

Project 2 Readme Team dsimmon4

1	Team Name: dsimmon4																		
2	Team members names and netids: N/A																		
3	Overall project attempted, with sub-projects: Tracing NTM Behavior																		
4	Overall success of the project: Successful																		
5	Approximately total time (in hours) to complete: 4																		
6	Link to github repository: https://github.com/d-simmons/traceNTM_dsimmon4																		
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>traceNTM_dsimmons/traceTM_dsimmon4.py</td><td>Code to trace and simulate NTM as well as print out results.</td></tr> <tr> <td colspan="2">Test Files</td></tr> <tr> <td>traceNTM_dsimmons/aplus.csv</td><td>Outlines the a+ NTM so that the python program can interpret it.</td></tr> <tr> <td colspan="2">Output Files</td></tr> <tr> <td>traceNTM_dsimmons/output__accept_dsimmon4.png traceNTM_dsimmons/output__reject_dsimmon4.png</td><td>Terminal output of running a+ traces.</td></tr> <tr> <td colspan="2">Plots (as needed)</td></tr> <tr> <td></td><td></td></tr> </tbody> </table>	File/folder Name	File Contents and Use	Code Files		traceNTM_dsimmons/traceTM_dsimmon4.py	Code to trace and simulate NTM as well as print out results.	Test Files		traceNTM_dsimmons/aplus.csv	Outlines the a+ NTM so that the python program can interpret it.	Output Files		traceNTM_dsimmons/output__accept_dsimmon4.png traceNTM_dsimmons/output__reject_dsimmon4.png	Terminal output of running a+ traces.	Plots (as needed)			
File/folder Name	File Contents and Use																		
Code Files																			
traceNTM_dsimmons/traceTM_dsimmon4.py	Code to trace and simulate NTM as well as print out results.																		
Test Files																			
traceNTM_dsimmons/aplus.csv	Outlines the a+ NTM so that the python program can interpret it.																		
Output Files																			
traceNTM_dsimmons/output__accept_dsimmon4.png traceNTM_dsimmons/output__reject_dsimmon4.png	Terminal output of running a+ traces.																		
Plots (as needed)																			
8	Programming languages used, and associated libraries: Python with csv library																		
9	Key data structures (for each sub-project): A tree was used to hold the possible nondeterministic paths the machine could take. Other arrays were used as needed, but the tree was the primary data structure of this project.																		
10	General operation of code (for each subproject): This code generally works by parsing a																		

	given “machine file” or any csv file that contains a machine definition. The program then simulates the possible paths that the machine could go given the input string. When it finds one that reaches an accept state, it returns it and prints the path.																		
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>The test cases I used came from changing the input string in the python program. By just changing this I could run as many different strings through the machine as I wanted. By doing this with several strings, it is clear to me that the a+ machine only accepts when the string consists only of a’s and there must be at least one a.</p>																		
12	<p>How you managed the code development.</p> <p>I started by getting a big picture idea of what I needed to complete this project. After that I began setting up the object oriented structure that I wanted to use with the NTM class and its init variables. After that I wrote the code that would fill in the class variables I had just made with information from the machine csv file. Next, I wrote the main functionality of the program with the function that simulates the entire non-determinism of the machine with the tree data structure discussed above. Finally, the output was finalized and I moved on to more testing with bug fixes as needed.</p>																		
13	<p>Detailed discussion of results:</p> <p>The program works as expected and accurately simulates a NTM. When a string is provided that is accepted, the path to that accept state is accurately given. Likewise, when a string that is not accepted by the machine is given, the program accurately detects this reject and reports it.</p> <p>Here are the results of a couple traces:</p> <table><tr><td>Machine</td><td>a+</td><td>a+</td></tr><tr><td>Input String</td><td>aaaaa</td><td>aaaaab</td></tr><tr><td>Result</td><td>accept</td><td>reject</td></tr><tr><td>Depth</td><td>6</td><td></td></tr><tr><td>Number of configurations</td><td>12</td><td></td></tr><tr><td>Average Non-determinism</td><td>1.71</td><td></td></tr></table>	Machine	a+	a+	Input String	aaaaa	aaaaab	Result	accept	reject	Depth	6		Number of configurations	12		Average Non-determinism	1.71	
Machine	a+	a+																	
Input String	aaaaa	aaaaab																	
Result	accept	reject																	
Depth	6																		
Number of configurations	12																		
Average Non-determinism	1.71																		
14	<p>How team was organized.</p> <p>I did this solo.</p>																		
15	<p>What you might do differently if you did the project again:</p> <p>I might have made more machines to see how it would work with various NTMs.</p>																		
16	<p>Any additional material:</p>																		