Proving Correctness of DFAs and Lower Bounds

Mahesh Viswanathan

Induction is a proof principle that is often used to establish a statement of the form "for all natural numbers i, some property $P(i)$ holds", i.e., $\forall i \in \mathbb{N}.\ P(i)$. In this class, there will be many occassions where we will need to prove that some property holds for all strings, especially when proving the correctness of a DFA design, i.e., $\forall w \in \Sigma^*.\ S(w)$. We will often prove such statements "by induction on the length of w". What that means is "We will prove $\forall w.\ S(w)$ by proving $\forall i \in \mathbb{N}. \forall w \in \Sigma^i.\ S(w)$". That is, we will take ith statement to be proved by induction to be $\forall w \in \Sigma^i.\ S(w)$. Before giving examples of such proofs, we will begin by establishing some basic properties of DFAs that will be useful.

1 Properties of DFAs

Let us fix a DFA $M = (Q, \Sigma, \delta, s, A)$ for the rest of this section. Recall the following definition of computation $p \xrightarrow{w}_M q$ that captures the notion that M, when started in state p, on input w, could end up in state q.

Definition 1. *For states $p, q \in Q$, and string $w = w_1 w_2 \cdots w_k$, where for each i, $w_i \in \Sigma$, we say $p \xrightarrow{w}_M q$ if there is a sequence of states $r_0, r_1, \ldots r_k$ such that*

1. *$r_0 = p$,*
2. *for each $i > 0$, $\delta(r_i, w_{i+1}) = r_{i+1}$, and*
3. *$r_k = q$.*

Thus, a computation from p to q on input w is a sequence of states (of length $|w|+1$), where the first state in the sequence is p (condition 1 above), last state is q (condition 3), and every state in the sequence other than the first, is obtained by taking a transition from the previous state in the sequence on the corresponding input symbol from w (condition 2). Notice that is naturally ensures that for any p, $p \xrightarrow{\epsilon}_M q$ iff $p = q$ and $p \xrightarrow{a}_M q$ for $a \in \Sigma$ iff $\delta(p, a) = q$.

One important property about DFAs is that for any state p and input string w, there is a *unique* state q such that $p \xrightarrow{w}_M q$. This is the reason why DFAs are *deterministic*, and we state and prove this observation next.

Proposition 1. *For any p and $w \in \Sigma^*$,*

$$|\{q \in Q \mid p \xrightarrow{w}_M q\}| = 1$$

Proof. Proof is by induction on $|w|$. Thus, the ith statement proved by induction is taken to be

For every $p \in Q$, and $w \in \Sigma^i$, $|\{q \in Q \mid p \xrightarrow{w}_M q\}| = 1$.

Base Case: We need to prove the case when $w \in \Sigma^0$. Thus, $w = \epsilon$. By definition, $p \xrightarrow{w}_M q$ if and only $q = p$ which establishes the claim.

Induction Hypothesis: Suppose for every $p \in Q$, and $w \in \Sigma^*$ such that $|w| < i$, we have

$$|\{q \in Q \mid p \xrightarrow{w}_M q\}| = 1$$

Induction Step: Consider (without loss of generality) $w = a_1 a_2 \cdots a_i$, such that $a_j \in \Sigma$ (for $1 \leq j \leq i$). Take $u = a_1 \cdots a_{i-1}$

$$\begin{array}{ll} p \xrightarrow{w}_M q & \text{iff there are } r_0, r_1, \ldots, r_i \text{ such that } r_0 = p,\ r_i = q,\ \text{and } \delta(r_j, a_{j+1}) = r_{j+1} \\ & \text{iff there is } r_{i-1} \text{ such that } p \xrightarrow{u}_M r_{i-1} \text{ and } \delta(r_{i-1}, a_i) = q \end{array}$$

Now, by induction hypothesis, since $|\{q \in Q \mid p \xrightarrow{u}_M q\}| = 1$, there is a unique r_{i-1} such that $p \xrightarrow{u}_M r_{i-1}$. Also, since from any state r_{i-1} on symbol a_i the next state is uniquely determined, $|\{q \in Q \mid p \xrightarrow{w}_M q\}| = 1$.

