Alex Hubers
CSC:341 Spring 2024

HW #2

Due: February 06, 2022

Learning Outcomes Completion of this assignment will contribute to your ability to fulfill the following learning outcomes:
1-1: Represent a problem using a regular model of computation.

Instructions. You are allowed to collaborate with others, however you should write up solutions independently. Copying an answer from another source (e.g. the Web) or from another student may yield few or zero points. Write solutions neatly and legibly, or type your solutions in LaTeX. Be sure to number each problem, and indicate a final solution (if relevant). Answers to problems should include justification (show your work).

Acknowledgments. Problems from this homework come from published sources. The specific sources are withheld due to the nature of this assignment.

Academic Honesty. Include the following information at the top of your submission, along with your name.

- Written sources used: (Include textbook(s), complete citations for web or other written sources. Write *none* if no sources used)
- Help obtained: (Include names of anyone other than the instructor.)

Rubric. Problems will be graded on a 4-point *EMBN rubric*. Graded answers will be assigned an appropriate letter descriptor (below) and provided some justification of this assignment. Points will be awarded in the range associated with each descriptor.

- **E (Excellent)**. 4 pts. Complete understanding of the material is evident; exhibits no errors and can serve as an exemplar solution for the course.
- **M (Meets Expectation)**. $[3, 4)$ pts. Complete understanding of the material is evident, but exhibits some minor errors that warrant revision.
- **B (Below Expectation)**. $[1, 3)$ pts. Limited understanding of the material is evident; exhibits many minor errors or one or more major errors that necessitate revision.
- **N (Not Completed)**. 0 pts. Not completed to a degree where understanding is evident.

Pointed DFAs

Formally define a *pointed DFA* $D = (Q, \Sigma^*, q_0, \delta, F)$ as a DFA except that the type of the transition function is:

$$\delta : Q \times \Sigma \to Q \cup \{\perp\}$$

The symbol $\perp$ is read “bot” and is commonly used in programming language semantics to denote erroneous program runtime behavior, such as non-termination or stuck computation. You may think of a “pointed DFA” as having this “bottom” point $\perp$ that represents system failure. We assume that $\perp \notin Q$.

Intuitively, if a state-character pair transitions to $\perp$ then no additional input is consumed and the machine rejects the input string.

1. (4 pts.) Recall that M accepts w if there exists a sequence of states $r_0, r_1, \ldots r_n \in Q$ such that:

   - $r_0 = q_0$,
   - $\delta(r_i, w_{i+1}) = r_{i+1}$ for $i = 0, \ldots n - 1$, and
   - $r_n \in F$

   Does the definition of acceptance need to change for a DFA with $\perp$? If so, give a modified definition, explaining what changes you made in a sentence or two. If not, explain why in a sentence or two.

2. Show that a DFA with $\perp$ is equivalent to a DFA. That is to say:

   (a) (4 pts.) give a procedure by which a DFA D may be transformed into a pointed DFA $D_\perp$.

   (b) (4 pts.) Let A be a regular language accepted by D. Argue that $D_\perp$ accepts A as well.

   (c) (4 pts.) In the other direction, give a procedure that transforms a pointed DFA $D_\perp$ into a DFA D.

   (d) (4 pts.) Let A be a regular language accepted by $D_\perp$. Argue that D accepts A as well.