

## CENG 442 Project (Part 1)

**Assigned:** March 04, 2015

**Due:** March 29, 2015

### Definition and Lexical Analysis of a Robot Programming Language

This year's project is about the design of a new programming language for controlling a mobile robot. Suppose that you are working as part of a research team building a robotic platform. Your colleagues in the mechanical engineering department are almost done with the construction of a robot to be deployed in Mars. This robot will be capable of moving forward and back, turning in place, grabbing/releasing objects as well as sending/receiving data to/from a supervisor. The robot also has various sensors to perceive the environment, returning floating point values associated with their measurements. The robot can turn 1 degree at a time, and move forward or back in steps of 1mm. Your team in the CS department is asked to design a Programming Language to for this robot.

The project will be completed in two parts. The first part (this assignment) involves the design of the syntax for your programming language including at least the following components:

- Syntactic components for the top level program definition
- Primitive functions: move (both forward and back), turn, grab, release, read sensor given the sensor ID, send, receive.
- Variable names,
- Assignment operator,
- Arithmetic expressions,
- Conditional statements (e.g., if-then, if-then,else)
- Loops (e.g., for, while),
- Syntactic components for program and function definitions and function calls,
- Comments

Your design should take the form of a BNF grammar, precisely defining the language based on your token definitions used as terminal symbols. We expect your group to do the following for your solution to this project:

- Give a name to your robot programming language
- Include in your report, a summary of tokens that are used by the syntax of your language as terminal symbols
- Include in your report, the grammar for your language in BNF form.
- With the main components of your language in place, write a scanner (lexical analyzer) using the lex tool available on Linux/Unix systems. The scanner is expected to read the input stream and output a sequence of tokens corresponding to the lexemes as defined above. But, since at this point, you do not have the necessary tools to write a full-blown parser, test your scanner by having it print messages on the screen corresponding to recognized tokens. For example, if your language supports an expression such as ... 1.45+453.43\*32.2 ..., it would generate the output ... FLOAT PLUS FLOAT MULT FLOAT ... on the screen.

The second part of the project (to be assigned later) will involve the yacc implementation of a parser for your robot programming language. None of these project stages require you to implement any execution facilities for your language. Both projects are mainly about SYNTAX.

- You will be working on this project in groups of at most three students. Form your group and email me by March 16, 2015.
- For both parts of the project, the members of the groups will be the same.

Email your program (the lex description file) along with the report to me by 23:59, March 29, 2015.

Your report should contain the following components:

- Name, ID and sections of ALL group members
- Verbal definitions of lexical components included in your language.
- Description of how the definitions of these components (nontrivial ones such as identifiers etc.) are implemented in lex.
- The complete BNF grammar for your language, where the tokens and lexemes for the above components are used as terminal symbols.
- An example program, written in your programming language, designed to make the robot do the following:
  1. Read the value of sensor 0 until it returns 1
  2. Turn right (Your program should include a function/method to turn right, and call that function at this point) until sensor 1 returns a value of 0.5
  3. Go ahead until sensor 1 returns 1.0
  4. Compute the product of the sum of the sensors 2, 3, 4 and sum of the sensors 5, 6, 7
  5. Grab the object at that point
  6. Go back to the starting point by going back until sensor 1 reads 0.5 and turning left until sensor 1 reads 0
  7. Release the object.
  8. Send a message to the master

### **What to hand in:**

(Following three items must be in electronic format. Mailed to me ([melih@cankaya.edu.tr](mailto:melih@cankaya.edu.tr)))

1. Your project report in PDF or Word format. Project report that includes:
  - Name and ID of the group members
  - Name of your programming language
  - The descriptions of tokens that you use in your lexical analyzer, together with their implementations
  - The complete BNF grammar for your language with the tokens you have defined used as terminal symbols
  - A brief explanation about why you have chosen that syntax for representing your tokens, and how much your syntax satisfies language evaluation criteria, such as readability, writability, reliability, etc.
2. Lex file for your language.
3. Test input that you have generated in your language.

A *\*very\** simple example test file (a C-like syntax): while (sensor[0]== 0) do  
And its output: WHILE LP SENSOR LB NUMBER RB EQUALS NUMBER RP DO

### **How to use lex and yacc**

Install Cygwin (A Unix like environment for windows)

<http://www.cygwin.com/>

While installing search for flex, bison and gcc, and add these tools.

As an editor you can use Textpad.

<http://www.textpad.com/>