## Week2 friday

**Warmup**: Design a DFA (deterministic finite automaton) and an NFA (nondeterministic finite automaton) that each recognize each of the following languages over  $\{a, b\}$ 

 $\{w \mid w \text{ has an } a \text{ and ends in } b\}$ 

 $\{w \mid w \text{ has an } a \text{ or ends in } b\}$ 

**Strategy**: To design DFA or NFA for a given language, identify patterns that can be built up as we process strings and create states for intermediate stages. Or: decompose the language to a simpler one that we already know how to recognize with a DFA or NFA.

Recall (from Wednesday of last week, and in textbook Exercise 1.14): if there is a DFA M such that L(M) = A then there is another DFA, let's call it M', such that  $L(M') = \overline{A}$ , the complement of A, defined as  $\{w \in \Sigma^* \mid w \notin A\}$ .

Let's practice defining automata constructions by coming up with other ways to get new automata from old.

Suppose  $A_1, A_2$  are languages over an alphabet  $\Sigma$ . Claim: if there is a NFA  $N_1$  such that  $L(N_1) = A_1$  and NFA  $N_2$  such that  $L(N_2) = A_2$ , then there is another NFA, let's call it N, such that  $L(N) = A_1 \cup A_2$ .

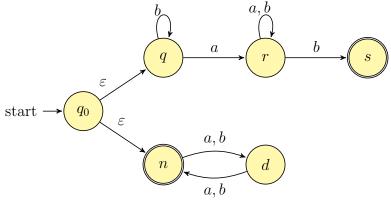
**Proof idea**: Use nondeterminism to choose which of  $N_1$ ,  $N_2$  to run.

Formal construction: Let  $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$  and  $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$  and assume  $Q_1 \cap Q_2 = \emptyset$  and that  $q_0 \notin Q_1 \cup Q_2$ . Construct  $N = (Q, \Sigma, \delta, q_0, F_1 \cup F_2)$  where

- Q =
- $\delta: Q \times \Sigma_{\varepsilon} \to \mathcal{P}(Q)$  is defined by, for  $q \in Q$  and  $x \in \Sigma_{\varepsilon}$ :

Proof of correctness would prove that  $L(N) = A_1 \cup A_2$  by considering an arbitrary string accepted by N, tracing an accepting computation of N on it, and using that trace to prove the string is in at least one of  $A_1$ ,  $A_2$ ; then, taking an arbitrary string in  $A_1 \cup A_2$  and proving that it is accepted by N. Details left for extra practice.

**Example:** The language recognized by the NFA over  $\{a, b\}$  with state diagram



is:

Could we do the same construction with DFA?

Happily, though, an analogous claim is true!

Suppose  $A_1$ ,  $A_2$  are languages over an alphabet  $\Sigma$ . Claim: if there is a DFA  $M_1$  such that  $L(M_1) = A_1$  and DFA  $M_2$  such that  $L(M_2) = A_2$ , then there is another DFA, let's call it M, such that  $L(M) = A_1 \cup A_2$ . Theorem 1.25 in Sipser, page 45

## Proof idea:

Formal construction:

**Example**: When  $A_1 = \{w \mid w \text{ has an } a \text{ and ends in } b\}$  and  $A_2 = \{w \mid w \text{ is of even length}\}.$ 



Suppose  $A_1, A_2$  are languages over an alphabet  $\Sigma$ . Claim: if there is a DFA  $M_1$  such that  $L(M_1) = A_1$  and DFA  $M_2$  such that  $L(M_2) = A_2$ , then there is another DFA, let's call it M, such that  $L(M) = A_1 \cap A_2$ . Sipser Theorem 1.25, page 45

Proof idea:

Formal construction: