CS390 – Assignment 2

Matt Hartigan

12-May-2019

Big-Oh Analysis for Assignment 2

The functions included in Hartigan\_Assignment\_2.c are:

* **findTransitionFunction**

This function includes a single while loop that is used to iterate through a linked list, checking for a match condition based on the function input parameters. The linked list represents the transition functions of a DFA. As the number of transition functions grows, the number of operations performed by this function will also grow linearly. **Therefore, it is O(n).**

* **createDFA**

This function includes the following:

* + A for loop to initialize the elements of two state arrays to null values. This loop is unaffected by the size of the input and is there for O(1).
  + A while loop to read in the states of the DFA. The number of operations performed in this loop is directly proportional to the number of input states, and therefore is O(n).
  + A while loop to read in the final states of the DFA. This is O(n) for the same reason as the original state while loop.
  + A while loop to read in the transition functions of the DFA. This is O(n) for the same reason as the original state while loop.

Taking these four loops into consideration, O(1 + n + n + n) simplifies to O(1+3n), which we can capture as O(n). **Therefore, this function is O(n).**

* **loadDFA**

This function uses three while loops (similar to those discussed in createDFA above) to read in the states, initial state, and final states from an input file in O(n) time. The bigger the contents of the input file, the more statements these loops will execute. The transition functions are read in using a nested while loop. The outer loop reads in each field from the input file while the inner loop parses the three values associated with each field (starting state, transition function, ending state). Because the inner loop will execute three times for each field regardless of the size of the input, we conclude that the nested while loop structure is O(3n).

Taking these four loops into consideration, O(n + n + n +3n) simplifies to O(6n), which we can capture as O(n). **Therefore, this function is O(n).**

* **executeDFA**

This function employs two while loops. The first iterates through every character of the user’s input test string so they can be checked for compatibility with the DFA. This takes O(n) time since more statements will follow a longer user input. The second while loop iterates through the list of final states of the DFA looking for a match. This again takes O(n) time since the amount of statements executed will grow proportionally to the number of final states in the input file.

Taking these two loops into consideration, O(n + n) simplifies to O(2n), which we can capture as O(n). **Therefore, this function is O(n).**

* **main**

The main function contains no loops. **We conclude by inspection that it is O(1).**