### 计算机学院、网安学院2023－2024学年第一学期

### 本科生编译系统原理期末考试试卷(A卷)

专业：\_\_\_\_\_\_\_\_\_\_\_\_\_\_年级：\_\_\_\_\_\_\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_

姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_成绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **得 分** |
|  |

1. 单项选择题（每空2分，共24分）

（空格末尾的编号与答题卡中编号对应）

* + - 1. LLVM编译器生成IR代码是在 C ⑴阶段，发现函数调用的实参与函数定义的形参不匹配是在 B ⑵阶段，消除无用代码是在 D ⑶阶段。

A．词法分析 B．语义分析

C．中间代码生成 D．代码优化

* + - 1. 将可重定位机器码放置到内存并调整程序中地址是 D ⑷的工作，解决程序中的外部引用是 C ⑸的工作。

A．预处理器 B．编译器

C．链接器 D．加载器

* + - 1. 用Thompson算法将一个有a个符号（包括****）、b个运算符的正则表达式r转换为NFA N，则下面说法**正确**的是 B ⑹。

A．N中每个状态只可能有一条****边

B．N中每个状态不会有标记字母表中相同符号的多条边

C．N可能有多个终态 D．N最多有(a+b)个状态

* + - 1. 下面可用上下文无关文法描述的语言是\_\_\_\_B\_\_\_\_\_⑺。

A．**{w2w | w是任意0、1串}** B．**{ 0n1n | n≥0 }**

C．**{ 0n1m2n3m | n≥1且m≥1 }** D．以上皆错

* + - 1. 对LR(1)项目[**A→XYZ•, a**]，说法**错误的**是 C ⑻。

A．它表示在某个最右句型中**a**跟随**A**

B．它表示在某个最右句型中**a**跟随**XYZ**

C．它表示可以进行归约了，输入缓冲区首符号可以是任意的

D．它表示可以进行归约了，输入缓冲区首符号可以是**a**

* + - 1. 对下面CFG，后面给出的第一个推导是 B ⑼。，第二个推导是 C ⑽。

**S → TU T → 0T1 |  U→1U0 | **

**S⇒TU⇒T1U0⇒T10⇒0T110⇒00T1110⇒001110**

**S⇒TU⇒T1U0⇒0T11U0⇒0T110⇒00T1110⇒001110**

A．最左推导

B．最右推导

C．以上皆错

* + - 1. 在进行代码优化时，应重点考虑哪种程序结构？ D ⑾。

A．变量声明 B．赋值

C．条件分支 D．循环

* + - 1. 在采用名字等价判定的高级语言程序中，如果披着羊皮的灰太狼（对原始类型“狼”定义了类型别名“羊”）被认为是“羊”，是否会与喜羊羊（原始类型就是“羊”）无法区分呢？ B ⑿。

A．转用结构等价判定来区分两者

B．同一作用域中只允许有一个名为“羊”的类型，后定义者非法

C．通过它们的二进制编码表示来进行区分

D．禁止邪恶的灰太狼，保留善良的喜羊羊

|  |
| --- |
| **得 分** |
|  |

1. 设计题（每题5分，共20分）

1. 设计正则表达式，与下面DFA接受完全相同的符号串集合。



答：a((c | bc)da)\*

2. 设计正则表达式，接受所有十进制定点数，无多余的前导零和末尾零。即，0.0、123.01、123005.0这样的串是合法的，但00.0、001.000、002345.1000是非法的。

答：([0-9] | [1-9][0-9]\*) . ([0-9] | [0-9]\*[1-9])

3. 证明下面文法生成的所有二进制串都能被3整除。

***num* → 11 | 1001 | *num* 0 | *num num***

证明：采用数学归纳法

对一步推导*num*⇒11和*num*⇒1001，生成的二进制串都能被3整数。

假设对所有<n步的推导，生成的二进制串都能被3整除，下面证明=n步的推导生成的二进制串也都能被3整除。有两种情况：

1. 推导过程为*num*⇒ *num* 0⇒\* *w* 0，*w*为二进制串。则*num*⇒\* *w*为<n步的推导，因此*w*能被3整除，而*w* 0的值为*w*的2倍，因此也能被3整除。
2. 推导过程为*num*⇒ *num* *num* ⇒\* *w* *num*⇒\* *w* *v*，*w、v*为二进制串。则*num*⇒\* *w*和*num*⇒\* *v*为<n步的推导，因此*w*和*v*都能被3整除，而*w* *v*的值为*w*<<log *v*+*v*，因此也能被3整除。

