Lexical Analysis in COSMOS

Team 10 Members

1. Avula Dinesh : CS20BTECH11005
2. Ayush Jha : CS20BTECH11006
3. Yuvraj Singh Shekhawat : CS20BTECH11057
4. Rohan Atkurkar : CS20BTECH11041
5. Shashank Shanbag : CS20BTECH11061

COSMOS

- COSMOS is predominantly built for space scientists and learning students. It is a statically typed programming language, and it serves as a scientific calculator for astronomical problems.

- It aims at assisting people with no background in programming. Various familiar data types and functions implementing a variety of formulas make COSMOS a user-friendly language.

Example for COSMOS

Q. The apparent and absolute magnitude of a star from earth are 18 and -4. Find its parallax angle observed from earth?

Ans. This could be solved very simply using COSMOS.

```
#/ Preprocessor directives and libraries must be included before main procedure /#
proc int main()
{
        int m = 18, Mv = -4;
        #/ We can calculate distance of star from earth using star dist() procedure /#
        #/ Declaring a variable of type parsec to store distance /#
        parsec d = star dist(m,Mv);
        #/ Now we can calculate parallax using parallax() procedure /#
        arc sec ans;
        ans = parallax(d);
        output(ans);
}
```

What is lexical analysis ?

1. It converts the input program into a sequence of tokens.
2. A lexical token is a sequence of characters that can be treated as a unit in the grammar of the programming languages.
3. Lexer does these 4 tasks : Tokenization, Removing whitespace characters, remove comments, generates error messages by providing row and column numbers.

What does our lexer do?

1. “lexer.l” takes an input cosmos program(.cos) and returns stream of tokens .
2. Our lexical analyzer uses flex.
3. Generated stream of tokens will be used by the parser for further processing.

How to use lexer?

1. To run lexer using terminal commands :

   ```
   lex lexer.l
   gcc -o lexer_out lex.yy.c
   ./lexer_out <input_file_name> output_file_name
   ```

2. To simply run our lexer on all test cases using makefile, the command is:

   ```
   make
   ```

3. To remove all the output files and intermediate generated code files, the command is :

   ```
   make clean
   ```

Thank You!