华南师范大学

计算机学院 2019—2020 学年第（二）学期期末考试

《编译原理》试卷（A）

年级_____ 班级_____ 姓名_____学号_____

题号	一	二	三	四	五	六	总分
得分							

一、基础知识题（1 题，16 分）

1.请写出实验一 C/C++语言中单词“数”的正则表达式、直接画出其对应的 DFA 并写出其相应的词法分析程序代码段。（提示，C 语言中数包括有整数、正整数、负整数、十进制数整数、八进制整数、十六进制整数、带科学计算的浮点数）

二、 正则表达式→DFA（1 题，16 分）

1.我们知道一个单词的正则表达式可以转换为DFA,接着就可以根据该DFA 图写出实现该单词的词法分析程序。现请写出将 DFA 图转换为词法分析的转换程序。【提示：你需要先写出 DFA 图的存储结构，接着才写出基于该结构下的转换程序。】

三、 自顶向下分析设计题（1 题，共 17 分）

1.如果要采用递归下降分析方法（或称递归子程序分析法）生成逻辑表达式对应的语法树，那我们需要解决的问题有：

（1）定义逻辑表达式的文法规则;

（2）逻辑表达式语法树的存储结构;

（3）写出文法规则对应的语法树生成算法。

现请你解决以上的三个问题。

四、LR 分析题（1 题，18 分）

1.如果我们为教科书中的 TINY 语言增加书写格式类似于 C 语言的 for 循环语句。那么请完成以下问题:【TINY 语言文法规则见后面附录】

（1）请写出所添加语句对应的文法规则。

（2）判断该文法是否为 LR(0)文法。请说明原因。

（3）判断该文法是否为 SLR(1)文法。请说明原因。

五、语义分析题（ 1 题，18 分 ）

1.改写逻辑表达式的语义函数，使得逻辑变量 A 以及关系运算 $I^{(1)}$ rop $I^{(2)}$不是按通常的方式翻译成两个相继的四元组：

（1）（ JNZ， A， ，0 ）或（ J_{rop}， $I^{(1)}$ ， $I^{(2)}$ ，0 ）

（2）（ J，，，0 ）

而是翻译成如下的一个四元组：

（ JZ， A， ，0 ） 或 （ J_{nrop}， $I^{(1)}$ ， $I^{(2)}$ ，0 ）

使得当逻辑变量 A 和关系运算 $I^{(1)}$ rop $I^{(2)}$为真的情况下不发生转移（即自动往下执行），当逻辑变量 A 和关系运算 $I^{(1)}$ rop $I^{(2)}$为假是才发生转移，从而产生较高的目标代码。请写出要改写的逻辑表达式文法规则以及相应的语义函数。

[参考翻译示例] 如有逻辑表达式： ！A || ((B<C) && ！(D>E))
则翻译结果为：

100 (JZ, A , , 0)
101 (J>=,B,C,103)
102 (J<=,D,E,100)
103

六、综合分析设计（1 题，15 分）

1.在实验三中,我们为教科书中的 TINY 语言进行了语法的扩充。我们知道,该 TINY 语言不支持类似于 C 语言的数据类型定义语句。现想为其做扩充，让其能支持这语句。请你根据实验三的实践经验，完成以下的内容:【TINY 语言文法规则见后面附录】

（1） 文法规则该如何改写扩充。【说明:支持数据类型为 int,bool,char,real】
（2） 根据实际的需要，该如何改写词法、语法的分析程序，请简明扼要地描述你的做法。
（3） 根据实际的需要，写出该扩充文法的语义分析程序。

附录：教科书中有关 TINY 语言的文法规则列表

```
program→stmt-sequence
stmt-sequence→ stmt-sequence ; statement  |statement
statement→ if-stmt |repeat-stmt |assign-stmt | read-stmt|write-stmt
if-stmt→if exp then stmt-sequence end
        | if exp then stmt-sequence else stmt-sequence  end
repeat-stmt→repeat stmt-sequence until exp
assign-stmt→identifier := exp
read-stmt→read identifier
write-stmt→write exp
exp→simple-exp comparison-op simple-exp |simple-exp
comparison-op→ < | =
simple-exp→simple-exp addop term  | term
addop→ + |-
term→term mulop factor | factor
mulop→ * | /
factor→(exp) | number | identifier
```