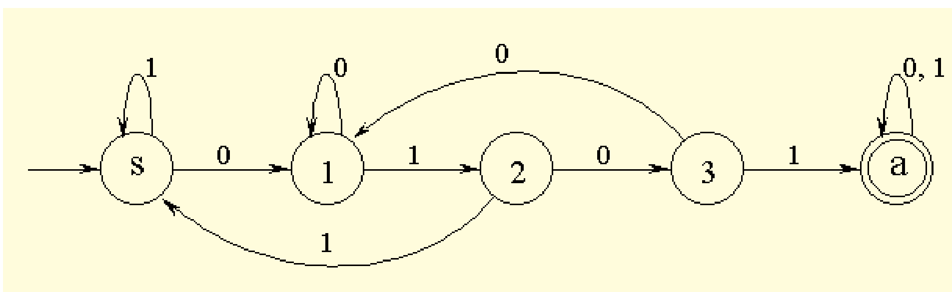
Solutions for HW1, Honors Theory, Fall 2002

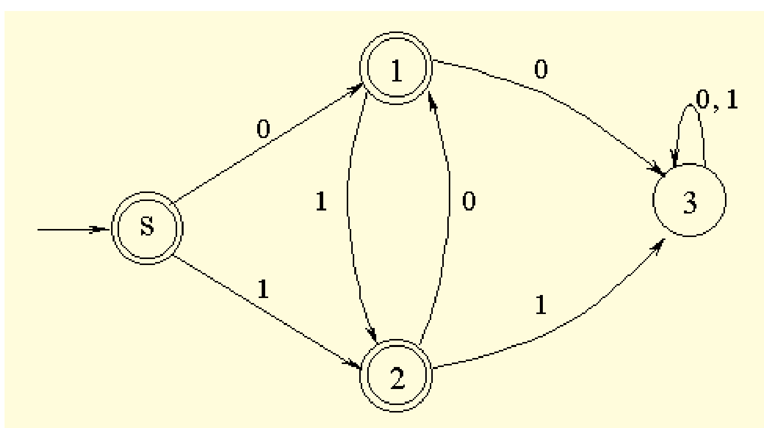
February 27, 2002

1. (10+10 Points) Construct dfa's to accept the following languages:
   (i) $\{w \in \{0,1\}^* : w$ has 0101 as substring$\}$.
   (ii) $\{w \in \{0,1\}^* : w$ has neither 00 nor 11 as substring$\}$.

   SOLUTION: (i)



   SOLUTION: (ii)

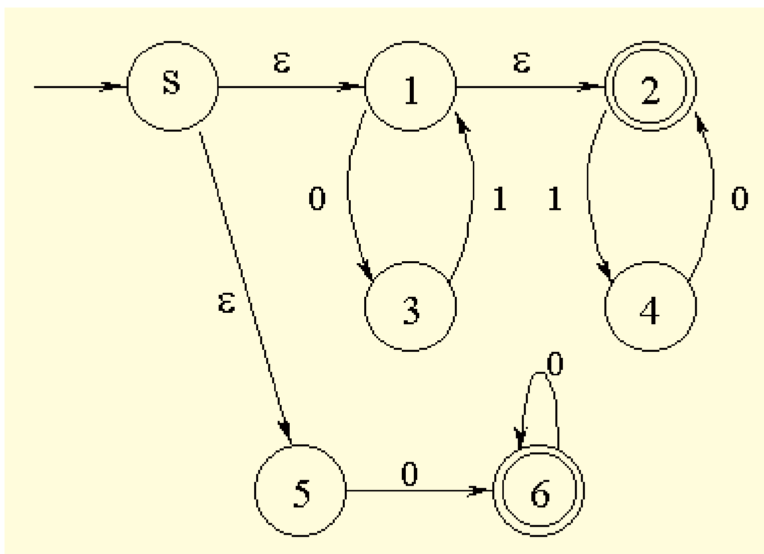


2. (10+10 Points)
   Write regular expressions for (i) and (ii) in the previous question.