综上，文法的任何推导过程生成的二进制串都能被3整除。

4. 设计CFG，接受与第1小题DFA完全相同的符号串集合。

答：**A0→a A1 A1→b A2 | c A3 |  A2→c A3 A3→d A0**

|  |
| --- |
| **得 分** |
|  |

1. 词法分析算法（24分）

对下面的正则表达式：

**(a\*bc) | (ab\*c) | (abc\*)**

1. 用**Thompson构造法**将其转换为NFA（NFA状态从0开始编号），识别abbc。（6分）

答：



识别abbc的状态转换过程0🡪7🡪8🡪9🡪10🡪9🡪10🡪11🡪12🡪19

2. 用**子集构造法**将得到的NFA转换为DFA，画出最终的状态转换图，识别abbc。（10分）

答：

A=closure({0})={0, 1, 2, 4, 7, 13}

move(A, a)={2, 3, 4, 8, 9, 11,14}=B

move(A, b)={5}=C

move(B, a)={2, 3, 4}=D

move(B, b)={5, 9, 10, 11, 15, 16, 18, 19}=E

move(B, c)={12, 19}=F

move(C, c)={6, 19}=G

move(D, a)=D

move(D, b)=C

move(E, b)={9, 10, 11}=H

move(E, c)={6, 12, 16, 17, 18, 19}=I

move(H, b)=H

move(H, c)=F

move(I, c)={16, 17, 18, 19}=J

move(J, c)=J



识别abbc状态转换过程A🡪B🡪E🡪H🡪F

3. 将DFA最小化，画出最终的状态转换图。（8分）

答：

加入死状态K，初始状态分组{A, B, C, D, H, K}、{E, F, G, I, J}

move({A, B, C, D, H, K}, a)={B, D, K, D, K, K}，无法分裂

move({A, B, C, D, H, K}, b)={C, E, K, C, H, K}，分裂{A, C, D, H, K}、{B}

move({A, C, D, H, K}, a)={B, K, D, K, K}，分裂{A}、{C, D, H, K}

move({C, D, H, K}, c)={G, K, F, K}，分裂{C, H}、{D, K}

move({C, H}, b)={K, H}，分裂{C}、{H}

move({D, K}, b)={C, K}，分裂{D}、{K}

move({E, F, G, I, J}, a)={K, K, K, K, K}，无法分裂

move({E, F, G, I, J}, b)={H, K, K, K, K}，分裂{E}、{F, G, I, J}

move({F, G, I, J}, c)={K, K, J, J}，分裂{F, G}、{I, J}，无法再分



|  |
| --- |
| **得 分** |
|  |

1. 语法分析算法（16分）

对下面文法：

**S → A a | B b A → d A | d B→ c B | c**

1. 指出其终结符集合、非终结符集合、开始符号（2分）

答：

终结符：a、b、c、d

非终结符：S、A、B

开始符号：S

2. 计算所有非终结符的First函数和Follow函数结果（2分）

答：First(S)={c, d}，First(A)={d}，First(B)={c}

Follow(S)={$}，Folow(A)={a}，Follow(B)={b}

3. 构造SLR分析表（从拓广产生式开始、所有产生式从0开始编号），对句子cccb进行分析，画出语法树（12分）。

答：计算LR(0)项目集规范族

I0=closure({S’🡪•S})={ S’🡪•S, S🡪•Aa, S🡪•Bb, A🡪•dA, A🡪•d, B🡪•cB, B🡪•c}

I1=goto(I0, S)={ S’🡪S• }

I2=goto(I0, A)={ S🡪A•a }

I3=goto(I0, B)={ S🡪B•b }

I4=goto(I0, d)={ A🡪d•A, A🡪d•, A🡪•dA, A🡪•d }

I5=goto(I0, c)={ B🡪c•B, B🡪c•, B🡪•cB, B🡪•c }

