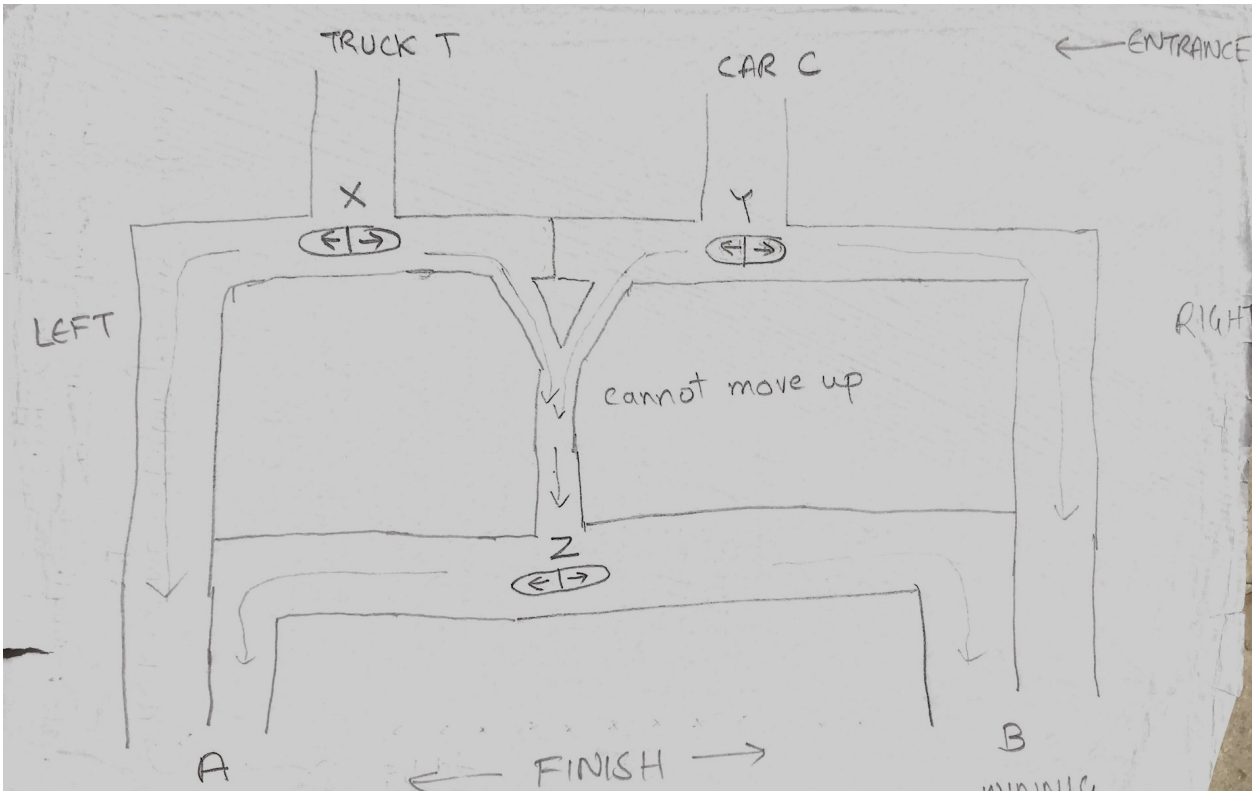
CS 301: Theory of Automata
Assignment 1 Solutions
Fall 2019



Due: Thursday 12th September, 2019

Problem 1

The diagram shows a car game, where the car has to reach the finish point A or B. There is a separate entrance for trucks T and cars C. Only one car or truck can remain on the track at any one time. There are 3 light sections, X, Y and Z with a digital sign for either left L or right R. At one time only one arrow is on. The cars move in the direction pointed to by the arrow. Once a car passes through that point the direction of arrow reverses. For example:

1. Arrow direction at X,Y,Z is L,L,R, and truck enters then it will go to finish point A and will change the arrow signal to R,L,R. (Note as it did not pass through Y and Z, there direction did not change.).
2. Arrow direction at X,Y,Z is L,L,R, and car enters then it will go to finish point B and will change the arrow signal to L,R,L. (Note as it did not pass through X and so its direction does not change.)
3. Arrow direction at X,Y,Z is L,R,L and car enters then car finishes at B and arrow changes to L,L,L. Only the arrow at Y changes as the car passes only through Y.

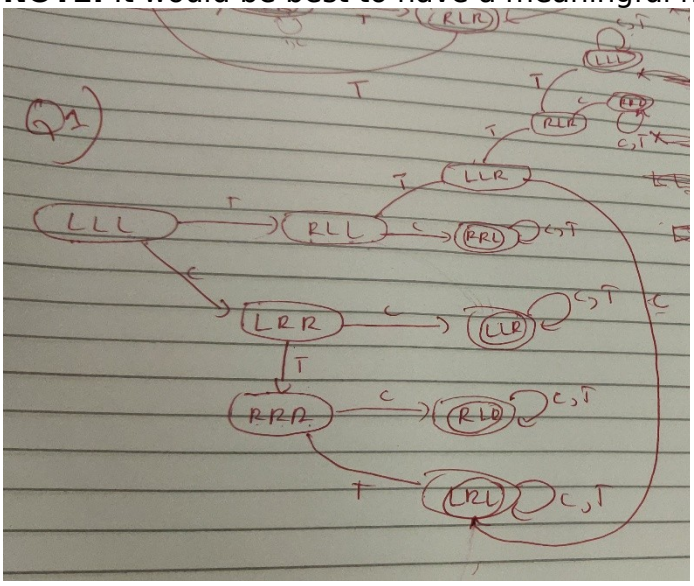
The initial configuration of arrows at X,Y,Z is L,L,L. A winning sequence is when any vehicle in the sequence goes to finish point B. If any vehicle in the sequence is unable to reach the finish point B then it is a losing sequence. For example:

1. CCT is a winning sequence (second car finished at B)

2. TTTC is a winning sequence (C enters B)
3. TTTCTTT is also a winning sequence as C enters B after that no matter which vehicle comes its a winning sequence
4. TTT is a losing sequence as no vehicle was able to reach B

Make an NFA or DFA to represent the above. Consider a sequence of vehicles as a string. Winning sequence should lead to acceptance and non-winning sequence should lead to rejection. Carefully think about what the states would represent.

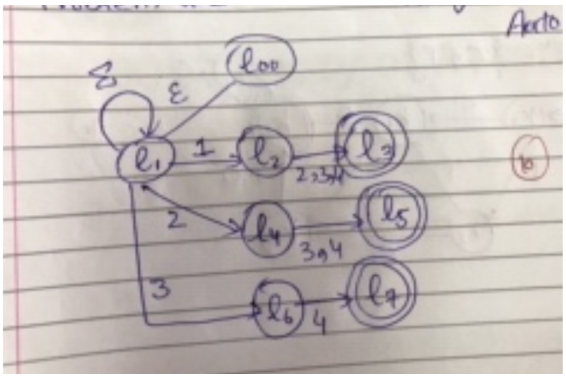
SOLUTION
NOTE: It would be best to have a meaningful name for each state.



Problem 2
Let $\Sigma = \{1,2,3,4\}$. The strings in the language have the last 2 characters such that the second last character is smaller than the last character. For example: strings in the language are 14313, 34224412, 123114 etc. Strings not a part of the language are 32, 321 211 122 etc.

Make an NFA or DFA to represent this language.

Solution by Kamran
Note: The null transition to l_{00} is not necessary



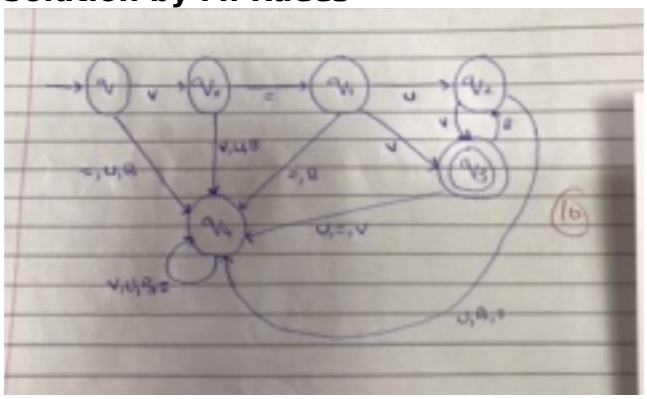
Problem 3
You are designing a new programming language. Here you define expressions as:

1. variable = variable binary_operator variable
2. variable = variable
3. variable = unary_operator variable
4. variable = unary_operator variable binary_operator variable
5. Variable = variable binary_operator variable binary_operator variable … (must end with variable)

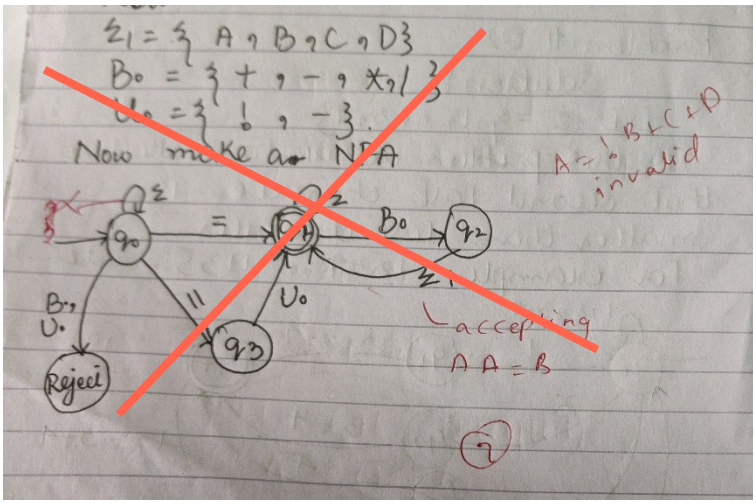
Variable ∈ {A,B,C,D}
binary_operator ∈ {+,-,*,/}
unary_operator ∈ {!,-}
Examples of valid expressions are:
1. A= B+C
2. A = !B
3. A = B
4. A = !B*C
5. A = B+A+C+D+A (ok because of rule 5)

Make an NFA or DFA to express a valid expression.

Solution by M. Raees



NOTE: This was a common mistake. This is NOT the correct solution.


Σ1 = { A, B, C, D }
B0 = { +, -, *, / }
U0 = { !, - }.
Now make an NFA
A = B+C+D
invalid
q0
q1
q2
q3
Reject
B0
U0
Accepting
A A = B