CS712/CS812 Project Phase 4 Spring 2016 Due Sunday May 8

Use your Tscript system to implement LL(1) grammar analysis in Tscript:

1. (50 points) Read a grammar from stdin. The grammar will be listed one production per line. The first production is a production for the start symbol. Other than this restriction that the first production will be for the start symbol, the productions can appear in any order. In particular, the productions do not need to be grouped by left-hand-side symbol. Each production is given as a (single) space separated list of symbols. The first symbol on the line is the left-hand-side symbol of the production. The symbols on the right-hand side, if any, are listed in order following the left-hand-side symbol. (Note that null rules will be given as having an empty right-hand side.) I will only test using grammars that are formatted correctly.

   The nonterminal symbols are the set of symbols that appear on the left-hand side of the productions.

   The terminal symbols are the set of symbols that appear in the right-hand-side of at least one production, but do not appear as the left-hand side of any production.

   Print a report to stdout. Start by printing a line containing only "Start Symbol", followed by a blank line, followed by a line containing the start symbol, followed by a blank line. Next print a line containing only "Nonterminals", followed by a blank line, followed by a line containing the nonterminals, in any order, separated by a (single) space, followed by a blank line. Next print a line containing only "Terminals", followed by a blank line, followed by a line containing the terminals, in any order, separated by a (single) space, followed by a blank line.

2. (10 points) Add a display of the null-deriving nonterminals to the report. Print a line containing only "Null-Deriving Nonterminals", followed by a blank line, followed by a line containing the null-deriving nonterminals, in any order, separated by a single space. Finally, print a blank line.

3. (10 points) Add a display of the First sets to the report. Print a line containing only "First Sets", followed by a blank line, followed by the First sets, given one per line, in any order. Each First set should be displayed by the nonterminal followed immediately by a colon and a space, followed by the members of the set, in any order, separated by a (single) space. Finally, print a blank line.

4. (10 points) Add a display of the Follow sets to the report. Print a line containing only "Follow Sets", followed by a blank line, followed by the Follow sets, in any order, given one per line. Each Follow set should be displayed by the nonterminal followed immediately by a colon and a space, followed by the members of the set, in any order, separated by a (single) space. Finally, print a blank line.

5. (10 points) Add a display of the Predict sets for the productions to the report. Print a line containing only "Predict Sets", followed by a blank line, followed by the Predict sets, in any order. Each Predict set should be displayed using three lines. The first line should contain the production, displayed as it was in the input. The second line should contain the members of the Predict Set for the production, in any order, separated by a (single) space. The third line should be blank.

6. (10 points) If the grammar is LL(1), add a line containing only "The grammar is LL(1).". If the grammar is not LL(1), add a line containing only "The grammar is NOT LL(1).". Then print a blank line.

You should implement items in the above list in the order that they are listed.

You can add features to your Tscript system, but they must be features described in the ECMAScript 5.1 standard specification. This restriction includes built-in functions. Any built-in function that you add, and you will need to add some to manipulate Strings, must be described in the spec.

To get full credit, your code must be adequately documented and structured. If I can't easily read and understand your code, you may lose points. This includes both your compiler and your Tscript code for the LL(1) grammar analysis.

You must give me back the system in the same form that I gave it to you. I must be able to install it and run it in the exact same way as when it was delivered to you. I would also like you to keep the source code organized in the same directory hierarchy. If you fail to do this, a significant deduction will be made to your grade.

Note that there is a script, *distribute.sh*, in *tscript/bin*, which will generate a tar file for you. This tar file is what I require you to submit to me for grading.

Your Tscript code should be placed in a single file called "LL1.ts".

To turn in this assignment, type:
~cs712/bin/submit phase4 tscript.tar LL1.ts

Submissions can be checked by typing:
~cs712/bin/scheck phase4

The assignment is due on Sunday May 8. Submissions between 8am May 9 and 8am May 10 will have a late penalty of 10 points. Submissions between 8am May 10 and 8am May 11 will have a late penalty of 30 points. **No program may be turned in after 8am on Wednesday May 11.**

Remember: you are expected to do your own work on this assignment.

Last modified on March 11, 2016.

Comments and questions should be directed to pjh@cs.unh.edu