Lexical Analysis 09/09/15 08:56

Lexical Analysis

This assignment is based on the project described in *Modern Compiler Implementation in ML*, section **PROGRAM**, pages 31-32. The source code mentioned in this section (as being available from \$Tiger/chap2) should be taken from the Documents

(https://bb.au.dk/webapps/blackboard/content/listContent.jsp?course_id=_43271_1&content_id=_355480_1&mode=reset) part of this website, stored in the tar archive part2.tgz. This archive should be unpacked and used for an initial test as follows (where myPrompt stands for your command line prompt, such as dovs@linux-en0j:~, and it is assumed that you downloaded part2.tgz to your home directory, /home/dovs):

```
myPrompt> cd ~/dovsproject
myPrompt> tar xzf ~/part2.tgz
myPrompt> cd part2
myPrompt> make standalone
...
myPrompt> ./compiletests
...
```

Note that the structure called Parse in the book is renamed to Scan in part2.tgz, and the function called parse renamed to scan, because that structure and function do scanning rather than parsing. The definition of the lexical analysis is given in tiger. Lex and the main task in this project is to implement the full lexer for the Tiger language by extending this file.

Lexer requirements

On nesting and escape sequences

At the *very minimum*, the lexer should support all basic constructs of the Tiger language, without escape sequences for strings and un-nested comments. This can be accomplished in the following way: the comment-start token " /* " should start a comment and the first comment-end token " */ " should end the comment, no matter how many comment-start tokens the comment so far has contained.

To get *maximum points*, the lexer needs to handle nested comments and escape sequences for strings, including caret notation ^ – see this ASCII Chart (http://www.finseth.com/craft/#aE) for the full list of appropriate symbols under "Graphic" column. If not all escape sequences are implemented, you should show how you would be able to handle them all. While there are several approaches to implementing this, we highly encourage using lexer states here, and not just hacking everything in SML (after all, this it what the states are there for; in addition, the lexer source will likely end up much more readable).

Exponentiation arithmetic expression

This year's version of Tiger extends the basic arithmetic expressions of Tiger language (as given in the textbook appendix) with one additional operator of arithmetic exponentiation, written as ^ . Your lexer must recognize this token.

Error handling

The lexer must do proper error handling for unfinished strings, unfinished comments, and multi-line strings without using special constructs for this particular case. Error handling associated with ill-formatted numbers is illustrated by the function s2i. Implementation of this function can be used a source of inspiration for other types of error handling that you choose to include.

Test cases

Write 5 interesting test files containing Tiger code with comments. By interesting we mean that you should test specific things about the lexer. Additionally, make sure that your implementation works on the provided test files ../testcases/test{0,1}?.tig.

Report

The project should be documented by a short report, only a few pages. It may be formatted using LaTeX or other document processing systems and delivered as a PDF file. The report must briefly describe your approach, especially insofar as it differs from the project description due to, e.g., problems making it work, or interesting experiments. Use the report template provided for this hand-in as a starting point.

Deliverables

• part2.tgz This is a tar archive file that will contain your solution. This file can be obtained by running the command

```
myPrompt> cd ~/dovsproject
myPrompt> tar czf ~/part2.tgz part2
```

This file should include the source code to your solution.

- part2-tests.tqz This is a tar file (created in a similar manner as above) that should contain the five test cases that you created.
- Project report as a PDF file. See the Documents section of the site for a template, and description above for what to include in the report.

Lexical Analysis 09/09/15 08:56

Submission instructions

 $\label{thm:course_submission} Upload\ three\ file:\ part2-tests.tgz\ ,\ and\ part2-report.pdf\ using\ the\ course\ submission\ system\ (http://dovs.cs.au.dk/).$

Deadline

Wednesday, September 16th, 14:00.

Content on this page is maintained by Aslan Askarov.