Project 2 Readme cb cbarco

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"

1	Team Name: cb
c	Cassandra barco (cbarco)
3	Overall project attempted, with sub-projects: NTM trace
4	Overall success of the project: I had a running machine but it follows some of the assumptions described in the project file. It may be limited because of these assumptions, so it
5	Approximately total time (in hours) to complete: 6-8 hrs
6	Link to github repository:https://github.com/cb-214/Project2-TOC
7	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.

File/folder Name	File Contents and Use
Code Files	
CB_tracer.py/src	This is the main file for NTM trace program. In this program I work with functions to complete this. I parse the csv file to turn into NTM. Then I actually simulated it using the tracer function.
Test Files	
a_plus_.csv/input composite.csv/input one_a_checker.csv/input	Files as described in the project description.
Output Files	

	Run on terminal | (blank) Plots (as needed) Not needed | (blank)
8	Programming languages used, and associated libraries: python, csv
9	Key data structures (for each sub-project): dictionary, tree , list[list]
10	General operation of code (for each subproject) As Lax’s comments suggest the program relies on BFS to simulate a NTM. The csv input files are parsed within the tracer program and converts it to the appropriate TM properties. The actual tracing of the NTM is done in the function ntm_tracer. This function explores all the possible configurations using BFS to go through each transition.
11	What test cases you used/added, why you used them, what did they tell you about the correctness of your code. No test cases added.
12	How you managed the code development I initially was going to revert my code to Lax’s but since i didn't rely on classes like him, I decided to revert to mine. I use git commands to keep track of what Ive completed and need to complete (listed as TBD in my commit messages).
13	Detailed discussion of results: The NTM simulator was tested using a machine based on the input file the user enters, same with the input string. The NTM simulation explores all possible configurations using BFS and prints each configuration level by level, clearly showing the machine’s behavior. When the input is accepted, the simulator halts at the accepting configurations and shows the steps taken. When rejected, it shows the max depth reached. These results show that the simulator correctly handles nondeterministic branching, following same logic in the skeleton code.
14	How team was organized : NA
15	What you might do differently if you did the project again I might try to add make it more Pythonic and add shortcuts for my code since i think the nested loops were a confusing part for me since i had to work with multiple variables. If I understood LAx’s comments, I woudlve just worked with the skelton code since it follows the same logic and already had many of the initializations done. However, I also think I couldve split my ntm trace function into two and maybe tested the function out separately rather than putting it in here so its more concise and clear what each function does.
16	Any additional material: