

# Lecture #2: DFA Design and Verification

## Another Example

### Problem To Be Solved

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

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

### Designing a DFA

#### 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 $\Sigma^*$

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

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

and

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

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  $\Sigma^*$  been identified?

This condition is satisfied, for this example, because only **two** subsets of  $\Sigma^*$ ,  $S_0$  and  $S_1$ , have been identified.

2. **Sanity Check #2:** Is it true that every string in  $\Sigma^*$  belongs to **exactly one** of the subsets of  $\Sigma^*$  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  $\Sigma^*$  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  $\Sigma^*$  that *do not* satisfy the same condition. It follows from this that every string in  $\Sigma^*$  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  $\Sigma^*$ , that we wish to establish can now be described as follows: For every string  $\omega \in \Sigma^*$ ,

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

and

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

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  $\Sigma^*$  that corresponds to a state?

This condition is satisfied for this example, since  $S_1 = L$  (so that  $S_1 \subseteq L$ ) and since  $S_0 = \Sigma^* \setminus L$  (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  $\omega$  such that  $\omega \in S_1$  — so that  $\omega$  includes at least one “a”.

- If  $\tau \in \Sigma$  then the string  $\omega \cdot \tau$  certainly includes at least one “a”, since  $\omega$  does. That is  $\omega \cdot \tau \in S_1$ . Since  $\omega$  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,

$$\{\omega \cdot a \mid \omega \in S_1\} \subseteq S_1, \quad (1)$$

$$\{\omega \cdot b \mid \omega \in S_1\} \subseteq S_1, \quad (2)$$

and

$$\{\omega \cdot c \mid \omega \in S_1\} \subseteq S_1. \quad (3)$$

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

Now let us consider the state  $q_0$ , and consider a string  $\omega$  such that  $\omega \in S_0$  — so that  $\omega$  does not include an “a”.

- The string  $\omega \cdot a$  certainly does include an “a” (since it ends with this symbol); that is,  $\omega \cdot a \in S_1$ . Since  $\omega$  was arbitrarily chosen from  $S_0$  it follows that

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

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

$$\delta(q_0, a) = q_1.$$

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

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

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

$$\delta(q_0, b) = q_0.$$

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

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

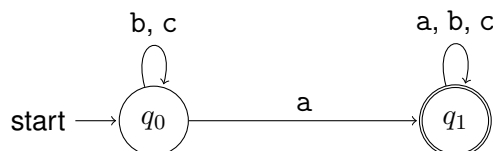
and the transition out of  $q_0$  for the symbol c is well-defined. In particular,

$$\delta(q_0, c) = q_0.$$

Thus all transitions out of state  $q_{n0}$  are well-defined.

## The DFA That Has Been Produced

A deterministic finite automaton  $M = (Q, \Sigma, \delta, q_{no}, F)$  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.



## Proving the Correctness of This DFA

Consider, now, the problem of *proving* that the deterministic finite automaton, given above, has the above language  $L$ .

### Confirming That We Have a DFA with Alphabet $\Sigma$

To begin, let us recall that

$$M = (Q, \Sigma, \delta, q_1, F)$$

where the components of  $M$  are as follows.

- $\Sigma = \{a, b, c\}$  — the same **alphabet** as used to define the language  $L$ .
- $Q = \{q_0, q_1\}$  — a finite, nonempty set of **states** such that  $Q \cap \Sigma = \{\emptyset\}$ .
- The **start state**,  $q_0$ , is a state in  $Q$ .
- The set  $F = \{q_1\}$  of **accepting states** is a subset of  $Q$ .
- The **transition function** is a well-defined total function  $\delta : Q \times \Sigma \rightarrow Q$ .

Indeed, an inspection of the transition function shows that this function can also be described using the following transition table.

	a	b	c
$q_0$	$q_1$	$q_0$	$q_0$
$q_1$	$q_1$	$q_1$	$q_1$

Thus this is a well-defined deterministic finite automaton with alphabet  $\Sigma$  — so its language,  $L(M)$ , is a subset of  $\Sigma^*$  — just like  $L$  is. It remains only to prove that these are the *same* subset of  $\Sigma^*$ , that is,  $L(M) = L$ .

- (a) Every string in  $\Sigma^*$  belongs to **exactly one** of  $S_0$  or  $S_1$  (needed since  $Q = \{q_0, q_1\}$ , and  $S_0$  and  $S_1$  are the subsets of  $\Sigma^*$  corresponding to  $q_0$  and  $q_1$ , respectively).
- (b)  $\lambda \in S_0$  (needed since  $S_0$  corresponds to the start state,  $q_0$ ).
- (c)  $S_1 \subseteq L$  (needed since the state  $q_1$ , corresponding to the set  $S_1$ , belongs to  $F$ ).
- (d)  $S_0 \cap L = \emptyset$  (needed since the state  $q_0$ , corresponding to the set  $S_0$ , does not belong to  $F$ ).
- (e) The following properties are satisfied.
  - (i)  $\{\omega \cdot a \mid \omega \in S_0\} \subseteq S_1$  (needed since  $\delta(q_0, a) = q_1$ ).
  - (ii)  $\{\omega \cdot b \mid \omega \in S_0\} \subseteq S_0$  (needed since  $\delta(q_0, b) = q_0$ ).
  - (iii)  $\{\omega \cdot c \mid \omega \in S_0\} \subseteq S_0$  (needed since  $\delta(q_0, c) = q_0$ ).
  - (iv)  $\{\omega \cdot a \mid \omega \in S_1\} \subseteq S_1$  (needed since  $\delta(q_1, a) = q_1$ ).
  - (v)  $\{\omega \cdot b \mid \omega \in S_1\} \subseteq S_1$  (needed since  $\delta(q_1, b) = q_1$ ).
  - (vi)  $\{\omega \cdot c \mid \omega \in S_1\} \subseteq S_1$  (needed since  $\delta(q_1, c) = q_1$ ).