   SOLUTION: (i) $(0+1)^*0101(0+1)^*$
   SOLUTION: (ii) $(01)^* + (01)^*0 + (10)^* + (10)^*1$

3. (10+10 Points) Construct nfa's to accept the languages indicated by the following expressions:
   (i) $(01)^*(10)^* + 00^*$.
   (ii) $((01 + 001)^*0^*)^*$.
   Be sure to parse these regular expressions correctly when we paretheses are omitted.

   SOLUTION: (i)



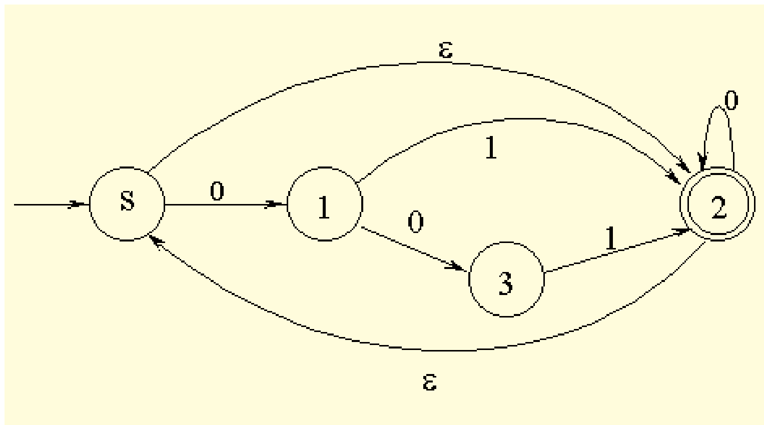
   SOLUTION: (ii)

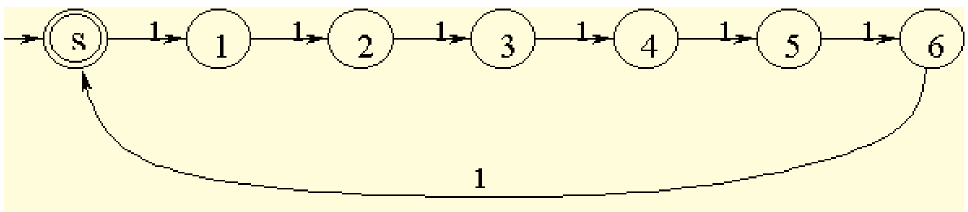
4. (10+10+10+10 Points) Prove or disprove that each of the following languages is regular:
   (i) $\{1^n : n \text{ is divisible by } 7\}$.
   (ii) $\{w \in \{0,1\}^* : w \text{ is a binary number divisible by } 7\}$.
   (iii) $\{0^n 1 0^m 1 0^{m+n} : n, m \geq 1\}$.
   (iv) $\{ww' \in \{0,1\}^* : w' \text{ is obtained from } w \text{ by interchanging } 0 \text{ and } 1\}$.

   SOLUTION: (i) It's regular, as there is a NFA M1 that recognizes it. The regular expression $(1111111)*$ describes it.

   SOLUTION: (ii) It's regular, as there is a DFA M2 that recognizes it.

   To understand this, note that the state labeled i (for $i = 0, 1, \ldots, 6$) indicates the binary string read so far is equal to $i \bmod 7$. It is easy to