□

Proposition 1 allows us to introduce a notation for the (unique) state of the DFA reached on input w from p. Since this is often used we will formally define it.

Definition 2. *$\delta^*_M(p, w) = q$ where q is the unique state such that $p \xrightarrow{w}_M q$.*

We could have defined $\delta^*_M(\cdot)$ inductively as follows.

$$\delta^*_M(p, w) = \begin{cases} p & \text{if } w = \epsilon \\ \delta^*_M(\delta(p, a), u) & \text{if } w = au \text{ with } a \in \Sigma,\ u \in \Sigma^* \end{cases}$$

This inductive definition is equivalent to the way we have defined $\delta^*_M(\cdot)$ in these notes. In addition the following observations are a simple consequence of the definition of $\delta^*_M(\cdot)$.

- For every $q \in Q$, $\delta^*_M(q, \epsilon) = q$, and
- For every $q \in Q$, and $a \in \Sigma$, $\delta^*_M(q, a) = \delta(q, a)$.

Consider an input string $u \bullet v$ that is the concatenation of two strings u and v. The state reached by the DFA M on $u \bullet v$ when started in state p is the same as the state reached by M on input v when started in q, where $q = \delta^*_M(p, u)$. This is a straightforward observation, but it is very useful.

Proposition 2. *For every $u, v \in \Sigma^*$ and $p \in Q$, $\delta^*_M(p, uv) = \delta^*_M(\delta^*_M(p, u), v)$.*

Proof. Let $u = a_1 a_2 \ldots a_i$ and $v = a_{i+1} \cdots a_{i+k}$, where $a_j \in \Sigma$ for every $1 \leq j \leq i + k$. Observe that,

$$\begin{array}{ll} q = \delta^*_M(p, uv) & \text{iff } p \xrightarrow{uv}_M q \\ & \text{iff there are } r_0, r_1, \ldots, r_{i+k} \text{ such that } r_0 = p,\ r_{i+k} = q,\ \text{and } \delta(r_j, a_{j+1}) = r_{j+1} \\ & \text{iff } p \xrightarrow{u}_M r_i \text{ and } r_i \xrightarrow{v}_M q \\ & \text{iff } r_i = \delta^*_M(p, u) \text{ and } q = \delta^*_M(r_i, v) \\ & \text{iff } q = \delta^*_M(\delta^*_M(p, u), v) \end{array}$$

□

2 Proving Correctness of DFA Constructions

To show that a DFA $M = (Q, \Sigma, \delta, s, A)$ accepts/recognizes a language L, we need to prove

$$\begin{array}{rl} & L = \mathbf{L}(M) \\ \textit{i.e.,} & \forall w.\ w \in \mathbf{L}(M) \text{ iff } w \in L \\ \textit{i.e.,} & \forall w.\ \delta^*_M(s, w) \in A \text{ iff } w \in L \end{array}$$

This last statement ($\forall w.\ \delta^*_M(s, w) \in A$ iff $w \in L$) is often proved by induction on $|w|$.

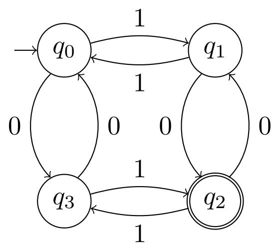

Figure 1: Transition Diagram of M_1

2.1 Example: Odd zeros and ones

Consider the DFA M_1 shown in Figure 1. We will prove that

$$\mathbf{L}(M_1) = \{w \in \{0,1\} \mid w \text{ has an odd number of 1s and an odd number of 0s}\}$$

Unrolling what it means for a string w to be in $\mathbf{L}(M_1)$, and taking A to stand for the accepting states of M_1, the above statement requires us to prove

$$\forall w.\ \delta^*_{M_1}(q_0, w) \in A \text{ iff } w \text{ has an odd number of 1s and an odd number of 0s}$$

Observing that there is only one accepting state (q_2) we could further simplify what we need to prove as follows.

$$\forall w.\ \delta^*_{M_1}(q_0, w) = q_2 \text{ iff } w \text{ has an odd number of 1s and an odd number of 0s}$$

We will prove the above statement by induction on $|w|$.

