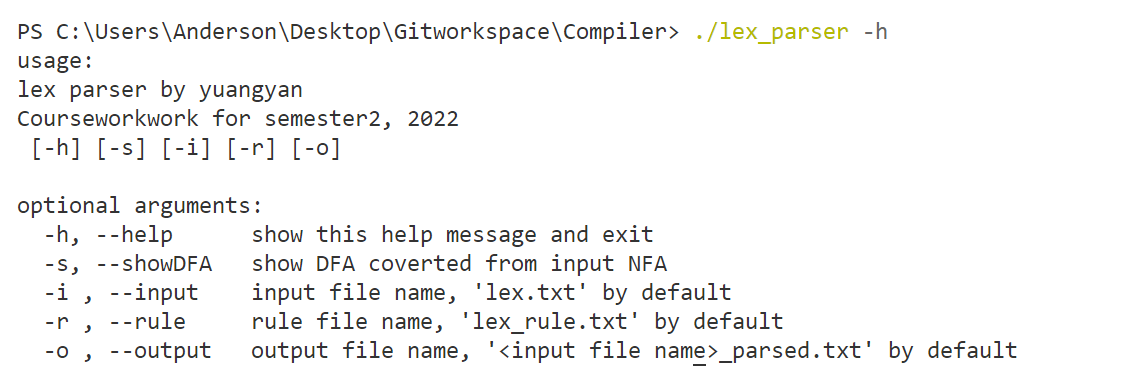
实验一 词法分析

一 程序说明

词法分析的程序的可执行文件名称为lex\_parser.exe，源代码文件为lex\_parser.py

* 1. 操作说明

在命令行输入命令 ./lex\_parser -h获取帮助信息：



在词法分析程序中共有5个可选参数，分别为：

-h, --help: 显示帮助信息

-s, --showDFA: 在程序运行的过程中打印从输入NFA转换成的DFA

-i, --input: 选择输入待分析文本路径，不选该参数则默认为同目录下’lex.txt’文件

-r, --rule: 选择词法分析规则路径，不选该参数则默认为同目录下’lex\_rule.txt’文件

-o, --output: 选择输出token表路径，不选该参数则默认为同目录下’<输入文件名>\_parsed.txt’文件

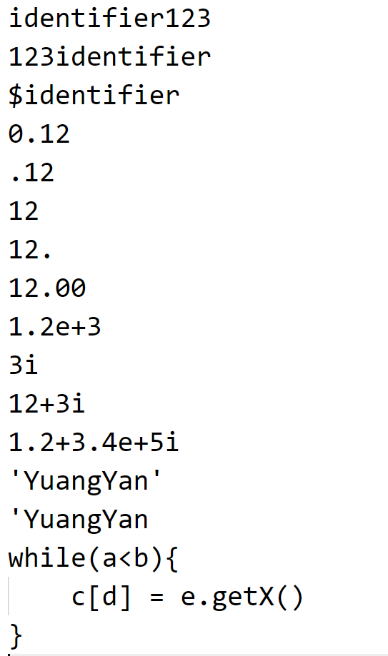
输入命令格式为

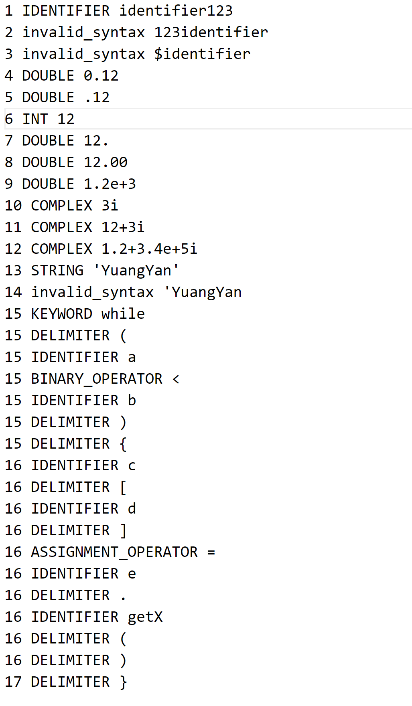
./lex\_parser (-i [待分析文本路径]) (-r [词法分析规则文件]) (-o [输出token路径]) (–showDFA)

