Problem Set 2

Spring 10

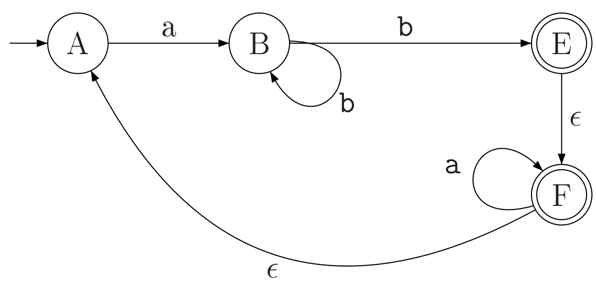
Due: Thursday Feb 11 in class before the lecture.

Please <u>follow</u> the homework format guidelines posted on the class web page:

`http://www.cs.uiuc.edu/class/sp10/cs373/`

1. NFA comprehension [**Category**: Comprehension, **Points**: 20]

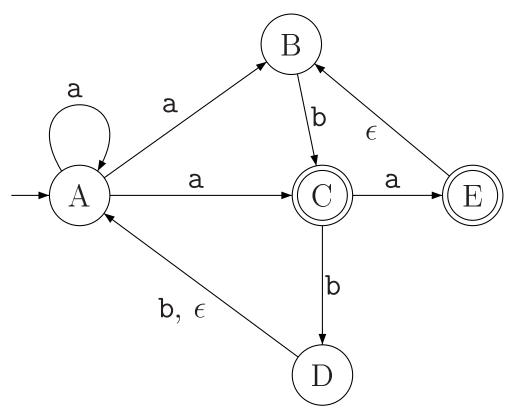
   Consider the following NFA M.



   (a) Give a regular expression that represents the language of M. Explain briefly why it is correct. (6 Points)

   (b) Recall the definition of an NFA accepting a string w (Sipser p. 54). Show formally that M accepts the string $w =$ `abbab` (6 Points)

   (c) Let $\Sigma = \{\texttt{a}, \texttt{b}\}$. Give the formal definition of the following NFA N (in tuple notation). Make sure you describe the transition function completely (for every state and every letter). (8 Points)



Solution:

(a) The language of M is $\texttt{ab}(\texttt{a} \cup \texttt{b})^*$. Each string in this language can be viewed as $\texttt{ab}$ followed by a sequence of substrings, either $\texttt{a}$'s or $\texttt{b}$'s. If this sequence starts from a substring of $\texttt{a}$'s and ends with a substring of $\texttt{b}$'s, then there is a state sequence $A \xrightarrow{\texttt{a}} B \xrightarrow{\texttt{b}} E \xrightarrow{\texttt{a}} \ldots \xrightarrow{\texttt{a}} B \xrightarrow{\texttt{b}} \ldots \xrightarrow{\texttt{b}} E \to \cdots \to E$. Other situations are easy to verify similarly.

(b) Consider the sequence of states $ABEFABE$ and a sequence of inputs $\texttt{abb}\epsilon\epsilon\texttt{ab}$. Note that $\texttt{abbab} = \texttt{abb}\epsilon\epsilon\texttt{ab}$. The first state A is the start state and the last state E is an accept state. Moreover, it can be verified that for each two adjacent states s, s' in the sequence, there is a transition from s to s' reading the corresponding inputs. Then by the definition of acceptance, M indeed accepts $\texttt{abbab}$.

(c) Formally, $N = (\text{Q}, \Sigma, \delta, \text{A}, \text{F})$, where

$Q = \{A, B, C, D, E\}$;

$\Sigma = \{\texttt{a}, \texttt{b}\}$;

δ is defined as follows:

δ	$\texttt{a}$	$\texttt{b}$	ϵ
A	$\{A, B, C\}$	$\varnothing$	$\varnothing$
B	$\varnothing$	$\{\text{C}\}$	$\varnothing$
C	$\{E\}$	$\{D\}$	$\varnothing$
D	$\varnothing$	$\{A\}$	$\{A\}$
E	$\varnothing$	$\varnothing$	$\{B\}$

$F = \{C, E\}$.