Base Case Since we are doing induction on $|w|$, the base case is when $|w| = 0$ or $w = \epsilon$. Observe that $\delta^*_{M_1}(q_0, \epsilon) = q_0 \neq q_2$. Further $w = \epsilon$ neither has an odd number of 1s nor an odd number of 0s. Thus, we have established the base case.

Induction Hypothesis Let us assume that the claim holds for all w, such that $|w| < i$. That is,

$$\forall w.\ \text{if } |w| < i \text{ then } \delta^*_{M_1}(q_0, w) = q_2 \text{ iff } w \text{ has an odd number of 1s and an odd number of 0s}$$

Induction Step Consider a string w such that $|w| = i$, where $i > 0$. Any such string can be assumed to be of the form ua, where $u \in \{0,1\}^*$ and $a \in \{0,1\}$. Based on what a is we have two subcases to consider.

If $a = 0$ then we have $w = u0$. Using Proposition 2, we have $\delta^*_{M_1}(q_0, u0) = \delta^*_{M_1}(\delta^*_{M_1}(q_0, u), 0)$. Since the only transition labeled 0 coming into state q_2 is from q_1, we have

$$\delta^*_{M_1}(q_0, u0) = \delta^*_{M_1}(\delta^*_{M_1}(q_0, u), 0) = q_2 \text{ iff } \delta^*_{M_1}(q_0, u) = q_1$$

Now, $|u| < i$, but can we use the induction hypothesis to conclude anything about u? Unfortunately, we cannot. The induction hypothesis only tells us that if on an input u, M_1 goes q_0 to q_2 then u has an odd number of 1s and 0s; the induction hypothesis says nothing about an input that takes M_1 to state q_1. Our induction proof cannot be completed and has failed.

The only way for us to succeed, is to prove (surprisingly) a stronger statement than what is needed to prove the correctness of M_1. This is often called *strengthening the induction hypothesis* and is typical of many induction proofs. The strengthening will explicitly characterize the strings that lead to q, for *each state* q (and not just the accepting state).

How do we determine what is true about strings that lead to a state q? This is based on our intuition about what each state "remembers" of the string it has seen so far. For the specific example at hand, we know that q_0 remembers that the input so far has an even number of 0s and an even number of 1s; q_1 remembers that the input so far has an even number of 0s but an odd number of 1s; q_2 remembers that the input has an odd number of 0s and 1s; and finally, q_3 remembers that the input has an odd number 0s and an even number of 1s.

Armed with this intuition, we will prove the following (stronger) statement by induction on $|w|$. For every string w,

(a) $\delta^*_{M_1}(q_0, w) = q_0$ iff w has an even number of 0s and even number of 1s,

(b) $\delta^*_{M_1}(q_0, w) = q_1$ iff w has an even number of 0s and an odd number of 1s,

(c) $\delta^*_{M_1}(q_0, w) = q_2$ iff w has an odd number of 0s and an odd number of 1s, and

(d) $\delta^*_{M_1}(q_0, w) = q_3$ iff w has an odd number of 0s and an even number of 1s.

Observe that if we manage to prove the above statement, the correctness of M_1 follows immediately because the strings accepted by M_1 are those that reach q_2.

Notice that we are proving, that all four conditions (a),(b),(c), and (d) hold for all strings. When we prove such a statement by induction on $|w|$, the ith statement (i.e., $P(i)$ in the induction template) is that for every string w of length i, (a),(b),(c), and (d) hold.

Base Case When $|w| = 0$, $w = \epsilon$. We make the following two observations: $\delta^*_{M_1}(q_0, \epsilon) = q_0$, and $w = \epsilon$ has even number of 0s and 1s. This shows that condition (a) holds. Further (b), (c), and (d) hold vaccuously. Thus, we have established the base case.

Induction Hypothesis Assume that for any string w of length $< i$, conditions (a), (b), (c), and (d) hold.

