CS 481 FA01 HW1 Solutions

by Allen Wang

(A) Let \( N = (Q, \Sigma, \Delta, S, F) \) be an NFA where

\[
Q = \{A, B, C, D\}
\]
\[
\Sigma = \{0, 1\}
\]
\[
\Delta \text{ is defined as follows:}
\]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A,B</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

\( S = \{A\} \)

\( F = \{D\} \)

The diagram is very similar to the one on page 27 of the text.

(B) From the definition of \( L \), we can represent it as \( \{w1ab | w \in \Sigma^*, a, b \in \Sigma\} \)

We will do the proof by double inclusion.

First, show that \( x \in L \Rightarrow x \in L(N) \).

\( x \) is of the form \( w1ab \) as defined. We'll put this through \( \Delta \):

\( \hat{\Delta}(A, w1ab) = \Delta(\Delta(A, w), 1ab) \)

From the definition of \( \Delta \) (with the assistance of some induction, not shown), we know that

\( \hat{\Delta}(A, y) \cap A \neq NULL, \forall y \in \Sigma^* \)

So, using nondeterminism (all "Guessable" states equivalent at any one given transition), we can say

\( \hat{\Delta}(\hat{\Delta}(A, w), 1ab) = \hat{\Delta}(A, 1ab) \)

\( = \hat{\Delta}(\Delta(\Delta(A, 1), a), b) \) Definition of \( \hat{\Delta} \)

\( = \Delta(\Delta(B, a), b) \) Definition of \( \Delta \)

\( = \Delta(C, b) \) Definition of \( \Delta \)

\( = D \) Definition of \( \Delta \)

\( \subseteq F \)

This proves what we wanted to show.
Now we need to show \( x \in L(N) \Rightarrow x \in L \).

\( x \in L(N) \Rightarrow \hat{\Delta}(S, x) \cap F \neq NULL \)

working backwards, using the definition of \( \Delta \):

\[
\Delta(\Delta(S, x'), b) \cap F \neq NULL \quad b \in \Sigma, x'b = x
\]

\[
\Delta(\Delta(S, x''), a, b) \cap F \neq NULL \quad a, b \in \Sigma, x''a = x' \\
\text{since we bridge from } S \text{ to } F \text{ in 2 transitions,}
\]

\[
\hat{\Delta}(S, x'') \supseteq B \quad B \text{ is 2 transitions from } F
\]

Thus

\[
\Delta(\Delta(S, x'''), 1) \subseteq \hat{\Delta}(S, x')
\]

\[
\Delta(\Delta(\Delta(S, x'''), 1), a, b) \cap F \neq NULL
\]

only transition to B, \( x'''1 = x'' \)

Note that \( x''' \in \Sigma^* \) is the only restriction according to definition of \( \Delta \)

Therefore \( x \) has a form of \( wlab \), which is what we want.

We’ve shown both sides of the inclusion so we have proven \( L(N) = L \).

(C) We will write the subset-states in the following form.

example: The state for \( \{A, B\} \) will be written as the state \( AB \).

This is to prevent confusion with a set of states.

Let \( M = (Q', \Sigma, \delta, s, F') \) be a DFA where

\( Q' = 2^Q \) (dropping unnecessary states if needed)

\( \Sigma \) is the same as before

\( \delta \) is defined as follows:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>AB</td>
</tr>
<tr>
<td>AB</td>
<td>AC</td>
<td>ABC</td>
</tr>
<tr>
<td>AC</td>
<td>AD</td>
<td>ABD</td>
</tr>
<tr>
<td>AD</td>
<td>A</td>
<td>AB</td>
</tr>
<tr>
<td>ABC</td>
<td>ACD</td>
<td>ABCD</td>
</tr>
<tr>
<td>ABD</td>
<td>AC</td>
<td>ABC</td>
</tr>
<tr>
<td>ACD</td>
<td>AD</td>
<td>ABD</td>
</tr>
<tr>
<td>ABCD</td>
<td>ACD</td>
<td>ABCD</td>
</tr>
</tbody>
</table>

All other states unreachable.

\( s = A \)

\( F' = \{ AD, ABD, ACD, ABCD \} \)

2.

by Anirban Dasgupta
(A) The language $L_a$ is not regular. A simple counterexample justifies our claim. If $L = 1^*$, then $L_a = \{0^i1^i | i \geq 0\}$, and from the proof done in class, we know that this is not regular.

(B) The language $L_b$ is regular. We prove this by a construction. Let $N = (Q, \Sigma, \delta, s, F)$ be the finite automaton accepting the language $L$. WLOG, we assume that $\Sigma = \{0,1\}$. We construct an automaton accepting the language $L_b$. Define the NFA $N' = (Q', \Sigma', \Delta, s', F')$ as $Q' = Q$, $\Sigma' = \Sigma$, $s' = s$, $F' = F$. Also, define the transition function as $\Delta(q,0) = \{\delta(q,0), \delta(q,1)\}$.

The following lemma then proves that $L(N') = L_b$, once we couple it with the definition of acceptance by a finite automaton.

