

Tanoja Sunkam Sunka004
Jingxiang Li Lixx3899

WRITEUP

The problems that we had to fix were Question 2 and 4.

The problems that we did not have to fix were Question 1, 3, 5, and 6.

The type of testing that was used was Blackbox testing and this was done by checking to make sure that the test cases that were already written passed correctly.

DIVISION OF LABOR

We both did questions 2 and 4 together and it took us 15 minutes and 30 minutes respectively.

ANSWERS to QUESTIONS

Question 1:

No there is not any duplication in the testing code we have written to test individual token regular expressions and the code by scan. For testing each regular expression, we directly call member function "int matchToken(const char *text, Token *&matchedToken)" in the class Scanner which is designed to generate one Token instance given the text. Then it's easy to finish the test by comparing the generated Token instance with the correct "terminal" and "lexeme" value.

Question 2:

Yes our scan function does make calls to makeRegex every time it is called.

This initialization should be done inside the constructor of class Scanner and we can make an array of regular expressions as a private member of the Scanner instance.

Question 3:

No we do not have a redundant array of tokenType values because it is unnecessary and just repeats the tokenEnumType enumerator. Also these types can be made into integers so we do not have to make them into an array to access them. Note that each member of a Enumerator is implemented as an unique integer, and those representations are assigned continuously from 0 on default. Then it's safe to cast a tokenType into an integer, and vice versa. Hence there is no have to make them into an array to access them, we can directly do that by using the corresponding integer.

Question 4:

Yes, changing the order of intkwd and printkwd in the definition of enum tokenEnumType in scanner.h will have an adverse effect on the rest of the code. The change that needs to be made to prevent this from happening is that in the for loops for all tokenTypes the initial index needs to be changed to 0 in place of intKwd.

Question 5:

No we did not create a named regex_t pointer for each regex instead of only putting them in an array.

No the expressions are not necessary and we can get rid of these. We created an array called "regex_array" that holds all the pointers to each regular expression for each token type.

Question 6:

No there are no places in which enumerated `TokenType` values should be used instead of integer literals.