COMPSCI/SFWRENG 2FA3
Discrete Mathematics with Applications II
Winter 2021

Extra Credit Assignment 3

Dr. William M. Farmer and Dr. Mehrnoosh Askarpour
McMaster University

Revised: March 6, 2021

Extra Credit Assignment 3 consists of three problems concerning Kleene algebas. You must write your solution to the problem using LaTeX.

Please submit Extra Credit Assignment 3 as two files, `EC_Assignment_3_`*`YourMacID`*`.tex` and `EC_Assignment_3_`*`YourMacID`*`.pdf`, to the Extra Credit Assignment 3 folder on Avenue under Assessments/Assignments. *`YourMacID`* must be your personal MacID (written without capitalization). The `EC_Assignment_3_`*`YourMacID`*`.tex` file is a copy of the LaTeX source file for this assignment (`EC_Assignment_3.tex` found on Avenue under Contents/Assignments) with your solution entered after the problem. The `EC_Assignment_3_`*`YourMacID`*`.pdf` is the PDF output produced by executing

```
pdflatex EC_Assignment_3_YourMacID
```

This assignment is due **Sunday, March 14, 2021 before midnight.** You are allow to submit the assignment multiple times, but only the last submission will be marked. **Late submissions and files that are not named exactly as specified above will not be accepted!** It is suggested that you submit your preliminary `EC_Assignment_3_`*`YourMacID`*`.tex` and `EC_Assignment_3_`*`YourMacID`*`.pdf` files well before the deadline so that your mark is not zero if, e.g., your computer fails at 11:50 PM on March 14.

Although you are allowed to receive help from the instructional staff and other students, your submission must be your own work. Copying will be treated as academic dishonesty! If any of the ideas used in your submission were obtained from other students or sources outside of the lectures and tutorials, you must acknowledge where or from whom these ideas were obtained.

Background

1. A two-tape deterministic finite automata (2-DFA) is a deterministic finite state automata that instead of only one input tape, has two input tapes and accepts pairs of strings. Its set of states has two subsets. For example, 2-DFA A below accepts $L(A) = \{(w_1, w_2) \in \{a, b\}^* \times \{a, b\}^* | \, |w_1| = 2 \cdot |w_2|\}$ which is a languages of pairs of strings where the first one is two times longer than the second one. for example, then ("*bab*", "*abbaba*") is an acceptable pair in $L(A)$.

   Formally, a 2-DFA is defined with a tuple $M = (Q, \Sigma, \sigma, s, F)$ where: $Q = Q_1 \cup Q_2$ and $Q = Q_1 \bigcap Q_2 = \emptyset$.
   Depending on the current state, the input is read from on of the input tapes; if the current state is q and $q \in Q_1$ then a symbol is read from tape 1, otherwise if $q \in Q_2$ then a symbol is read from tape 2.

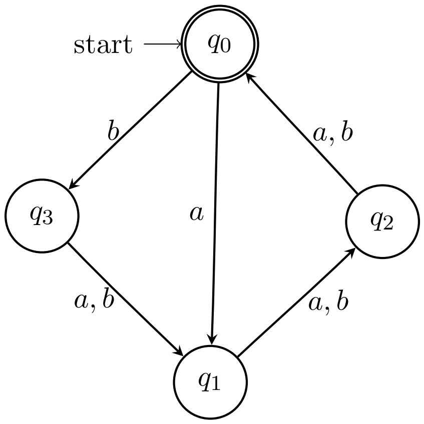
Extra Credit Problem [2 bonus points]

1. Define a 2-DFA for the following language: all pairs of strings (w_1, w_2) in $\{a, b\}^* \times \{a, b\}^*$ such that the length of w_2 is twice the number of a's in w_1 plus three times the number of b's in w_1.

2. Compare the computational power of DFA and 2-DFA, and explain why one might be able to accept languages that the other one cannot? Justify your answer.

Mohammad Omar Zahir, zahirm1, March 21, 2021

1. The constructed 2-DFA is below. It is to be noted that the single state on the top, q_0 is a part of Q_1, and the three states on the bottom, right, and left, q_1, q_2, and q_3 make up Q_2, respectively. We can see that when we accept a b from the first tape, we must accept any three more letters from the second tape. Similarly, when we accept an a from the first tape, we must accept at least two more letters from the second tape before the word can be accepted.



2. The advantage that the 2DFA has over the regular DFA is the fact that it allows for the reading of two separate input tapes, meaning that the 2DFA allows for the possibility of making a word from one input tape to conform and adapt to the other. This is starkly different from that of a regular DFA where we can only accept a single word from a single input tape. With the 2DFA, as we can see in the example above, we can clearly base the acceptance of one word on the other. The acceptance of the amount of letters in the second word was designed directly based on the types of letters accepted by the first word. While it is important to note that the addition of a second input tape, and consecutively a 2DFA, does not necessarily have more computational power than a regular DFA, as they both have finite memory. That being said, the fact that the 2DFA accepts inputs from two separate input tapes means that we have an extra capability of creating machines that can interact based on multiple inputs, and would better model a more practical purpose in the real-world, where multiple areas of input can affect the single output that we obtain. A 2DFA, as a result, can be argued to have more computational power and the possibility of accepting everything that a regular DFA can, as well as the added benefit as described above. As stated, a 2DFA behaves a step more closer to a modern computer or embedded system that decides its final state or output by the inputs that it takes from multiple facets, which in our case are other input tapes in the 2DFA.