课程期末考试参考答案

考试中心填写：

|  |
| --- |
| \_\_\_\_年\_\_\_月\_\_\_日  考 试 用 |

|  |
| --- |
| 专业班级： |
| 学号： |
| 姓名： |

装订线（题目不得超过此线）

湖南大学课程考试试卷

**湖南大学教务处考试中心**

课程名称： 编译技术 ；课程编码：CS05067 试卷编号： A ；

**考试形式：** 闭卷 ；考试时间：120分钟

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题 号 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | 十 | 总分 |
| 应得分 | 4 | 18 | 18 | 18 | 14 | 10 | 18 |  |  |  | 100 |
| 实得分 |  |  |  |  |  |  |  |  |  |  |  |
| 评卷人 |  |  |  |  |  |  |  |  |  |  |  |

1. 对于仅只含a和b两个字符的字符串中，a和b出现的次数相等。请为其定义文法。**（4分）**

S→aSb | bSa | ab | ba 或者：S→aSb | bSa | ε

1. 数值常量的例子有123, 123.01，123E2， 123.01E3，其中第一个为整数，第二个为实数，第三个和第四个为科学计数法表达方式。1）写出数值常量的正则表达式**(6分)**；2）画出所得正则表达式的NFA **(6分)**，再由子集构造法得出DFA 的状态转换表Dtran，再画出其DFA图。**(6分)**

digit → [0 - 9]

digits → digit+

integer → digits

float →digits.digits

number → digits (.digits )?( E digits)?

**ε**

**ε**

**ε**

**ε**

**[0-9]**

**E**

5

**ε**

**.**

[0-9]

[0-9]

0

7

**6**

**4**

**3**

**2**

**1**

1. **start**

**ε**

**ε**

1. **ε**

状态转换表Dtran

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NFA中对应的状态集合 | 状态集合序号  (即DFA状态序号) | 驱动字符下的目标状态序号 | | |
| digit | ‘.’ | ‘E’ |
| {**0**} | 0 | 1 |  |  |
| ｛0,**1**,4,7｝ | 1 | 1 | 2 | 3 |
| ｛**2**｝ | 2 | 4 |  |  |
| ｛**5**｝ | 3 | 5 |  |  |
| ｛2,**3**,4,7｝ | 4 | 4 |  | 3 |
| ｛5,**6**,7｝ | 5 | 5 |  |  |

‘E’

2

*digit*

*digit*

1

‘.’

*digit*

4

‘E’

3

*digit*

5

*digit*

*digit*

0

1. 现有文法G[S]: S→ S+S | SS | (S) | S\* | a。其中*S*为非终结符，+，\*，（，），a为终结符。输入串(a+a)\*a满足该文法。
2. 请对该文法做消左递归处理，得到一个不含左递归的等价文法。**（6分）**
3. 对由(1)得到的等价文法，计算其中每个非终结符的FOLLOW函数值，得出该文法的LL预测分析表(也叫LL语法分析表)，说明该文法不为LL（1）文法的理由。（**6分）**
4. 对于由(2)得到的LL预测分析表，如果某个格中含有两个产生式，且其中一个为S’→ε，则去掉S’→ε。于是消除了文法的二义性。然后对输入串(a+a)\*a执行最左推导，只要求写出每一步推导后的句型（也叫格局）。**（6分）**

答：

①消除左递归后得到如下的文法产生式：

1. S**→(S)**S’
2. S**→a**S’
3. S’**→+S**S’
4. S’**→S**S’
5. S’**→\***S’
6. S’**→ε**

②计算FIRST():

S**→(S)**S’ **⇒ FIRST(**S) = {**(**}

S**→a**S’ **⇒ FIRST(**S) += { **a** } = {a, **(**}

S’**→+S**S’ **⇒ FIRST(**S’) = { **+** }

S’**→ε** **⇒ FIRST(**S’) += { **ε** } = { +, **ε** }

S’**→S**S’ **⇒ FIRST(**S’) += **FIRST(**S) ={a, **(,** +, **ε** }

S’**→\***S’**⇒ FIRST(**S’) +={\*} ={a, **(,** +, \*, **ε** }

因此：

**FIRST(**S) ={ **(**, **a** }

**FIRST(**S’) = {a, **(,** +, \*, **ε** }

**③**计算FOLLOW():

1）由高到低排列：S，S’;