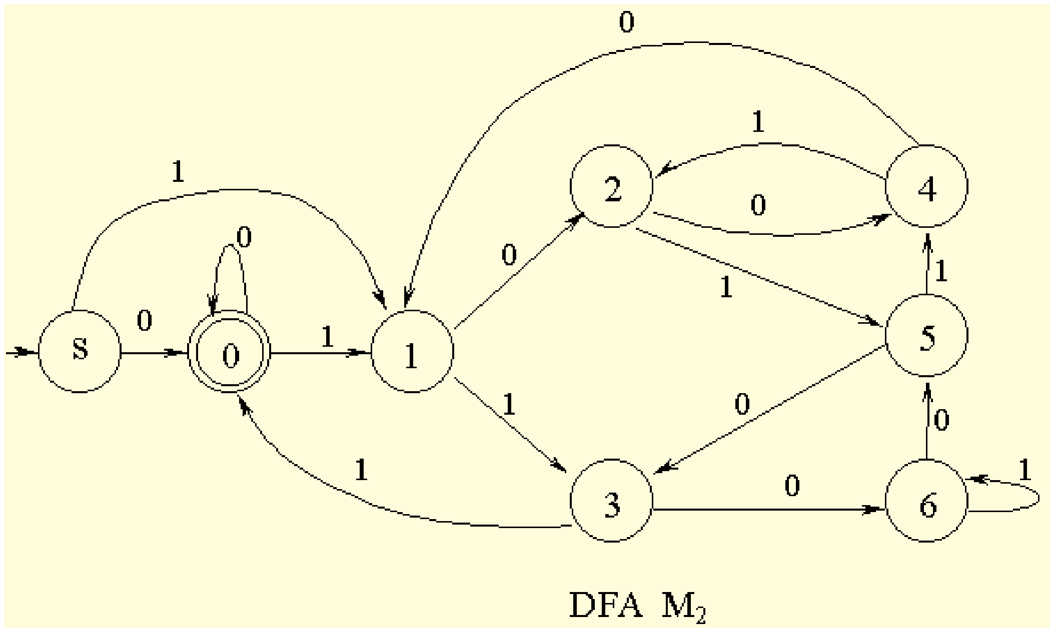


NFA M_1

verify this for each of the arrows. For instance, from state 1, if we see 0, then we go to state 2. This is because if the current string w has a binary value of $7m+1$, then the new string $w0$ has value $14m+2$ which is congruent to 2 mod 7.

COMMENTS: 1. Most student's solution is slightly defective in that their dfa accepts the empty string, which does not represent any binary number. 2. Some students solve a slightly different problem, by assuming that the dfa reads the binary number from the least significant bit (lsb) instead of the msb. We do not take off any points for this, but your dfa will have more states that the above.

SOLUTION: (iii) Use the pumping lemma to prove the language A is not regular. Assume to the contrary that the language A is regular. Let p be the pumping length given by the pumping lemma. Choose s to be the string $0^p10^m10^{p+m}$. Because s is a member of A and s has length more than p, the pumping lemma guarantees that s can be split into three pieces, $s = xyz$, where for any $i \geq 1$, the string xy^iz is in A. By conditin 3, y consists only of 0's, let $y = 0^k, k > 0$. The string $xy^2z = 0^{p+k}10^m10^{p+m}$ can not be a member of A, as $p+k+m \neq p+m$ for any $k > 0$. Thus we obtain a contradiction.



SOLUTION: (iv) Use the pumping lemma to prove the language B is not regular. Assume to the contrary that B is regular. Let p be the pumping length given by the pumping lemma. Choose s to be the string 0^p1^p. Because s is a member of B and s has length more than p, the pumping lemma guarantees that s can be split into three pieces, $s = xyz$, where for any $i \geq 0$ the string xy^iz is in B. By condition 3, y consists only 0's, let it be 0^k where $k > 0$. The string $xy^2z = 0^{p+k}1^p$ can not be a member of B. Thus we obtain a contradiction.

5. (20+20 Points) A **letter homomorphism** is a function $h : \Sigma \to \Gamma^*$ where Σ, Γ are alphabets. We can extend h to the function

$$h : \Sigma^* \to \Gamma^*$$

in the natural way, so that $h(vw) = h(v)h(w)$ for all $v, w \in \Sigma^*$. In particular, $h(\epsilon) = \epsilon$. If $A \subseteq \Sigma^*$, then define $h[A] = \{h(w) : w \in A\} \subseteq \Gamma^*$. Show the following.
(i) If $A \subseteq \Sigma^*$ is regular, so is $h[A]$.
(ii) If $B \subseteq \Gamma^*$ is regular, so is that $h^{-1}[B] = \{w \in \Sigma^* : h(w) \in B\}$. HINT: start from a dfa M for B and construct one that, when it reads an input symbol a, tries to simulate M on $h(a)$.

