Project 2 Readme Team drew

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

machine name	input string	result	depth	configurations explored	average ndm
abc_star_drew.csv	abcabab	reject	4	13	3.25
abc_star_drew.csv	abc	accept	5	14	2.50
aplus_drew.csv	y	reject	1	1	1.00
aplus_drew.csv	aaaaa	accept	7	7	1.00

1	Team Name: drew
2	Team members names and netids: Drew DiGuglielmo - ddigugli
3	Overall project attempted, with sub-projects: traceTM
4	Overall success of the project: successful!
5	Approximately total time (in hours) to complete: 10
6	Link to github repository: https://github.com/ddigugli/theory_project_2
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	
traceTM_drew.py	main code to perform the traceTM project, usage is `$ python3 traceTM_drew.py`
Test Files	
aplus_drew.csv	this is the csv file to perform test cases on strings

	with the a+ machine
abc_star_drew.csv	this is the csv file to perform test cases on strings with the a*b*c* machine
Output Files	
output_drew.txt	this just saves what is printed in terminal, the result of the test cases that were run

8	Programming languages used, and associated libraries: I used python and the libraries I used were os and csv.
9	Key data structures (for each sub-project): the key data structures I have in this project is the dictionary transitions, the list of lists tree, and a set visitedconfigs. Transitions is a dictionary where the keys are tuples, (state, character), and values are lists of tuples, (new state, write symbol, move direction). This stores my transition rules for the ntm. Tree is a list of lists where the sublists are a level of configurations in the bfs simulation. Finally, visitedconfigs is a set which tracks configurations I have already looked at to avoid repetition and revisiting.
10	General operation of code (for each subproject): The code simulates the behavior of an ntm based on the rules provided in the file. sim_ntm is a function which takes an input and simulates the ntm by first performing bfs on the configurations, for each configuration, possible transitions are applied from the current state/character, it continues simulating until accept is reached or there are no more possible transitions, and it also tracks the depth of the tree and the total number of transitions we have simulated.
11	What test cases you used/added, why you used them, what did they tell you about the correctness of your code. I used abc_star_drew.csv and aplus_drew.csv to test my code. I used the strings abcabab and abc for abc_star to check the machines ability to accept or reject strings following the a*b*c* pattern. I used aaaaa and y for aplus to verify my codes ability to accept or reject strings based of the a+ pattern.These cases helped to verify my machines ability to accept or reject and the output I provided shows both cases where we have an accept and a reject for each simulation.
12	How you managed the code development: I developed the code incrementally, I started with parsing and transition simulation and then refined the handling of configuration trees. I periodically saved to track progress and used many print statements throughout my code (now removed) to help with debugging when my machine was not passing the test cases I knew it was supposed to.
13	Detailed discussion of results: - abc_star_drew.csv with input abcabab, the string is rejected after several transitions due to failure to reach an accepting state - abc_star_drew.csv with input abc, the string is accepted after five transitions, demonstrating that the machine correctly handles the pattern a*b*c*. - aplus_drew.csv with input y, the string is rejected as it does not conform to the

	grammar expected by the machine - aplus_drew.csv with input aaaaa, the string is accepted, and the simulation tracks each step correctly
14	How team was organized: Since individual, team organization was not considered.
15	What you might do differently if you did the project again: Potentially work with a partner, even though it is more work, it would aid in times when I got stuck and did not know what to do while staring at the code.
16	Any additional material: NA