## Assignment F - Turing Machines

*Reminder of the recommended approach to best prepare you for success on the exams:*

1. Attempt each of the primary activities *without* reference to others' solutions or use of an automated computational tool (e.g., JFLAP).
2. Develop a solution and a set of data that helps validate the solution.
3. After you have deemed a solution satisfactory, implement that solution in JFLAP where appropriate and use JFLAP's features to analyze and test your work.
4. Clean up or annotate your solution as necessary for submission.

F1. Define a Turing machine TM1 that accepts the language L1 = { ww | w ∈ Σ\*, |w| > 0 } over the alphabet Σ = {a, b}.

Implement TM1 in JFLAP and upload the Turing machine as a file named f1tm.jff

Use Online Text to describe observations and insights associated with the learning experience of creating TM1.

F2. Define a Turing machine TM2 that transforms a unary representation of a number into the unary representation of that number modulo 3. The input alphabet is Σ = {1}. For example, if the starting configuration is "...□1111111□..." the output configuration will be "...□1□..." after halting. Similarly, starting with "...□11111□..." will result in "...□11□...". (Note carefully the positions of the read/write head.)

Implement TM2 in JFLAP and upload it as a file named f2tm.jff

Use Online Text to describe observations regarding the output shown by JFLAP related to the positioning of the read/write head. Also report insights associated with the learning experience of creating TM2.

F3. Define a Turing machine TM3 that decides language L3 = { w | w ∈ Σ\*, #a(w) = #b(w) } over the alphabet Σ = {a, b}.

Implement TM3 in JFLAP and upload the Turing machine as a file named f3tm.jff

Use Online Text to describe differences between your experiences with creating TM3 and each of TM1 and TM2.

F4. Define a Turing machine TM4 that transforms input of the form **0**i**1**k with 0 ≤ k ≤ i by appending **1** symbols as necessary to make the resulting string of the form **0**i**1**i. For example, if the input to TM4 was "...□0001□□..." then the output will be "...□000111□□..." after halting. (Note carefully the positions of the read/write head.)

Implement TM4 in JFLAP and upload it as a file named **f4tm.jff**

Use Online Text to describe any particular difficulties encountered while developing TM4. Reflect and report insights gained while testing and modifying your implementation of TM4.

(F5 involves the use of JFLAP's Turing machine building blocks. Review Turing machines in the [JFLAP Tutorial](http://www.jflap.org/tutorial/) and the notes on [Changes to Turing Machines in 7.1](http://www.jflap.org/tutorial/turing/changes7.1/index.html).)

F5. Define the following three simple Turing machines over input alphabet {□, 0, 1} to be used as components for another Turing machine.

* TMScanR : Move right. If the character under the read/write head is not blank (□), repeat. If blank, halt. This finds the first blank square to the right of the current square and leaves the read/write head at that location.
* TMScanL: Move left. If the character under the read/write head is not blank (□), repeat. If blank, halt. This finds the first blank square to the left of the current square and leaves the read/write head at that location.
* TMShiftL: Transform input "...□u□w□..." into output "□uw□". This shifts the string that is to the right of the read/write head over one square to the left by copying each symbol onto the square immediately to its left. Note that the read/write head begins on a blank between two strings and ends on the blank to the right of the concatenated strings.

F5.a. Implement each of the three building blocks in JFLAP and upload them as **f5TMScanR.jff**, **f5TMScanL.jff**, and **f5TMShiftL.jff** respectively.

F5.b. Use these three Turing machines as building blocks for a machine TM5 that concatenates multiple strings on the input tape. For example, if the input configuration to TM5 was "...□001□1110□101□□..." then the output configuration will be "...□0011110101□□..." after halting.

Implement TM5 in JFLAP using the previously defined Turing Machine building blocks and upload that implementation as **f5tm.jff**

Use Online text to report your confidence in the correctness of your implementation of TM5 and to provide a reflection on the experience of developing tests for TM5.

#### Submission Summary

JFLAP files: f1tm.jff, f2tm.jff, f3tm.jff, f4tm.jff, f5TMScanR.jff, f5TMScanL.jff, f5TMShiftL.jff, f5tm.jff

Online text: √