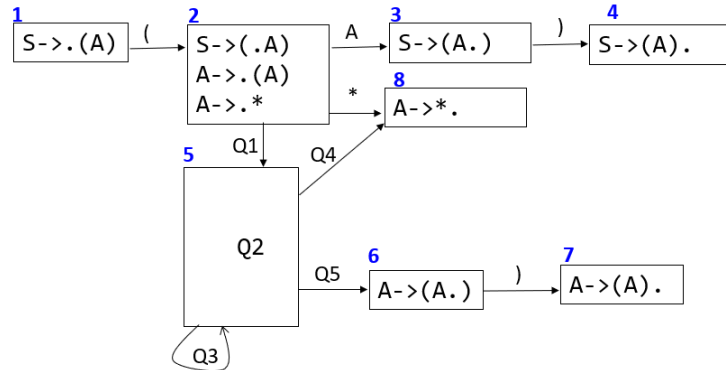


1. Consider the Language  $L_1$  defined by the expression  $a^{m+n}b^m c^n$ ,  $m, n \geq 0$  (i.e.  $L_1$  contains any string where the number of as are equal to number of bs and cs combined and as, bs, Cs appear in that order).
  - a. Can a regular expression accept all strings of the language  $L_1$  and nothing else? If not, why? **(2 points)**
  - b. Express  $L_1$  using a CFG with S as start symbol (hint: you can do this with 5 productions in total). **(6 points)**
2. A partial LR(0) CFSM is shown below for a grammar  $G_1$ .
  - a. Complete the CFSM by answering questions Q1-Q5 i.e., you need to fill state 5 and label edges coming into or going out of state 5. **(3.5 points)**



- b. Given your CFSM in the previous question, complete the table below by answering Q6-Q9 **(4 points)**

Parse Stack	Prefix matched	Parser Action
12556	(((A	Q6
125567	(((A)	Q7
Q8	Q9	

The parser action Q6 results in parse stack shown in the next row. Parser action Q7 results in parse stack of Q8. For parser action, write accept, shift or reduce, and the symbol shifted or production reduced.

- c. Suppose the parse stack contains 1 2 5 5 5 5 8. How many "(" are there in the string? **(2 points)**
      - d. From the previous state of the parse stack, suppose the input string is accepted eventually, how many ")" are there in the input string? **(0.5 points)**
3. Consider the expression  $a := ((b+c)-d)/e$ 
  - a. Draw the AST for the expression **(2 points)**
  - b. Generate 3 address code for the expression **(4 points)**
4. Consider the code snippet shown. When the call to fact(3) in main returns and the main has not yet returned, the stack contains the frame(s) of \_\_\_\_\_ **(1 point)**

```
main() {
    fact(3);
}
fact(int n) {
    if (n=0) return 1;
    return n*fact(n-1);
}
```