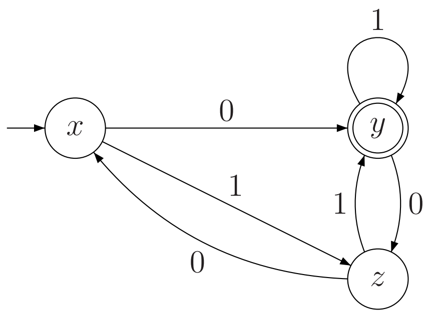
2. DFA Transformation [**Category**: Comprehension, **Points**: 20]

Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$, we define $\mathcal{T}(M)$ to be a new DFA $(Q', \Sigma, \delta', q_0', F')$ such that

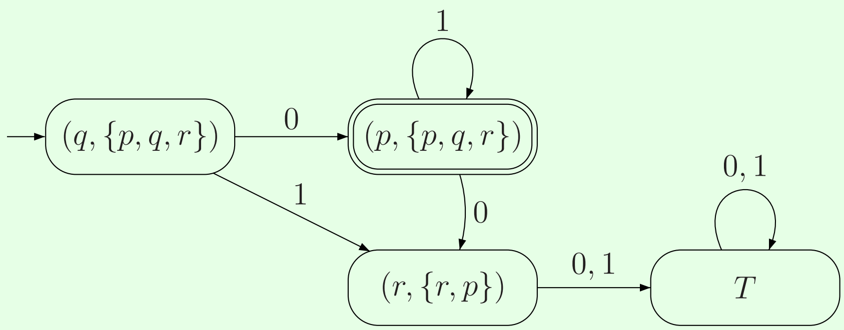
$$
\begin{aligned}
&\forall q \in Q,\ N_q = \{\delta^*(q, ab) : a, b \in \Sigma\} \\
&Q' = \{(q, N_q) : q \in Q\} \cup \{T\} \\
&F' = \{(q, N_q) : q \in F\} \\
&q_0' = (q_0, N_{q_0}) \\
&\forall q \in Q', \forall a \in \Sigma,
\end{aligned}
$$

$$
\delta'(q, a) = \begin{cases} T & q = T \\ \left(\delta(p, a), N_{\delta(p,a)}\right) & \exists p \in Q,\ q = (p, N_p) \text{ and } N_{\delta(p,a)} \subseteq \{\delta(r, a) : r \in N_p\} \\ T & \text{otherwise} \end{cases}
$$

Let M be the following DFA. Draw $\mathcal{T}(M)$ (label the states).



Solution:



3. Language Projection [**Category**: Proof, **Points**: 20]

   Let's define $w \downarrow \Sigma$ to be a word w', such that w' is equal to w with all symbols not in Σ removed. For example, $abcdbdcad \downarrow \{a, b, c\} = abcbca$.
   Let L_1 be a regular language over the alphabet Σ_1 and L_2 be a regular language over the alphabet Σ_2.

   Prove that $L = \{w \in (\Sigma_1 \cup \Sigma_2)^* \mid (w \downarrow \Sigma_1) \in L_1 \ \wedge \ (w \downarrow \Sigma_2) \in L_2\}$ is also regular.
   Note: Σ_1 and Σ_2 may have common symbols.

Solution:

Theorem 0.1 *If language L over alphabet Σ is regular, then $L' = \{w \in \Sigma \cup \Gamma \parallel (w \downarrow \Sigma) \in L\}$ is also regular.*

Since L is regular, there exists a DFA $A = (Q, \Sigma, \delta, q_0, F)$ that accepts L. Let's construct a new DFA $A' = (Q, \Sigma \cup \Gamma, \delta', q_0, F)$ with

$$\delta'(q, c) = \begin{cases} \delta(q, c) & : c \in \Sigma \\ q & : c \notin \Sigma \end{cases}$$

That is, we stay in the current state if the letter is not in the source alphabet and move according to the δ-function of the source DFA otherwise.

To prove that this DFA accepts exactly L' we need to prove that for any $w \in \Sigma \cup \Gamma$, such that $w = x_1 \ldots x_n$ and $w \downarrow \Sigma = y_1 \ldots y_m = y$ where $m \leq n$, $y_1 = x_{i_1}, \ldots, y_m = x_{i_m}$ and $i_1 < i_2 < \cdots < i_m$:

(a) $(w \downarrow \Sigma) \in L \Rightarrow w \in L(A')$
$(w \downarrow \Sigma) \in L$ means that there exists a subsequence y in w such that $w \downarrow \Sigma = y$ and $y \in L(A)$. By construction, A' will accept any string $w' \in (\Sigma \cup \Gamma)$ that contains y as a subsequence. w contains y as a subsequence, hence $w \in L(A')$.

