

Project 2 Results Discussion

Luca Watson

CODE DISCUSSION:

My code works by first taking command line arguments and using these to generate an object to represent an NTM, complete with a starting state, accept state, reject state, and list of transitions. The code then creates a representation of the starting configuration, with the state and starting string, and adds it to a list to serve as the first level of my tree. The code then loops through each level of the tree, then loops through each configuration in this level, and then loops through all of the NTMs transitions for each configuration to check if each could be taken. If a transition can be taken, as dictated by the state and input symbol described by the transition, a new configuration will be generated from the current configuration and transition description and added to the next level. This continues until the loop is broken by an accept state being reached, or every configuration entering the reject state by an inability to make any other transitions.

Non-determinism is modelled in my code by a single configuration being able to produce multiple other configurations in the next level of the tree if it is able to take multiple transitions. My test cases demonstrate this behavior as they produced trees that have more transitions taken than levels.

I traced my machine's execution by adding a fourth element to each configuration list to represent its parent configuration within the tree, or the configuration that it transitioned from. This makes it much easier to trace the path taken by any given string.

Non-determinism in my machine is calculated according to the description provided - it is the average number of new configurations generated by an average configuration, which is calculated by averaging the number of configurations in each level divided by the number of configurations in the previous level. The non-determinism values output by the machines make sense, as completely deterministic machines all output a determinism value of 1.0, while more non-deterministic machines have values well above 1.

I tested my code by running test scripts with varying levels of determinism. I first started with the example provided and made sure the output matched, then I used relatively short strings for each machine and checked that the input was correct by comparing it to what I got by hand. I then used much longer strings and modified my code to print out

each configuration and transition taken, and verified that the path taken made sense given the machine.

TEST CASES + OUTPUT

Machine: a_equal_b.csv

Input string: abbabababbab

Output:

Machine name: {w | w has the same number of a's and b's} - deterministic

Initial string: abbabababbab

Tree depth: 75

Transitions simulated: 74

String rejected in: 75

Degree of non-determinism: 1.0

Input string: abbbaabbabaabb

Output:

Machine name: {w | w has the same number of a's and b's} - deterministic

Initial string: abbbaabbabaabb

Tree depth: 132

Transitions simulated: 131

String accepted in: 132

Path taken:

['____', 'q0', 'abbbaabbabaabb____']

['____x', 'q1', 'bbbaabbabaabb____']

['____', 'q3', 'xxbaaabbabaabb____']

['____', 'q3', '_xxbaaabbabaabb____']

['____', 'q0', 'xxbaaabbabaabb____']

['____x', 'q0', 'xbaaabbabaabb____']

['____xx', 'q0', 'baaabbabaabb____']

['____xxx', 'q2', 'aaabbabaabb____']

['____xx', 'q3', 'xxaabbabaabb____']

['____x', 'q3', 'xxxaabbabaabb____']

['____', 'q3', 'xxxxaabbabaabb____']

['_', 'q3', '_xxxxaabbabababb_']
['_', 'q0', 'xxxxaabbabababb_']
['_x', 'q0', 'xxxaabbabababb_']
['_xx', 'q0', 'xxaabbabababb_']
['_xxx', 'q0', 'xaabbabababb_']
['_xxxx', 'q0', 'aabbabababb_']
['_xxxxx', 'q1', 'abbabababb_']
['_xxxxxa', 'q1', 'bbabababb_']
['_xxxxx', 'q3', 'axbabababb_']
['_xxxx', 'q3', 'xaxbabababb_']
['_xxx', 'q3', 'xxaxbabababb_']
['_xx', 'q3', 'xxxaxbabababb_']
['_x', 'q3', 'xxxxaxbabababb_']
['_', 'q3', 'xxxxxaxbabababb_']
['_', 'q3', '_xxxxxaxbabababb_']
['_', 'q0', 'xxxxxaxbabababb_']
['_x', 'q0', 'xxxxaxbabababb_']
['_xx', 'q0', 'xxxaxbabababb_']
['_xxx', 'q0', 'xxaxbabababb_']
['_xxxx', 'q0', 'xaxbabababb_']
['_xxxxx', 'q0', 'axbabababb_']
['_xxxxxx', 'q1', 'xbabababb_']
['_xxxxxxx', 'q1', 'babababb_']
['_xxxxxxx', 'q3', 'xxabababb_']
['_xxxxx', 'q3', 'xxxabababb_']
['_xxxx', 'q3', 'xxxxabababb_']
['_xxx', 'q3', 'xxxxxabababb_']
['_xx', 'q3', 'xxxxxxabababb_']
['_x', 'q3', 'xxxxxxxabababb_']
['_', 'q3', 'xxxxxxxxxabababb_']
['_', 'q3', '_xxxxxxxxxabababb_']
['_', 'q0', 'xxxxxxxxxabababb_']
['_x', 'q0', 'xxxxxxxxxabababb_']
['_xx', 'q0', 'xxxxxxabababb_']
['_xxx', 'q0', 'xxxxxabababb_']
['_xxxx', 'q0', 'xxxxabababb_']
['_xxxxx', 'q0', 'xxxabababb_']
['_xxxxxx', 'q0', 'xxabababb_']
['_xxxxxxx', 'q0', 'xabababb_']
['_xxxxxxxx', 'q0', 'abababb_']

