Problem 5

We give a detailed induction argument for (a), re. key ideas
for other induction arguments for (b), (c), or (d).

(a) Prove \( \forall n \geq 0 \) \( |x_n| = |y_n| \)

Induction on \( n \).

Base: \( n = 0 \): Prove \( P(0) \), i.e., \( |x_0| = |y_0| \).

The basis of the induction definition of \( x_n \) and \( y_n \) for \( n = 0 \),
gives \( x_0 = 0 \) and \( y_0 = 1 \).

So \( |x_0| = 1 \) and \( |y_0| = 1 \).

Induction step: Prove \( \forall n \geq 0 \) \( P(n) \Rightarrow P(n+1) \).

Let \( n \geq 0 \) be arbitrary.

Assume \( P(0), P(1), \ldots, P(n) \).

To prove \( P(n+1) \), i.e., \( |x_{n+1}| = |y_{n+1}| \),
we consider \( x_{n+1} = x_n y_n \) by inductive definition.

So \( |x_{n+1}| = |x_n y_n| \)

\[ = |x_n| \cdot |y_n| \leq \text{property of } 0 \text{ and } 1 \]

on strings.

If strings \( u, v \)

\( |uv| = |u| + |v| \)

Similarly \( |y_{n+1}| = |y_n + x_n| \)

\[ = |x_n| + |y_n| \]

\[ \text{commutativity of } + \]

Hence, \( |x_{n+1}| = |y_{n+1}| \) as desired.

Hence we applied any induction hypothesis
\( P(0), P(1), \ldots, P(n) \) ?

This completes the induction step.

By induction, we have \( \forall n \geq 0 \) \( P(n) \).
(b) \( V_n > 0 \) in all \( n \) differ in every position.

Idea: In the induction step:

\[ V_n > 0 \land V_1 > 0 \quad \cdots \quad \land V_n \Rightarrow V_{n+1} \cdot \]

To prove \( P(n) \), i.e., \( x_{n+1} \neq y_{n+1} \) in every position, we consider \( x_{n+1} \) versus \( y_{n+1} \):

\[ x_{n+1} = x_{n+1} \] by inductive definitions of

\[ y_{n+1} = y_{n+1} \quad \text{for } n = 0, 1, \ldots \]

Now,

\[ x_{n} = x_{n}, \quad y_{n} = y_{n} \]

\[ \text{By (a), } V_{m} > 0 \]

\[ |x_{m}| = |y_{m}| \]

\[ \text{in particular, } |x_{m}| = |y_{m}| \]

\[ \text{By induction hypothesis } P(n) :) \]

\[ x_{n+1} \neq y_{n+1} \] in every position.

\[ \text{(Same conclusion)} \]

Hence, \( x_{n+1} \neq y_{n+1} \) in every position.
(c) Idea:
The strings $x_n$ and $y_n$ are clearly palindromes.
For $n \geq 0$,
\[ x_{2(n+1)} = x_{n+1} = x_{2n+1} \]
\[ = x_n y_n z_n x_n \]
why do we express $x_{2(n+1)}$ in terms of $x_n, y_n, z_n$?

The above expression of $x_{2(n+1)}$ allows us to apply induction hypothesis to find that
\[ x_{2(n+1)} = x_{2(n+1)} \]
so $x_{2(n+1)}$ is a palindrome.

Similar argument holds for $y_{2(n+1)}$.

(d) Idea:
Clearly, for every $n \geq 2$,
$x_n$ begins with 01 or ends with 10,
and $y_n$ begins with 10 or ends with 01.

Use the above observation in the induction step
do show that neither $x_n$ nor $y_n$ contains
000 or 111.
Problem 6

We show a stronger statement:

\( \forall x, y \in \mathbb{Z}^* \ (xy = yx \iff \exists d \in \mathbb{Z}^* \ x = x_d \ y = y_d \text{ for some } j, k \geq 0) \)

\((\Rightarrow)\) is obvious.

\((\Leftarrow)\) Assume that \(xy = yx\).

Notice that, if there exist \(x \in \mathbb{Z}^*\) such that \(x = x_j x_k\) and \(y = y_j y_k\) for some \(j, k \geq 0\), then obviously \(|x| \) divides \(|x_j|\) and \(|y| \) divides \(|y_j|\).

With this observation, we let \(d = \gcd(|x|, |y|)\).

