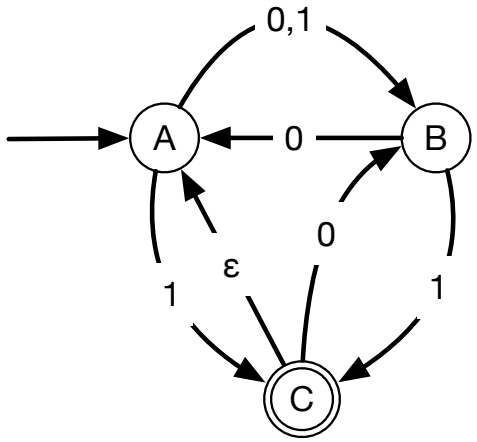
Sample Midterm 1

(1) Given the following NFA, create an equivalent DFA. Give the transition diagram of the DFA. (Don't forget the empty set should it arise). Use double stroke to indicate an accepting state. The initial state is given by the arrow pointing from nowhere.



(2) Use the Master Theorem in order to solve the recurrences for size n integer multiplication:

$T(n) = 4T(n/2) + c$

$T(n) = 3T(n/2) + c$

(3) Indicate whether the Master Theorem can be applied to the following recurrences and if it can, then solve them.

a. $T(n) = (n-1)T(n-1) + c$

b. $T(n) = 5T(n/25) + \sqrt{n}$

c. $T(n) = 2T(n/2) + \log(n)n$

(4) What is the recurrence for the run time of the following algorithm (presented in Pseudo-Python)?

```
def alg(array, lo, hi):
 """ array is an array of integers, lo and hi are indices """
     if lo == hi:  #one element in the array
         return array[lo]
     if lo > hi:
          return None
     else:
          suma = 0
          for i in range(lo, hi):
                suma += alg(array(lo, hi-1))
     return suma // (hi-lo)
```

(5) Prove the loop invariant in the following code by induction on the number of iterations

```
Algorithm sort(A)
    Input: an array A storing n integers
    Output: the same array with the elements sorted in ascending order
    for j = 1 to n-1
        // Invariant: A[0..j-1] contains the same elements as
        // the original subarray A[0..j-1], but in sorted order.
        key ← A[j]
        i ← j - 1
        while i ≥ 0 and A[i] > key
            A[i+1] ← A[i]
            i ← i - 1
        A[i+1] ← key
```