I6=goto(I2, a)={ S🡪Aa• }

I7=goto(I3, b)={ S🡪Bb• }

goto(I4, d)= I4

I8=goto(I4, A)={ A🡪dA• }

goto(I5, c)= I5

I9=goto(I5, B)={ B🡪cB• }

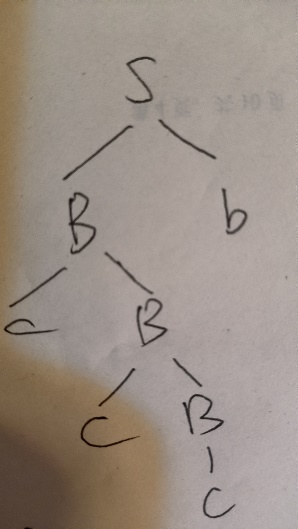
SLR分析表：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | $ | S | A | B |
| 0 |  |  | s5 | s4 |  | 1 | 2 | 3 |
| 1 |  |  |  |  | acc |  |  |  |
| 2 | s6 |  |  |  |  |  |  |  |
| 3 |  | s7 |  |  |  |  |  |  |
| 4 | r4 |  |  | s4 |  |  | 8 |  |
| 5 |  | r6 | s5 |  |  |  |  | 9 |
| 6 |  |  |  |  | r1 |  |  |  |
| 7 |  |  |  |  | r2 |  |  |  |
| 8 | r3 |  |  |  |  |  |  |  |
| 9 |  | r5 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

分析过程：

|  |  |  |
| --- | --- | --- |
| 状态栈 | 输入 | 动作 |
| $0 | cccb$ |  |
| $0c5 | ccb$ |  |
| $0c5c5 | cb$ |  |
| $0c5c5c5 | b$ | B🡪c归约 |
| $0c5c5B9 | b$ | B🡪cB归约 |
| $0c5B9 | b$ | B🡪cB归约 |
| $0B3 | b$ |  |
| $0B3b7 | $ | S🡪Bb归约 |
| $0S1 | $ | acc |
|  |  |  |

语法树：



|  |
| --- |
| **得 分** |
|  |

1. 语法制导翻译（10分）

写出描述类型表达式（支持基本类型char、int、real、bool以及类型构造符array、×、record、pointer和🡪）的CFG，为其设计语法制导定义，构造对应的抽象语法树（叶节点为基本类型、非叶节点为类型构造符，假设已有创建叶节点的辅助函数mkleaf和创建非叶节点的辅助函数mknode，在语法制导定义中直接调用即可）。

答：

T 🡪array ‘(’ I , T1 ‘)’ { T.nptr=mknode(“array”, I.nptr, T1.nptr); }

| T1 ‘×’ T2 { T.nptr=mknode(“×”, T1.nptr, T2.nptr); }

| pointer ‘(’ T1 ‘)’ { T.nptr=mknode(“pointer”, T1.nptr); }

| record ‘(’ T1 ‘)’ { T.nptr=mknode(“record”, T1.nptr); }

| T1 ‘🡪’ T2 { T.nptr=mknode(“🡪”, T1.nptr, T2.nptr); }

| boolean { T.nptr=mkleaf(“boolean”); }

| char { T.nptr=mkleaf(“char”); }

| integer { T.nptr=mkleaf(“integer”); }

| real { T.nptr=mkleaf(“real”); }

I🡪num1 .. num2 { I.nptr=mkleaf(“range”, num1.val, num2.val); }

|  |
| --- |
| **得 分** |
|  |

1. 中间代码生成（6分）

采用临时变量重用算法将下面表达式翻译为三地址码，临时变量名为t1、t2、…。

**a+a\*(a+(a+a)\*(a+a+a) +a)**

答：

t1 := a + a;

t2 := a + a;

t2 := t2 + a;

t1 := t1 \* t2;

t1 := a + t1;

t1 := t1 + a;

t1 := a \* t1;

t1 := a + t1;