2）FOLLOW(S)={$}； FOLLOW(S’)=∅；

3）右边含S的产生式有：S**→(S)**S’； S’**→+S**S’; S’**→S**S’

由S**→(S)**S’⇒ FOLLOW(S)+= {)} = {), $};

由S’**→+S**S’⇒ FOLLOW(S)+= **FIRST(**S’) - ε = {a, +, \*, **(,** **)**, $};

又由ε∈**FIRST(**S’) ⇒ FOLLOW(S)+= **FOLLOW(**S’) ；

由S’**→S**S’ ⇒ FOLLOW(S)+= **FIRST(**S’)- ε: 前面已经有了，这里没意义了;

又由ε∈**FIRST(**S’) ⇒ FOLLOW(S)+= **FOLLOW(**S’)；前面已经有了，这里没意义

5）右边含S’的产生式：S**→(S)**S’; S**→a**S’；S’**→+S**S’；S’**→S**S’； S’**→\***S’

上述5个产生式都是 ⇒ FOLLOW(S’)+= FOLLOW(S) ;

因此： **FOLLOW(S)={a, +, \*,(, ), $};**

**FOLLOW(S’)={a, +, \*, (, ), $};**

④构建预测分析表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 非终结符号 | 输入符号 | | | | | |
| ( | ) | **+** | **\*** | a | $ |
| S | S**→(S)**S’ |  |  |  | S**→a**S’ |  |
| S’ | S’**→S**S’  S’**→ε** | S’**→ε** | S’**→+S**S’  S’**→ε** | S’**→\***S’  S’**→ε** | S’**→S**S’  S’**→ε** | S’**→ε** |

**由此可知，该文法不是LL（1）文法。**

**(a+a)\*a的推导：**

S⇒ **(S)S’**⇒ **(a**S’**)**S’ ⇒ **(a**+SS’**)**S’ ⇒ **(a+a**S’S’**)**S’ ⇒ **(a+a**S’**)**S’ ⇒ **(a+a)**S’ ⇒ **(a+a)\***S’

⇒ **(a+a)\*S**S’ ⇒ **(a+a)\*a**S’S’ ⇒ **(a+a)\*a**S’ ⇒ **(a+a)\*a**

1. 对于文法G[E]: E→ E+E | E\*E | (E) | id。*E*为非终结符，+，\*，（，），id为终结符。该文法描述了算术运算表达式。该文法没有表达出运算的优先级，不为SLR(1)文法，是二义性文法。不过基于运算优先级（乘法高于加法，括号高于乘法）来构造该文法的DFA，可消除其二义性，使其成为SLR(1)文法。
2. 构造该文法的DFA，要求写出DFA中每个状态的LR(0)项集。构造中要体现运算的优先级，即基于优先级来确定DFA的状态中应包含的非核心项，把与运算优先级不符的非核心项去掉。**(9分)**
3. 由（1）所得的DFA，得出无二义的LR语法分析表。**(9分)**
4. *I*0

E'→•E

E→ •E + E

E→ •E\*E

E→ •(E)

E→ •id

1. *I*1

E'→E•

E→ E• + E

E→ E•\*E

1. *I*4

E→ E +• E

E→ •E\*E

E→ •(E)

E→ •id

1. *I*7

E→ E + E•

E→ E•\*E

E

+

E

Accept

$

1. *I*5

E→ E\*•E

E→ •(E)

E→ •id

1. *I*8

E→ E \* E•

\*

E

1. \*

\*

\*

1. *I*3

E→ id•

1. *I*2

F→ (•E)

E→ •E + E

E→ •E\*E

E→ •(E)

E→ •id

1. *I*6

F→ (E• )