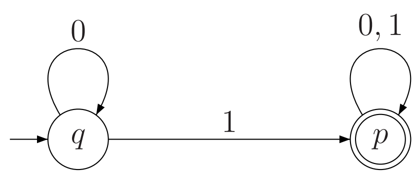
(b) $w \in L(A') \Rightarrow (w \downarrow \Sigma) \in L$
Since $w \in L(A')$, there exists a trace $r_0, \ldots, r_n$ that leads from the start state to the final state ($q_0 = r_0$ and $r_n \in F$). If for some symbol x_i in w, $x_i \notin \Sigma$, then, by construction, $\delta'(r_{i-1}, x_i) = r_i = r_{i-1}$, that is, we stay in the same state. $w \downarrow \Sigma$ will remove all such symbols from w, but the trace will start and end in the same state as before. That means that on $w \downarrow \Sigma$, A' will remain in the final state, or $\delta'^*(q_0, w) = \delta'^*(q_0, w \downarrow \Sigma) = r_n \in F$. But since all symbols in $w \downarrow \Sigma$ are in Σ, *delta* is defined on all of these symbols and is equal to *delta*$'$. This means that $\delta^*(q_0, w \downarrow \Sigma) = r_n \in F$, and hence $(w \downarrow \Sigma) \in L$.

By the above theorem $L_1' = \{w \in \Sigma_1 \cup \Sigma_2 \parallel (w \downarrow \Sigma_1) \in L_1\}$ and $L_2' = \{w \in \Sigma_1 \cup \Sigma_2 \parallel (w \downarrow \Sigma_2) \in L_2\}$ are regular.
Since $L = \{w \in \Sigma_1 \cup \Sigma_2 \parallel (w \downarrow \Sigma_1) \in L_1 \ \wedge \ (w \downarrow \Sigma_1) \in L_1\} = L_1' \cap L_2'$, L is regular as an intersection of two regular languages.

4. Language of a DFA [**Category**: Proof, **Points**: 20]

First have a look at the following claim and its formal proof. The proof uses induction. You may think that since the claim is an easy fact you don't need such a heavy technique for proving it and in fact you are right! We could avoid induction and build a much easier proof for the claim. The reason that we have applied induction to prove this claim is to introduce this technique to you.

Claim: The language of the DFA D below is $A = \{0^n 1x : x \in \{0,1\}^*, n \geq 0\}$.



Proof: Let $L(p)$ represent the set of all strings that if we feed them to the DFA D, then D will stop in state p. Similarly define $L(q)$ for state q. Note that since p is the only final state, we have $L(D) = L(p)$. Instead of proving the Claim directly, we will introduce a stronger claim and we will prove that stronger claim using induction (and this stronger claim is easier to attack using induction).

The Stronger Claim: $L(q) = C = \{0^n : n \geq 0\}$ and $L(p) = A = \{0^n 1x : x \in \{0,1\}^*, n \geq 0\}$.

Note that the stronger claim asks for everything in the previous old claim and also asks for something more; this is why sometimes it is called *overloaded claim.*

Proof of the Stronger Claim: Let B_k represent the set of all binary strings of length at most k. Using induction on k, we will prove that for every value of k, we have $L(q) \cap B_k = C \cap B_k$ and $L(p) \cap B_k = A \cap B_k$ (as an easy exercise, please justify for yourself that if we prove this, then we have proved the stronger claim).

Base case: When $k = 0$. We have $B_0 = \{\epsilon\}$. When we feed ϵ to D, it stops in state q and therefore $L(q) \cap B_0 = \{\epsilon\}$ and $L(p) \cap B_0 = \varnothing$. It is trivial to see that $C \cap B_0 = \{\epsilon\}$ and $A \cap B_0 = \varnothing$. Therefore we have $L(q) \cap B_0 = C \cap B_0$ and $L(p) \cap B_0 = A \cap B_0$.

Inductive Step: Assume that for some $k \geq 0$ we have $L(q) \cap B_k = C \cap B_k$ and $L(p) \cap B_k = A \cap B_k$, then we prove that $L(q) \cap B_{k+1} = C \cap B_{k+1}$ and $L(p) \cap B_{k+1} = A \cap B_{k+1}$.

