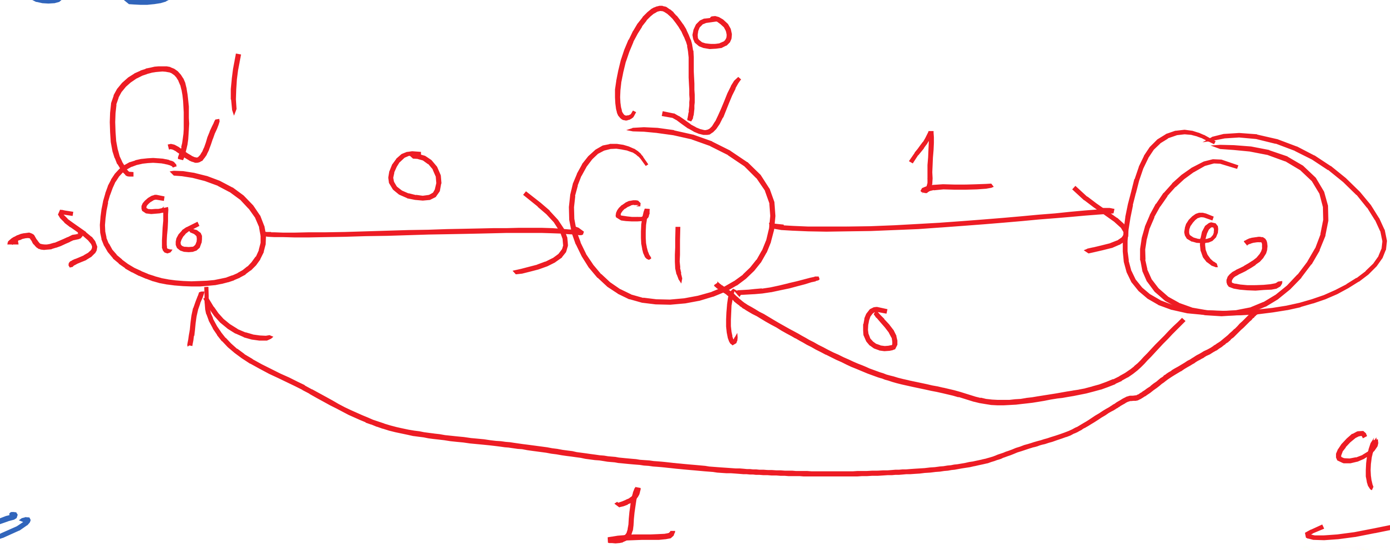
Distinguishable Strings with respect to Language L

$x, y \in \Sigma^*$

$(0+1)^*01$



q_0

$x = 1$

$xz = 11 \notin L$

$xz \notin L$

$yz \in L$

q_1

$y = 0$

$z = 1$

$L/y = \{1, 101, \ldots\}$

$L/x = \{01, 0101, 001, \ldots\}$

$L/x \neq L/y$

Distinguishable Strings with respect to Language L

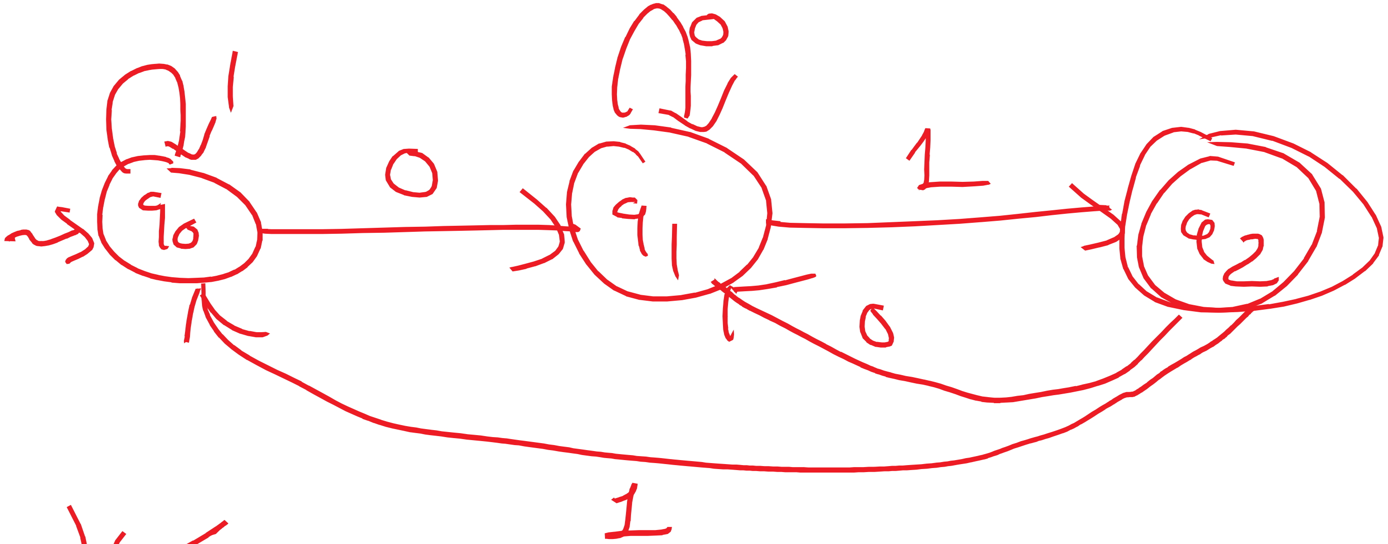
If L is a language over the alphabet Σ, and x and y are strings in Σ^*, then x and y are *distinguishable with respect to L*, or *L-distinguishable*, if there is a string $z \in \Sigma^*$ such that either $xz \in L$ and $yz \notin L$, or $xz \notin L$ and $yz \in L$. A string z having this property is said to distinguish x and y with respect to L. An equivalent formulation is to say that x and y are L-distinguishable if $L/x \neq L/y$, where

$$L/x = \{z \in \Sigma^* \mid xz \in L\}$$

The two strings x and y are L-indistinguishable if $L/x = L/y$, which means that for every $z \in \Sigma^*$, $xz \in L$ if and only if $yz \in L$.

The strings in a set $S \subseteq \Sigma^*$ are *pairwise L-distinguishable* if for every pair x, y of distinct strings in S, x and y are L-distinguishable.

Distinguishable Strings with respect to Language L



$x = 1$

$y = 01$

$z = \underline{\Lambda}$

$(1, 0)$ ✓

$(1, 01)$ ✓

$(0, 01)$ ✓

Distinguishable Strings with respect to Language L

Suppose $M = (Q, \Sigma, q_0, A, \delta)$ is an FA accepting the language $L \subseteq \Sigma^*$. If x and y are two strings in Σ^* that are L-distinguishable, then $\delta^*(q_0, x) \neq \delta^*(q_0, y)$. For every $n \geq 2$, if there is a set of n pairwise L-distinguishable strings in Σ^*, then Q must contain at least n states.

Proof

If x and y are L-distinguishable, then for some string z, one of the strings xz, yz is in L and the other isn't. Because M accepts L, this means that one of the states $\delta^*(q_0, xz)$, $\delta^*(q_0, yz)$ is an accepting state and the other isn't. In particular,

$$\delta^*(q_0, xz) \neq \delta^*(q_0, yz)$$

According to Exercise 2.5, however,

$$\delta^*(q_0, xz) = \delta^*(\delta^*(q_0, x), z)$$
$$\delta^*(q_0, yz) = \delta^*(\delta^*(q_0, y), z)$$

Because the left sides are different, the right sides must be also, and so $\delta^*(q_0, x) \neq \delta^*(q_0, y)$.

The second statement in the theorem follows from the first: If M had fewer than n states, then at least two of the n strings would cause M to end up in the same state, but this is impossible if the two strings are L-distinguishable.

For Every Pair *x*, *y* of Distinct Strings in $\{a,b\}^*$, *x* and *y* Are Distinguishable with Respect to *Pal*

Case - I

$x \neq y$

$|x| = |y|$

$\exists z \in \Sigma^*$,

$z = x^{rev}$

$xz \in L$

$yz \notin L$

$x = ab$

$y = ba$

$z = ba$

$xz = x\, x^{rev} = abba \in Pal$

$yz = y\, x^{rev} = baba \notin Pal$

For Every Pair x, y of Distinct Strings in $\{a,b\}^*$, x and y Are Distinguishable with Respect to *Pal*

Case - II $\quad x \neq y \quad |x| < |y|$ and x is not a prefix of y

$x = ab \qquad y = bbb$

$\exists z \in \Sigma^*, \quad z = x^{rev}$

$xz = x\,x^{rev} = abba \in Pal$

$yz = y\,x^{rev} = bbbba \notin Pal$

For Every Pair x, y of Distinct Strings in $\{a,b\}^*$, x and y Are Distinguishable with Respect to *Pal*

Case - III $\qquad |x| < |y| \qquad x$ is a Prefix of y

$x = ab \qquad y = \underbrace{ab}_{y_1}\underbrace{a}_{y_2} \qquad y_1 = x \quad y_2 = a$

$\exists z \in \Sigma^*, \quad z = \sigma x^{rev}$

$\boxed{\begin{array}{c} y_2\sigma \notin Pal \\ \sigma = b \\ ab \notin Pal \end{array}}$

$xz = x\sigma x^{rev} = abbba \in Pal$

$yz = y\sigma x^{rev} = ababba \notin Pal$

Distinguishable Strings with respect to Language L

For Every Pair x, y of Distinct Strings in $\{a,b\}^*$, x and y Are Distinguishable with Respect to *Pal*

First suppose that $x \neq y$ and $|x| = |y|$. Then x^r, the reverse of x, distinguishes the two with respect to *Pal*, because $xx^r \in Pal$ and $yx^r \notin Pal$. If $|x| \neq |y|$, we assume x is shorter. If x is not a prefix of y, then $xx^r \in Pal$ and $yx^r \notin Pal$. If x is a prefix of y, then $y = xz$ for some nonnull string z. If we choose the symbol σ (either a or b) so that $z\sigma$ is not a palindrome, then $x\sigma x^r \in Pal$ and $y\sigma x^r = xz\sigma x^r \notin Pal$.

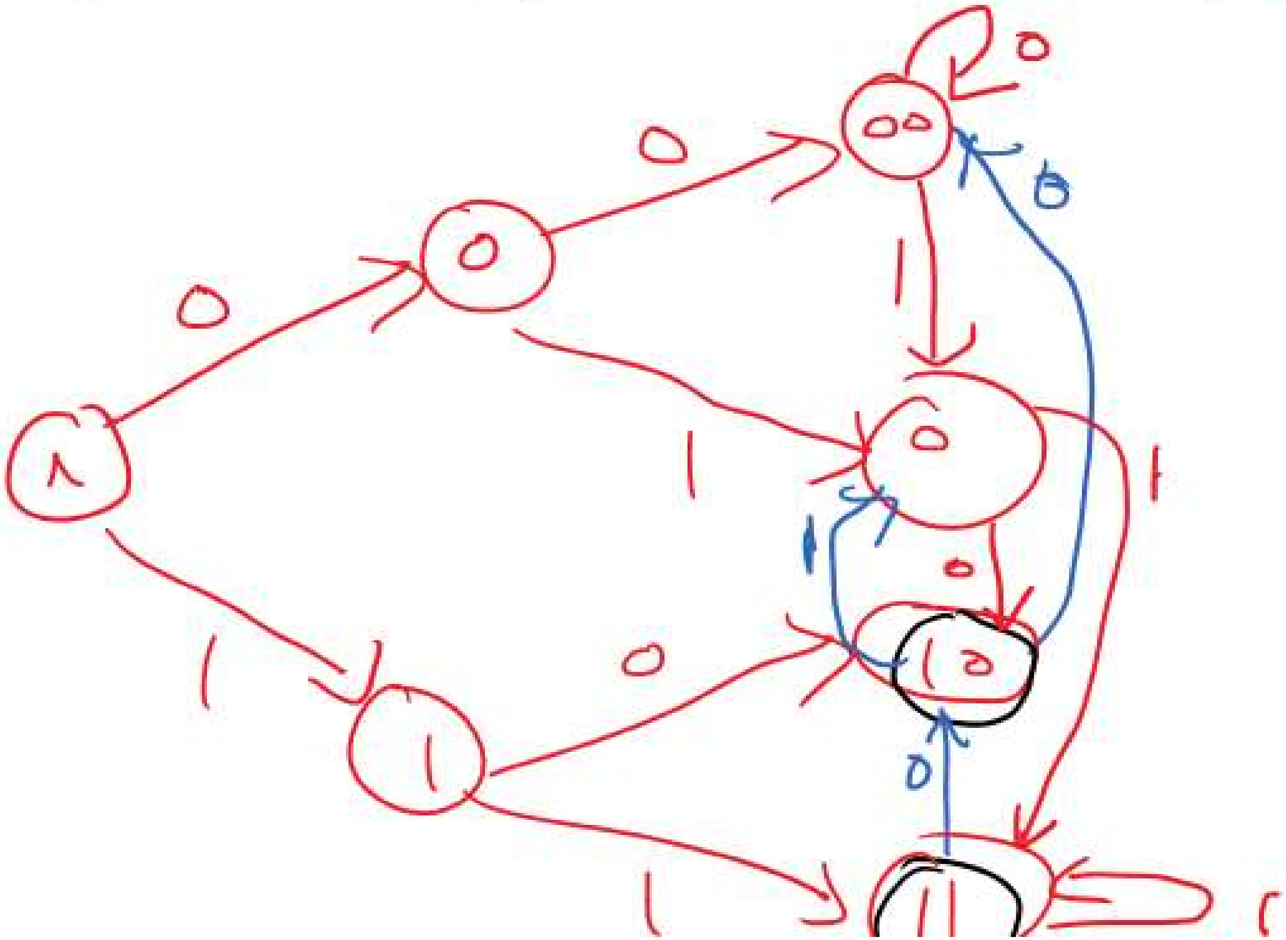
An explanation for this property of *Pal* is easy to find. If a computer is trying to accept *Pal*, has read the string x, and starts to receive the symbols of another string z, it can't be expected to decide whether z is the reverse of x unless it can actually remember every symbol of x. The only thing a finite automaton M can remember is what state it's in, and there are only a finite number of states. If x is a sufficiently long string, remembering every symbol of x is too much to expect of M.

Suppose n is a positive integer, and L_n is the language of strings in $\{a, b\}^*$ with at least n symbols and an a in the nth position from the end.

$n = 2$, L_2 Lang.

$a = 1$

$b = 0$



Suppose n is a positive integer, and L_n is the language of strings in $\{a, b\}^*$ with at least n symbols and an a in the nth position from the end.

$a=1 \quad b=0$

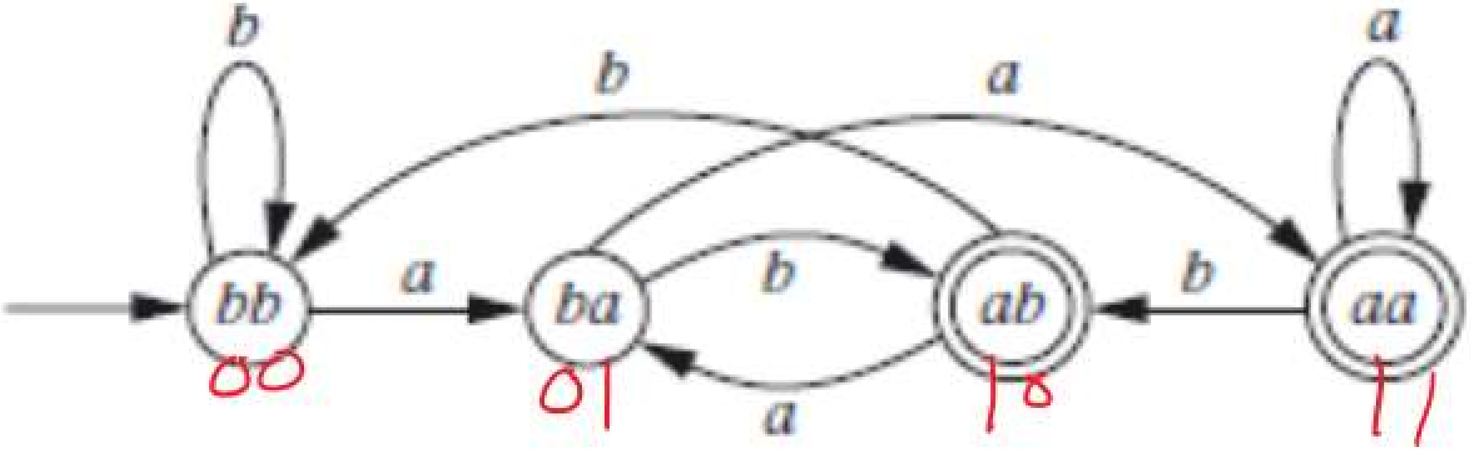


Figure 2.25 |

An FA accepting L_n in the case $n = 2$.