SOLUTION: (i) If A is regular, then some regular expression R describes it. We construct a regular expression R' from R by replacing each symbol $a \in \Sigma$ in R with the string $h(a)$. We must now show $L(R')$ describes $h[L(R)]$.

BASIS: (a) If $R = \emptyset$ which means A and $h[A]$ are empty sets, then $R' = \emptyset$, so $h[A] = L(R')$.

(b) If $R = \epsilon$ which means $A = \{\epsilon\}$, and $h[A] = \{h(\epsilon)\} = \{\epsilon\}$, then $R' = \epsilon$, so $h[A] = L(R')$.

(c) If $R = a$ where $a \in \Sigma$, which means $A = \{a\}, h[A] = \{h(a)\}$, then $R' = h(a)$, so $h[A] = L(R')$.

INDUCTION: Suppose R, R_1, R_2 are regular expressions, and R', R_1', R_2' are their transformations as above. Assume that inductively, $h[L(R_i)] = L(R_i')$ for $i = 1, 2$. If R is constructed from R_1, R_2 using one of the three regular operators, we will now show that $h[L(R)] = L(R')$.

a) If $R = R_1^*$ then $R' = R_1'^*$. Then

$$\begin{aligned}
w \in h[L(R)] \quad &\Leftrightarrow \quad w = h(w_1), w_1 \in L(R) = L(R_1^*) \\
&\Leftrightarrow \quad w = h(w_1), w_1 = u_1 u_2 \cdots u_m, u_j \in L(R_1)(\forall j) \\
&\Leftrightarrow \quad w = h(u_1)h(u_2)\cdots h(u_m), u_j \in L(R_1)(\forall j) \\
&\Leftrightarrow \quad w = h(u_1)h(u_2)\cdots h(u_m), h(u_j) \in h[L(R_1)] = L(R_1')(\forall j) \\
&\Leftrightarrow \quad w \in L((R_1')^*) = L(R').
\end{aligned}$$

b) If $R = R_1 + R_2$ then $R' = R_1' + R_2'$. Suppose $w \in h[L(R)]$. Then $w \in h[L(R_i)]$ for some $i = 1, 2$. Hence $w \in L(R_i')$, and therefore $w \in L(R')$. Conversely, if $w \in L(R')$ then $w \in L(R_i')$ for some $i = 1, 2$. Then $w \subseteq h[L(R_i)] \subseteq h[L(R)]$.

c) If $R = R_1 R_2$ then $R' = R_1' R_2'$. This is somewhat similar to b), and is omitted here.

SOLUTION: (ii) If B is regular, then there is a DFA $M = (Q, \Gamma, \delta, q_0, F)$ that accepts B. We construct DFA

$$M' = (Q', \Sigma, \delta', q_0, F')$$

such that $Q = Q', F = F'$ and $\delta'(q, a) = \delta^*(q, h(a))$ for each $q \in Q, a \in \Sigma$.

We show that $\delta'^*(q, w) = \delta^*(q, h(w))$ by induction on $|w|$: Basis: If $|w| = 0$, $\delta'(q, \epsilon) = q = \delta(q, \epsilon) = \delta(q, h(\epsilon))$.

Induction: Let $w = w'a$ for some $a \in \Sigma$. Then

$$\begin{aligned}
\delta'^*(q, w) &= \delta'(\delta'^*(q, w'), a) \\
&= \delta'(\delta^*(q, h(w')), a), \qquad \text{(by induction)} \\
&= \delta(\delta^*(q, h(w')), h(a)), \qquad \text{(by definition of } \delta') \\
&= \delta^*(q, h(w')h(a)) \\
&= \delta^*(q, h(w'a)) = \delta^*(q, h(w))
\end{aligned}$$

as desired.

Finally, let us show that $L(M') = h^{-1}[B]$: $w \in L(M')$ iff $\delta'^*(q_0, w) \in F$ iff $\delta^*(q_0, h(w)) \in F$, iff $h(w) \in B$, iff $w \in h^{-1}(B)$.