Project x Teamwork <phoebe> - <chuang26>

Version 1 9/11/24

A ***separate copy*** of this template should be filled out and submitted by each student, regardless of the number of students on the team. Also change the title of this template to "Project x Teamwork <team> - <netid>"

<table>
<tr><td>1</td><td>Team Name: phoebe</td></tr>
<tr><td>2</td><td>Individual name: Phoebe Huang</td></tr>
<tr><td>3</td><td>Individual netid: chuang26</td></tr>
<tr><td>4</td><td>Other team members names and netids: N/A</td></tr>
<tr><td>5</td><td>Link to github repository: https://github.com/chuang26-hue/TM_Project</td></tr>
<tr><td>6</td><td>Overall project attempted, with sub-projects:
Tracing NTM Behavior</td></tr>
<tr><td>7</td><td>List of included files (if you have many files of a certain type, such as test files of different sizes, list just the folder): (Add more rows as necessary)
<table>
<tr><td>File/folder Name</td><td>File Contents and Use</td></tr>
<tr><td colspan="2">Code Files</td></tr>
<tr><td>traceNTM_phoebe.py</td><td>This file is the main code file that traces and simulates a NTM by using BFS</td></tr>
<tr><td colspan="2">Test Files</td></tr>
<tr><td>input.txt</td><td>Includes the name of the file describing the machine, the input strings to test, a ”termination” flag that will stop execution under some circumstance such as if the depth of the configuration tree exceeds a limit</td></tr>
<tr><td>NTM.csv</td><td>Defines the NTM, with the header lines being
Line 1: Name of machine 5
• Line 2: List of state names for Q
• Line 3: List of characters from Σ
• Line 4: List of characters from Γ
• Line 5: The start state
• Line 6: Accept state
• Line 7: Reject state
The rest are transition lines</td></tr>
<tr><td colspan="2">Output Files</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>output_phoebe.txt</td><td>The output echoes the name of the machine, the initial string, the depth of the tree of configurations, and the total number of transitions simulated.
1. If the simulation halts because of reaching an accept configuration, print out the following:
a. ”String accepted in ” and then the number of transitions from the start to the accept on just the accepting path (the depth of the tree).
b. Starting at the starting configuration, print out each configuration in the format: left of head string, state, head character and right of string.
2. If the simulation halts because of all paths lead to reject, ” String rejected in ”, followed the number of steps from the start to the last reject
3. If the step limit is exceeded, print out something like ”Execution stopped after” the max step limit.</td></tr>
</table>
</td></tr>
<tr><td>8</td><td>Individual Student time (in hours) to complete: 6-7 hours</td></tr>
<tr><td>9</td><td>Your specific activities and responsibilities: I completed everything as this as a 1-person project</td></tr>
<tr><td>10</td><td>What was personally learned (topic, programming, algorithms)

Theory of Computing Concepts:

● Deepened understanding of Non-Deterministic Turing Machines (NTMs), their operation, and their role in computational theory.

Programming Skills:

● Improved Python skills, such as file handling
● Gained experience handling edge cases

Algorithm Design:

● Learned to implement depth and step limits to manage non-determinism
● Learned about metrics to evaluate NTMs such as average non-determinism

Debugging and Testing:

● Strengthened debugging skills and the importance of continuous testing</td></tr>
<tr><td>11</td><td>How team was organized, and what might be improved: no team organization as it was a one person team and I handled everything. One improvement is I will start doing the</td></tr>
</table>

	project earlier and divide the work load over a longer period of time instead of procrastinating to the last second.
12	Any additional material: N/A