['____xxxxxxxx', 'q1', 'baabb____']
['____xxxxxxxx', 'q3', 'xxaabb____']
['____xxxxxxx', 'q3', 'xxxaabb____']
['____xxxxxxx', 'q3', 'xxxxaabb____']
['____xxxxx', 'q3', 'xxxxxaabb____']
['____xxxx', 'q3', 'xxxxxxaabb____']
['____xxx', 'q3', 'xxxxxxxaabb____']
['____xx', 'q3', 'xxxxxxxxaabb____']
['____x', 'q3', 'xxxxxxxxxaabb____']
['____', 'q3', 'xxxxxxxxxxaabb____']
['____', 'q3', '_xxxxxxxxxxaabb____']
['____', 'q0', 'xxxxxxxxxxaabb____']
['____x', 'q0', 'xxxxxxxxxxaabb____']
['____xx', 'q0', 'xxxxxxxxxaabb____']
['____xxx', 'q0', 'xxxxxxxxaabb____']
['____xxxx', 'q0', 'xxxxxxaabb____']
['____xxxxx', 'q0', 'xxxxxaabb____']
['____xxxxxx', 'q0', 'xxxaabb____']
['____xxxxxxx', 'q0', 'xxaabb____']
['____xxxxxxxx', 'q0', 'xaabb____']
['____xxxxxxxx', 'q0', 'aabb____']
['____xxxxxxxxxxx', 'q1', 'abb____']
['____xxxxxxxxxxxa', 'q1', 'bb____']
['____xxxxxxxxxxx', 'q3', 'axb____']
['____xxxxxxxxxxx', 'q3', 'xaxb____']
['____xxxxxxxxxxx', 'q3', 'xxaxb____']
['____xxxxxxxx', 'q3', 'xxxaxb____']
['____xxxxxxx', 'q3', 'xxxxaxb____']
['____xxxxxx', 'q3', 'xxxxxaxb____']
['____xxxxx', 'q3', 'xxxxxxaxb____']
['____xxxxx', 'q3', 'xxxxxxxaxb____']
['____xxx', 'q3', 'xxxxxxxxxaxb____']
['____xx', 'q3', 'xxxxxxxxxxaxb____']
['____x', 'q3', 'xxxxxxxxxxxaxb____']
['____', 'q3', 'xxxxxxxxxxxxaxb____']
['____', 'q3', '_xxxxxxxxxxxxaxb____']
['____', 'q0', 'xxxxxxxxxxxxaxb____']
['____x', 'q0', 'xxxxxxxxxxxaxb____']
['____xx', 'q0', 'xxxxxxxxxxaxb____']

['__xxx', 'q0', 'xxxxxxxxaxb__']
['__xxxx', 'q0', 'xxxxxxxxaxb__']
['__xxxxx', 'q0', 'xxxxxxxxaxb__']
['__xxxxxx', 'q0', 'xxxxxaxb__']
['__xxxxxxx', 'q0', 'xxxxaxb__']
['__xxxxxxxx', 'q0', 'xxxaxb__']
['__xxxxxxxxx', 'q0', 'xxaxb__']
['__xxxxxxxxxx', 'q0', 'xaxb__']
['__xxxxxxxxxxx', 'q0', 'axb__']
['__xxxxxxxxxxxx', 'q1', 'xb__']
['__xxxxxxxxxxxxx', 'q1', 'b__']
['__xxxxxxxxxxxxx', 'q3', 'xx__']
['__xxxxxxxxxxxxx', 'q3', 'xxx__']
['__xxxxxxxxxxxxx', 'q3', 'xxxx__']
['__xxxxxxxxxxx', 'q3', 'xxxxx__']
['__xxxxxxxxxx', 'q3', 'xxxxxx__']
['__xxxxxxx', 'q3', 'xxxxxxx__']
['__xxxxxx', 'q3', 'xxxxxxx__']
['__xxxxx', 'q3', 'xxxxxxxxxx__']
['__xxx', 'q3', 'xxxxxxxxxxx__']
['__xx', 'q3', 'xxxxxxxxxxxxx__']
['__x', 'q3', 'xxxxxxxxxxxxxxxxx__']
['__', 'q3', 'xxxxxxxxxxxxxxxxxxx__']
['__', 'q3', '_xxxxxxxxxxxxxxxxxxx__']
['__', 'q0', 'xxxxxxxxxxxxxxxxxxx__']
['__x', 'q0', 'xxxxxxxxxxxxxxxxxxx__']
['__xx', 'q0', 'xxxxxxxxxxxxxxxxxxx__']
['__xxx', 'q0', 'xxxxxxxxxxxxxxxxxxx__']
['__xxxx', 'q0', 'xxxxxxxxxxxxx__']
['__xxxxx', 'q0', 'xxxxxxxxxxx__']
['__xxxxxxx', 'q0', 'xxxxxxx__']
['__xxxxxxxxx', 'q0', 'xxxxxx__']
['__xxxxxxxxxx', 'q0', 'xxxxx__']
['__xxxxxxxxxxx', 'q0', 'xxxx__']
['__xxxxxxxxxxxxx', 'q0', 'xxx__']
['__xxxxxxxxxxxxxxx', 'q0', 'xx__']
['__xxxxxxxxxxxxxxxxx', 'q0', 'x__']
['__xxxxxxxxxxxxxxxxxxx', 'q0', '__']

['____xxxxxxxxxxxxx_', 'q4', '____']
Degree of non-determinism: 1.0

Machine: a_plus.csv

Input string: aaa

Output:

Machine name: a plus

Initial string: aaa

Tree depth: 5

Transitions simulated: 10

String accepted in: 5

Path taken:

['____', 'q1', 'aaa____']

['____a', 'q1', 'aa____']

['____aa', 'q1', 'a____']

['____aaa', 'q2', '____']

['____aaa_', 'qacc', '____']

Degree of non-determinism: 1.2333333333333334

Input string: aaaaaaab

Output:

Machine name: a plus

Initial string: aaaaaaab

Tree depth: 9

Transitions simulated: 23

String rejected in: 9

Degree of non-determinism: 1.1666666666666667

Machine: abc_star.csv

Input string: aaabbbccccc

Output:

Machine name: a*b*c* Nondeterministic

Initial string: aaabbbccccc

Tree depth: 13
Transitions simulated: 51
String accepted in: 13
Path taken:
['__', 'q0', 'aaabbbccccc__']
['__a', 'q0', 'aabbccccc__']
['__aa', 'q0', 'abbbccccc__']
['__aaa', 'q0', 'bbbccccc__']
['__aaab', 'q1', 'bbccccc__']
['__aaabb', 'q1', 'bccccc__']
['__aaabbb', 'q2', 'ccccc__']
['__aaabbbc', 'q2', 'cccc__']
['__aaabbbcc', 'q2', 'ccc__']
['__aaabbbccc', 'q2', 'cc__']
['__aaabbbccccc', 'q2', 'c__']
['__aaabbbccccc', 'q3', '___']
['__aaabbbccccc_', 'qacc', '___']
Degree of non-determinism: 1.185897435897436

Input string: aaabbbcccccbc

Output:

