Lecture #3: DFA Design and Verification — Part One
Another Example

Problem To Be Solved

Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Our goal will be to design a deterministic finite automaton for the language

$$L = \{\omega \in \Sigma^\star \mid \omega \text{ includes at least one “a”}\}.$$

What Needs To Be Remembered?

To begin we should try to decide what information, about the part of the string that has been processed so far, must be kept track of — that is, ***what the DFA must remember***.

For this problem, let us start by choosing the minimal amount of information that seems to be necessary and relevant: We will try to design a DFA that (only) remembers ***whether the string, that has been seen so far, includes at least one “***$\mathtt{a}$***”***.

Identification of Subsets of $\Sigma^\star$

This suggests that two subsets of $\Sigma^\star$ should be considered, namely, the sets

$$S_{\text{no}} = \{\omega \in \Sigma^\star \mid \omega \text{ does not include an “a”}\}$$

and

$$S_{\text{yes}} = \{\omega \in \Sigma^\star \mid \omega \text{ includes at least one “a”}\}.$$

We will, therefore, try to design a deterministic finite automaton with two states — so that

$$Q = \{q_{\text{no}}, q_{\text{yes}}\}$$

— where the state q_{no} corresponds to the subset S_{no} (in a way that will be described more completely, later on) and where the state q_{yes} corresponds to the subset S_{yes}.

Initial "Sanity Checks"

The following conditions are generally satisfied, but they should still be checked.

1. ***Sanity Check #1:*** Have only a *finite* number of subsets of $\Sigma^\star$ been identified?

   This condition is satisfied, for this example, because only ***two*** subsets of $\Sigma^\star$, S_{yes} and S_{no}, have been identified.

2. ***Sanity Check #2:*** Is it true that every string in $\Sigma^\star$ belongs to ***exactly one*** of the subsets of $\Sigma^\star$ that have been described?

   For this example, the condition is satisfied if and only if every string in $\Sigma^\star$ belongs to at least one of S_{no} or S_{yes} — so that

   $$S_{\text{no}} \cup S_{\text{yes}} = \Sigma^\star$$

   — *and* no string in $\Sigma^\star$ belongs to *both* of S_{no} and S_{yes} — that is,

   $$S_{\text{no}} \cap S_{\text{yes}} = \emptyset.$$

   Note: $\lambda \in S_{\text{no}}$ so our DFA's start state should be the state, q_{no}, that corresponds to the subset S_{no}. The correspondence between states in the DFA and subsets of $\Sigma^\star$, that we wish to establish can now be described as follows: For every string $\omega \in \Sigma^\star$,

   $$\omega \in S_{\text{yes}} \text{ if and only if } \delta^\star(q_{\text{no}}, \omega) = q_{\text{yes}}$$

   and

   $$\omega \in S_{\text{no}} \text{ if and only if } \delta^\star(q_{\text{no}}, \omega) = q_{\text{no}}.$$

3. ***Sanity Check #3:*** Are "accepting states" well defined? Is it true that either $S \subseteq L$ or $S \cap L = \emptyset$ for every subset S of $\Sigma^\star$ that corresponds to a state?

   This condition is satisfied for this example, since $S_{\text{yes}} = L$ (so that $S_{\text{yes}} \subseteq L$) and since $S_{\text{no}} = \Sigma^\star \setminus L$ (so that $S_{\text{no}} \cap L = \emptyset$).

   Since $S_{\text{yes}} \subseteq L$ S_{yes} is an accepting state. Since $S_{\text{no}} \cap L = \emptyset$, S_{no} is *not* an accepting state.

Checking that Transitions will be Well-Defined

Let us consider the state q_{yes}, and consider a string ω such that $\omega \in S_{\text{yes}}$ — so that ω includes at least one "a".

- If $\tau \in \Sigma$ then the string $\omega \cdot \tau$ certainly includes at least one "a", since ω does. That is $\omega \cdot \tau \in S_{\text{yes}}$. Since ω was arbitrarily chosen from S_{yes}, it follows that

  $$\{\omega \cdot \tau \mid \omega \in S_{\text{yes}}\} \subseteq S_{\text{yes}}$$

  for every symbol $\tau \in \Sigma$. In particular,

  $$\{\omega \cdot \mathtt{a} \mid \omega \in S_{\text{yes}}\} \subseteq S_{\text{yes}}, \tag{1}$$

  $$\{\omega \cdot \mathtt{b} \mid \omega \in S_{\text{yes}}\} \subseteq S_{\text{yes}}, \tag{2}$$

  and

  $$\{\omega \cdot \mathtt{c} \mid \omega \in S_{\text{yes}}\} \subseteq S_{\text{yes}}. \tag{3}$$

  Thus the transitions out of state q_{yes} are well-defined. In particular, it follows by the equation at line (1) that $\delta(q_{\text{yes}}, \mathtt{a}) = q_{\text{yes}}$. It follows by the equation at line (2) that $\delta(q_{\text{yes}}, \mathtt{b}) = q_{\text{yes}}$, and it follows by the equation at line (3) that $\delta(q_{\text{yes}}, \mathtt{c}) = q_{\text{yes}}$.

Now let us consider the state q_{no}, and consider a string ω such that $\omega \in S_{\text{no}}$ — so that ω does not include an "a".

- The string $\omega \cdot \mathtt{a}$ certainly does include an "a" (since it ends with this symbol); that is, $\omega \cdot \mathtt{a} \in S_{\text{yes}}$. Since ω was arbitrarily chosen from S_{no} it follows that

  $$\{\omega \cdot \mathtt{a} \mid \omega \in S_{\text{no}}\} \subseteq S_{\text{yes}}$$

  and the transition out of q_{no} for the symbol $\mathtt{a}$ is well-defined. In particular,

  $$\delta(q_{\text{no}}, \mathtt{a}) = q_{\text{yes}}.$$

- Since ω does not include an "a" the string $\omega \cdot \mathtt{b}$ cannot include an "a" either; that is, $\omega \cdot \mathtt{b} \in S_{\text{no}}$. Since ω was arbitrarily chosen from S_{no} it follows that

  $$\{\omega \cdot \mathtt{b} \mid \omega \in S_{\text{no}}\} \subseteq S_{\text{no}}$$

  and the transition out of q_{no} for the symbol $\mathtt{b}$ is well-defined. In particular,

  $$\delta(q_{\text{no}}, \mathtt{b}) = q_{\text{no}}.$$

- Similarly, since ω does not include an "a" the string $\omega \cdot \mathtt{c}$ cannot include an "a" either; that is, $\omega \cdot \mathtt{c} \in S_{\text{no}}$. Since ω was arbitrarily chosen from S_{no} it follows that

  $$\{\omega \cdot \mathtt{c} \mid \omega \in S_{\text{no}}\} \subseteq S_{\text{no}}$$

  and the transition out of q_{no} for the symbol $\mathtt{c}$ is well-defined. In particular,

  $$\delta(q_{\text{no}}, \mathtt{c}) = q_{\text{no}}.$$

Thus all transitions out of state q_{no} are well-defined.

The DFA That Has Been Produced

A deterministic finite automaton $M = (Q, \Sigma, \delta, q_{\text{no}}, F)$ has now been developed such that $Q = \{q_{\text{yes}}, q_{\text{no}}\}$, q_{no} is the start state, $F = \{q_{\text{yes}}\}$, and the transitions for M are as shown below.