University of Nevada, Las Vegas Computer Science 456/656 Spring 2023

Assignment 2: Due Friday February 3, 2023, 11:59 PM

Name:_____

You are permitted to work in groups, get help from others, read books, and use the internet. You will receive a message from the graduage assistant, Sandeep Maharjan, telling you how to turn in the assignment.

1. Prove that the grammar G given below is ambiguous by giving two different parse trees of the string iiaea. Both of those parse trees are correct, but only one of them is consistent with the usual rule for resolving ambiguity of if-then-else statements. Which one? 1. S a 2. S iS 3. S iSeS

2. Solve problems given in the handout `finiteAutomata.pdf` associated with the following figures.

   (a) Figure 7.

   (b) Figure 8.

   (c) Figure 10.

(d) Figure 12.

(e) Figure 13.

3. Write (do not prove) the pumping lemma for regular languages. (You will need to do this on the first examination, so try to understand it as you write it.)

4. Please help me make a very important decision! As you know, the question of whether there is a $\mathcal{P}$–TIME algorithm which finds the factors of an integer is one of the most important unsolved problems in computation theory. In fact, the security of RSA encryption, which is used many times every day, depends on the hypothesis that no such algorithm exists.

   After many years of work, I have found an algorithm for the problem. Here is my code:

```
void primefactors(int n)
 // lists prime factors of n separated by commas
 {
  int f = 2;
  while (n%f != 0) f++;
   if(f == n) cout << n << endl; // n is prime
   else if(n%f == 0) // f is a prime factor of n
    {
     cout << f << ",";
     primefactors(n/f);
    }
 }
```

You can easily see that the running time of the code is $O(n)$, which is certainly polynomial!

I could either publish this result and become insanely famous, or keep it to myself (I'm sure you won't tell) and become insanely rich breaking RSA encryption for enormous fees. What should I do?