
**Solution.**

(a) \( L = \{0^n1^n2^n \mid n \geq 0\} \): Let \( y = 0 \) in the statement of the NPL. Then let \( m, n \) be distinct positive integers, and let \( z = 1^m2^m \). We have \( 0^mz = 0^m1^m2^m \in L \), but \( 0^nz = 0^n1^m2^m \notin L \). By the NPL, \( L \) is not regular.

(b) \( L = \{www \mid w \in \{a,b\}^*\} \). Let \( y = a \) in the NPL, and let \( m, n \) be distinct. Choose \( z = ba^mba^mb \). Then \( y^n \in L \), but \( a^mba^mb \notin L \). So \( L \) is not regular.

(c) \( L = \{a^{2^n} \mid n \geq 0\} \). Let \( y = a \), and let \( m < n \) be distinct. Choose an integer \( k \) such that \( 2^k + (n - m) \) is not a power of 2. This is possible since the gaps between successive powers of 2 get as large as you wish. Then \( a^m a^{2^k-m} \in L \), but \( a^m a^{2^k-m} = a^{k+n-m} \notin L \).

2. Refer to Problem 4 on HW 2. Prove that the class of regular languages is closed under reversal. In other words, if \( L \) is regular, then \( L^R \) is regular. Do this using regular expressions – you should give an inductive definition of the expression \( \alpha^R \) using the inductive definition of the regular expression \( \alpha \). Then prove using induction on expressions that \( L(\alpha^R) = (L(\alpha))^R \).

**Solution.** The inductive definition of the expression \( \alpha^R \) is the following.

\[
\begin{align*}
\emptyset^R &= \emptyset, \\
a^R &= a, \\
e^R &= e, \\
(\alpha \beta)^R &= \beta^R \alpha^R, \\
(\alpha \cup \beta)^R &= \alpha^R \cup \beta^R, \\
(\alpha^*)^R &= (\alpha^R)^*. \\
\end{align*}
\]

The proof that this construction works involves the following facts about languages in general:

\[
\begin{align*}
(L_1 L_2)^R &= L_2^R L_1^R, & (1) \\
(L_1 \cup L_2)^R &= L_1^R \cup L_2^R, & (2) \\
(L^*)^R &= (L^R)^*. & (3)
\end{align*}
\]

I’ll just prove (3). Using (1) and induction on \( k \) it follows that for any \( k \), \( (L^k)^R = (L^R)^k \).

Therefore, using (2) and its clear generalization to infinite unions,

\[
(L^*)^R = \bigcup_{k \geq 0} (L^k)^R = \bigcup_{k \geq 0} (L^R)^k = (L^R)^*.
\]

Now using (1), (2), and (3), we prove by induction on regular expressions that \( L(\alpha^R) = (L(\alpha))^R \).

I will illustrate just one base case:

\[
L(\alpha^R) = L(\alpha) = \{a\} = \{a\}^R = L(\alpha)^R.
\]

Assume that \( L(\alpha^R) = (L(\alpha))^R \) and the same equation for \( \beta \) Then

\[
L((\alpha \beta)^R) = L(\beta^R \alpha^R) = L(\beta^R)^R L(\alpha^R) = (L(\alpha) L(\beta))^R \text{ (by (1))} = L(\alpha \beta)^R.
\]

This is the inductive assertion for \( \alpha \beta \). The other 2 cases work the same way using (2) and (3) respectively.

3. Do problem 1.16 in Sipser using the method involving Arden’s lemma from class.

**Solution (part b).** From the state diagram, letting \( X_1, X_2, \) and \( X_3 \) be the languages accepted starting in 1,2,3, respectively, we have the equations:

\[
\begin{align*}
X_1 &= (a \cup b) X_2 \cup e; \\
X_2 &= aX_2 \cup bX_3; \\
X_3 &= aX_1 \cup bX_2 \cup e.
\end{align*}
\]
There is a choice about which variables to solve for using Arden’s Lemma. I started by solving for $X_2$, because it is the shortest equation. This gives

$$X_2 = a^*bX_3.$$ 

Substituting this into the first equation for $X_1$; you get

$$X_1 = (a \cup b)(a^*bX_3) \cup e.$$ 

Then you can substitute the solutions in the previous 2 equations into the third equation, getting

$$X_3 = a((a \cup b)(a^*bX_3) \cup e) \cup b(a^*bX_3) \cup e.$$ 

Now simplify this equation for $X_3$ so you can use Arden one more time. You get

$$X_3 = a((a \cup b)(a^*bX_3) \cup e) \cup b(a^*bX_3) \cup e$$

This lets you solve for $X_3$ using Arden:

$$X_3 = ((a(a \cup b)(a^*b) \cup ba^*b))^*(a \cup e).$$

But now you need the expression for $X_1$ since 1 is the start state. You can get this via

$$X_2 = a^*bX_3 = a^*b((a(a \cup b)(a^*b) \cup ba^*b))^*(a \cup e)).$$

Finally

$$X_1 = (a \cup b)X_2 \cup e = (a \cup b)(a^*b((a(a \cup b)(a^*b) \cup ba^*b))^*(a \cup e))) \cup e.$$ 

This is a very complicated expression, but it doesn’t matter. It doesn’t even matter too much that there’s a mistake in the algebra, as long as you see how to eliminate variables.

4. Use the partition refinement algorithm to find the minimum-size machine equivalent to the one below.
**Solution.** Here is a picture of the successive refinements:

![Diagram](image)

- **Partition 0:**
  - $q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4\rightarrow q_5$
  - $B_0 \rightarrow B_1$

- **B0 does not split.**
- **q1 splits from q2, q3, and q4.**
- **q2 splits from q3 and q4.**
- **q3 splits from q4.**

- **Partition 1:**

- **q3 splits from q4.**

- **Partition 2:**

- **No further need to refine.**

- **Partition 2 represents the identity relation; all states are distinguishable, and the given machine is minimal.**