \rightarrow Y_1 Y_2 \dots$ 是一个产生式，若 $\epsilon$ 在  $\text{FIRST}(Y_1) \dots \text{FIRST}(Y_{i-1})$ 中，则  $\text{FIRST}(Y_i)$ 中的所有符号一定在  $\text{FIRST}(\alpha)$ 中
- ③如果 $\alpha \rightarrow \epsilon$ ，那便将 $\epsilon$ 加入到  $\text{FIRST}(\alpha)$ 中

### ● FOLLOW 集

定义：FOLLOW( $\alpha$ )定义为可能在某些句型中紧跟在 $\alpha$ 右边的终结符的集合。

算法：

- ①将  $\$$  放到 FOLLOW( $S$ )中，其中  $S$  是开始符号， $\$$  是结束标志
- ②如果存在  $A \rightarrow \alpha B \beta$ ，那么  $\text{FIRST}(\beta)$ 中除 $\epsilon$ 以外的所有符号都在 FOLLOW( $B$ )中
- ③如果存在  $A \rightarrow \alpha B$ ，或  $A \rightarrow \alpha B \beta$ 且  $\text{FIRST}(\beta)$ 包含 $\epsilon$ ，那么 FOLLOW( $A$ )中的所有符号都在 FOLLOW( $B$ )中

Eg. 该题相对比较复杂，涉及到 first 集中有空的情况

一、考虑以下文法：

$S \rightarrow aTUV \mid bV$

$T \rightarrow U \mid UU$

$U \rightarrow \varepsilon \mid bV$

$V \rightarrow \varepsilon \mid cV$

写出每个非终端符号的 FIRST 集和 FOLLOW 集。

An.

$\text{First}(S) = \{a, b\}$

$\text{First}(T) = \text{First}(U) = \{b, \text{空}\}$

$\text{Follow}(S) = \{\$ \}$

$\text{Follow}(T) = \text{Follow}(U) = \text{Follow}(V) = \{\$, c, b\}$

预测分析表的使用 & 给出一个输入串要会画出语法识别的详细分析过程！  
要会做 4.4 节所有的例题 ...

Eg.

7. (12 points) The LR parsing table for an unknown grammar is given as follows:

	b	A	\$	S	A
0	s3	s2		1	5
1			accept		
2	r4	s2	r4		4
3			r1		
4	r3		r3		
5	s6				
6		s2			7
7			r2		

The grammar is known to have the following productions:

(r1)  $S \rightarrow$  \_\_\_\_\_ (1)

(r2)  $S \rightarrow$  \_\_\_\_\_ (2)

(r3)  $A \rightarrow$  \_\_\_\_\_ (3)

(r4)  $A \rightarrow$  \_\_\_\_\_ (4)

Please fill in the blank (1)~(4) to complete the grammar according to the LR table above.

An.

Please fill in the blank (1)~(4) to complete the grammar according to the LR table above.

(1) b

(2) AbA

(3) aA

(4) a

\*) 4.5 结合 4.4 一定会详细考！

\*) 4.6 对 LR 文法项集的分析过程！另外：一定要会证明一个文法到底是不是属于 LR(0)?  
LR(1)? SLR? 中的哪个文法！

平时作业大把，不粘题了，但是这一题：

## 2. 对文法

$$\begin{aligned} S &\rightarrow a A c B e \\ A &\rightarrow A b \mid b \\ B &\rightarrow d \end{aligned}$$

构造 LR(1)分析表，要求给出构造表的各个详细步骤。

第一步（1分），拓广文法：

$$\begin{aligned} (0) \quad S' &\rightarrow S \\ (1) \quad S &\rightarrow a A c B e \\ (2) \quad A &\rightarrow A b \\ (3) \quad A &\rightarrow b \\ (4) \quad B &\rightarrow d \end{aligned}$$

第二步（2分），求所有非终结符号的 FIRST 集合：

$$\begin{aligned} \text{FIRST}(S) &= \{ a \} \\ \text{FIRST}(A) &= \{ b \} \\ \text{FIRST}(B) &= \{ d \} \end{aligned}$$

第三步（5分），构造 LR(1)有效项目集族：

令  $I_0 = \text{closure}(\{ [S' \rightarrow \cdot S, \$] \}) = \{ [S' \rightarrow \cdot S, \$], [S \rightarrow \cdot a A c B e, \$] \}$ ，其他状态如下：

$$\begin{aligned} \text{go}(I_0, S) &= \{ [S' \rightarrow S \cdot, \$] \} = I_1 \\ \text{go}(I_0, a) &= \{ [S \rightarrow a \cdot A c B e, \$], [A \rightarrow \cdot A b, c/b], [A \rightarrow \cdot b, c/b] \} = I_2 \\ \text{go}(I_2, A) &= \{ [S \rightarrow a A \cdot c B e, \$], [A \rightarrow A \cdot b, c/b] \} = I_3 \\ \text{go}(I_2, b) &= \{ [A \rightarrow b \cdot, c/b] \} = I_4 \end{aligned}$$

错则扣3分！

中的红字标出部分还是需要注意；

Eg.

一. 文法  $G = (\{U, T, S\}, \{a, b, c, d, e\}, P, S)$

其中 P 为：

$$S \rightarrow U T a \mid T b$$
$$T \rightarrow S \mid S c \mid d$$
$$U \rightarrow U S \mid e$$

(1) 判断 G 是 LR(0)，SLR(1)，LALR(1) 还是 LR(1)，说明理由。

(2) 构造相应的分析表。

An. 个人感觉不会这样变态吧，这样考的话一题就得一个小时？但是是万海出题，就是这个题型了，个人意见，从 LR(1) → LALR(1) → ... 顺序分析，一旦一个不行，后面的肯定不行了，因为他们的表达能力是递减的，一定要全部都会分析!!!! 祝君好运

## 第五章：语法制导的翻译

\*) 5.1 综合属性、继承属性的定义；

八、（8 分）考虑以下语法制导定义（SDD）：

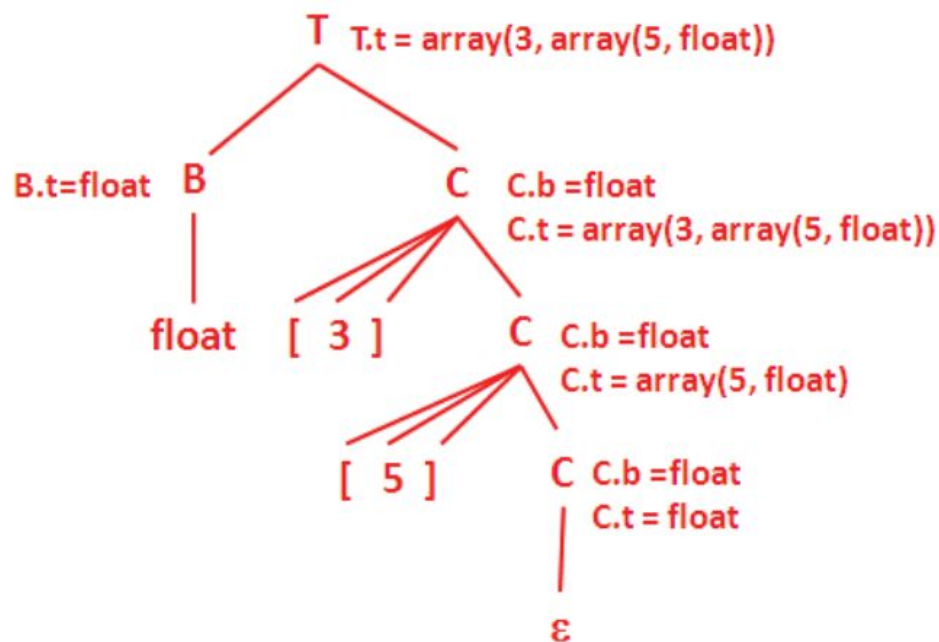
语法规则	语义规则
$T \rightarrow BC$	$T.t = C.t$ $C.b = B.t$
$B \rightarrow \text{int}$	$B.t = \text{integer}$
$B \rightarrow \text{float}$	$B.t = \text{float}$
$C \rightarrow [\text{num}] C_l$	$C.t = \text{array}(\text{num.val}, C_l.t)$ $C_l.b = C.b$
$C \rightarrow \varepsilon$	$C.t = C.b$