Machine name: $a^*b^*c^*$ Nondeterministic
Initial string: aaabbbcccccbc
Tree depth: 12
Transitions simulated: 50
String rejected in: 12
Degree of non-determinism: 1.2638888888888888

Machine: equal_01s_DTM.csv

Input: 01010101010101010101010101010001110

Output:

Machine name: $\{w \mid w \text{ has the same number of 0's and 1's}\}$ Deterministic
Initial string: 01010101010101010101010101010001110
Tree depth: 552
Transitions simulated: 551
String rejected in: 552
Degree of non-determinism: 1.0

Input: 01010011

Output:

Machine name: {w | w has the same number of 0's and 1's} Deterministic

Initial string: 01010011

Tree depth: 44

Transitions simulated: 43

String accepted in: 44

Path taken:

['__', 'q0', '01010011__']
['___', 'q1', '1010011__']
['___', 'q3', ' _x010011__']
['___', 'q4', 'x010011__']
['___x', 'q4', '010011__']
['___xx', 'q1', '10011__']
['___x', 'q3', 'xx0011__']
['___', 'q3', 'xxx0011__']
['___', 'q3', ' _xxx0011__']
['___', 'q4', 'xxx0011__']
['___x', 'q4', 'xx0011__']
['___xx', 'q4', 'x0011__']
['___xxx', 'q4', '0011__']
['___xxxx', 'q1', '011__']
['___xxxx0', 'q1', '11__']
['___xxxx', 'q3', '0x1__']
['___xxx', 'q3', 'x0x1__']
['___xx', 'q3', 'xx0x1__']
['___x', 'q3', 'xxx0x1__']
['___', 'q3', 'xxxx0x1__']
['___', 'q3', ' _xxxx0x1__']
['___', 'q4', 'xxxx0x1__']
['___x', 'q4', 'xxx0x1__']
['___xx', 'q4', 'xx0x1__']
['___xxx', 'q4', 'x0x1__']
['___xxxx', 'q4', '0x1__']
['___xxxxx', 'q1', 'x1__']
['___xxxxxx', 'q1', '1__']
['___xxxxx', 'q3', 'xx__']
['___xxxx', 'q3', 'xxx__']


```

['___xxx', 'q3', 'xxxx___']
['___xx', 'q3', 'xxxxx___']
['___x', 'q3', 'xxxxxx___']
['___', 'q3', 'xxxxxxx___']
['___', 'q3', '_xxxxxxx___']
['___', 'q4', 'xxxxxxx___']
['___x', 'q4', 'xxxxxx___']
['___xx', 'q4', 'xxxxx___']
['___xxx', 'q4', 'xxxx___']
['___xxxx', 'q4', 'xxx___']
['___xxxxx', 'q4', 'xx___']
['___xxxxxx', 'q4', 'x___']
['___xxxxxxx', 'q4', '___']
['___xxxxxxx_', 'qacc', '___']

```

Degree of non-determinism: 1.0

Machine name: equal_01s.csv

Input string: 01010101010101010101010101010011

Output:

Machine name: {w | w has the same number of 0's and 1's} Nondeterministic

Initial string: 01010101010101010101010101010011

Tree depth: 484

Transitions simulated: 723

String accepted in: 484

Path taken:

```

['___', 'q0', '01010101010101010101010101010011___']
['___', 'q1', '101010101010101010101010101010011___']
['___', 'q3', '_x01010101010101010101010101010011___']
['___', 'q4', 'x01010101010101010101010101010011___']
['___x', 'q4', '01010101010101010101010101010011___']
['___xx', 'q1', '101010101010101010101010101010011___']
['___x', 'q3', 'xx01010101010101010101010101010011___']
['___', 'q3', 'xxx01010101010101010101010101010011___']
['___', 'q3', '_xxx01010101010101010101010101010011___']
['___', 'q4', 'xxx01010101010101010101010101010011___']
['___x', 'q4', 'xx01010101010101010101010101010011___']
['___xx', 'q4', 'x01010101010101010101010101010011___']
['___xxx', 'q4', '01010101010101010101010101010011___']

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Degree of non-determinism: 1.0165289256198347

Output:

Initial string: 01010101010101010

Tree depth: 146

Transitions simulated: 217

String rejected in: 146

Degree of non-determinism: 1.0273972602739727