Lab Assignment 6

LR Parsing

Consider the CFG fragment with non-terminal symbols **{D, T, L}**, with start symbol **S**, terminal symbols { **id**, **:**, **;**, **,**, **var**, **integer**, **real** } and the productions **P** listed below.

D → var L : T ; $

T → integer

T → real

L → L , id

L → id

Q1. Construct the **LR(0)** DFA for the above grammar. Clearly mention what are the items in each state. Based on the LR(0) DFA, provide the LR(0) parse table for the grammar. Briefly describe the process followed to build the automaton and the parse table.
Is the grammar **LR(0)**?

Submit a document describing your solutions to Q1.

Q2. Develop a program that takes any grammar "G" as input and has modules/functionalities to generate the **LR(0)** automaton and the **LR(0)** parse table for the given grammar.

NOTE: State any restrictions/assumptions on the input grammar that you may consider (e.g. if any limit on the number of terminals/non-terminals/productions etc).