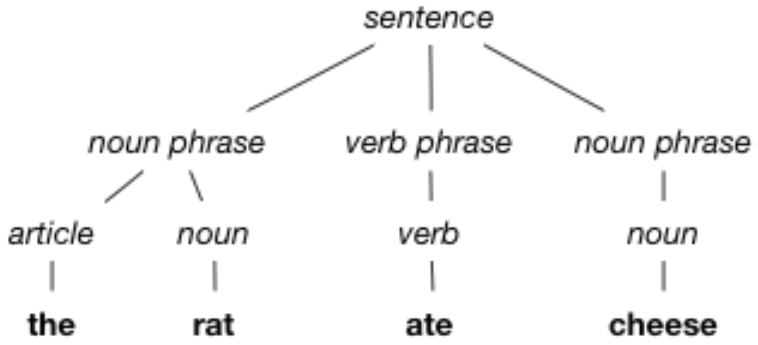

Homework 4: Parser Implementation

Instructor: Zhiyao Liang
Macau University of Science and Technology
2020 FAll

Purpose

Construct a *syntax analyzer*, also called a *parser* for the specified source language. The above picture is found from the internet and shows some ideas of parsing a sentence.

The source language

The source language is Pym, which is described in the previous homeworks.

Some Helpful code

Some code files are provided in the folder: `some_parser_code` : Some code files of a parser for another C-like language are provided. These files are not complete. You can use or modify these code files for a parser of language Pym.

The code provided by the textbook `textbook_code.zip` is already uploaded to the FTP site of this course, which could also be helpful.

The provided file `Pym-grammar.txt` provides a set of grammar rules for Pym. These grammar rules can be used to design the corresponding recursive-descendent parser functions.

If you prefer to use your own grammar rules designed in Homework 2, you can also do that. But make sure that your own grammar rules are proper.

The tasks

1. Write a program using to implement a parser for the language Pym, using the recursive-descendent method, i.e., translate each non-trivial grammar rule as a function. The behavior of such a function include the following aspects:

- Read a sequence of tokens, starting from the current position in the token list, as required by the corresponding grammar rule.
- If the sequence of tokens satisfy the description of the RHS of the grammar rule, build a parse tree using the information of these tokens. The shape and structure of the tree should be properly designed, which is a task of homework. Return the address of the top node of the tree.
- If cannot find a sequence of tokens that satisfy the RHS of grammar rule, report some error. Return NULL, or the address of the top node of a partially built tree.

3. Test your parser with some correct Pym source program files, and some `.n` files with syntax errors (modify the correct files, like delete a colon, a keyword, and so on.)
    - When a Pym file is correct, some syntax tree (also called parse tree) should be printed.
    - When some syntax error is found (at least one) error message should be printed.

How to submit

- Upload your files at the webpage address of this homework on Moodle.
- The uploaded files should include the following:
    - All of the source code files.
    - A document (.txt, .md, .docx), for example `readme.txt` , that describes:
        - How to compile and run your file.
        - What has been done, and what are the remaining problems. For example, which syntax errors can be detected, and which cannot be detected by your program.
        - What are the extra features of your homework that deserve extra points.
        - How did your team mates cooperate with each other. What did they do.
        - Any other information describing your work. Like your experience of solving problems. Some screen capture or record of running or compiling your program.
    - Any relevent file that is needed or used by your program. For exmple, if you created some testing source files of Pym to test your parser, these files should also be uploaded.
- About homework submission in a group:

- At most 3 students can form a group to submit the homework together. You can surely do the homework alone.
- One group only need to submit the homework by once by one student. The other members do not need to (better not) submit the homework again; or, just submit one .txt file saying who are the members of the group and who submitted the homework.
- In each file that you submit, record the registered names in Chinese (if you are not an international student) and English letters (PinYin), classes of each student, last 5 digits of student ID, as comments. For example:

```
/*   homework 3. Group members:
 李白      Li, Bai
 杜甫      Du, Fu
 李清照    Li, Qingzhao
*/
```

- Deadline: Wednesday Dec 30 2020 11:00 pm
- Plagiarism is not allowed

References

1. Kenneth C. Louden. Compiler Construction Principles and Practice. PWS Publishing Company, 1997. ISBN 0-534-93972-4 http://www.cs.sjsu.edu/~louden/cmptext/.
2. Andrew W. Appel. Modern Compiler Implementation in C: Basic Tech- niques. Cambridge University Press, New York, NY, USA, 1997.