Project Readme Rademacher

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: Rademacher</td></tr>
<tr><td>2</td><td>Team members names and netids: Grace Rademacher, grademac</td></tr>
<tr><td>3</td><td>Overall project attempted, with sub-projects: Program 1</td></tr>
<tr><td>4</td><td>Overall success of the project: Correctly identify valid and invalid strings, correctly calculate degree of nondeterminism for machine, correct accepting path provided</td></tr>
<tr><td>5</td><td>Approximately total time (in hours) to complete: 6 hours</td></tr>
<tr><td>6</td><td>Link to github repository: https://github.com/grademac/Theory_Project3</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_Rademacher.py</td><td>Code for program 1</td></tr>
<tr><td colspan="2">Test Files</td></tr>
<tr><td>TestFiles_Rademacher</td><td>Folder with all of the different test machines used (a+, a+ DTM, equal 01s, equal 01s DTM, a*b*c*, a*b*c* DTM)</td></tr>
<tr><td colspan="2">Output Files</td></tr>
<tr><td>TestOutput_Rademacher</td><td>Folder with all the different outputs from each of the test machines</td></tr>
<tr><td colspan="2">Plots (as needed)</td></tr>
<tr><td>Project 3 Analytics_Rademacher</td><td>Table of different inputs and outputs for different machines to analyze the accuracy of the code and what the outputs mean</td></tr>
</table></td></tr>
<tr><td>8</td><td>Programming languages used, and associated libraries: Python, import csv</td></tr>
</table>

9	Key data structures (for each sub-project): Tree, List, Tuple
10	General operation of code (for each subproject): Defined a Turing Machine class and used that to trace the path of the program through the given machine. Trace path works by creating a tree and a parent map to trace the paths of the machine via BFS to trace level by level. After either traversing the whole tree or exceeding max depth, the result will either be the string was accepted, rejected, or the program ran too long. Once done, the accepted path is printed out using the parent map to eliminate repeated steps and the degree of nondeterminism is calculated. All information is then output into the desired text file.
11	What test cases you used/added, why you used them, what did they tell you about the correctness of your code: I used the machines a+, a+ DTM, equal 01s, equal 01s DTM, a*b*c*, and a*b*c* DTM. See Project 3 Analytics for more explanation
12	How you managed the code development: I started by defining the turing machine class and from there writing and basic BFS outline. It took a lot of trial and error to figure out what I needed to add, adding the lines of code to account for calculating the degree of nondeterminism and adding parent_map to fix issues of multiple repeated lines printing.
13	Detailed discussion of results: The results showed consistency in the difference between NTM and DTM, where NTMs generally explored more configurations and had a degree of nondeterminism greater than 1. DTMs, on the other hand, explored less configurations and had a degree of nondeterminism equal to 1. This makes sense as NTMs have multiple different transitions for each configuration as DTMs only have one transition for each configuration, hence degree of nondeterminism equals 1 for DTMs.
14	How team was organized: N/A
15	What you might do differently if you did the project again: Probably play around with different machines and try and print out an actual tree to better understand the different configurations of each machine
16	Any additional material: N/A