注：假设 num 对应非负整数，num.val 代表这一非负整数的值。

- （2 分）在这一 SDD 中，哪些属性是综合属性？哪些属性是继承属性？
- （6 分）对输入串 float[3][5]构造带注释的分析树（annotated parse tree）。

**【参考答案】**

- T.t, C.t 和 B.t 是综合属性，C.b 是继承属性。
- 



\*) 5.2 依赖图 以及 求值顺序  
书上 198 页

Eg.

一. 考虑以下语法制导定义 (Syntax Directed Definition):

语法规则	语义规则
$S \rightarrow ABCD$	$S.val = A.val + B.val + C.val + D.val$
$A \rightarrow gBa$	$A.val = B.val * 5$
$B \rightarrow B_1b$	$B.val = B_1.val * 2$
$B \rightarrow b$	$B.val = 2$
$C \rightarrow C_1c$	$C.val = C_1.val * 3$
$C \rightarrow c$	$C.val = 3$
$D \rightarrow d$	$D.val = 1$

对于输入串 gbbabbccd 构造带注释的分析树 (annotated parse tree) .

如何写出 SDD? (这个必考!)

Eg.

二. 以下文法定义了二进制浮点数常量的语法规则:

$S \rightarrow L.L | L$

$L \rightarrow LB | B$

$B \rightarrow 0 | 1$

试给出一个 S 属性的语法制导定义, 其作用是求出该二进制浮点数的十进制值, 并存放在开始符号 S 相关联的一个综合属性 value 中。

例如, 对于输入串 101.101, S 的 value 属性值结果应该是 5.625。

要求在编写语法制导定义时, 不得改写文法!

An. 略

Eg.

8. (10 points) Design a Syntax-Directed Translation scheme (SDT) for the following expression grammar, such that the non-terminal E has an attribute E.type which keeps the type of E. The value of E.type is BOOL, INTEGER or ERROR, where BOOL denotes the boolean type, INTEGER denotes the integer type, and ERROR denotes an illegal type.

$E \rightarrow E_1 + E_2$	{	(1)	}
$E_1 \text{ and } E_2$	{	(2)	}
$E_1 = E_2$	{	(3)	}
$(E_1)$	{	(4)	}
true	{	(5)	}
false	{	(6)	}
int	{	(7)	}

Please fill in the blanks (1) ~ (7) with appropriate semantic actions.

An.



Please fill semantic actions in the blank (1) ~ (7).

Note: The operator '+' must be applied to two integer expressions; the operator 'and' must be applied to two boolean expressions; the relational operator '=' must be applied to either two integer expressions or two boolean expressions, but cannot be applied to an integer expression and a boolean expression; if any operand is of illegal type, the resulting type is also illegal.

- (1)  $E.type := ((E1.type = INT) \text{ and } (E2.type = INT) ? INT : ERROR);$
- (2)  $E.type := ((E1.type = BOOL) \text{ and } (E2.type = BOOL) ? BOOL : ERROR);$
- (3)  $E.type := ((E1.type = E2.type) \text{ and } (E2.type \neq ERROR) ? BOOL : ERROR);$
- (4)  $E.type := E1.type;$
- (5)  $E.type := BOOL;$
- (6)  $E.type := BOOL;$
- (7)  $E.type := INT;$

8. (11 points) The following grammar generates binary fractions.

$F \rightarrow 0.B$

$B \rightarrow 0B$

| 1B

| 0

| 1

Design a syntax-directed definition (SDD) for the above grammar such that the nonterminal  $F$  has an attribute  $F.val$  which keeps the decimal value of the binary fraction generated by  $F$ . Please use as few attributes as possible and do NOT modify the grammar.

【参考答案】

产生式	语义规则
$F \rightarrow 0.B$	$F.val = B.val$
$B \rightarrow 0B_1$	$B.val = B_1.val * 0.5$
$B \rightarrow 1B_1$	$B.val = 0.5 + B_1.val * 0.5$

$B \rightarrow 0$	$B.val = 0$
$B \rightarrow 1$	$B.val = 0.5$

\*) 5.5.4 Marker 技术的使用 例题 5.25 - 5.26  
书上 224 页的定义

## 第六章：中间代码的生成

### \*) 6.1 会绘制 DAG 图

Eg.

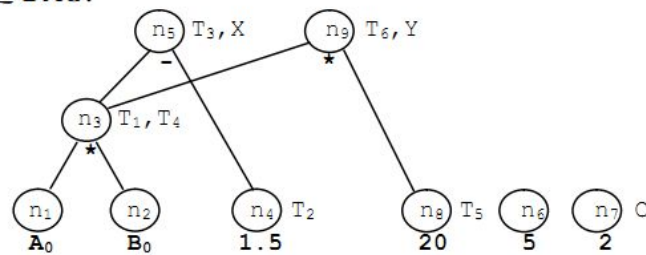
5. 设某基本块的代码序列如下：

- (1)  $T_1 := A * B$
- (2)  $T_2 := 3 / 2$
- (3)  $T_3 := T_1 - T_2$
- (4)  $X := T_3$
- (5)  $C := 5$
- (6)  $T_4 := A * B$
- (7)  $C := 2$
- (8)  $T_5 := 18 + C$
- (9)  $T_6 := T_4 * T_5$
- (10)  $Y := T_6$

利用 DAG 优化该基本块。要求首先根据基本块产生 DAG，然后将 DAG 转换为基本块的代码序列。

Answer:

第一步，构造 DAG：



第二步，根据 DAG 生成基本块的代码序列：

- (1)  $T_1 := A * B$
- (2)  $T_4 := T_1$  // 这类复写语句的次序可有多种
- (3)  $T_2 := 1.5$
- (4)  $T_3 := T_1 - 1.5$  // 若为  $T_3 := T_1 - T_2$  则扣 1 分
- (5)  $X := T_3$
- (6)  $C := 2$
- (7)  $T_5 := 20$
- (8)  $T_6 := T_1 * 20$  // 若为  $T_6 := T_1 * T_5$  则扣 1 分
- (9)  $Y := T_6$

思考：为何  $C=2$  与  $C=5$  分开写？

Eg.

9. (12 points) Consider the following basic block:

