

# Project 2 Readme Team mjimNTM

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: mjimNTM															
2	Team members names and netids: Mariya Jim (mjim)															
3	Overall project attempted, with sub-projects: Tracing NTM Behavior															
4	Overall success of the project: I successfully created a program that traces the possible paths for NTMs and calculate the nondeterminism.															
5	Approximately total time (in hours) to complete: 8 hours															
6	Link to github repository: <a href="https://github.com/mjim18/mjim">https://github.com/mjim18/mjim</a>															
7	<p>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.</p> <table border="1"><thead><tr><th>File/folder Name</th><th>File Contents and Use</th></tr></thead><tbody><tr><td colspan="2">Code Files</td></tr><tr><td>traceTM_mjimNTM.py</td><td>This file includes the algorithm and reads in the csv file and tracks the NTM performance.</td></tr><tr><td colspan="2">Test Files</td></tr><tr><td>check_a_plus_mjimNTM.csv check_abc_star_mjimNTM.csv check_equal_01s_mjimNTM.csv</td><td>This file includes the NTMs describing the machine as a csv file.</td></tr><tr><td colspan="2">Output Files</td></tr><tr><td>output_mjimNTM</td><td>This file is a screenshot of the output from the terminal which includes the depth, acceptance/rejected, and average</td></tr></tbody></table>		File/folder Name	File Contents and Use	Code Files		traceTM_mjimNTM.py	This file includes the algorithm and reads in the csv file and tracks the NTM performance.	Test Files		check_a_plus_mjimNTM.csv check_abc_star_mjimNTM.csv check_equal_01s_mjimNTM.csv	This file includes the NTMs describing the machine as a csv file.	Output Files		output_mjimNTM	This file is a screenshot of the output from the terminal which includes the depth, acceptance/rejected, and average
File/folder Name	File Contents and Use															
Code Files																
traceTM_mjimNTM.py	This file includes the algorithm and reads in the csv file and tracks the NTM performance.															
Test Files																
check_a_plus_mjimNTM.csv check_abc_star_mjimNTM.csv check_equal_01s_mjimNTM.csv	This file includes the NTMs describing the machine as a csv file.															
Output Files																
output_mjimNTM	This file is a screenshot of the output from the terminal which includes the depth, acceptance/rejected, and average															

		nondeterminism.
	Plots (as needed)	
8	Programming languages used, and associated libraries: I used the csv library in Python.	
9	Key data structures (for each sub-project): I used a dictionary for the transitions to allow a real time lookup based on the current state and tape symbol. The tree represented by a list of lists where each sublist uses BFS for possible configurations. Tuples were used to create keys in the dictionary.	
10	General operation of code (for each subproject): The code simulates a NTM using breadth first search by reading from a file with TM definition, initializations, and explores the possible paths until the input string is accepted, rejected, or maximum number of steps is reached. The parse_tm_file reads the csv file pairs the possible transitions in the dictionary. The simulate function initializes a tree using lists with the starting state, input string, and parent. At each level, the current_level is popped and processed by checking whether it is accepted or rejected. For transitions, the current symbol under the head is found and the dictionary is used to find all possible next states. The average nondeterminism is also calculates using the step_count. The print function moves through the accept state and prints the path. If the machine accepts, the path is printed with the sequence or if rejected or maximum transitions are reached, a rejection message is printed. The depth is also printed.	
11	<p>What test cases you used/added, why you used them, what did they tell you about the correctness of your code.</p> <p>I used the a plus, abc star, and equal 01s test cases to analyze the machine's behavior for a variety of input strings to see that it can process different language types. The a+ test case checks if the machine can handle a repetitive pattern. The abc star case checks if the TM can handle multiple symbol types in a sequence. The equal 01s case checks if the machine can handle binary inputs and process accurately and check against requirements.</p>	
12	<p>How you managed the code development: I looked at NTM problems I've done prior to understand it conceptually and then I looked into how turing machines are implemented codewise. I broke down the different components of the algorithm into functions to simplify the main function and keep it organized. I incrementally tested simple NTMs as I went along to ensure that the algorithm was running properly. Then I made the function to read in the csv file and started using testing using the test files.</p>	
13	<p>Detailed discussion of results: The machine accepted the strings aaa and aabc and rejected 10101, which matches the expected behavior based on the definitions. Despite the non-deterministic design of the machine, it effectively operated in deterministically for these test cases. The transitions were perhaps defined in a way that the machine didn't need to branch non-deterministically. The depth indicates the number of levels that were explored during the simulation.</p>	

14	How team was organized : I did the project solo.
15	What you might do differently if you did the project again: I would make more test cases to see if my program works accurately for a more ambiguous input string and test if it can handle truly non-deterministic scenarios.
16	Any additional material:

NTM used:	A plus	abc star	equal 01s
String	aaa	aabc	10101
Results (Accept/reject/too long):	accept	accept	reject
Depth of tree	4	5	0
Number of config explored:	4	5	13
Average nondeterminism:	1.0	1.05	1.0
Comments	The machine behaved deterministically	The machine behaved non deterministically	The machine behaved deterministically