First we prove $L(q) \cap B_{k+1} = C \cap B_{k+1}$. Since from induction hypothesis we know $L(q) \cap B_k = C \cap B_k$, we just need to show that $L(q) \cap \{0,1\}^{k+1} = C \cap \{0,1\}^{k+1}$ (justify this for yourself). Let $x \in L(q) \cap \{0,1\}^{k+1}$, write $x = x'a$ where $x' \in B_k$ and $a = 0$ or 1. Since $x \in L(q)$, we have $q = \delta^*(q,x) = \delta(\delta^*(q,x'),a)$. From this equation we have $\delta^*(q,x') = q$ and $a = 0$ (Why?). Since $\delta^*(q,x') = q$ by definition of $L(q)$ we have $x' \in L(q)$, and since $x' \in B_k$ we have $x' \in L(q) \cap B_k$. Since by induction hypothesis $L(q) \cap B_k = C \cap B_k$, we have $x' \in C \cap B_k$, and since we know that x' is of length k, we have $x' = 0^k$. But this means that $x = x'a = 0^k0 = 0^{k+1}$. Since x was an arbitrary member of $L(q) \cap \{0,1\}^{k+1}$, we have $L(q) \cap \{0,1\}^{k+1} = \{0^{k+1}\}$. It is also trivial to see that $C \cap \{0,1\}^{k+1} = \{0^{k+1}\}$, therefore we have proved that $L(q) \cap \{0,1\}^{k+1} = C \cap \{0,1\}^{k+1}$.

Now we prove that $L(p) \cap B_{k+1} = A \cap B_{k+1}$ in a similar way. Since from induction hypothesis we know $L(p) \cap B_k = A \cap B_k$, we just need to show that $L(p) \cap \{0,1\}^{k+1} = A \cap \{0,1\}^{k+1}$ (again justify this for yourself). Let $x \in L(p) \cap \{0,1\}^{k+1}$, write $x = x'a$ where $x' \in B_k$ and $a = 0$ or 1. Since $x \in L(p)$, we have $p = \delta^*(q,x) = \delta(\delta^*(q,x'),a)$. From this last equation we have that either $\delta^*(q,x') = q$ and $a = 1$, or $\delta^*(q,x') = p$ and $a = 0$ or 1 (why?).

Case1: When $\delta^*(q,x') = q$ and $a = 1$. From definition of $L(q)$ we have that $x' \in L(q)$ and since $|x'| = k$ we have $x' \in L(q) \cap B_k$. By the induction hypothesis $L(q) \cap B_k = C \cap B_k$ and therefore $x' \in C \cap B_k$. Therefore $x' = 0^k$ and $x = x'a = 0^k1 \in A \cap B_{k+1}$.

Case2: When $\delta^*(q,x') = p$ and $a = 0$ or 1. By definition of $L(p)$ we have $x' \in L(p)$ and hence $x' \in L(p) \cap B_k$. By induction hypothesis we have $L(p) \cap B_k = A \cap B_k$ and therefore $x' \in A \cap B_k$ and hence $x' = 0^n1y$ for some $n \geq 0$ and $y \in \{0,1\}^*$ (such that $n + 1 + |y| = k$). Hence $x = x'a = 0^n1ya \in A \cap B_{k+1}$.

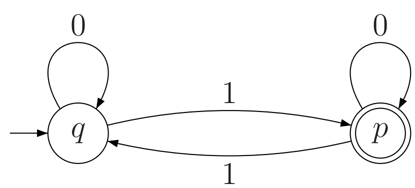
So up to this point we have proved that $L(p) \cap B_{k+1} \subseteq A \cap B_{k+1}$. Now we prove that $A \cap B_{k+1} \subseteq L(p) \cap B_{k+1}$. Let $x \in A \cap B_{k+1}$ we have $x = 0^n1y$ for some $n \geq 0$ and $y \in \{0,1\}^*$ (such that $n + 1 + |y| = k + 1$). Now:

$$\delta^*(q,x) = \delta^*(q,0^n1y) = \delta^*(\delta^*(q,0^n),1y) = \delta^*(q,1y) = \delta^*(\delta(q,1),y) = \delta^*(p,y) = p$$

Therefore $x \in L(p)$ and since already $x \in B_{k+1}$, we have $x \in L(p) \cap B_{k+1}$, therefore $A \cap B_{k+1} \subseteq L(p) \cap B_{k+1}$. So we have proved that $A \cap B_{k+1} = L(p) \cap B_{k+1}$ and the induction is complete. □

