**Summary of Computer Theory Phase #3 Requirements**

Davis described a primitive recursive predicate (the proof is not presented in the book) of the Step Counter Predicate for Deterministic Turing Machines. Given a text file that contains transition rules that describe a Turing Machine and a text file that contains the initial input to the machine, and given a positive integer as to the maximum number of transitions allowed during the execution of the Turing machine, the Step Counter Predicate states whether or not the given Turing machine will HALT on the given input to the machine within the Number of steps inputted in. An output file is maintained that dumps the contents of the tape that were used during the transitions of the machine. Hint: Since a machine cannot move more than one cell per transition, the number of steps inputted would be a bound on the length of the tape necessary. The output file should also state whether or not the machine halted properly within the maximum number of transitions or not.

A transition of the Turing machine will be a current state and alphabet character as input and this pair maps to the remainder of the Turing machine information for the given transition. All of the states will be numbered and the alphabet will be single characters. The state transitions table will be stored in a hashtable where the concatenated string <State#><AlphabetCharacter> will be hashed to obtain the tape movement of the Turing machine (if any), output to the tape (if any) and destination state for each such state/character pair.

In terms of the text files and inputted number, these should be read in using flags at the command line. The names of these flags should be as follows:

TransitionsFile

InitialTapeInput

FinalOutput

Steps

Note: The first three flags require names of the file containing the requested information. The fourth flag is expecting a positive integer. For debugging purposes, Steps=0 will output the names of the text files with their corresponding contents into the FinalOutput file.

Search the web for the transitions of any Turing machine and data for the input file to test out your code.

In terms of submission, this will be done on the Brightspace System under the Assignments tab where a directory will be provided there with instructions of the various "tasks" (questions) you may need to answer.

**Computer Theory Phase 3 Submission Protocol on the Brightspace System**

**TC.P3.Task#1) Complete Java Code**. Collect all the .java files necessary for your system into a single temporary directory on your computer system. Compress/zip the directory all as one file and upload it to the Brightspace system for **TC.P3.Task#1**. The compressed file to be uploaded should not be particularly large and you will be able to upload as is (and do NOT provide a shared or cloud file or access via file link.)

**TC.P3.Task#2)** **Commenting Statistics.** Comment your code appropriately and use meaningful names for classes, methods, variables and constants. You are expected to report on the number of lines of code using cloc found at https://github.com/AlDanial/cloc/releases/tag/v2.02 This command must be run on the command line (cmd.exe). When you collect all .java files into the same temporary directory (see TC.P3.Task#1 above), then the following command will report on the contents of each .java file. The report generated by cloc program will report on the number of actual java code lines, blank lines, comment lines for each of the .java files that is part of your project. The command to obtain the report data is as follows (assuming you are using the above version):

cloc-2.02.exe --by-file \*.java

NOTE: This command MUST have two minus signs as a prefix to the word “by” and one minus after the word “by” and not intervening spaces before/after the word “by”.

This report should be copied into a text (.txt) file which you will upload to Brightspace system for **TC.P3.Task#2.**

**TC.P3.Task#3) Coding Statistics (Part1).** How many classes in total did you code in your submitted project? How many methods in total did you code in your submitted project? This report should be typed into a text (.txt) file which you will upload to Brightspace system for **TC.P3.Task#3.**

**TC.P3.Task#4) Coding Statistics (Part2).** Estimate how many hours of coding you put into this phase. This report should be typed into a text (.txt) file which you will upload to Brightspace system for **TC.P3.Task#4.**

**TC.P3.Task#5) Turing Machine transitions.** In a text (.txt) file, provide the actual transitions for the Turing Machine(s) you tested this code with. Indicate where you obtained the set(s) of transitions you are providing here. Upload this text file to Brightspace system for **TC.P3.Task#5.** As mentioned above,the transition table will be stored in a hash table. If a number of sets of transitions are being provided for multiple machines, then collect all of the input files into a single temporary directory on your computer system. Compress/zip the directory all as one file and upload it to the Brightspace system for **TC.P3.Task#5.**

**TC.P3.Task#6) Turing Machine inputs.** In a text (.txt) file, provide the actual inputs used to test the set of transitions you provided in **TC.P3.Task#5**. If a number of sets of transitions were provided in **TC.P3.Task#5**, then collect all of the input files into a single temporary directory on your computer system. Compress/zip the directory all as one file and upload it to the Brightspace system for **TC.P3.Task#6.**

**TC.P3.Task#7) Output Files.** An output file was to be maintained that records the contents of the tape that were used during the transitions of the machine. Upload this text file to Brightspace system for **TC.P3.Task#7.** If a number of sets of transitions were provided in **TC.P3.Task#5**, then collect all of the output files into a single temporary directory on your computer system. Compress/zip the directory all as one file and upload it to the Brightspace system for **TC.P3.Task#7.**

**TC.P3.Task#8) Halting Predicate.** In a text (.txt) file, type a simple report on whether or not the set of transitions halted for each of the inputs provided in **TC.P3.Task#6. ALSO, collect all pieces of .java code** that determines that the transitions halted into a temporary directory. Place the text file report into this directory as well. Compress/zip the directory all as one file and upload it to the Brightspace system for **TC.P3.Task#8.**

**TC.P3.Task#9) Turing Machine Transitions.** Collect all the .java code in your system necessary to interpret a transition and place these pieces of code (each in its own text file) into a single temporary directory on your computer system. Compress/zip the directory all as one file and upload it to the Brightspace system for **TC.P3.Task#9.**

**TC.P3.Task#10) Overall Project Assessment.** Read over the specifications for this phase. Type into a text (.txt) file a report on what was accomplished and what was not accomplished in your final project submission (**cf. TC.P3.Task#1 above**). What functionality that you submitted works and which functionality in your code does not work. For the latter, what do you think is causing the problem? This report should be typed into a text (.txt) file which you will upload to Brightspace system for **TC.P3.Task#10**.