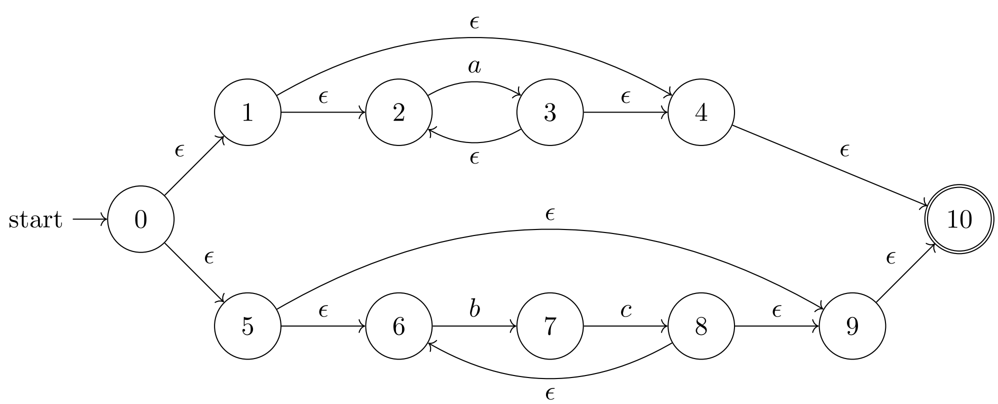
ECE 467 Midterm 1

University of Toronto

2022 October 11

1. (3 points) Draw an NFA for **(open|close)*** (where the parentheses indicate the order of operations). Draw states as circles. Label the start state. Indicate final state(s) with a double-circle. See the next question for an example.

2. (4 points) Draw the DFA for the following NFA. Show your work.



3. (10 points) Consider the following grammar, where E is the start symbol.

$$
\begin{aligned}
E &\to \text{T C} \\
T &\to \text{lparen E rparen} \mid \text{id} \\
C &\to \text{C op T} \mid \epsilon
\end{aligned}
$$

   (a) (3 points) Compute the NULLABLE, FIRST, and FOLLOW functions for all nonterminals in the grammar.

   (b) (5 points) Draw the LR(0) states for the grammar (with labeled arrows between the states).

   (c) (2 points) Write the SLR parsing table for the grammar **based on your previous two parts** (you will not lose marks in this part if your previous computations were incorrect).

4. (3 points) Based on the course lab project.

   (a) (1 point) You wish to add support for augmented assignment operators (e.g. `+=` and `-=`). The instructor says you don't need to introduce new tokens as augmented assignment operators consist of existing tokens, e.g. `+=` is a `PLUS` token followed by an `ASSIGN` token. What could go wrong?

   (b) (2 points) Suppose you have a sorted vector `v` of the offsets of all newline characters in the input buffer (starting from 0). Given an arbitrary offset `k`, you wish to use the binary search function `partition` in the standard library of your favourite language to find which line it is on. Suppose `partition` returns the *smallest* index `i` such that `v[i] >= k`. Given the vector of newline offsets `v`, an input offset `k`, write the code to compute the line and column of `k` using `partition`, assuming lines and columns both start at 1.

5. (8 points) Let a_k be the string of length $2k$ consisting of k left parentheses followed by k right parentheses. Consider the language $L = \{\, a_k \mid k \in \mathbb{N} \,\}$ (i.e. all strings a_k for non-negative integers k). **Suppose** there exists a DFA that defines L; call it D.

   (a) (1 point) Define *reachable* of a state v of D to be true if there exists a string $s_v \in L$ such that the execution of the DFA is at state v at some point during the input of s_v, and false otherwise. Consider the subset V of *reachable* states of D. Argue that V is finite. *(Hint: the argument is very simple.)*

   (b) (1 point) For each state $v \in V$, choose some string s_v such that the execution of D given s_v (at some point) reaches v. Consider the set S of all such chosen strings s_v (one string for each $v \in V$). What is the most you can say about the size of S?

   (c) (3 points) Argue that there exists a string **not in** L that D accepts. *(Hint: let n be the greatest integer such that $a_n \in S$; i.e. a_n is the longest string in S. Consider the execution of D given the string a_{n+1}.)*

   (d) (1 point) Give a definition for a regular language.

   (e) (2 points) Combine the previous parts to prove that L is not regular.