Homework 2

LR table for regexp:
https://docs.google.com/document/d/1nc9g5v55HqBjKYDdE5OnLJdi_rRvPlb3XS6KOXoMJ8g/edit?usp=sharing

Programming Languages - Homework 2

Write C/C++ functions that builds and runs an LR parser.

Skeleton codes are provided with the assignment.

DO NOT change the lr_parser_main.cc and regexp_main.cc file.

1.1. Write C/C++ functions : a function that loads an LR parsing table to build an LR parser, and a function that runs it on input token strings and returns the acceptance (true/false). [60pts]

The LR parser table file structure is as follows:

num_table_elements num_rules
state symbol action next_state
...
rule_id lhs_symbol num_rhs
...

Refer description in the header file lr_parser.h for more details.

Design the LRParser structure in lr_parser.h.

Implement the BuildLRParser function that builds LRParser structure using the given table elements.

Implement the RunLRParser function so that it returns the acceptance of the given token string.

1.2. Write a C/C++ function that builds DFA from a regular expression string. [40pts]

Design and implement the parser that builds NFA from the regular expression, consisting of single characters (abc), any character (.), set of characters ([abc]), OR (a|b), zero-or-more repetition ((a*), and group ((ab c)).

ab|cd : ab, cd
a(b|c)d : abd, acd
a.*b : ab, acb, axyzb, ...
(a(b.c)*|de)f : af, def, abxcf, abxcbycf, ...
[abc]*def : adef, bdef, cdef, aadef, abdef, ...

Convert the built NFA into DFA and match the input string with the regular expression.

Due: May 20 (Fri) 11:59 pm

– Zip the source code (ONLY .h, .cc and Makefile; absolutely no executable or object files) and submit it in ezhub (portal).

– The program must run on the Linux server (csedev.hanyang.ac.kr).