Closure and Equivalence Properties

These first two weeks of content have largely had proofs concerning the equivalence of models of computation or the closure of operations over regular languages. Please fill in the blanks:

1. The language A is regular if...
2. Two regular models of computation are equivalent if...
3. **If A is regular and $f(A)$ is an operation on A, then f is a *regular operator*...**

(drumroll please:)

1. ... We exhibit a DFA (or equivalent model of computation) that recognizes A.
2. ... We exhibit two dual constructions that translate each model of computation to one another and show that these constructions preserve regularity.
3. **... We show that $f(a)$ is regular. We also say that f *preserves regularity* or that f is *closed. N.b. f here is unary, but this idea extends to binary operators such as $\cup$.***

By now you should be becoming more comfortable recognizing these different styles of claim and their corresponding proof techniques. For today's lecture, I want to give you more practice/familiarity with *closure* proofs. In particular, we will complete our proof that each regular expression corresponds to an NFA by showing the NFAs we construct when a regular expression is the union of two subexpressions; the concatenation of two subexpressions; or the Kleene closure (star) of a subexpression.

Recall the following from Lecture 3A.

Claim 1. *If a language is described by a regular expression, then it is regular.*

Proof. Let A be a language described by the regular expression R. Proceed by case analysis on R.

Case ($R = a$). Suppose $R = a$ for some letter $a \in \Sigma$. Then $L(R) = \{a\}$, which can be recognized by... (See lecture 3A).

Case ($R = \epsilon$). Suppose $R = \epsilon$. Then $L(R) = \epsilon$ and see Lecture 3A.

Case ($R = \emptyset$). Suppose $R = \emptyset$. Then $L(R) = \emptyset$ and see Lecture 3A.

Case ($R = R_1 \cup R_2$). By the IH, R_1 and R_2 are regular. As union is a regular operator, it follows immediately that $R_1 \cup R_2$ is regular.

Case ($R = R_1 \circ R_2$). Ditto.

Case ($R = R_1^*$). Ditto. □

We complete our proof by showing that union, concatenation, and star are regular operators.

Regularity of Union

For this and the following proofs, suppose A_1 and A_2 are regular languages. In particular, suppose they are recognized by N_1 and N_2, respectively. Decompose N_1 and N_2 into the following.

$$N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$$
$$N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$$

For simplicity, presume $Q_1 \cap Q_2 = \emptyset$; if the two do have any overlap in states, we can simply rename one.

Claim 2. *The class of regular languages is closed under the union operator.*

Proof. Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \cup A_2$.

1. $Q = \{q_s\} \cup Q_1 \cup Q_2$. (Presume $q_s \notin (Q_1 \cup Q_2)$, otherwise rename it).
2. The start state is q_s.
3. The set of accept states is $F_1 \cup F_2$.
4. Define δ according to the following equations.

$$\delta(q_s, \epsilon) = \{q_1, q_2\} \tag{1}$$
$$\delta(q, a) = \delta_1(q, a) \qquad \text{if } q \in Q_1 \tag{2}$$
$$\delta(q, a) = \delta_2(q, a) \qquad \text{if } q \in Q_2 \tag{3}$$

Presume inputs left unspecified map to $\emptyset$ (for example, $\delta(q_s, a) = \emptyset$ for all $a \neq \epsilon$).

See Figure 1.46 in your text for a picture.

We next need to prove that if x is in the language $A_1 \cup A_2$ that x is accepted by N. In other words, we need to argue that N does what it says it does. (Sipser omits this step). I will give this explanation once; it is similarly structured for the concatenation and star cases, and thus omitted.

To complete our proof, suppose x is in the language $A_1 \cup A_2$. Then, by definition, x is an element of A_1, A_2, or both. WLOG, assume $x \in A_1$. Then x is accepted by N_1 along some trace $q_1 r_2..r_n$ for some $r_n \in F_1$. But this trace exists in N, prefixed by q_0. Further, we know $r_n \in F$. Thus x will be recognized by N along the trace $q_0 q_1 r_2..r_n$. □

Regularity of Concatenation

Claim 3. *The class of regular languages is closed under concatenation.*

Proof. See figure 1.48 for a picture. The intuition is that we glue the ends of N_1 onto the start of N_2. More formally, suppose that N_1 and N_2 are as described previously and construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$.
2. The start state is q_1.
3. The accept states are F_2.
4. Define δ according to the following equations:

$$\delta(q, \epsilon) = \delta_1(q, \epsilon) \cup \{q_2\} \qquad \text{if } q \in F_1 \tag{4}$$
$$\delta(q, a) = \delta_2(q, a) \qquad \text{if } q \in Q_2 \tag{5}$$
$$\delta(q, a) = \delta_1(q, a) \qquad \text{if } q \in Q_1 \tag{6}$$

□

Regularity of Star

Claim 4. *The class of regular languages is closed under star.*

Proof. See figure 1.49 for a picture. The intuition is that we (i) prefix N_1 with a new accept state q_s that accepts the empty string and (ii) add a loop back from all accept states back to the accept state to make N_2 match *more than 1 repitition* of the pattern. More formally, suppose that N_1 is as described previously and construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize A_1^*.

1. $Q = Q_1 \cup \{q_s\}$.
2. The start state is q_s.
3. The accept states are $F_1 \cup \{q_s\}$.
4. Define δ identically to δ_1 except on the following specific inputs:

$$\delta(q_s, \epsilon) = \{q_1\} \tag{7}$$
$$\delta(q_f, \epsilon) = \delta_1(q_f, \epsilon) \cup \{q_1\} \qquad \text{if } q_f \in F_1 \tag{8}$$

□

Tying things up

Go back to our claim that regular expressions correspond to NFAs and ask why these proofs are sufficient to complete the claim.