E→ E •+ E

E→ E•\*E

1. *I*9

E→ (E) •

1. id

id

(

1. (
2. (

id

E

)

id

(

+

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 状态 | ACTION | | | | | | GOTO |
| id | + | \* | ( | ) | ＄ | E |
| 0 | s3 |  |  | s2 |  |  | 1 |
| 1 |  | s4 | s5 |  |  | acc |  |
| 2 | s3 |  |  | s2 |  |  | 6 |
| 3 |  | r4 | r4 |  | r4 | r4 |  |
| 4 | s3 |  |  | s2 |  |  | 7 |
| 5 | s3 |  |  | s2 |  |  | 8 |
| 6 |  | s4 | s5 |  | s9 |  |  |
| 7 |  | r1 | s5 |  | r1 | r1 |  |
| 8 |  | r2 | r2 |  | r2 | r2 |  |
| 9 |  | r3 | r3 |  | r3 | r3 |  |

1. 给定文法G[E]： E→E1+E2 | E1 / E2 | (E1) | id。该文法表达了运算表达式，其中只有除法，加法，括号这三种运算，非终结符只有E，终结符有+，/, (, ), id。翻译目标是在画布上绘出运算表达式（即可视化）。现举例说明：对于输入串a / (b + c) + d, 其可视化图为：

a

b+c

+d

注意：a在横向上要居中。无论是终结符还是非终结符，都有综合属性w和h，表示宽度和高度，另有继承属性x和y，表示左上角的坐标值。最终要得出输入串中所有非终结符+, /, id 的x,y,w,h属性值，于是就可画出可视化图。思路是先执行LR语法分析，得出语法分析树中每个结点的w和h属性值，再自顶向下扫描语法分析树，得出每个结点的x和y 属性值。已知终结符中的+和id的h = h0，w= w0。终结符/的h = 0.2h0， w则取两个运算数的大者。已知树根的x =x0，y= y0。

1）请针对该文法，设计出该翻译目标的SDD。**(8分)**

2）对输入串a / (b + c) + d，画出其注释语法分析树。**(6分)**

|  |  |
| --- | --- |
| 产生式 | 语义规则 |
| E→E1 / E2 | E.w = max(E1.w, E2.w) |
| E.h = E1.h +0.2h0 +E2.h |
| E1.x = E.x + (max(E1.w, E2.w) - E1.w) /2 |
| E1.y = E.y |
| /. x = E.x + (max(E1.w, E2.w) ) |
| /. y = E.y + E1.h |
| E2.x =E.x + (max(E1.w, E2.w) - E2.w) /2 |
| E2.y = E.y +E1.h +0.2h0 |
| E→E1 + E2 | E.w = E1.w +h0 + E2.w |
| E.h = max(E1.h , E2.h) |
| E1.x = E.x |
| E1.y = E.y+ (max(E1.h, E2.h) - E1.h) / 2 |
| +. x = E.x + E1.x |
| +. y = E.y + (max(E1.h, E2.h) - h0) / 2 |
| E2.x = E.x + E1.w + w0 |
| E2.y = E.y + (max(E1.h, E2.h) - E2.h) / 2 |
| E→(E1) | E.w = E1.w |
| E.h = E1.h |
| E1.x = E.x |
| E1.y = E.y |
| E→id | E.w = w0 |
| E.h = h0 |
| id.x = E.x |
| id.y = E.y |

**E**

**E / E**

**E**

**+**

**E**

**id**

**( E )**

**E + E**

**id**

**id**

**id**

**x = x0**

**y = y0+1.2h0**

**w =w0**

**h = h0**

**x = x0+2w0**

**y = y0+1.2h0**

**w =w0**

**h = h0**

**x =x0+w0**

**y = y0+1.2h0**

**w =w0**

**h = h0**

**x = x0**

**y = y0+1.2h0**

**w =3w0**

**h = h0**

**x = x0+w0**

**y = y0**

**w =w0**

**h = h0**

**x = x0**

**y =y0+ h0**

**w =3w0**

**h = 0.2h0**

**x = x0**

**y = y0**

**w =3w0**

**h = 2.2h0**

**x = x0+3w0**

**y = y0+0.6h0**

**w =w0**

**h = h0**

**x = x0**

**y = y0**

**w =5w0**

**h = 2.2h0**

**x = x0+4w0**

**y = y0+0.6h0**

**w =w0**

**h = h0**

1. C语言的文法中，非终结符S表示语句和语句序列，有综合属性nextList和code，非终结符B表示逻辑运算表达式，有综合属性trueList，falseList，和code。基于LR语法分析，就中间代码生成这一翻译目标，画出while语句（其文法为S→while(B) S1）的中间代码布局图 **(4分)**；针对产生式S→while(B) S1和S→S1S2, 写出其中间代码生成的SDT。提示：goto回填通过调用函数backPatch来完成。**(6分)**