**Lemma 0.1** $\hat{\Delta}(S,0^k) = \bigcup_{x,|x| = k} \{\hat{\delta}(q, x) | q \in S\}$

**Proof.** We just follow the definition of the multistep transition function and do the proof by induction.

**Base** For $k = 0$, the proof is done by just the definition of the function $\Delta$.

**Inductive step** Let the lemma be assumed to be proved till $k = m$. So, $\hat{\Delta}(S,0^{m+1}) = \{\delta(q,0) | q \in \hat{\Delta}(S,0^m)\} = \{\delta(q,0) | q \in \hat{\Delta}(S,0^m)\} \cup \{\delta(q,1) | q \in \hat{\Delta}(S,0^m)\} = \{\delta(q,0) \cup \delta(q,1) | q \in \hat{\delta}(S,x), |x| = m\} = \bigcup_{x,|x| = k} \{\hat{\delta}(q, x) | q \in S\}.$

After this, we note that, a string $x \in L_b$ iff $\hat{\delta}(q, x) \in F \neq \emptyset$. Hence $L(N') = L_b$.

(C) This language is regular. We construct a finite automaton in order to prove our claim.

Let $N = (Q, \Sigma, \delta, s, F)$ be the finite automaton accepting the language $L$. We construct a new finite automaton $N' = (Q', \Sigma, \Delta, S', F')$ where $Q' = Q \times Q$, $S' = \{(s,f) | f \in F\}$ and $F' = \{(q,q) | q \in Q\}$.

Define $\Delta((p,q),0) = \{(p',q') | p' = \delta(p,0), \exists a \in \Sigma : \delta(q',a) = q\}$.

**Lemma 0.2** $\hat{\Delta}(R,0^i) = \{(p',q') | \exists (p,q) \in R, x \in \Sigma^* : |x| = i \land \hat{\delta}(p,0^i) = p' \land \hat{\delta}(q',x) = q\}$

**Proof.** By induction.

**Base.** When $i = 0$ i.e. $x = \epsilon$. The equality is trivial.

**Inductive Step.** Let $A = \hat{\Delta}(R,0^i)$. So, $\hat{\Delta}(R,0^{i+1}) = \hat{\Delta}(A,0) = \{(p',q') | \exists (p,q) \in A : p' = \delta(p,0) \land \exists a \in \Sigma : \delta(q',a) = q\}$. Putting in the definition of $A$, we get the lemma.

After this lemma we just need two more steps to show that $L_c = L(N')$. 

3
• $L_c \subseteq L(N')$: If $0^i \in L_c$, then, by definition we know that $\exists x : |x| = i \land 0^ix \in L$. Hence, $\exists q \in Q : \hat{\delta}(s, 0^i) = q \land \delta(q, x) \in F$. That is, $q$ is the “midway” state in the path that the string takes. So, in the run on $N'$, we have, $(q, q) \in \hat{\Delta}(S', 0^i) \cap F'$. Looking at the definition of $F'$, we obtain that $0^i \in L(N')$.

• $L(N') \subseteq L_c$: If $0^i \in L(N')$, then for some $q \in Q, (q, q) \in \hat{\Delta}(S', 0^i)$.
Hence, from the above lemma that we proved, $\hat{\delta}(s, 0^i) = q \land \exists x : |x| = i \land \delta(q, x) \in F$. From the definition of $L_c$, we again have $0^i \in L_c$.

3.

by Misha Zatsman

We’ll pick an arbitrary $\alpha \in \Sigma$ and define $L_k = \{a^i \mid 1 \leq i \leq k\}$. $L_k$ is regular because it is finite (it has $k$ elements).

Assume that $L_k = L(M_k)$, where $M_k = (Q_k, \Sigma, \delta_k, s_k, F_k)$. We’ll prove by contradiction that $|F_k| \geq k$:

Assume $|F_k| < k$. $L(M_k) = L_k \Rightarrow \forall i \leq k, \hat{\delta}(s_k, a^i) \in F_k$.
Since $|\{a^i| q \leq i \leq k\}| = k > |F_k|$, and $\{\hat{\delta}(s_k, a^i)| 1 \leq i \leq k\} \subseteq F_k$ the pigeonhole principle tells us $\exists i, j < k, i \neq j : \hat{\delta}(s_k, a^i) = \hat{\delta}(s_k, a^j)$.
Now we assume without loss of generality (wlog) that $i < j$.
We define $d = j - i \geq 1$ and $q = \hat{\delta}(s_k, a^i)$.
$\hat{\delta}(q, a^d) = \hat{\delta}(\hat{\delta}(s_k, a^i), a^d) = \hat{\delta}(s_k, a^i a^d) = \hat{\delta}(s_k, a^{i+d}) = \hat{\delta}(s_k, a^i) = q$,
so we’ve discovered a loop.
Now we exploit our loop by noticing that $\hat{\delta}(s_k, a^{i+kd}) = \hat{\delta}(s_k, a^i a^{kd}) = \hat{\delta}(q, a^{kd}) = \hat{\delta}(s_k, a^{i+kd}) \in L_k$,
but $d \geq 1 \Rightarrow i + kd > k \Rightarrow a^{i+kd} \notin L_k$.
So we’ve reached a contradiction, and our assumption($|F_k| < k$) must be false.