Induction Step Consider w of length i, where $i > 0$. Without loss of generality, w is of the form ua, where $a \in \{0, 1\}$ and $u \in \{0, 1\}^{i-1}$ [1]. We can complete the induction step through a case analysis.

- *Case $q = q_0$, $a = 0$:* $\delta^*_{M_1}(q_0, u0) = q_0$ iff $\delta^*_{M_1}(q_0, u) = q_3$ (because the only incoming 0 transition into q_0 is from q_3) iff by induction hypothesis (condition (d)) u has odd number of 0s and even number of 1s iff $u0$ has even number of 0s and an even number of 1s. Thus (a) has been established for the induction step when $a = 0$.
- *Case $q = q_0$, $a = 1$:* $\delta^*_{M_1}(q_0, u1) = q_0$ iff $\delta^*_{M_1}(q_0, u) = q_1$ (because the only incoming 1 transition into q_0 is from q_1) iff by induction hypothesis (condition (b)) u has even number of 0s and odd number of 1s iff $u1$ has even number of 0s and an even number of 1s. Thus (a) has been established for the induction step when $a = 1$.
- *Case $q = q_1$, $a = 0$:* $\delta^*_{M_1}(q_0, u0) = q_1$ iff $\delta^*_{M_1}(q_0, u) = q_2$ (because the only incoming 0 transition into q_1 is from q_2) iff by induction hypothesis (condition (c)) u has odd number of 0s and odd number of 1s iff $u0$ has even number of 0s and an odd number of 1s. Thus (b) has been established for the induction step when $a = 0$.
- *Case $q = q_1$, $a = 1$:* $\delta^*_{M_1}(q_0, u1) = q_1$ iff $\delta^*_{M_1}(q_0, u) = q_0$ (because the only incoming 1 transition into q_1 is from q_0) iff by induction hypothesis (condition (a)) u has even number of 0s and even number of 1s iff $u1$ has even number of 0s and an odd number of 1s. Thus (b) has been established for the induction step when $a = 1$.
- *Case $q = q_2$, $a = 0$:* $\delta^*_{M_1}(q_0, u0) = q_2$ iff $\delta^*_{M_1}(q_0, u) = q_1$ (because the only incoming 0 transition into q_2 is from q_1) iff by induction hypothesis (condition (b)) u has even number of 0s and odd number of 1s iff $u0$ has odd number of 0s and an odd number of 1s. Thus (c) has been established for the induction step when $a = 0$.

[1] **Further thought:** Why do we assume that w is of the form ua, and not of the form au? Will the induction proof, as stated go through if we assumed w to be of the form au?

- *Case $q = q_2$, $a = 1$:* $\delta^*_{M_1}(q_0, u1) = q_2$ iff $\delta^*_{M_1}(q_0, u) = q_3$ (because the only incoming 1 transition into q_2 is from q_3) iff by induction hypothesis (condition (d)) u has odd number of 0s and even number of 1s iff $u1$ has odd number of 0s and an odd number of 1s. Thus (c) has been established for the induction step when $a = 1$.
- *Case $q = q_3$, $a = 0$:* $\delta^*_{M_1}(q_0, u0) = q_3$ iff $\delta^*_{M_1}(q_0, u) = q_0$ (because the only incoming 0 transition into q_3 is from q_0) iff by induction hypothesis (condition (a)) u has even number of 0s and even number of 1s iff $u0$ has odd number of 0s and an even number of 1s. Thus (d) has been established for the induction step when $a = 0$.
- *Case $q = q_3$, $a = 1$:* $\delta^*_{M_1}(q_0, u1) = q_3$ iff $\delta^*_{M_1}(q_0, u) = q_2$ (because the only incoming 1 transition into q_3 is from q_2) iff by induction hypothesis (condition (c)) u has odd number of 0s and odd number of 1s iff $u1$ has odd number of 0s and an even number of 1s. Thus (d) has been established for the induction step when $a = 1$.

The above loooong case analysis can be simplified and shortened by carefully renaming the states and introducing a new notation. For $w \in \{0,1\}^*$ and $a \in \{0,1\}$, let us denote by $\#_a(w)$ the number of times the symbol a appears in w. Let us rename q_0 as $(0,0)$, q_1 by $(0,1)$, q_2 by $(1,1)$ and q_3 as $(1,0)$. Under the new naming, we could define $M_1 = (Q, \Sigma, \delta, s, A)$ as follows.

- $Q = \{0,1\} \times \{0,1\}$
- $\Sigma = \{0,1\}$
- $s = (0,0)$
- $A = \{(1,1)\}$
- And δ defined as
$$\delta((i,j), a) = \begin{cases} ((i+1) \bmod 2, j) & \text{if } a = 0 \\ (i, (j+1) \bmod 2) & \text{if } a = 1 \end{cases}$$
We could make the definition δ even more succinct as
$$\delta((i,j), a) = ((i + (1-a)) \bmod 2, (j+a) \bmod 2)$$

The strengthened statement that we will prove by induction can be now written as

$$\forall w.\ \delta^*_{M_1}((0,0), w) = (\#_0(w) \bmod 2, \#_1(w) \bmod 2)$$

Notice how much simpler this statement is when compared with conditions (a), (b), (c), and (d). The induction proof is also suitably much shorter.

Base Case When $|w| = 0$, $w = \epsilon$. We have

$$\delta^*_{M_1}((0,0), \epsilon) = (0,0) = (\#_0(\epsilon) \bmod 2, \#_1(\epsilon) \bmod 2)$$

Induction Hypothesis Assume that for every w with $|w| < i$, we have $\delta^*_{M_1}((0,0), w) = (\#_0(w) \bmod 2, \#_1(w) \bmod 2)$

Induction Step Consider w such that $|w| = i$, where $i > 0$. Without loss of generality, we can again assume that $w = ua$, where $a \in \{0,1\}$ and $u \in \{0,1\}^{i-1}$. The proof is then completed as follows.

$$\begin{aligned}
\delta^*_{M_1}((0,0), w = ua) &= \delta^*_{M_1}(\delta^*_{M_1}((0,0), u), a) && \text{(Proposition 2)} \\
&= \delta(\delta^*_{M_1}((0,0), u), a) && (\delta^*_{M_1}(q, a) = \delta(q, a) \text{ for } a \in \{0,1\}) \\
&= \delta((\#_0(u) \bmod 2, \#_1(u) \bmod 2), a) && \text{(induction hypothesis on } u) \\
&= ((\#_0(u) + (1-a)) \bmod 2, (\#_1(u) + a) \bmod 2) && \text{(definition of } \delta) \\
&= (\#_0(ua) \bmod 2, \#_1(ua) \bmod 2)
\end{aligned}$$

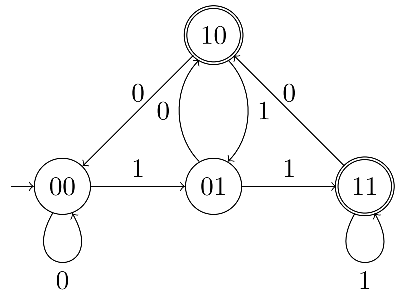

Figure 2: Transition Diagram of M_2

2.2 Example: One in second last position

Consider the DFA M_2 shown Figure 2. For a string $w \in \{0,1\}$ let $\text{last}_2(w)$ be the last two symbols in w defined precisely as follows.

$$\text{last}_2(w) = \begin{cases} w & \text{if } |w| < 2 \\ ab & \text{if } w = uab \text{ where } u \in \{0,1\}^*,\ a,b \in \{0,1\} \end{cases}$$

We will prove that

$$\mathbf{L}(M_2) = L_2 = \{w \in \{0,1\}^* \mid \text{last}_2(w) \in \{10, 11\}\}$$

Again, unrolling the definition of $\mathbf{L}(M_2)$, and observing that the accepting states of M_2 are $\{10, 11\}$, the above statement requires us to prove

$$\forall w.\ \delta^*_{M_2}(00, w) \in \{10, 11\} \text{ iff } \text{last}_2(w) \in \{10, 11\} \tag{1}$$

Once again, if we try to prove this statement by induction on $|w|$ we will fail in the induction step because it is too weak; it does not characterize when a string reaches 00 or 01.