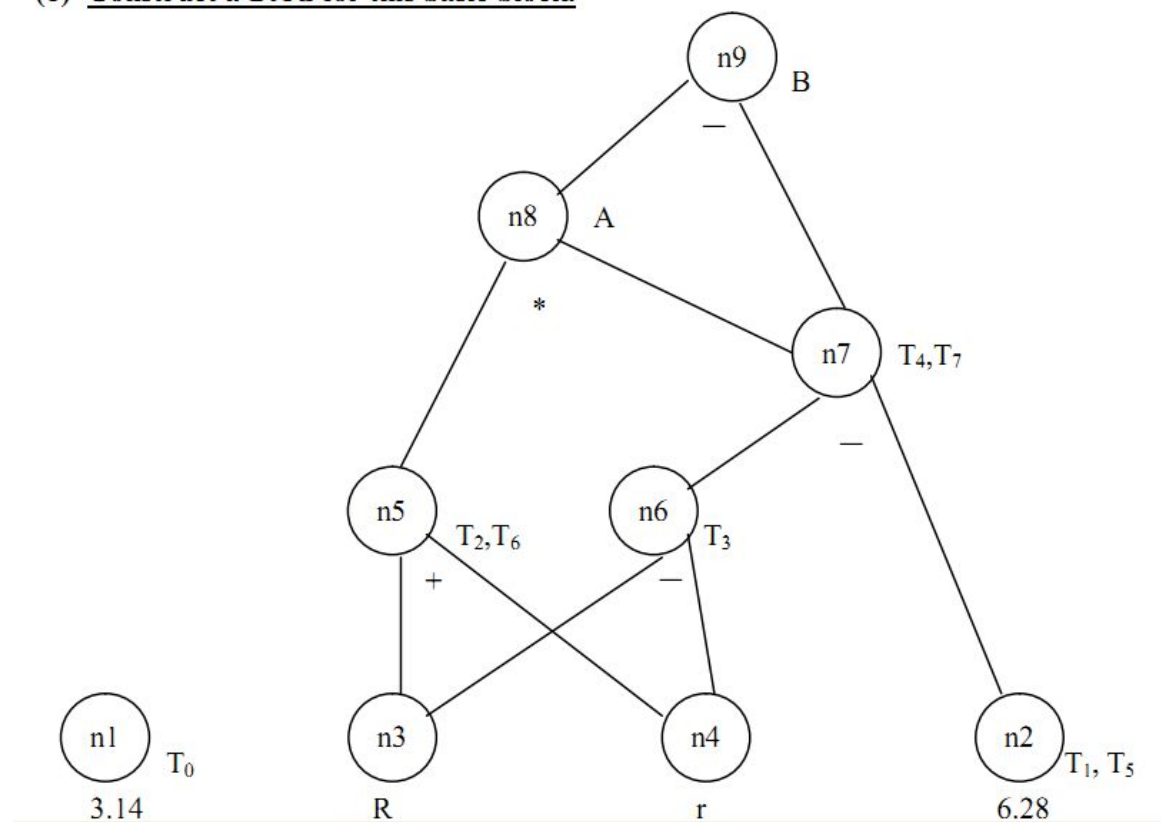
- |                        |                         |
|------------------------|-------------------------|
| (1) $T_0 := 3.14$      | (7) $B := A$            |
| (2) $T_1 := 2 * T_0$   | (8) $T_5 := 2 * T_0$    |
| (3) $T_2 := R + r$     | (9) $T_6 := R + r$      |
| (4) $T_3 := R - r$     | (10) $T_7 := T_3 - T_5$ |
| (5) $T_4 := T_3 - T_1$ | (11) $B := A - T_7$     |
| (6) $A := T_2 * T_4$   |                         |

(1) Construct a DAG for this basic block.

(2) Assuming that only A and B are live on exit from this basic block, simplify the three-address code.

Answer:

(1) Construct a DAG for this basic block.



(2) Assuming that only A and B are live on exit from this basic block, simplify the three-address code.

$S1 := R + r$

$S2 := R - r$

$S3 := S2 - 6.28$

$A := S1 * S3$

$B := A - S3$

注意本题中的临时变量 S1~S3 可以替换为其它名称.

思考:  $B=A$  与  $B=A-T7$  为何直接合并了?

\*) 6.2 三地址码 (必考!!) 补充!!! 区分三元式, 四元式!!!

Eg.

4. 产生赋值语句的三地址代码时, 可采用如下的语法制导定义:

产生式	语义规则
$S \rightarrow id := E$	$S.code := E.code \parallel gen(id.place := E.place)$
$E \rightarrow E_1 + E_2$	$E.place := newtemp;$ $E.code := E_1.code \parallel E_2.code \parallel gen(E.place := E_1.place + E_2.place)$
$E \rightarrow E_1 * E_2$	$E.place := newtemp;$ $E.code := E_1.code \parallel E_2.code \parallel gen(E.place := E_1.place * E_2.place)$
$E \rightarrow - E_1$	$E.place := newtemp;$ $E.code := E_1.code \parallel gen(E.place := - E_1.place)$
$E \rightarrow ( E_1 )$	$E.place := E_1.place;$ $E.code := E_1.code;$
$E \rightarrow id$	$E.place := id.place;$ $E.code := ''$

