CS536: Homework 2

Keith Funkhouser

September 22nd, 2015

1

Below are 5 ***incorrect*** JLex patterns for the kind of comment defined above. For each pattern, explain what is wrong. For example, you might say *The pattern does not allow the comment body to include stars*, or *The pattern does allow the comment body to include */*. If the pattern both disallows some "good" comments and allows some "bad" comments, give two explanations, one for each problem with the pattern. Then, for each problem with the pattern, give a string that illustrates the problem; i.e., give a string that is ***not*** matched by the pattern but is a "good" comment and/or give a string that ***is*** matched by the pattern but is a "bad" comment. ***Be sure that it is clear whether your example strings are intended to be "good" or "bad" comments.***

- `{SLASH}{STAR}[^(*/)]*{STAR}{SLASH}`

  This pattern disallows some "good" comments, namely those which include any number of * or / within them. An example comment which should be matched by the pattern but is not is:

  ```
  /* *** */
  ```

- `{SLASH}{STAR}(.)*{STAR}{SLASH}`

  This pattern allows some "bad" comments, namely those which include */ within them. An example comment which should not be matched by the pattern but is is:

  ```
  /* */ */
  ```

- `{SLASH}{STAR}([^*]*{STAR}+[^*/])*{STAR}+{SLASH}`

  This pattern disallows some "good" comments, namely within the outer comment markers, any sequence of non-* must be followed by at least one * and then exactly one non-*, non-/ (which is not a constraint that comments have). An example comment which should be matched by the pattern but is not is:

  ```
  /* */
  ```

- `{SLASH}{STAR}([^*]|[^/])+{STAR}{SLASH}`

  This pattern allows some "bad" comments, namely those which include */ within them. An example comment which should not be matched by the pattern but is is:

  ```
  /* */ */
  ```

  This pattern also disallows a "good" comment, namely the empty comment:

  ```
  /**/
  ```

- `{SLASH}{STAR}[^*]*{STAR}+{SLASH}+`

  This pattern allows some "bad" comments, namely those which end with more than one /. An example comment which should not be matched by the pattern but is is:

  ```
  /* *//
  ```

  This pattern also disallows some "good" comments, namely those which include * within the middle of the comment. An example comment which should be matched by the pattern but is not is:

  ```
  /* *** */
  ```

2

Give a correct JLex pattern for comments as defined above. You may define and use additional macros. Make your pattern as readable as possible.

Consider the following macros:

```
STARFOLLOWEDBYNOTSLASH  = [*][^/]
NOTSTAR                 = [^*]
```

And the following JLex pattern:

```
{SLASH}{STAR}({STARFOLLOWEDBYNOTSLASH}|{NOTSTAR})*{STAR}{SLASH}
```

In plain English, this corresponds to: "a comment starts with a /*, followed by: (EITHER a * followed by a non-/, OR a non-*) 0 or more times, and ends with a */."

We ran the following unit tests on this JLex pattern in order to better assess its correctness:

```
*************************
   THESE SHOULD PASS
*************************
PASSED: /**/
PASSED: /* */
PASSED: /* * */
PASSED: /* **** */
PASSED: /* / */
PASSED: /* //// */
PASSED: /* /* */
PASSED: /* //////****** */
PASSED: /* this is a comment */
PASSED: /* this* is a comment */
PASSED: /*a**k*/
PASSED: /*a**ka**k*/
PASSED: /*a**ka**k*****/
PASSED: /*   asdlk any text here **** */
PASSED: /*   asdlk any text here * */
PASSED: /*   asdlk any text here ** */
PASSED: /*   asdlk any text here **z*/
PASSED: /*comment* */
*************************
   THESE SHOULD FAIL
*************************
FAILED: /* */ */
FAILED: /* ////*/ */
FAILED: /* **////
FAILED: /* *//
```