Project Readme Template

Version 1 9/11/24

A single copy of this template should be filled out and submitted with each project submission, regardless of the number of students on the team. It should have the name readme_"teamname"

Also change the title of this template to "Project x Readme Team xxx"

<table>
<tr><td>1</td><td>Team Name: cmassman</td></tr>
<tr><td>2</td><td>Team members names and netids
Katie Massman
Netid - cmassman</td></tr>
<tr><td>3</td><td>Overall project attempted, with sub-projects:
Program 1: Tracing NTM behavior</td></tr>
<tr><td>4</td><td>Overall success of the project: Successfully implemented the Non-Deterministic Turing Machine (NTM) tracing behavior using breadth-first exploration of configuration trees. The project correctly identifies accept/reject states and outputs the configuration paths and transitions. Minor issues were debugged and resolved to improve accuracy.</td></tr>
<tr><td>5</td><td>Approximately total time (in hours) to complete: Around 5 hours spent working on the algorithm implementation, testing, and performance analysis.</td></tr>
<tr><td>6</td><td>Link to github repository: https://github.com/cmassman1/theory_proj2</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). 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>traceTM_cmassman.py</td><td>Main code file for simulating NTM behavior.</td></tr>
<tr><td colspan="2">Test Files</td></tr>
<tr><td>a_plus_cmassman.csv</td><td>Defines the NTM for the language a+a^+a+.</td></tr>
<tr><td>abc_plus_cmassman.csv</td><td>Defines the NTM for the language a*b*c*</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>abcd_star_cmassman.csv</td><td>Defines the NTM for the language (abcd)*</td></tr>
<tr><td colspan="2">Output Files</td></tr>
<tr><td>test_outputs_cmassman.txt</td><td>Outputs from running various test cases.</td></tr>
<tr><td colspan="2">Plots (as needed) - replaced with analysis document</td></tr>
<tr><td>Project 2 Analysis cmassman - cmassman</td><td>Analysis of how code works and how nondeterminism was measured</td></tr>
</table>
</td></tr>
<tr><td>8</td><td>Programming languages used, and associated libraries:

Programming Language: Python
Libraries Used:

● <code>csv</code> (to parse NTM definitions)
● <code>os</code> (to check file existence)</td></tr>
<tr><td>9</td><td>Key data structures (for each sub-project):

NTM Configuration Tree:

● A list of lists, where each inner list represents a level of configurations at a given depth.

Transitions:

● A dictionary with keys as <code>(state, input)</code> pairs and values as lists of possible transitions <code>(new_state, write, move)</code>.</td></tr>
<tr><td>10</td><td>General operation of code (for each subproject)

Program 1: Tracing NTM Behavior

● Parses an NTM definition from a <code>.csv</code> file.
● Uses breadth-first search (BFS) to explore all possible paths.
● Stops when an accept state is reached or all configurations reject.
● Outputs the configuration tree and details of transitions.</td></tr>
</table>

11	What test cases you used/added, why you used them, what did they tell you about the correctness of your code. **Test Cases:** ● `"aaa"` and `""` for `a_plus.csv`: Validates a+a^+a+ for a non-deterministic machine. ● `"abc"` and `"abcab"` for `abc_star.csv`: Tests non-deterministic transitions with a*b*c* ● `"Adcd"and "Abcd"` for `abcd_star.csv`: Confirms correct traversal for (abcd)∗. **Purpose:** ● To verify correct handling of accept states, reject paths, and configuration depth. **Results:** ● The test cases demonstrated that the code correctly handles non-deterministic transitions and terminates when appropriate.
12	How you managed the code development ● Incremental development using a combination of: ○ **Function-based structure:** Individual functions for parsing, simulating, and printing results. ○ **Debugging and iteration:** Testing after implementing each sub-feature (e.g., transition parsing, BFS logic). ○ **GitHub repository:** For version control and collaboration.
13	Detailed discussion of results: ● The code successfully simulated and traced NTMs for all provided `.csv` files. ● Depths and transitions were correctly calculated. ● Edge cases (e.g., empty input for a+a^+a+) were handled as expected. ● Redundant configurations in the accept state were resolved.
14	How team was organized Single-member team (Katie Massman) managing all aspects, including: ● Writing code. ● Debugging issues. ● Designing and running test cases. ● Documenting results

15	What you might do differently if you did the project again Implement a more modular design for transition logic to easily extend support for multi-tape Turing Machines. Optimize handling of large configuration trees for complex NTMs. Add more extensive automated testing for edge cases.
16	Any additional material: Outputs for all test cases are saved in `test_outputs.txt` for reference. Comments are included in the code for better readability and maintenance.

Summary Table of Simulation Results

The following table summarizes the performance and results of the NTM simulation for each input string. It includes key metrics such as the depth of the tree, the number of configurations explored, and the average non-determinism per depth level.

NTM used	String Used	Result	Depth of tree	Configurations explored	Average non-determinism	Comments
abc_star_cmassman.csv	abcab	Rejected	4	13	3.25	No valid path found
abc_star_cmassman.csv	abc	Accepted	5	14	2.8	Explored multiple paths
a_plus_cmassman.cvs	“”	Rejected	0	0	0	Immediate rejection
a_plus_cmassman.csv	aaa	Accepted	5	5	1.0	Linear path followed
abcd_star_cmassman.csv	adcd	Rejected	2	2	1.0	Incorrect input format
abcd_star_cmassman	abcd	Accepted	6	6	1.0	Correct pattern

.csv						found

Metric Explanation

- **1.0:** Fully deterministic (no alternative paths).
- **1.0 - 1.5:** Slight nondeterminism.
- **1.5 - 3.0:** Moderate nondeterminism.
- **3.0 or higher:** Highly nondeterministic with many alternative paths.