根据以上语法制导定义给出赋值语句  $a := b * c + b * d$  的三地址代码 (假设过程 *newtemp* 产生的临时变量名字依次为  $t_1, t_2, t_3, \dots$ ; 以 LR 分析次序执行语义规则)。

An. 据大神称上面的表格只有左半边“产生式”对解答有作用, 越向下优先级越高; 右半边表格“语义规则”里面的内容看看即可

4. 错一条语句扣 3-4 分, 变量下标错扣 1 分。

- (1)  $t_1 := b * c$
- (2)  $t_2 := b * d$
- (3)  $t_3 := t_1 + t_2$
- (4)  $a := t_3$

9. (15 points) Based on the syntax-directed definitions in the textbook, translate the following program into quadruples.

```

WHILE (C < D) ∧ (E < D) DO
  IF D = 10 THEN
    WHILE C < E DO C := C+1
  ELSE E := E*2

```

【参考答案】

- (1) (j<, C, D, 3)
- (2) (j, -, -, 0)
- (3) (j<, E, D, 5)
- (4) (j, -, -, 2) ← S.CHAIN
- (5) (j=, D, 10, 7)
- (6) (j, -, -, 13)
- (7) (j<, C, E, 9)
- (8) (j, -, -, 1)
- (9) (+, C, 1, T<sub>1</sub>)
- (10) (:=, T<sub>1</sub>, -, C)
- (11) (j, -, -, 7)
- (12) (j, -, -, 1)
- (13) (\*, E, 2, T<sub>2</sub>)
- (14) (:=, T<sub>2</sub>, -, E)
- (15) (j, -, -, 1)

\*) 6.4 表达式的翻译, 特别注意: 数组元素的运算方法!

\*) 6.6.2 短路代码 在代码优化中的运用

\*) 6.7 一定要理解一下回填技术 注意例题: 例子 6.24

第七、八、九章:

\*) 理解两个图中的详细过程: 图 7.1 图 7.7(调用与被调用)

注意例题: 例子 7.6 要会分析!

\*) 8.4 基本块 和 流图 (必考!!)

9、(4 分) 考虑以下三地址码片断:

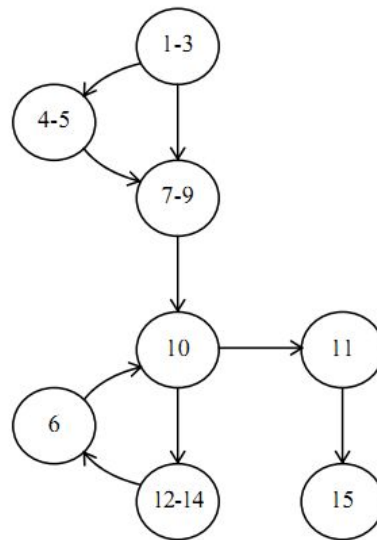
```
(1)      b := 1
(2)      b := 2
(3)      if w <= x goto B
(4)      e := b
(5)      jump B
(6)  A:   jump D
(7)  B:   c := 3
(8)      b := 4
(9)      c := 6
(10) D:   if y <= z goto E
(11)      jump End
(12) E:   g := g + 1
(13)      h := 8
(14)      jump A
(15) End: h := 9
```

为上述代码片断划分基本块, 并画出该代码片断的控制流图 (Control Flow Graph, 简称 CFG)。在答案中你可以直接画出 CFG, 但请务必在 CFG 的每一结点中用  $n \sim m$  表示该基本块由第  $n$  至  $m$  条指令组成。

An.



