Homework 8

Note: You may collaborate on the assignment. If you do collaborate on the assignment, list your collaborators. All duplicate assignments without collaborators listed will be flagged for plagiarism. Additionally, presenting others work – including a Chegg expert answer – as your own work *is* a violation of the academic dishonesty policy.

This homework assignment is designed to prepare you for the in class quiz. For the following graph:

Problem 1: Consider the following BNF grammar:

```
<postalcode>            ::= <forwardsortationarea> <space> <localdeliveryunit>
<forwardsortationarea> ::= <provarea> <loctype> <letter>
<localdeliveryunit>    ::= <digit> <letter> <digit>
<provarea>             ::= A | B | C | E | G | H | J | K | L | M | N |
                           P | R | S | T | V | X | Y
<loctype>              ::= <rural> | <urban>
<rural>                ::= 0
<urban>                ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<letter>               ::= A | B | C | E | G | H | J | K | L | M | N |
                           P | R | S | T | V | W | X | Y | Z
<digit>                ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

(a) How many non-terminal symbols are in the grammar?

(b) How many terminal symbols are in the grammar?

(c) Write four strings that are valid according to the BNF.

(d) For each of your four strings, give two valid mutants of the string.

(e) For each of your four strings, give two invalid mutants of the string.

Problem 2: Write a BNF grammar for the language of NC A&T course codes.

Example sentences:

```
COMP 285
ENGL 100
AGER 796
READ 735
```

Problem 3: Mutants Over Code, Take 1

```
int countOccurrences(int arr[], int n, int x) {
    int cnt = 0;
    //for (int i = 0; i < n; i++)
    for (int i = 0; i < n; i += 2 ) //mutant
        if (x == arr[i])
          cnt++;
    return cnt;
}
```

(a) If possible, find test inputs that do not reach the mutant.

(b) If possible, find test inputs that satisfy reachability but not infection for the mutant.

(c) If possible, find test inputs that satisfy reachability and infection, but not propagation for the mutant.

(d) If possible, find test inputs that strongly kill the mutants.

Problem 4: Mutants Over Code, Take 2

```
public String reverse(char [] str) {
    String reverse = "";

    for (int i = str.length; i > 0; i--) {
        //reverse += str[i];
        reverse = str[i]; //mutant
    }
    return result;
}
```

(a) If possible, find test inputs that do not reach the mutant.

(b) If possible, find test inputs that satisfy reachability but not infection for the mutant.

(c) If possible, find test inputs that satisfy reachability and infection, but not propagation for the mutant.

(d) If possible, find test inputs that strongly kill the mutants.