Department of Computer Science and Engineering

IIT Guwahati

Implementation of Programming Languages Lab: CS 348

Assignment - 4: Parser for nanoC *Marks: 100*

Use the lexical and the phase structure grammar of nanoC as given in Assignment 3.

1 The Assignment

1. Write a Bison specification for defining the tokens of nanoC and generate the required `y.tab.h` file.
2. Write a Bison specification for the language of nanoC using the phase structure grammar given in Assignment 3. Use the Flex specification that you had developed for Assignment 3 (if required, you may fix your Flex specification).
3. While writing the Bison specification, you may need to make some changes to the grammar. For example, some non-terminals like
   argument-expression-list$_{opt}$
   are shown as optional on the right-hand-side as:
   postfix-expression:
   postfix-expression `(` *argument-expression-list*$_{opt}$ `)`
   One way to handle them would be to introduce a new non-terminal, *argument-expression-list-opt*, and a pair of new productions:
   argument-expression-list-opt:
   argument-expression-list
   ϵ
   and change the above rule as:
   postfix-expression:
   postfix-expression `(` *argument-expression-list-opt* `)`
4. Names of your `.l` and `.y` files should be `A4_`*`group`*`.l` and `A4_`*`group`*`.y` respectively. *The* `.y` *or the* `.l` *file should not contain the function* `main()`. Write your `main()` (in a separate file `A4_`*`group`*`.c`) to test your lexer and parser.
5. Prepare a Makefile to compile the specifications and generate the lexer and the parser.
6. Prepare a test input file `A4_`*`group`*`.nc` that will test all the rules that you have coded.
7. Prepare a tar-archive with the name `A4_`*`group`*`.tar` containing all the files and upload to Moodle.

2 Credits

1. Specifications and testing: **70**
2. Main file and makefile: **10**
3. Test file: **20**