CS 374 **Spring 2015**

Homework 2

Nikhil Unni (nunni2)

2. First, to do this problem we must define a new δ function, which we'll call δ_2. It is to represent the state transitions for L^R, and it's defined as : $\delta_2(p, a) = \{q | \delta(q, a) = p, q \in Q\}$. Then our new NFA is defined as follows:

$$Q^N = Q \times Q \cup q_0^N$$
$$\Sigma^N = \Sigma$$
$$\delta^N(q_0^N, \epsilon) = \{(q_0, r), r \in F\}$$
$$\delta^N((p, q), a) = (\delta(p, a), \delta_2(q, a))$$
$$q_0^N$$
$$F^N = \{(p, q) | \delta(p, a) = q, \delta_2(q, a) = p, p \in Q, q \in Q \text{ for some } a \in \Sigma\}$$

Essentially all the NFA is doing is that it has several "starting" points, where it marks two indices: one is the index of q_0, and the other is an element from F, and the idea is that the two will walk towards each other.

To actually implement multiple possible starting points [1], I just have an artificial q_0^N that ϵ transitions to all of the starting points that I want.

The two walk towards each other, with the first index incrementing with the forwards δ, and the second index decrementing with the backward δ_2. Eventually they reach a point where they're one transition away from one another, at the last character of w. At this point, we see that the set of final states has this case included, and we can return successfully.

[1] https://piazza.com/class/i4mrvddxr0h3sd?cid=265