Figure 1: Properties Used to Prove That  $L(M) = L$

### Confirming that $L(M) = L$

Recall that the lecture notes introduced a “key technical claim” that could be used to prove that a deterministic finite automaton has a given language. In order to apply this result, the properties shown in Figure 1, above, must all be shown to hold.

Since the above deterministic finite automaton has only two states — whose corresponding subsets of  $\Sigma^*$  are the language  $L$  and its complement — it is easy to show that properties (a) – (d) are satisfied. Proving condition (e) is not quite as straightforward — primarily because it includes quite a few relationships between sets that must be checked.

**Lemma 1.** *Each of the conditions shown in property (e) (as shown in Figure 1) is satisfied.*

*Proof.* Each of the claimed relationships can be established by considering the definitions of the subsets of  $\Sigma^*$  that are mentioned in the claims.

- Let  $\omega \in \Sigma^*$  such that  $\omega \in S_0$ , so that  $\omega$  does not include an “a”.

The string  $\omega \cdot a$  certainly does include an “a”, since it ends with a copy of this symbol, so  $\omega \cdot a \in S_1$ . On the other hand, neither of the strings  $\omega \cdot b$  or  $\omega \cdot c$  include an “a”, since there are no copies of “a” in  $\omega$ , so that  $\omega \cdot b \in S_0$  and  $\omega \cdot c \in S_0$ . Since the string  $\omega$  was arbitrarily chosen such that  $\omega \in S_0$  it follows that

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

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

and

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

That is, conditions (i), (ii) and (iii) are satisfied.

- Let  $\omega \in \Sigma^*$  such that  $\omega \in S_1$ , so that  $\omega$  includes at least one “a”. Then  $\omega \cdot \sigma$  must also include at least one “a”, for any symbol  $\sigma \in \Sigma$ , since  $\omega$  is a substring of  $\omega \cdot \sigma$ . Thus  $\omega \cdot \sigma \in S_1$ , for all  $\sigma \in \Sigma$ . That is,  $\omega \cdot a \in S_1$ ,  $\omega \cdot b \in S_1$ , and  $\omega \cdot c \in S_1$ .

Since the string  $\omega$  was arbitrarily chosen such that  $\omega \in S_1$ , it now follows that

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

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

and

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

That is, conditions (iv), (v) and (vi) are satisfied, as needed to establish the claim.  $\square$

**Theorem 2.**  $L(M) = L$ .

*Proof.* This result will be established by applying the “Correctness of a DFA” theorem that was introduced in the notes for this lecture. An examination of the statement of that theorem, the language  $L$ , and the above DFA confirms that it is necessary, and sufficient, to show that each of the properties that are listed in Figure 1 are satisfied, in order to confirm that  $L(M) = L$ . “Correctness of a DFA” theorem that was introduced in the notes for this lecture.

- Let  $\omega \in \Sigma^*$ . Then either  $\omega$  does not include a copy of “a”, so that  $\omega \in S_0$ , or  $\omega$  includes at least one copy of “a”, so that  $\omega \in S_1$ . That is,  $\omega \in S_0 \cup S_1$ . Since  $\omega$  was arbitrarily chosen from  $\Sigma^*$ , it follows that every string in  $\Sigma^*$  belongs to *at least* one of the sets  $S_0$  or  $S_1$ .

On the other hand,  $S_0 \cap S_1 = \emptyset$  — for if  $\omega$  belonged to both  $S_0$  and  $S_1$  then  $\omega$  would have to include at least one “a” (since  $\omega \in S_1$ ) and no a’s at all (since  $\omega \in S_0$ ) — and both of these conditions cannot be satisfied at the same time. Thus every string in  $\Sigma^*$  belongs to *exactly* one of the subsets  $S_0$  and  $S_1$ . That is, property (a) is satisfied.

- Since the empty string,  $\lambda$  does not include any a’s,  $\lambda \in S_0$  — and property (b) is satisfied.
- Since  $S_1 = \{\omega \in \Sigma^* \mid \omega \text{ includes at least one “a”}\} = L$ ,  $S_1 \subseteq L$ , and property (c) is satisfied.
- As noted above,  $S_0 \cap S_1 = \emptyset$  — so that  $S_1 \cap L = \emptyset$ , since  $S_1 = L$ . Thus property (d) is satisfied.

- Lemma 1 implies that property (e) is satisfied.

Since all the properties listed in Figure 1 are satisfied, it now follows by the theorem concerning the “correctness of a DFA”, from the lecture notes, that  $L(M) = L$ , as claimed.  $\square$

## More About This Example

This example is considerably shorter and simpler than the one in the preparatory material (even though more details are given than is necessary, once again) — because the deterministic finite automaton, being considered, here, has fewer states than the deterministic finite automaton in the first example did.

***It is certainly not a mistake*** to present a different (correct) deterministic finite automaton than the one given as a solution for a problem — but a proof of the correctness of the deterministic finite automaton will generally be longer, and more complicated, for a deterministic finite automaton with lots of states than it is for a deterministic finite automaton with only a few states.