Project 1 Report

Author: Sadia Afreen

Abstract: The project demonstrates implementation of finite automata and evaluations. In this project, non-deterministic finite automata (NFA), the conversion of non-deterministic finite automata to deterministic finite automata (DFA), minimization of deterministic finite automata, and parallelization of deterministic finite automata should be implemented and processed with given inputs. The project should also evaluate each model's size, runtime and their comparison against the change of input size. However, the author could only implement NFA, and a standalone DFA but not the conversion from NFA to DFA. The evaluation for benchmark files for the NFA is also demonstrated by the author in this report.

Implementation:

The project is implemented for NFA and DFA using Java. However, the DFA cannot process epsilon inputs so evaluations for benchmark files were not performed for this model. The NFA model successfully handles the epsilon and non-epsilon input strings and processed the test inputs given in the test folder. It performed successfully for the test inputs. However, it fell short while processing 'bm1' file. Certain limitations in the NFA implementation led it to fail processing for eval_1, eval_2, eval_3 inputs for 'bm1' benchmark NFA. It did process eval_4 input for 'bm1' and all the other inputs for 'bm2' and 'bm3'.

Experimentation Results:

Table 1: Empirical evaluation for 'bm1' benchmark file

Model	Size	eval_1	eval_2	eval_3	eval_4
			bm1		
NFA	|Q| = 6, |δ| = 10				234.67 ms

Table 2: Empirical evaluation for 'bm2' benchmark file

Model	Size	eval_1	eval_2	eval_3	eval_4
			bm2		
NFA	|Q| = 8, |δ| = 14	15.67 ms	34 ms	106.67 ms	178.33 ms

Table 3: Empirical evaluation for 'bm3' benchmark file

Model	Size	eval_1	eval_2	eval_3	eval_4
			bm3		
NFA	|Q| = 9, |δ| = 20	16 ms	21 ms	74.33 ms	250.67 ms

Table 1, 2, 3 shows the empirical evaluation for the implemented NFA for benchmark 1,2,3 respectively. For each table the first column indicates the model, second column indicates size of the model which include the number of states (|Q|) and the number of transitions (|δ|), and rest of the columns are the runtime for input files eval_1, eval_2, eval_3, eval_4. It can be noted that the runtimes are counted in milliseconds unit using *System.currentTimeMillis()* function of Java.

Discussion:

It can be observed from Table 1,2,3 that as the input files are concurrently increased in size, the runtime for each file has also increased. Also, Benchmark 3 NFA model has the greatest number of states, and the greatest number of transitions among the three models. The runtime for benchmark 3's last input file which also has the greatest size among the input file is the largest, namely almost 250.67 ms. The eval_1 for 'bm2' and 'bm3' has almost the same runtime and they increased as the input size were also increased. So, it can be concluded that with increase of model size and input size, runtime is also increasing.

Limitations:

The project could have observed more evolved evaluations if it could compare the runtime of NFA with the DFA, DFA minimized, and parallel DFA.