CS 321: Homework #3 solution

Due: Monday Oct 16 at 9am, on Canvas

4. Let $M = (Q, \Sigma, \delta, s, F)$ be a DFA.

(a) Show that for any $q \in Q$ and $P \subseteq Q$, the following language is regular:

$$\{w \in \Sigma^* \mid \delta^*(q, w) \in P\}$$

Clearly describe a procedure to construct a DFA for this language, in terms of M.

Sol. In this problem we are given $M = (Q, \Sigma, \delta, s, F)$ and $q \in Q$ and $P \subseteq Q$. Just make a new DFA with the same states & transitions as M, but with q as its start state and P as its accept states. More precisely, the new DFA is $M' = (Q, \Sigma, \delta, q, P)$. The definition of M' accepting a string w is literally that

$$\delta^*(\underset{\substack{\uparrow \\ \text{start state of } M'}}{q}, w) \in \underset{\substack{\uparrow \\ \text{accept states of } M'}}{P}$$

which is exactly the condition we want to check.

(b) If A is a language over alphabet Σ, define:

$$\mathsf{undouble}(A) \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid ww \in A\}.$$

Show that if A is regular, then so is $\mathsf{undouble}(A)$.

Example: if $A = \{\varepsilon, 0, 11, 0010, 0101\}$ then $\mathsf{undouble}(A) = \{\varepsilon, 1, 01\}$.

Hint: First consider a simpler version where I fix the "middle" state q, so:

$$\{w \in \Sigma^* \mid ww \in A \text{ and } \delta^*(s, w) = q\}$$

In the "real" version of the problem, the "middle state" is not fixed.

Sol. The trick here is for a DFA/NFA to check whether $ww \in A$, given only w as input. Let $M = (Q, \Sigma, \delta, s, F)$ be a DFA that recognizes A. The idea is to cleverly split up M's computation on ww into a first half and second half, which can both be checked "in parallel."

Imagine running M on input ww. After reading just w, the DFA is in some state $\delta^*(s, w)$. Let's call that state the half-way point. Then,

$$[ww \in A \text{ with } q \text{ as half-way point}] \Leftrightarrow [\delta^*(s, w) = q \text{ and } \delta^*(\underline{q}, w) \in F]$$

This just says that reading w from the start state gets you to the half-way point q, and reading w from there gets you to an accept state. The purpose of splitting things up in this way is that the expression on the right-hand-side is the logical-and of two things, both of which can be checked by a DFA that is just given w as input (from the previous part).

Given a totally arbitrary string w, *any* $q \in Q$ can potentially be the half-way point when M is given input ww. Let's write $Q = \{q_1, \ldots, q_n\}$. Then:

$$w \in \mathsf{undouble}(A) \Leftrightarrow ww \in A$$

$$\begin{aligned}
&\Leftrightarrow [ww \in A \text{ with } q_1 \text{ as the half-way point}] \\
&\qquad \text{or } [ww \in A \text{ with } q_2 \text{ as the half-way point}] \\
&\qquad \vdots \\
&\qquad \text{or } [ww \in A \text{ with } q_n \text{ as the half-way point}] \\
&\Leftrightarrow [\delta^*(s,w) = q_1 \text{ and } \delta^*(q_1,w) \in F] \\
&\qquad \text{or } [\delta^*(s,w) = q_2 \text{ and } \delta^*(q_2,w) \in F] \\
&\qquad \vdots \\
&\qquad \text{or } [\delta^*(s,w) = q_n \text{ and } \delta^*(q_n,w) \in F] \\
&\Leftrightarrow w \in \left(\begin{array}{ll}
 & \Big(\{x \mid \delta^*(s,x) = q_1\} \cap \{x \mid \delta^*(q_1,x) \in F\}\Big) \\
\cup & \Big(\{x \mid \delta^*(s,x) = q_2\} \cap \{x \mid \delta^*(q_2,x) \in F\}\Big) \\
\vdots & \\
\cup & \Big(\{x \mid \delta^*(s,x) = q_n\} \cap \{x \mid \delta^*(q_n,x) \in F\}\Big)
\end{array} \right)
\end{aligned}$$

This chain of if-and-only-ifs shows that the final expression is just a different way of writing $\mathsf{undouble}(A)$. Importantly, we have written it as a union of intersections of *regular languages*, so we know the result is regular.

If you find this construction a little abstract, you might wonder what the NFA/DFA looks like when you actually unroll all these constructions. Here is a way to think of the problem in terms of 3 pebbles. First, an informal description of how an NFA can solve this problem:

- Imagine a drawing of the DFA $M = (Q, \Sigma, \delta, s, F)$.
- Before doing anything, place a red pebble on state s, guess a state $q \in Q$ and put a green and blue pebble on state q. This guess is the only non-deterministic step.
- Whenever you read a character c from input, advance the red pebble and the blue pebble according to character c. Always leave the green pebble where it is.
- You can accept any time (1) the red & green pebbles are on the same state, and (2) the blue pebble is on an accept state.

The green pebble is the "guess" of the half-way state, and it stays in place forever. The red pebble follows the path from the start state on input w. The blue pebble follows the path from the half-way state on input w. The NFA accepts if the red pebble's path indeed gets to the guess of the half-way state, and if the blue pebble's path indeed ends in an accepting state.

This construction could be made more formal as follows. The new NFA consists of:

- Set of states Q^3 plus a special start state s^*. Apart from the start state, each state is a triple (r, g, b) which gives the positions of the 3 pebbles.
- ε-transitions from the start state to every state of the form (s, q, q). This places the red pebble at the DFA's start state, and the other two pebbles in the same state.

- From state (r, g, b) on input c, a transition to state $(\delta(r, c), g, \delta(b, c))$, to move the red and blue pebbles, while keeping the green pebble in place.
- Accept states $\{(p, p, q) \mid p \in Q, q \in F\}$, to accept whenever the red and green pebbles are in the same place, and the blue pebble is in an accepting state of the DFA.