Then, we write \(x = x_1 x_2 \cdots x_p\) and \(y = y_1 y_2 \cdots y_q\) where all \(x_i\)s and \(y_i\)s are \(\mathbb{Z}^*\) with \(j \) length \(d\), and \(p\) and \(q\) have no common factor (\(i.e., \ p = \frac{|x|}{d} \) and \(q = \frac{|y|}{d}\)).

We want to show below that

\[x_1 = x_2 = \cdots = x_p = y_1 = y_2 = \cdots = y_q\]

Here is the trick!

Since \(xy = yx\), we must have \(z^{yp} = y^{xp}\) (by repeatedly transposing \(x\) and \(y\)).
Also observe that both $x^2 y^p \cdot x y^q x^2$ have the same length $2pqd$.

$$12^2 y^p = q_x x_1 + p(y_1
 = q(p \cdot d) + p(q \cdot d)
 = 2pqd$$

and the prefixes of length $pqd$ of $x^2 y^p$ and $y^q x^2$ are $x^2$ and $y^p$, respectively.

As $x^2 y^p = y^q x^2$ (see (1) last page)

we by (3) this page, the two strings $x^2$ and $y^p$ are equal.

Now $x^2 = (x_1 x_2 \cdots x_p)^2$.

This means that within the string $x^2$, the substring $x_i$ occurs starting at positions

$$1, pd+1, 2pd+1, \ldots, (q-1)pd+1.$$  

In the substring $y^p$, the substring of length $d$ starting at position $ipd+1$ is

$$y_{ip},$$  

where $i = ip \pmod{q} + 1$.

Since $p$ and $q$ have no common factor, we can check that all the indices $i, i_2, \ldots, i_{q-1}$ are distinct.

This means, however, that all the strings

$$y_{i_1}, y_{i_2}, \ldots, y_{i_{q-1}}$$

are the same, say $\lambda$, and this implies that all the $\lambda$s are equal to $\lambda$ as well.
Let $G$ be any graph with $n$ nodes where $n \geq 2$. The degree of every node in $G$ is one of the $n$ possible values from 0 to $n - 1$. We would like to use the pigeon hole principle to show that two of these values must be the same, but number of possible values is too great. However, not all of the values can occur in the same graph because a node of degree 0 cannot coexist with a node of degree $n - 1$. Hence $G$ can exhibit at most $n - 1$ degree values among its $n$ nodes, so two of the values must be the same.

Observe that $D \subseteq b^*a^*$ because $D$ doesn't contain strings that have $ab$ as a substring. Hence $D$ is generated by the regular expression $(aa)^*b(bb)^*$. From this description, finding the DFA for $D$ is more easily done.

By simulating binary division, we create a DFA $M$ with $n$ states that recognizes $C_n$. $M$ has $n$ states which keep track of the $n$ possible remainders of the division process. The start state is the only accept state and corresponds to remainder 0.

The input string is fed into $M$ starting from the most significant bit. For each input bit, $M$ doubles the remainder that its current state records, and then adds the input bit. Its new state is the sum modulo $n$. We double the remainder because that corresponds to the left shift of the computed remainder in the long division algorithm. If an input string ends at the accept state (corresponding to remainder 0), the binary number has no remainder on division by $n$ and is therefore a member of $C_n$.

The formal definition of $M$ is $(\{q_0, \ldots, q_{n-1}\}, \{0, 1\}, \delta, q_0, \{q_0\})$. For each $q_i \in Q$ and $b \in \{0, 1\}$, define $\delta(q_i, b) = q_j$ where $j = (2i + b \mod n)$.

We construct a DFA which alternately simulates the DFAs for $A$ and $B$, one step at a time. The new DFA keeps track of which DFA is being simulated. Let $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be DFAs for $A$ and $B$. We construct the following DFA $M = (Q, \Sigma, \delta, s_0, F)$ for the perfect shuffle of $A$ and $B$.

i) $Q = Q_1 \times Q_2 \times \{1, 2\}$.

ii) For $q_1 \in Q_1$, $q_2 \in Q_2$, $b \in \{1, 2\}$, and $a \in \Sigma$:

$$\delta((q_1, q_2, b), a) = \begin{cases} (\delta_1(q_1, a), q_2, 2) & b = 1 \\ (q_1, \delta_2(q_2, a), 1) & b = 2. \end{cases}$$

iii) $s_0 = (s_1, s_2, 1)$.

iv) $F = \{(q_1, q_2, 1) | q_1 \in F_1 \text{ and } q_2 \in F_2\}$. 