Project 2 Readme Team isadurni

Version 1 9/11/24

<table>
<tr><td>1</td><td>Team Name: isadurni</td></tr>
<tr><td>2</td><td>Team members names and netids:
● Ignacio Sadurni (isadurni)</td></tr>
<tr><td>3</td><td>Overall project attempted, with sub-projects: Tracing NTM Behavior
Problem: Given an input for an NTM, read in the machine and for a string explore all configurations and analyze the nondeterminism.</td></tr>
<tr><td>4</td><td>Overall success of the project:
The project was successful, managing to read in NTM and exploring the non deterministic nature for accepting and rejecting different strings.</td></tr>
<tr><td>5</td><td>Approximately total time (in hours) to complete: Around 8 hours</td></tr>
<tr><td>6</td><td>Link to github repository: https://github.com/isadurni/theory-project2</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">Test Files</td></tr>
<tr><td>● a*b*c*.csv
● a+.csv
● str1.txt
● str2.txt
● str3.txt
● str4.txt</td><td>.csv files are NTM with their headers and transitions.

.txt files are string inputs to simulate the NTMs.</td></tr>
<tr><td>ntm_isadurni.py</td><td>Code file to read and simulate the NTMs.</td></tr>
<tr><td>output_isadurni.pdf</td><td>Output file with details and evidence of running the simulations with results in a table.</td></tr>
<tr><td>readme_isadurni.pdf</td><td>readme file</td></tr>
</table>
</td></tr>
<tr><td>8</td><td>Programming languages used, and associated libraries:</td></tr>
</table>

	● Python ○ standard library
9	Key data structures (for each sub-project): ● Dictionary ○ The NTM is a dictionary with all of its parts consisting of string and lists of strings ● List of Lists ○ Used to organize lists of configurations by depth (tree).
10	General operation of code (for each subproject): The code works pretty simple. Running the python script with two files argv[1] being the NTM .csv files and the argv[2] being the .txt files containing the string. Then a function is used to read in the csv file and create a dictionary which contains all the header information from the NTM, as well as the transitions in a list. The big function to simulate the NTM runs a breadth first search which essentially explores all possible branches and creates a tree of configurations for the simulation of the NTM. When there are no more configurations to run we have reached an accept or reject state or when we exceed the limit of transitions we chose to hardcode (in this example I put 100), the simulation is ended and we input the transitions, depth of the tree, and its configurations.
11	What test cases you used/added, why you used them, what did they tell you about the correctness of your code. I utilized the test cases provided by the professor in the google drive. I slightly modified them but generally kept the same idea and by running different strings I could test the code by observing the output and manually checking if the transitions were correct. Also there is an example for the a plus NTM in the project details file which I could compare with.
12	How you managed the code development: The code development was written in python using vs code using a very similar format to my experience at a previous Notre Dame course, programming challenges, where we did various programs on reading csv files and writing a BFS algorithm.
13	Detailed discussion of results: Analyzing results we are able to conclude on the following. Simpler NTMs are less non deterministic. The a+ NTM generally is less nondeterministic for same inputs for the NTM a*b*c*. Additionally, longer strings make the nondeterminism higher as well. And finally, if strings are rejected, this increases non determinism as well. In conclusion the complexity of the NTM, the length of the string, and if the string is rejected all contribute to a higher non determinism.
14	How team was organized: Individual - n/a
15	What you might do differently if you did the project again: If I was to repeat this project I would have selected more complex NTMs which would have made the complexity of the project higher and more interesting.
16	Any additional material:

	n/a