例如：./lex\_parser -i input.txt -o output.txt -r rule.txt --showDFA

* 1. 输入输出样例

输入文件lex.txt: 输出token文件：lex\_parsed.txt





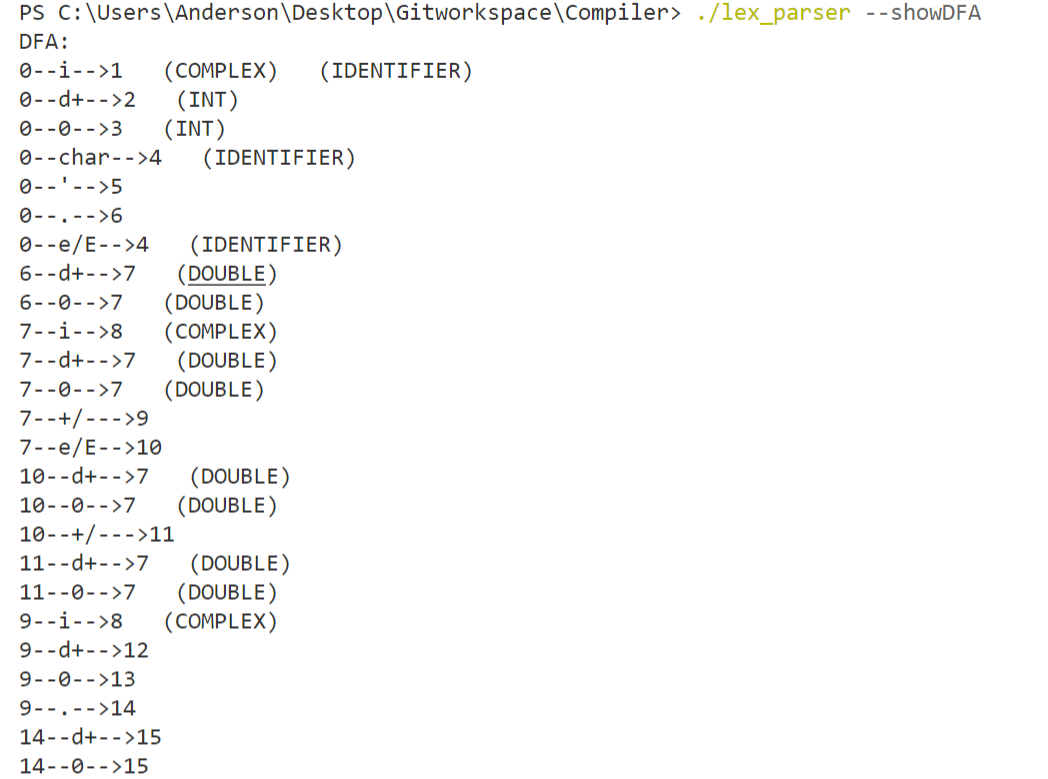
1.3 输入样例解释

在lex\_parsed.txt文件中每行格式为[行号] [类型] [内容]

在设计的文法规则中

1. 标识符只能由字母开头，因此123identifier和$identifier被识别为invalid\_syntax
2. 0.12，12.，12.00 12e+3均能识别为double类型
3. 复数的实部和虚部均可为int和double类型，兼容科学计数法，如1.2+3.4e+5i
4. 字符串常量由成对的单引号标识，’YuangYan缺少一个单引号，因此被识别为invalid\_syntax
5. 可以识别连在一起的不同类型，如while(a<b)被识别为关键字，限定符，标识符，赋值操作符，限定符

