Project 2 Readme Team Yokogaijin

Version 1 9/11/24

<table>
<tr><td>1</td><td>Team Name: Yokogaijin</td></tr>
<tr><td>2</td><td>jrhodus@nd.edu</td></tr>
<tr><td>3</td><td>Overall project attempted, with sub-projects:
K Tape Deterministic Turing Machine (DTM)</td></tr>
<tr><td>4</td><td>The overall success of the project:
I feel it was a struggle initially, however, I am proud of what I was able to make. The programs work, and the csv files that accompany it.</td></tr>
<tr><td>5</td><td>Approximately total time (in hours) to complete:
20</td></tr>
<tr><td>6</td><td>Link to github repository:
https://github.com/Yokogaijin/Project2_TheoryOfComputing</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><th>File/folder Name</th><th>File Contents and Use</th></tr>
<tr><td colspan="2">Code Files</td></tr>
<tr><td></td><td>ktape_Yokogaijin.py</td></tr>
<tr><td colspan="2">Test Files</td></tr>
<tr><td></td><td>check_k_tape_palindromic_DTM.csv-Yokogaijin
check_k_tape_0_plus_DTM.csv-Yokogaijin</td></tr>
<tr><td colspan="2">Output Files</td></tr>
<tr><td></td><td>output_0^+_resultgood_Yokogaijin
output_FinalrResultOfPalon.png_Yokogaijin
output_PalindromicWorks.png_Yokogaijin</td></tr>
<tr><td colspan="2">Plots (as needed)</td></tr>
<tr><td></td><td>N/A</td></tr>
</table>
</td></tr>
</table>

8	Programming languages used, and associated libraries: Python3
9	Key data structures (for each sub-project): The key data structure I guess would just be the k-tape, however, the csv files provided the necessary states and transitions for it to work.
10	General operation of code (for each subproject): For k-tape I want it to work in conjunction with the CSV files. It works by determining if the string is in the language. Then the code will read the csv file, from the top [{Tapes name}, {k tapes that the machine would run}] The states of the machine Tape alphabet Tape input alphabet Initial state Accept state Final state Transitions of the states. Then in the python file, at the bottom you would would see filename: this is where you put your desired csv file Then below that, you would put a input string. The k-tape TDM would determine if that string is in the language of the CSV file, if yes, then it would say input accepted. If not, then it would say halt, your string is not in the language.
11	What test cases you used/added, why you used them, what did they tell you about the correctness of your code.: I used Online Turing Machine Simulator to prove that my results were correct. I used this because many universities use this site, so it has to be good. https://turingmachinesimulator.com This site allowed me to confirm that my code functions properly.
12	How you managed the code development: I really rushed development at the end, I started several weeks before hand, however then ended up rushing the entire thing, that was my bad.
13	Detailed discussion of results: For 0^+, I am thoroughly happy that it finds the string is in the language or not accurately. However, for the palindromic code, I had to put $ in the front of the string for it not to be a headache, so the code knows if it's at the beginning or end. So I recommend doing the same.
14	How team was organized: Me, Myself, and I did the project.
15	What you might do differently if you did the project again: I should really have started earlier and spread out my project. It caused unwanted stress that entirely was entirely my fault. So, I guess the motivation to get started earlier.

16	Any additional material: