Recall the automaton that accepts the set of all strings $w$ over (alphabet) $\{a, b\}$ such that $w$ begins and ends with $a$:

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$a$</th>
<th>$b$</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$q_1$</td>
<td>$q_3$</td>
<td>start state</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_1$</td>
<td>$q_2$</td>
<td>previous symbol was $a$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_1$</td>
<td>$q_2$</td>
<td>previous symbol was $b$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$q_3$</td>
<td>$q_3$</td>
<td>error: didn’t start with $a$</td>
</tr>
</tbody>
</table>

We can formally prove correctness by verifying the assertions made about each state. Assertions take the form “$\delta(q_0, w) = q$ iff $w \in L(q)$”, that is, the automaton ends up in state $q$ when given input $w$ if and only if $w$ satisfies the assertion for state $q$. In our example,

$$L(q_0) = \{ w \in \{a, b\}^* \mid w = \varepsilon \}$$
$$L(q_1) = \{ w \in \{a, b\}^* \mid \exists x \text{ with } w = ax \text{ and } \exists y \text{ with } w = yb \}$$
$$L(q_2) = \{ w \in \{a, b\}^* \mid \exists x \text{ with } w = ax \text{ and } \exists y \text{ with } w = yb \}$$
$$L(q_3) = \{ w \in \{a, b\}^* \mid \exists x \text{ with } w = bx \}$$

Note that the assertions are simply formalizations of the comments I made next to the states in the table.

It is sufficient to prove for all $q \in Q$ the assertion

$$\text{if } \delta(q_0, w) = q \text{ then } w \in L(q)$$

We can prove the converse, namely that, for any $q$, $w \in L(q)$ implies $\delta(q_0, w) = q$, using the fact that the automaton is well-defined (a transition exists for every possible state/symbol pair) and that the languages in the assertions are disjoint. First note that $w \in L(q)$ means $w \notin L(p)$ for all $p \in Q - \{q\}$. Using the contrapositive of (1), we know that $\delta(q_0, w) \neq p$ for all $p \in Q - \{q\}$. By process of elimination, we can conclude that $\delta(q_0, w) = q$.

Verification of (1) is done by induction on $|w|$. The basis is $w = \varepsilon$ and it is obvious that $\varepsilon \in L(q_0)$ as desired (we know $\delta(q_0, \varepsilon) = q_0$ by definition).

For the induction step we assume $w = v\sigma$ and that (1) holds with $v$ in place of $w$. We then verify that (1) holds for $w$ and every $q \in Q$. So if $\delta(q_0, w) = q$ then $\delta(p, \sigma) = q$, where $p = \delta(q_0, v)$. In other words, we need to analyze all the transitions into state $q$. For $q = q_0$, the conclusion follows by default — there are no transitions into $q_0$. For $q = q_1$, note that $p$ is either $q_0$, $q_1$, or $q_2$ and $\sigma = a$. By induction (we can assume $v \in L(p)$), we see that $v$ does not begin with $b$ (true for $v \in L(q_0)$, $L(q_1)$ or $L(q_2)$), so $w = va$ is clearly in $L(q_1)$. When $q = q_2$, the incoming transitions have $p = q_1$ or $q_2$ and $\sigma = b$ and we know that $v$ begins with $a$; hence $w = vb \in L(q_2)$. For $q = q_3$, we see that either $p = q_0$ and $\sigma = b$, from which $v = \varepsilon$ and $w = b \in L(q_3)$, or $p = q_3$, from which $v$ begins with $b$ and $w = v\sigma$ also begins with $b$. 

For the induction case $q = q_3$, we can assume $v \in L(p)$ and if $w \in L(q)$ then $w = va$ or $w = vb$. The basis is $w = \varepsilon$ and it is obvious that $\varepsilon \in L(q_0)$ as desired (we know $\delta(q_0, \varepsilon) = q_0$ by definition).
Now consider the example in Figure 1. A reasonable first guess at $L(M)$ (which $= L(D)$) might be: “the set of all strings ending in 101”. We can check the validity of this hypothesis by deducing assertions for the various states from it. If the given English description were correct, then $L(D)$ would be the set of all strings ending in 101. Then $L(A)$ would have to include all strings ending in 1010, and $L(B)$ would include all strings ending in 10101. But the latter is a subset of $L(D)$, which contradicts the fact that the automaton is deterministic.

A better hypothesis can be formulated using the observation that $\delta(D,01) = B$ and $\delta(B,01) = D$. This suggests that $B$ and $D$ partition the strings ending in 101 into those that have an even number of occurrences of the trailing 01 and those that have an odd number. It is straightforward to verify the following:

\[
L(A) = \{w \in \{0,1\}^* \mid w = x(10)^k \text{ where } k \text{ is even and } x \text{ does not end in } 10, \text{ and, if } k = 0, \ x \text{ does not end in } 1\}
\]
\[
L(B) = \{w \in \{0,1\}^* \mid w = x(10)^k1, \text{ where } k \text{ is even and } x \text{ does not end in } 10\}
\]
\[
L(C) = \{w \in \{0,1\}^* \mid w = x(10)^k, \text{ where } k \text{ is odd and } x \text{ does not end in } 10\}
\]
\[
L(D) = \{w \in \{0,1\}^* \mid w = x(10)^k1, \text{ where } k \text{ is odd and } x \text{ does not end in } 10\}
\]

It’s hard to say this more simply in English without becoming imprecise. One possibility is: “The set of strings ending in a 1 preceded by an odd number of occurrences of 10”.

The extra complication in the definition of $L(A)$ is needed to avoid conflict with $L(B)$ in cases where $k = 0$.

A similar approach to proving the correctness of a DFA is shown in the solution to Exercise 2.7 on page 53 of the Hopcroft and Ullman book.