当加入—showDFA参数时，程序运行过程中会打印转换后的DFA，如图：

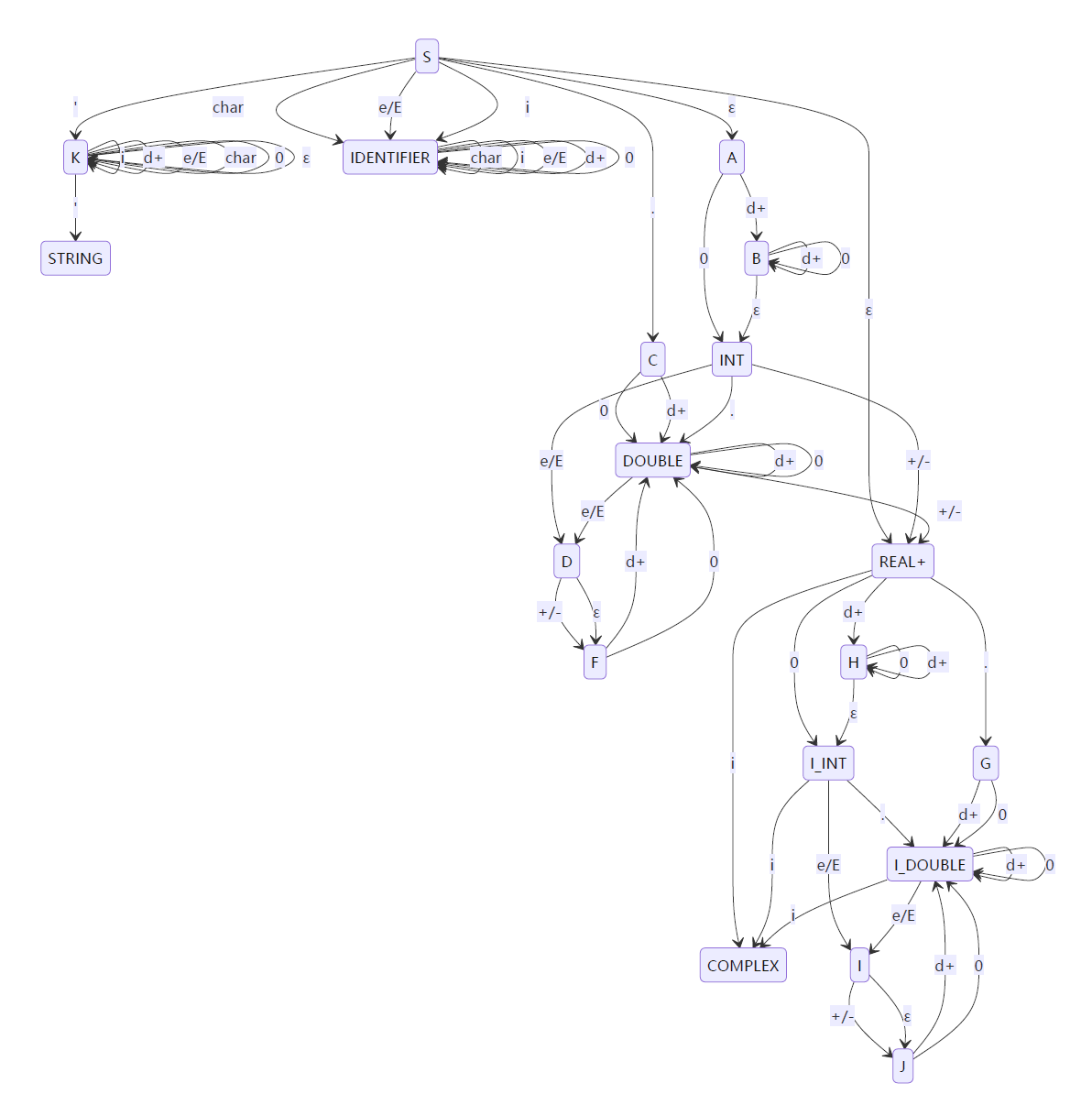


二 设计思路

2.1 文法设计

2.1.1NFA设计

在词法分析程序中，为分析器设计的文法NFA如图所示：



其中可接受状态为IDENTIFIER，STRING，DOUBLE，COMPLEX，分别为标识符，字符串，浮点数，复数

转换边解释：

i：字母i

e/E：小写字母e或大写字母E

char：其他字母字符

d+：1-9的数字字符

0：字符0

+/-：字符+或字符-

ε：空输入

在处理前先对所有字符进行分类，简化了NFA的复杂程度

2.1.2 关键词，操作符，限定符设置

关键词：for, while, if, else, return, break, continue, def, class, int, double

赋值操作符：+=，-=，\*=，/=，\*=， //=，%=，=

二元运算符：and，or，xor，==，*\*，+，-，*，/，//，%，in，<，>，<=，>=

单目运算符：！，not

限定符：. : , ; [ ] { } ( )

2.2 NFA到DFA转换

2.3 DFA和输入字符串的匹配

2.4 分析命令行语句

2.5 算法总流程图

2.6 数据结构