CS301P Compiler Design Laboratory Exercises Week-4

Date: Aug 26 2024

Objectives

- To implement parsers for different languages using *Yet Another Compiler Compiler (YACC).*
- To learn the error handling mechanism using *error* token.

Exercise Problems

1. Implement a parser to verify whether given complex numbers are in a valid format. The valid format for a complex number is either: ‘a + ib‘ or ‘a + bi‘, where: ‘a‘ and ‘b‘ are integers or floating-point numbers (including negative numbers), and ‘i‘ is the imaginary unit. Spaces between components should be ignored. Whenever an erroneous input string is detected, the parser should print appropriate error diagnostic message as shown in the following.

   Sample Input:

3 + 4i	→	Accepted
-2 + 5i	→	Accepted
3.5 + i2	→	Accepted
-4.7 - 6.8i	→	Accepted
0 + 1i	→	Accepted
3 + 4	→	Rejected - missing ‘i‘
i + 4b	→	Rejected - Invalid format
3 + i	→	Rejected - missing imaginary part coefficient

2. Implement a parser to verify whether the given list of dates are valid or not. The date format is **DD-MON-YYYY**, where DD, MON, and YYYY take string type input from the valid ranges, [1 . . . 31], {Jan, Feb, . . ., Dec}, and [0000 . . . 9999], respectively. For each input string, the parser should print either ”Accepted” message if the string is in a valid date format and the date is a correct one (that is, it must be correct both syntactically and sematically); otherwise ”Rejected” with approriate reason as shown below.

   Sample Input:

   10-Jan-2024 → Accepted

   30-Feb-2001 → Rejected - Date out of range

   29-Feb-1992 → Accepted

   Feb-29-1992 → Rejected - Invalid format

   29-Abc-1992 → Rejected - Invalid month

   29-dec-2000 → Accepted

3. Implement a parser to accept strings belonging to the language $a^i b^j c^k$ where $i \neq j \neq k$ and reject anything else.

References

1. Flex Manual https://westes.github.io/flex/manual/
2. Lex & Yacc by John R. Levine, Tony Mason, and Doug Brown

Sumission Guidelines

Same as the lab #3.