Now prove formally that the language of the following DFA is:

$$\{x \in \{0,1\}^* : x \text{ has odd number of 1's}\}$$



Solution:

Using the sample we saw in this problem, we claim that $L(p)$ and $L(q)$ (for the DFA above) are the following languages and we prove our claim using induction on the length of strings in $\{0,1\}^*$:

$$L(p) = \{x \in \{0,1\}^* : x \text{ has odd number of 1's}\}$$
$$L(q) = \{x \in \{0,1\}^* : x \text{ has even number of 1's}\}$$

Again we note that since $L(p)$ is the language of our DFA, the first equality above is basically what we are asked to prove and the second equality is the extra fact that we want to prove just because proving this stronger claim makes our induction easier!

Let's rewrite our claim in a more convenient way for applying induction,
Claim : For any string x of length $k \geq 0$, we have

$$x \in L(p) \iff x \in \{x \in \{0,1\}^* : x \text{ has odd number of 1's}\}$$
$$x \in L(q) \iff x \in \{x \in \{0,1\}^* : x \text{ has even number of 1's}\}$$

(Note that this claim is exactly our first two inequalities).

Base Case: The claim is true for $k = 0$.
proof: The only string of length 0 is ϵ. Looking at the DFA we observe that ϵ makes the DFA to finish in state q. Additionally ϵ has even number of 1's. Hence both iff ($\iff$) relations are true for $x = \epsilon$ and therefore the claim is true for $k = 0$.

Induction Hypothesis: If the claim is true for all x of length *less than* k, then it is also true for all strings of length k.
proof: Pick an arbitrary string x of length $k \geq 1$. Write $x = x'a$ where $a \in \{0,1\}$.
If $x \in L(p)$, then we have two possible subcases:

- $x' \in L(p)$ and $a = 0$: Since $|x'| = k - 1$, by the induction hypothesis the claim is true for x' and therefore x' has odd number of 1's. Hence $x = x'0$ has an odd number of $1's$.

- $x' \in L(q)$ and $a = 1$: Again, since $|x'| = k-1$, by the induction hypothesis the claim is true for x' and therefore x' has even number of 1's. Hence $x = x'1$ has an odd number of $1's$.

Therefore we have proved that for all strings of length k

$$x \in L(p) \implies x \in \{x \in \{0,1\}^* : x \text{ has odd number of 1's}\}.$$

Now we prove its reverse. Assume that x has odd number of $1's$. We have two possibilities:

- $a = 0$: Since $x = x'a$, this means that x' has odd number of 1's. Since $|x'| = k - 1$, the claim is true for x' and hence $x' \in L(p)$. But this means that $\delta^*(q, x) = \delta^*(q, x'0) = \delta(\delta^*(q, x'), 0) = \delta(p, 0) = p$. Hence $x \in L(p)$.
- $a = 1$: Since $x = x'a$, this means that x' has even number of 1's. Since $|x'| = k - 1$, the claim is true for x' and hence $x' \in L(q)$. But this means that $\delta^*(q, x) = \delta^*(q, x'1) = \delta(\delta^*(q, x'), 1) = \delta(q, 1) = p$. Hence $x \in L(p)$.

Therefore we have proved that for all strings of length k

$$x \in L(p) \Longleftarrow x \in \{x \in \{0,1\}^* : x \text{ has odd number of 1's}\}$$

and now we should concentrate on the second iff ($\iff$) condition in the claim.
If $x \in L(q)$, then we have two possible subcases:

- $x' \in L(p)$ and $a = 1$: Since $|x'| = k - 1$, by the induction hypothesis the claim is true for x' and therefore x' has odd number of 1's. Hence $x = x'1$ has an even number of $1's$.
- $x' \in L(q)$ and $a = 0$: Again, since $|x'| = k-1$, by the induction hypothesis the claim is true for x' and therefore x' has even number of 1's. Hence $x = x'0$ has an even number of $1's$.