**L1: B.code**

**L2: S1.code**

* 1. **goto L1**

*S*→while *M* ( *B*) **{** backpatch(B->truelist, nextinstr); **}** *S*1

**{** backpatch(S1->nextlist, M->instr);

gen('goto' M->instr);

1. >nextlist = *B*->falselist;

**}**

*M*→ε **{**  M->instr = nextInstr; **}**

*S*→*S*1 **{** backpatch(S1->nextlist, nextinstr); **}** *S*2 **{** S->nextlist = S2->nextlist; **}**

1. 有如下C语言源代码段。其中所有变量都为局部变量，且定义为：

int a, b, c, i, j, x[12][18]。

while( i < 12) {

i ++;

j ++;

c = x[i][j] + 2;

if (c > 0 || b < 10 && a > 20 )

b = b - c;

else

a = a + c;

}

i = 0;

1. 基于LR语法分析，将上述源代码翻译成中间代码（用三地址码表达），填入如下的中间代码表中。设翻译出的第一行中间代码的行号为50。**(10分)**

2）对(1)中生成的中间代码，将其翻译成目标代码。设int宽度为4Bytes，局部变量的基地址存于寄存器R0中，变量a的偏移量为0。另有寄存器R1，R2，R3。运算指令要求：一个操作数在寄存器中，另一操作数可在内存中，也可在寄存器中，运算结果在寄存器中。**(8分)**。

|  |  |  |  |
| --- | --- | --- | --- |
| RowId | 中间代码 | 位置标识 | 目标代码 |
| 50 | if i < 12 goto **52** | **50:** | LD R1, 12(R0) |
| 51 | goto **68** |  | SUB R2, R1, 12 |
| 52 | i = i + 1 |  | BLTZ R2, **52** |
| 53 | j = j + 1 |  | Br **68** |
| 54 | t0 = i \* WIDTH(int [18]) | **52:** | ADD R1, R1, 1 |
| 55 | t1 = j \* WIDTH(int) |  | ST 12(R0), R1 |
| 56 | t2 = t0 + t1 |  | LD R2, 16(R0) |
| 57 | c = a[t2] + 2 |  | ADD R2, R2, 1 |
| 58 | if c > 0 goto **64** |  | ST 16(R0), R2 |
| 59 | goto **60** |  | MUL R3, R1, 72 |
| 60 | if b < 10 goto **62** |  | MUL R2, R2, 4 |
| 61 | goto **66** |  | ADD R3, R3, R2 |
| 62 | if a > 20 goto **64** |  | ADD R3, R3, 20 |
| 63 | goto **66** |  | LD R2, R3(R0) |
| 64 | b = b - c |  | ADD R3, R2, 2 |
| 65 | goto **50** |  | ST 8(R0), R3 |
| 66 | a = a + c |  | SUB R2, 0，R3 |
| 67 | goto **50** |  | BLTZ R2, **64** |
| 68 | i = 0 |  | Br **60** |
|  |  | **60:** | LD R2, 4(R0) |
| ... |  |  | SUB R1, R2，10 |
|  |  |  | BLTZ R1, **62** |
|  |  |  | Br **66** |
|  |  | **62:** | LD R1, 0(R0) |
|  |  |  | SUB R1, 20，R1 |
|  |  |  | BLTZ R1, **64** |
|  |  |  | Br **66** |
|  |  | **64:** | SUB R2, R2，R3 |
|  |  |  | ST 4(R0), R2 |
|  |  |  | Br **50** |
|  |  | **66:** | LD R1, 0(R0) |
|  |  |  | ADD R2, R1, R3 |
|  |  |  | ST 0(R0), R2 |
|  |  |  | Br **50** |
|  |  | **68:** | ST 12(R0), 0 |