To obtain a strengthening that can be proved by induction, we rely on our intuition about how DFA M_2 works — it remembers the last two symbols seen. However, since the start state of M_2 is 00, after reading string w, the machine M_2 remembers the last two symbols of $00w$ (and not w). Thus, the strong correctness statement we will prove is the following.

$$\forall w.\ \delta^*_{M_2}(00, w) = \text{last}_2(00w) \tag{2}$$

Before we prove Equation 2 by induction on $|w|$, let us see how it implies Equation 1 or in other words the correctness of M_2. For this we need the following lemma.

Lemma 3. *For any* $w \in \{0,1\}^*$, $\text{last}_2(00w) \in \{10, 11\}$ *iff* $\text{last}_2(w) \in \{10, 11\}$.

Proof. There are two directions to establish. Observe that if $\text{last}_2(w) \in \{10, 11\}$ then $|w| \geq 2$ and hence $\text{last}_2(00w) = \text{last}_2(w)$. Conversely, observe that if $|w| < 2$ then $\text{last}_2(00w) \in \{00, 01\}$. Hence, if $\text{last}_2(00w) \in \{10, 11\}$ then $|w| \geq 2$ and hence (again) $\text{last}_2(00w) = \text{last}_2(w)$. □

We can now show that Equation 1 follows from Equation 2 because

$$\begin{array}{lll} \delta^*_{M_2}(00, w) \in \{10, 11\} & \text{iff } \text{last}_2(00w) \in \{10, 11\} & \text{(because of Equation 2)} \\ & \text{iff } \text{last}_2(w) \in \{10, 11\} & \text{(because of Lemma 3)} \end{array}$$

We now complete the proof by showing Equation 2 by induction on $|w|$.

Base Case When $|w| = 0$, $w = \epsilon$. Now, $\delta^*_{M_2}(00, w = \epsilon) = 00 = \text{last}_2(00\epsilon)$. This establishes the base case.

Induction Hypothesis Assume that $\delta^*_{M_2}(00, w) = \text{last}_2(00w)$ for all w such that $|w| < i$.

Induction Step Consider w such that $|w| = i$, for $i > 0$. Without loss of generality, w is of the form ua, where $u \in \{0,1\}^{i-1}$ and $a \in \{0,1\}$. Recall that we can write the transition function of M_2 as

$$\delta(ab, c) = bc = \text{last}_2(abc)$$

Now we can complete the proof as follows.

$$\begin{aligned}
\delta^*_{M_2}(00, w = ua) &= \delta^*_{M_2}(\delta^*_{M_2}(00, u), a) && \text{(Proposition 2)} \\
&= \delta(\delta^*_{M_2}((00), u), a) && (\delta^*_{M_2}(q, a) = \delta(q, a) \text{ for } a \in \{0,1\}) \\
&= \delta(\text{last}_2(00u), a) && \text{(induction hypothesis on } u) \\
&= \text{last}_2(00ua) && \text{(definition of } \delta)
\end{aligned}$$

2.3 Proof Template for Proving Correctness of DFAs

Based on the above examples, we can come up with a standard template for proving correctness of DFA constructions. Given a DFA $M = (Q, \Sigma, \delta, s, A)$, to prove that $\mathbf{L}(M) = L$ we take the following steps.

1. For each $q \in Q$, identify a language L_q.
2. Prove the following statement by induction on $|w|$

$$\forall w. \forall q \in Q.\ \delta^*_M(s, w) = q \text{ iff } w \in L_q$$

3. Finally prove that $L = \cup_{q \in A} L_q$

The language L_q maybe only implicitly identified in the correctness statement that we prove by induction. For example, in Section 2.1, after renaming states as (i, j) with $i, j \in \{0,1\}$, the language $L_{(i,j)} = \{w \in \{0,1\}^* \mid \#_0(w) = i \text{ and } \#_1(w) = j\}$ is implicit in the correctness statement.