Therefore we have proved that for all strings of length k

$$x \in L(q) \implies x \in \{x \in \{0,1\}^* : x \text{ has even number of 1's}\}.$$

Now we prove its reverse. Assume that x has even number of $1's$. We have two possibilities:

- $a = 0$: Since $x = x'a$, this means that x' has even number of 1's. Since $|x'| = k - 1$, the claim is true for x' and hence $x' \in L(q)$. But this means that $\delta^*(q, x) = \delta^*(q, x'0) = \delta(\delta^*(q, x'), 0) = \delta(q, 0) = q$. Hence $x \in L(q)$.
- $a = 1$: Since $x = x'a$, this means that x' has odd number of 1's. Since $|x'| = k - 1$, the claim is true for x' and hence $x' \in L(p)$. But this means that $\delta^*(q, x) = \delta^*(q, x'1) = \delta(\delta^*(q, x'), 1) = \delta(p, 1) = q$. Hence $x \in L(q)$.

Therefore we have proved that for all strings of length k

$$x \in L(q) \Longleftarrow x \in \{x \in \{0,1\}^* : x \text{ has even number of 1's}\}$$

which completes the proof.

5. Extra Credit/Honors [**Category**: Proof, **Points**: 20]

Let L be a regular language. Show that $L' = \{w : ww^R \in L\}$ is regular.

Solution:

Let $L \subseteq \Sigma^*$ be a regular language. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting L. We will show $L' = \{w : ww^R \in L\}$ is regular by exhibiting an NFA B accepting L'.

Intuitively, the NFA B for L' will run *two copies* of the automaton A (similar to a product construction), except that the first copy will start from the initial state of A and read the input w simulating A, while the second copy will start from a final state of A and simulate A *backwards* reading w^R. If the first copy and the second copy both reach the same state, after reading w, then we would know that A after reading ww^R will start from the initial state and reach a final state. Hence B can accept w if the two copies reach the same state. In order for B to begin with a pair of states (q_0, q_f) where $q_f \in F$, we create a new initial state q_{init} from which B can go to any state of the form (q_0, q_f) on reading ϵ.

Here is the formal construction. Let $B = ((Q \times Q) \cup \{q_{init}\}, \Sigma, \delta', q_{init}, F')$ be the NFA where:

- $\delta'(q_{init}, \epsilon) = \{(q_0, q_f) \mid q_f \in F\}$
- For every $a \in \Sigma$, $\delta'(q_{init}, a) = \emptyset$
- For every $a \in \Sigma$, $q, q' \in Q$, $\delta'((q, q'), a) = \{(p, p') \mid p = \delta(q, a), q' \in \delta(p', a)\}$.
- For every $q, q' \in Q$, $\delta'((q, q'), \epsilon) = \emptyset$
- $F' = \{(q, q) \mid q \in Q\}$

We can now prove that B accepts L'.

First, let us prove that $L' \subseteq L(B)$. Let $w \in L'$, i.e. let $ww' \in L = L(A)$. Let $w = a_1 a_2 \ldots a_n$, and let the (accepting) run in A on ww' be:

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n \xrightarrow{a_n} p_{n-1} \xrightarrow{a_{n-1}} p_{n-2} \ldots \xrightarrow{a_1} p_0$$

with $p_0 \in F$. Then it is easy to see that the following is an accepting run of B on w:

$$q_{init} \xrightarrow{\epsilon} (q_0, p_0) \xrightarrow{a_1} (q_1, p_1) \xrightarrow{a_2} (q_2, p_2) \ldots \xrightarrow{a_{n-1}} (q_{n-1}, p_{n-1}) \xrightarrow{a_n} (q_n, q_n)$$

Hence $w \in L(B)$.

Now let us show that $L(B) \subseteq L'$. Let $w \in L(B)$, and let $w = a_1, a_2, \ldots a_n$. Let an accepting run of B on w be:

$$q_{init} \xrightarrow{\epsilon} (q_0, p_0) \xrightarrow{a_1} (q_1, p_1) \xrightarrow{a_2} (q_2, p_2) \ldots \xrightarrow{a_{n-1}} (q_{n-1}, p_{n-1}) \xrightarrow{a_n} (q_n, p_n)$$

with $p_n = q_n$ (since the run must end in a final accepting state). Then it is easy to show that the following is a run in A:

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \ldots \xrightarrow{a_n} q_n \xrightarrow{a_n} p_{n-1} \xrightarrow{a_{n-1}} p_{n-2} \ldots \xrightarrow{a_1} p_0$$

Also, since $q_{init} \xrightarrow{\epsilon} (q_0, p_0)$, $p_0 \in F$, and hence the above is an accepting run in A on ww^R. Hence $ww^R \in L(A) = L$. Therefore $w \in L'$.