【参考答案】



【评分标准】

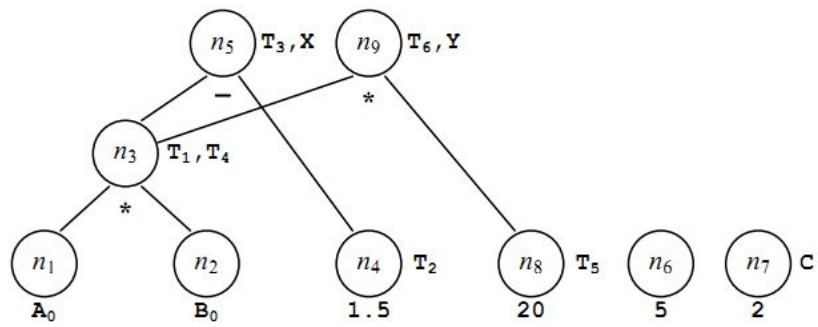
### 8.5 优化分析 并能写出优化后的代码

10、（6分）设某基本块的代码序列如下：

```
(1) T1 := A * B  
(2) T2 := 3 / 2  
(3) T3 := T1 - T2  
(4) X := T3  
(5) C := 5  
(6) T4 := A * B  
(7) C := 2  
(8) T5 := 18 + C  
(9) T6 := T4 * T5  
(10) Y := T6
```

利用 DAG 优化该基本块。要求首先根据基本块产生 DAG，然后将 DAG 转换为基本块的代码序列。

(1) 构造 DAG:



(2) 根据 DAG 生成基本块的代码序列:

```
(1) T1 := A * B
(2) T4 := T1           // 这类复写语句的次序可有多种
(3) T2 := 1.5
(4) T3 := T1 - 1.5    // 若为 T3 := T1 - T2 则错
(5) X := T3
(6) C := 2
(7) T5 := 20
(8) T6 := T1 * 20    // 若为 T6 := T1 * T5 则错
(9) Y := T6
```

4. 设有以下基本块

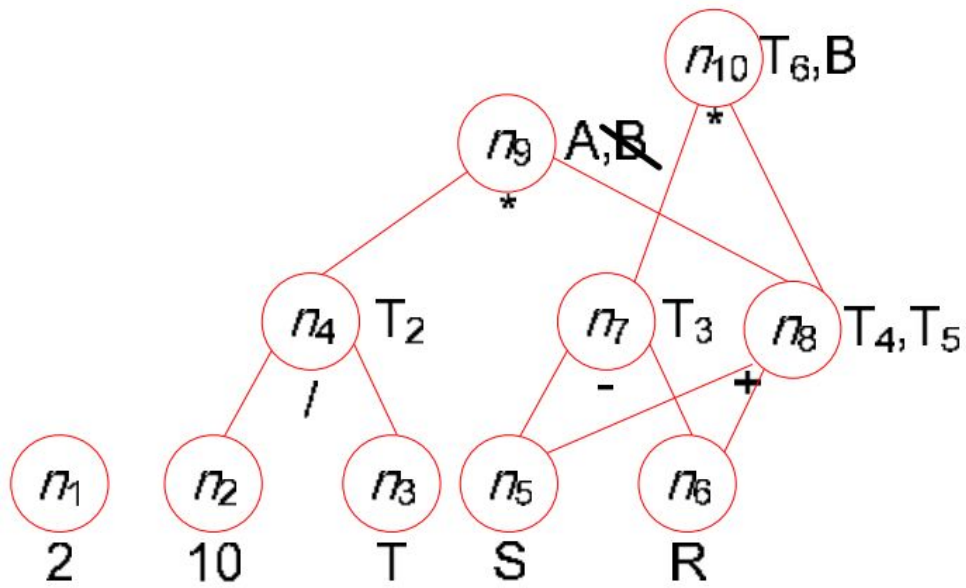
```
T1: =2
T2: =10/T
T3: =S-R
T4: =S+R
A: =T2 * T4
B: =A
T5: =S+R
T6: =T3 * T5
B: =T6
```

(1) 画出 DAG 图; (6 分)