3 Proving DFA Lower Bounds

Consider a DFA $M = (Q, \Sigma, \delta, s, A)$ that recognizes a language L. Suppose $u, v \in \Sigma^*$ are two strings such that $\delta^*_M(s, u) = \delta^*_M(s, v)$. Then for any string w, we have

$$\begin{aligned}
\delta^*_M(s, uw) &= \delta^*_M(\delta^*_M(s, u), w) && \text{(Proposition 2)} \\
&= \delta^*_M(\delta^*_M(s, v), w) && (\delta^*_M(s, u) = \delta^*_M(s, v)) \\
&= \delta^*_M(s, vw) && \text{(Proposition 2)}
\end{aligned}$$

Hence, for every w, either M accepts both uw and vw or rejects both uw and vw. Since M recognizes L then means that either both uw and vw are in L or neither one is.

The contrapositive of the above observation is the following. Suppose for a language L, and strings u, v, we have a string w such that $uw \in L$ but $vw \notin L$ then in *every* DFA M that recognizes L, u and v must go to different states. When this happens, w is said to *distinguish* u and v (with respect to L). This leads to the notion of a *fooling set*,

Definition 3. *A* fooling set *for $L \subseteq \Sigma^*$ is a set $F \subseteq \Sigma^*$ such that for every $u, v \in F$ such that $u \neq v$ there is a w such that either $uw \in L$ and $vw \notin L$ or $uw \notin L$ and $vw \in L$.*

Notice that based on our observations above we can conclude that no two strings in a fooling set F for L can go to the same state in *any* DFA recognizing L. Hence if L has a fooling set F of size k, every DFA recognizing L has at least k states. Identifying a fooling set for a language helps establish the optimality of certain DFA designs.

3.1 Example: Even length strings with 2 as

Consider the language

$$L_{\text{even}}^{\geq 2a} = \{w \in \{a,b\}^* \mid w \text{ has even length and contains at least 2 } a\text{s}\}$$

This language can be recognized by a DFA that keeps track of the number of as seen (either 0, 1, or ≥ 2), *and* the parity (odd or even) of the number of symbols we have seen. Thus the states of this DFA are of the form (n, p), where $n \in \{0, 1, 2\}$ is the number of as seen and $p \in \{e, o\}$ is the parity of the number of symbols seen. The transition function of this DFA is shown in Figure 3.

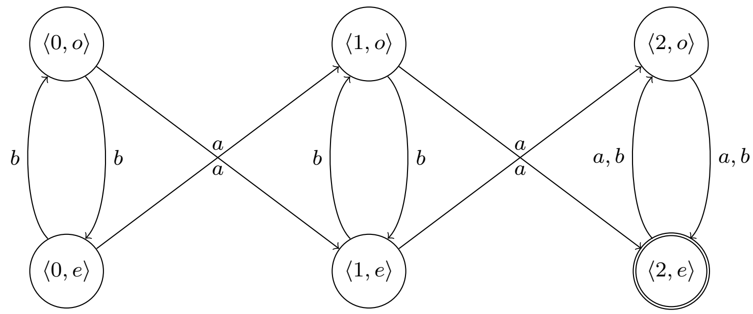

Figure 3: DFA recognizing $L_{\text{even}}^{\geq 2a}$

Now the above DFA seems to have the fewest states possible — any DFA recognizing $L_{\text{even}}^{\geq 2a}$ must keep track of the two pieces of information. We can turn this intuition into a mathematical proof by constructing a fooling set.

We can show that any DFA recognizing $L_{\text{even}}^{\geq 2a}$ has at least 6 states by constructing a fooling set F of size 6. We will come up with this fooling set based on our intuition that any DFA recognizing $L_{\text{even}}^{\geq 2a}$ must remember both the number of as and the parity of the length of the string. So the fooling set F will contain strings such that any two of them will either differ in the number of as or in the parity of the length.

Let us take $F = \{\epsilon, b, a, ab, aa, aab\}$. To finish the proof, we need to argue that F is a fooling set. For that we need to show that all possible 15 pairs are distinguishable.

