

Programming Assignment 1 (Part 2)

Automata Theory

Shikhar Saxena (2021121010)

q3.cpp

Compile using `g++ q3.cpp` and then run using `./a.out` on the terminal.

Input is assumed to be valid (all the constraints mentioned for the DFA) as well as the input format should be valid.

Code explanation

I have created a class for DFA which has data members namely, `n`, `k`, `a`, `accept_states`, `sigma` and `transitions`. Here, `n`, `k` and `a` have their usual meaning as in the input. `accept_states` is a set of the given accept states for the DFA. `sigma` stores the input alphabets used in the DFA. And `transitions` are a map from state, input-symbol pair to next state.

Firstly, the program takes input from the user. After taking input we call the `minimize` function that starts with the minimizing DFA procedure.

Firstly, we insert the unreachable states (not reachable from the start state) into a set `not_reachable_states`.

After this, I have used Myhill-Nerode Theorem (table-filling approach) to mark the pairs of states which are distinguishable from each other. The pairs of states left unmarked are equivalent to each other and hence not-distinguishable (and can be merged).

After this we create a vector `partitions` that stores sets of non-distinguishable states.

In my approach, all the unique states will not be included in this vector.

For example: if 3 is a unique state then it will not be there in the partitions (as a singleton set).

This is resolved in the next step, where I've given a partition-id in `_partition` to each state (in the original DFA). If two or more states are equivalent to each other then they essentially have the same partition number whereas for unique states the id is incremented.

Therefore, the final id is the total number of states n' in the minimized DFA.

After this we construct the minimized DFA, using the original DFA data-members, the set of not reachable states (not included in the final DFA) and the `partition_id` for each state (since the map is a collection of unique key to value pair. By standard library, the duplicate transitions get removed).

After this we display the minimized DFA and the program exits.

q4.cpp (INCOMPLETE)

Compile using `g++ q4.cpp` and the run using `./a.out <input_file> <output_file>`.

The file prints error messages if the input format (passing arguments through command line) is not valid.

The program internally uses IO redirection to redirect input from `<input_file>` and print directly to the given `<output_file>`. Output file is created (if it does not exist) else it is truncated for every test case run.

After taking input, program inserts explicit concatenation symbol `.` wherever is necessary. Now, it converts this regex into its postfix expression.

Converting the postfix expression to nfa was incomplete and hence has been commented out in the file.