(2) 假设只有 A、B 在基本块出口之后还被引用, 请写出优化后的四元式序列。(4 分)

【答案】:

(1) DAG 图如下:



(2)

$T_2 := 10 / T$

$T_3 := S - R$

$T_4 := S + R$

$A := T_2 * T_4$

$B := T_3 * T_4$

---

```

(1) m := 0
(2) v := 0
(3) if v >= n goto (19)
(4) r := v
(5) s := 0
(6) if r < n goto (9)
(7) v := v + 1
(8) goto (3)
(9) s := v + r
(10) y := 0 * x
(11) z := v - y
(12) x := z + r
(13) r := m - x
(14) if s <= m goto (17)
(15) m := s
(16) s := s + r
(17) r := r + 1
(18) goto (6)
(19) return m

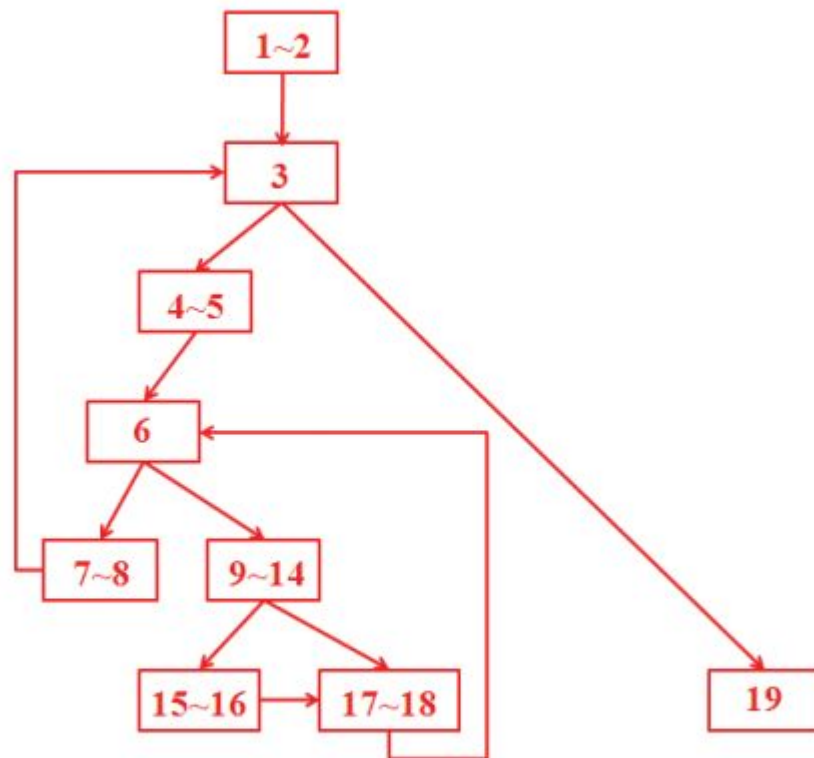
```

1/19/4/9/7/3/10/9/17/15/6/5/19  
 1/3/4/6/7/9/15/17/19

1. (4 分) 为这段代码划分基本块 (Basic Block), 并画出控制流图 (Control Flow Graph). 在答案中你可以直接画出控制流图, 但对图中的每个结点, 请用  $m \sim n$  表示相应的基本块由第  $m$  至第  $n$  条语句组成.
2. (4 分) 选择包括最多语句的基本块进行优化, 优化时注意:
  - a) 只在这一基本块中出现的变量均看作是这一基本块中的局部变量;
  - b) 可以使用任何你觉得适用的基本块优化方法进行优化.
  - c) 假设优化前后原来语句的标号不发生变化.
 给出这个基本块最终的优化结果即可, 不必写出中间过程, 也不必抄写其它基本块的代码.

【参考答案】

1.



2.  $s := v + r$   
 $r := m - s$   
if  $r \geq 0$  goto (17)

对 基本块 的分析（必是有一道大题的...）  
\*) 9.1 节的例题要看一下，另有可能设计到 9.6.1 的内容！