- Case $u = aa$ and $v \in F \setminus \{u\}$. The string $w = \epsilon$ distinguishes u and v. This is because $uw = w = aa \in L_{\text{even}}^{\geq 2a}$ and for any $v \in F \setminus \{u\}$, $vw = v \notin L_{\text{even}}^{\geq 2a}$.
- Case $u = \epsilon$, and $v \in \{b, a, aab\}$. The string $w = aa$ distinguishes any such pair. The reason is $uw = aa \in L_{\text{even}}^{\geq 2a}$ but $vw \notin L_{\text{even}}^{\geq 2a}$
- Case $u = \epsilon$ and $v = ab$. The string $w = a$ distinguishes u and v. This is because $uw = a \notin L_{\text{even}}^{\geq 2a}$ while $vw = aba \in L_{\text{even}}^{\geq 2a}$.
- Case $u = aab$ and $v \in \{a, b, ab\}$. Taking $w = b$, we observe that $uw = aabb \in L_{\text{even}}^{\geq 2a}$, while $vw \notin L_{\text{even}}^{\geq 2a}$.
- Case $u = a$ and $v \in \{b, ab\}$. Taking $w = a$, we have $uw = aa \in L_{\text{even}}^{\geq 2a}$ while $vw \notin L_{\text{even}}^{\geq 2a}$.
- Case $u = b$ and $v = ab$. Taking $w = aaa$ we have $uw = baaa \in L_{\text{even}}^{\geq 2a}$ but $vw = abaaa \notin L_{\text{even}}^{\geq 2a}$.

3.2 Example: One k positions from the end

The language L_2 in Section 2.2 was shown to have a 4 state DFA. One can show 4 is the fewest number of states needed to recognize L_2. In this section, we will prove a more general result — let L_k denote the set of binary strings having a 1 k positions from the end, and we will show that any DFA recognizing L_k has at least 2^k states.

For a string $w \in \{0,1\}^*$ define $\text{last}_k(w)$ to be last k symbols in w. That is

$$\text{last}_k(w) = \begin{cases} w & \text{if } |w| < k \\ v & \text{if } w = uv \text{ where } u \in \Sigma^* \text{ and } v \in \Sigma^k \end{cases}$$

Consider the language L_k as follows.

$$L_k = \{w \in \{0,1\}^* \mid \text{last}_k(w) = 1u \text{ where } u \in \{0,1\}^{k-1}\}$$

We can define a simple DFA M_k that recognizes L_k using the same intuition as M_2 for L_2 — M_k will remember the last k input symbols read. Thus formally, we have $M_k = (Q_k, \{0,1\}, \delta_k, s_k, A_k)$ where

- $Q_k = \{0,1\}^k$
- $\delta_k(w, a) = \text{last}_k(wa)$
- $s_k = 0^k$
- $A = \{w \in \{0,1\}^k \mid w = 1u \text{ where } u \in \{0,1\}^{k-1}\}$

We can prove that $\mathbf{L}(M_k) = L_k$ in a manner similar to Section 2.2 by showing

$$\forall w.\ \delta^*_{M_k}(0^k, w) = \text{last}_k(0^k w)$$

To show that every DFA recognizing L_k must have at least 2^k states, we will construct a fooling set F of size 2^k. Our fooling set will simply be the set of all binary strings of length k, i.e., $F = \{0,1\}^k$. Notice that F has 2^k elements. To prove that F is a fooling set, let us consider any $u, v \in F$ such that $u \neq v$. Since $u \neq v$, there must be a position where u and v have different symbols. Let i be the first such position. Without loss of generality, let us assume that u has 0 in position i, and v has 1 in position i.

Consider $w = 0^{i-1}$. The strings uw and vw are as follows.

$$\begin{aligned} u0^{i-1} &= \ \ldots\ \overbrace{0\ \ldots\ 0^{i-1}}^{k} \\ v0^{i-1} &= \underbrace{\ldots}_{i-1}\ 1 \underbrace{\ldots}_{k-i} 0^{i-1} \end{aligned}$$

Thus, $u0^{i-1} \notin L_k$ and $v0^{i-1} \in L_k$. Hence, w distinguishes u and v with respect to L_k.