COSC 480 – Compilers
Milestone 1

Phases Covered: Lexical Analysis, Syntactic Analysis

In this milestone you will utilize ANTLR to produce the AST (Abstract Syntax Tree) that is the result of the lexer and parser once run. Between the two modules that are automagically produced by ANTLR, several error conditions should be caught:

* Lexicographic Errors
* Character Pair
* Reducibility
* Stackability

In order to handle these (and a few other conditions noted below) you may need to adjust the baseline parser and lexer as produced by ANTLR. As a cautionary note, be very careful with what you adjust as you could easily break something in the lexer or parser that would render it unusable.

Conditions of Interest:

Comments can be in one of two basic types, the C type, and C++ type. C type comments are block based comments enclosed by /* and */. Inline comments, or C++ type, will be started by // and end with a newline.

Each lexical entity is terminated by a newline except for comments.

Sytactic errors should output as much information as possible, current stack contents, what was popped during the most recent token process and what is left in the input statement (i.e. what token did it fail on?).

Lexical errors to be caught: numeric constant containing too many significant digits (see below), invalid ASCII character, invalid reserved word, invalid identifiers. These should all output an informative message.

Reserved Words: END, PROGRAM, DECLARE, REAL, INTEGER, PROCEDURE, VALUE, REFERENCE, MAIN, INPUT, OUTPUT, CALL, ELSE, IF, THEN, DO

Identifiers start with a lower case letter and has at most 31 other characters, digits, -s and _s.

Integer values and real values have 32-bit accuracy. Integer values can be represented in the source language as d, +d or –d, where d is continuous sequence of 9 or less digits. Real values can be represented as d.d, +d.d, or –d.d, where the total number of digits does not exceed 7.

Single ASCII characters: ;, _, -, ,, [,], (,), :, !, <, >, +, -, *, /, {, }

Multiple ASCII characters: ==, !=, <=, >=, <-, ::, ||, &&

Code and output must be printed out to be turned in and each source file must contain a header with names, email addresses, date and time. In addition, all source files must be turned in via Digital Dropbox by the deadline noted below. Physical copies must be highlighted or otherwise flagged to show where your error conditions required are met. There will be no late milestones accepted. Physical copies are due at the start of the class following the deadline. If the physical copies and the Digital Dropbox copies differ in any way, it will result in a 0 for the milestone. Not following the proper format of the turnin will result in a 0 for the milestone.

All code must be well documented.

Evaluation of this Milestone:

Presentation of Source and Output
ASCII tokens
Double ASCII tokens
Comments
Reserved words
Identifiers
Integer values
Real values
Integer error catching
Real error catching
Invalid identifiers
Invalid reserved word
Invalid ASCII character

DUE: February 18th at 11:59pm in the Digital Dropbox. Physical copies due in class February 19th.

This and all the other milestone documents, grammars and other project information are adapted from course material by Dr. Harold Grossman, School of Computing, Clemson University.