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 "a"***.

Identification of Subsets of $\Sigma^\star$

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

$$S_0 = \{\omega \in \Sigma^\star \mid \omega \text{ does not include an "a"}\}$$

and

$$S_1 = \{\omega \in \Sigma^\star \mid \omega \text{ includes at least one "a"}\}.$$

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

$$Q = \{q_0, q_1\}$$

— where the state q_0 corresponds to the subset S_0 (in a way that will be described more completely, later on) and where the state q_1 corresponds to the subset S_1.

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_0 and S_1, 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 because S_1 has been defined to be the set of all strings in $\Sigma^\star$ that *do* satisfy a condition — that of *including at least one* "a" — and S_0 has been defined to be the set of strings in $\Sigma^\star$ that *do not* satisfy the same condition. It follows from this that every string in $\Sigma^\star$ must belong to *exactly one* of the sets S_0 and S_1.

   Note: $\lambda \in S_0$ so our DFA's start state should be the state, q_0, that corresponds to the subset S_0. 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 $\omega \in \Sigma^\star$,

   $$\omega \in S_1 \text{ if and only if } \delta^\star(q_0, \omega) = q_1$$

   and

   $$\omega \in S_0 \text{ if and only if } \delta^\star(q_0, \omega) = q_0.$$

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 $S_1 = L$ (so that $S_1 \subseteq L$) and since $S_0 = \Sigma^\star \setminus L$ (so that $S_0 \cap L = \emptyset$).

   Since $S_1 \subseteq L$ q_1 is an accepting state. Since $S_0 \cap L = \emptyset$, q_0 is *not* an accepting state. Thus $F = \{q_1\}$.

Checking that Transitions will be Well-Defined

Let us consider the state q_1, and consider a string ω such that $\omega \in S_1$ — 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 $\omega \cdot \tau \in S_1$. Since ω was arbitrarily chosen from S_1, it follows that

  $$\{\omega \cdot \tau \mid \omega \in S_1\} \subseteq S_1$$

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

$$\{\omega \cdot \mathtt{a} \mid \omega \in S_1\} \subseteq S_1, \tag{1}$$

$$\{\omega \cdot \mathtt{b} \mid \omega \in S_1\} \subseteq S_1, \tag{2}$$

and

$$\{\omega \cdot \mathtt{c} \mid \omega \in S_1\} \subseteq S_1. \tag{3}$$

Thus the transitions out of state q_1 are well-defined. In particular, it follows by the equation at line (1) that $\delta(q_1, \mathtt{a}) = q_1$. It follows by the equation at line (2) that $\delta(q_1, \mathtt{b}) = q_1$, and it follows by the equation at line (3) that $\delta(q_1, \mathtt{c}) = q_1$.

Now let us consider the state q_0, and consider a string ω such that $\omega \in S_0$ — 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, $\omega \cdot \mathtt{a} \in S_1$. Since ω was arbitrarily chosen from S_0 it follows that

  $$\{\omega \cdot \mathtt{a} \mid \omega \in S_0\} \subseteq S_1$$

  and the transition out of q_0 for the symbol $\mathtt{a}$ is well-defined. In particular,

  $$\delta(q_0, \mathtt{a}) = q_1.$$

- Since ω does not include an "a" the string $\omega \cdot \mathtt{b}$ cannot include an "a" either; that is, $\omega \cdot \mathtt{b} \in S_0$. Since ω was arbitrarily chosen from S_0 it follows that

  $$\{\omega \cdot \mathtt{b} \mid \omega \in S_0\} \subseteq S_0$$

  and the transition out of q_0 for the symbol $\mathtt{b}$ is well-defined. In particular,

  $$\delta(q_0, \mathtt{b}) = q_0.$$

- Similarly, since ω does not include an "a" the string $\omega \cdot \mathtt{c}$ cannot include an "a" either; that is, $\omega \cdot \mathtt{c} \in S_0$. Since ω was arbitrarily chosen from S_0 it follows that

  $$\{\omega \cdot \mathtt{c} \mid \omega \in S_0\} \subseteq S_0$$

  and the transition out of q_0 for the symbol $\mathtt{c}$ is well-defined. In particular,

  $$\delta(q_0, \mathtt{c}) = q_0.$$

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

The DFA That Has Been Produced

A deterministic finite automaton $M = (Q, \Sigma, \delta, q_{\mathsf{no}}, F)$ has now been developed such that $Q = \{q_0, q_1\}$, q_0 is the start state, $F = \{q_1\}$, and the transitions for M are as shown below.