**Department of Computing**
**Hong Kong Polytechnic University**

# COMP 3438 SYSTEM PROGRAMMING

## Assignment Three (Total marks: 100)
## Deadline: 23:55, 22 Nov. 2014

Note: This programming assignment needs to be done **individually**. Discussion with classmates is encouraged. But if the plagiarism was found, it would be reported to the department and the students involved would be kicked out the university.

In this assignment, you are required to write a lexical analyzer for the given example language – Simple Language (SL). The following is a sample program with SL.

*The sample test program:*

```
var  A, B, C;
begin
    A = 5;
    B = 6;
    C = (a + b) / 2.0;
end.
```

In SL, a program begins with the declaration for variables using the keyword “var” following variable names. The program segment starts with the keyword “begin” and ends with the keyword “end” and “.”. A program segment consists of statements. A statement has two cases: (1) It can be another program segment; (2) It can be an assignment statement consisting of identifiers, “=” and expressions. An expression consists of number (integer and real number), operators (+,-, *, /).

The syntactic elements and their corresponding tokens are listed as follows:

| Syntactic Element | Token Name | Syntactic Element | Token Name |
|---|---|---|---|
| var | KEYWORD | + | PLUS |
| begin | KEYWORD | - | MINUS |
| end | KEYWORD | * | MUL |
| , | COMMA | / | DIV |
| ; | SEMICOLON | ( | LBRACE |
| = | ASSIGN | ) | RBRACE |
| . | PERIOD | A string starting with letter and following with letter or digit | ID |
| Integer (e.g. 124) or real number (e.g. 1.2) | NUM | | |

**What you need to do:**

1. Give the regular expression describing each token (20%).
2. Construct Finite Automata to recognize the tokens (20%).
3. Write a program (a lexical analyzer) with **C/C++/Java** for SL based on the FA. For tokens with the types ID, KEYWORD and NUM, you should give the lexeme (value) of the token (see the sample output as an example). Describe each function of your program and its detailed procedure, and explain how the program works (how each token can be identified step by step). (30%)
4. Test the lexical analyzer with some example programs written in SL. The output of sample program above is given at the end of this document (30%).

**What to Submit**

**An assignment report includes the following four parts:**

**Part 1**: Regular expressions of all tokens (20 marks)
**Part 2**: Finite automata for each token (20 marks)
**Part 3**: Describe each function of your program and its detailed procedure, and explain how the program works (how each token can be identified step by step). Provide the source code of your program (30 marks).
**Part 4**: Give the source code (in a separate file), and a readme file to describe the procedure how to compile and run your program with the sample test program or any other programs made by you. (30 marks)

**From the Blackboard, please submit a zip file including your assignment report, the readme file and the source code.**

**Marking standards:**

1. For Parts 1 and 2, you need to provide regular expressions and finite automata for all tokens.
2. In Part 3, your program must match what you provided in Part 2. We want to see if you clearly understand how to generate a program based on the finite automata. Basically, each token should be correctly identified based on what you describe. **Note that you will get zero if you only provide the source code without the above description requested**.
3. For Part 4, the tutor will follow the procedure submitted to compile your code and test your program using the sample test program. If your program can be compiled successfully and can process the sample test program correctly, you will get 10 marks. Five other test programs will be inputted to your program, and each is specially designed to test some tokens (4 marks for each).

**The sample output for the above sample program:**

| **Token** | **Lexeme** |
|---|---|
| KEYWORD | var |
| ID | A |
| COMMA | |
| ID | B |
| COMMA | |
| ID | C |
| SEMICOLON | |
| KEYWORD | begin |
| ID | A |
| ASSIGN | |
| NUM | 5 |
| SEMICOLON | |
| ID | B |
| ASSIGN | |
| NUM | 6 |
| SEMICOLON | |
| ID | C |
| ASSIGN | |
| LBRACE | |
| ID | a |
| PLUS | |
| ID | b |
| RBRACE | |
| DIV | |
| NUM | 2.0 |
| SEMICOLON | |
| KEYWORD | end |
| PEROID | |