Type your answers to the following questions and submit a PDF file to Blackboard. One page per problem.

Problem 1. [5pts] Consider the sequence $\{A_n\}_{n=0}^{\infty}$ where each element of the sequence is a set A_n, defined by $A_0 = \varnothing$ and for $n \geq 1$, $A_n = \{|A_i| : 0 \leq i < n\}$. Prove that $|A_n| = n$ for all $n \geq 0$.

Proof. We will use strong induction.

Case $n = 0$: $|\varnothing| = 0$.

(Strong Induction Hypothesis) Let $N > 0$ and suppose that for all n where $0 \leq n < N$, we have $|A_n| = n$.

Case N: $A_N = \{|A_n| : 0 \leq n < N\}$. Thus, by the Strong Induction Hypothesis, $A_N = \{0, 1, 2, \ldots, N-1\}$. Since A_N contains N distinct elements, $|A_N| = N$. (Alternatively, $A_N = A_{N-1} \cup \{|A_{N-1}|\}$, so $|A_N| = |A_{N-1}| + 1 = (N-1) + 1$. This does not require strong induction.) $\square$

Problem 2. [5pts] Define a sequence $\{c_n\}_{n=0}^{\infty}$ by $c_0 = 1$ and for all $n \geq 1$, let $c_n = \sum_{i=0}^{n-1} \frac{c_i}{2^{n-i}}$. Prove that for all $n \geq 1$, $c_n = \frac{1}{2}$. [Note: There are two ways to prove this statement. One is by strong induction. The other is to use weak induction after proving an equivalent recurrence relation. Either will be accepted.]

Proof. We use strong induction. Note that we must prove this for $n \geq 1$, so our base case is $n = 1$.

Case $n = 1$: $c_1 = \frac{1}{2}c_0 = \frac{1}{2}$.

(Strong Induction Hypothesis) Let $N > 1$ and suppose that for all n where $1 \leq n < N$ we have $c_n = \frac{1}{2}$.

Case N: By the recurrence relation, $c_N = \sum_{i=0}^{N-1} \frac{c_i}{2^{N-i}} = \frac{1}{2^N} \sum_{i=0}^{N-1} c_i 2^i = \frac{1}{2^N} \left[c_0 + \sum_{i=1}^{N-1} c_i 2^i\right]$. By the strong induction hypothesis, $c_i = \frac{1}{2}$ when $1 \leq i < N$, and $c_0 = 1$. Thus,

$$\begin{aligned}
c_N &= \frac{1}{2^N}\left[1 + \sum_{i=1}^{N-1} 2^{i-1}\right] \\
&= \frac{1}{2^N}\left[1 + \sum_{j=0}^{N-2} 2^j\right] && \text{(j = i-1)} \\
&= \frac{1}{2^N}\left[1 + \frac{2^{N-1} - 1}{2 - 1}\right] \\
&= \frac{1}{2^N}\left[2^{N-1}\right] \\
&= \frac{2^{N-1}}{2^N} = \frac{1}{2}.
\end{aligned}$$

Therefore, by strong induction $c_n = \frac{1}{2}$ for all $n \geq 1$. $\square$

[Alternate Solution]

Proof. Observe that for $n \geq 2$, $c_n = \sum_{i=0}^{n-1} \frac{c_i}{2^{n-i}} = \frac{1}{2^n} \sum_{i=0}^{n-1} c_i 2^i$. Since $c_{n-1} = \frac{1}{2^{n-1}} \sum_{i=0}^{n-2} c_i 2^i$ by this formula, we have

$$c_n = \frac{1}{2^n} \sum_{i=0}^{n-1} c_i 2^i = \frac{1}{2^n}(2^{n-1}c_{n-1} + \sum_{i=0}^{n-2} c_i 2^i) = \frac{1}{2^n}(2^{n-1}c_{n-1} + 2^{n-1}c_{n-1}) = c_{n-1}.$$

We now use this simplified recurrence and induction.

Case $n = 1$: $c_1 = \frac{1}{2}$.

(Induction Hypothesis) Suppose $c_n = \frac{1}{2}$.

Case $n + 1$: By the recurrence above, $c_{n+1} = c_n = \frac{1}{2}$. $\square$

Problem 3. [10pts] The *merge sort algorithm* sorts a list of n numbers $x_1, \ldots, x_n$. The algorithm works by first testing if $n = 1$, and if so does nothing. Otherwise, the algorithm recursively calls itself on the first $\lfloor n/2 \rfloor$ entries, and then calls itself on the last $\lceil n/2 \rceil$ entries, then "shuffles" them together by iterating through the two parts, selecting the minimum elements from each part until creating a sorted list of n entries. If t_n is the time it takes to run the merge sort algorithm on a list of n numbers, then $t_1 = 1$ (only need to test one operation), and (roughly)

$$t_n = Ct_{\lceil n/2 \rceil} + n$$

Using this recurrence relation and strong induction, prove that $t_n \leq Cn \log_2(n + 1)$ for all $n \geq 1$.

(With using the ceiling function instead of the floor function and $C = 2$ this is false!)

Proof. We use strong induction on n to prove that $t_n \leq 2n \log_2(n + 1)$. We start with some base cases (to explore the value of C that is needed!).

Case $n = 1$: $t_1 = 1 \leq C \cdot 1 \cdot 1 = Cn \log_2(n + 1)$. ($C \geq 1$)
Case $n = 2$: $t_2 = 2 \cdot 1 + 2 = 4 \leq C \cdot 2 \cdot \log_2 3 = Cn \log_2(n + 1)$. ($C \geq 2$)
Case $n = 3$: $t_3 = 2t_2 + 3 = 11 \leq C \cdot 3 \cdot 2 = 6C = Cn \log_2(n + 1)$. ($C \geq 2$)
Case $n = 4$: $t_4 = 2t_2 + 4 = 12 \leq C \cdot 3 \cdot \log_2 5 = Cn \log_2(n + 1) \approx 6.9C$. ($C \geq 2$)
Case $n = 5$: $t_5 = 2t_3 + 5 = 27 \leq C \cdot 5 \cdot \log_2 6 = Cn \log_2(n + 1) \approx 12.95C$. ($C \geq 3$) [THIS is where the result is false for $C = 2$.]
Case $n = 6$: $t_6 = 2t_3 + 6 = 28 \leq C \cdot 6 \cdot \log_2 7 = Cn \log_2(n + 1) \approx 16.85C$. ($C \geq 2$)

(Strong Induction Hypothesis) Let $N > 6$ and suppose that for all n where $1 \leq n < N$ we have that $t_n \leq 2n \log_2(n + 1)$.

Case N: Consider t_N. Let $n = \lceil N/2 \rceil < N$ and note that $2n \leq N + 1$ and $2(n + 1) \leq N + 3$. By the strong induction hypothesis, $t_n \leq Cn \log_2(n + 1)$. By the recurrence relation, $t_N = 2t_n + N$, and so

$$\begin{aligned}
t_N &\leq 2(Cn \log_2(n + 1)) + N \\
&\leq C(N + 1) \log_2(n + 1) + N && \text{Since } 2n \leq N + 1 \\
&\leq CN \log_2 (n + 1) + \frac{C}{2}N && \text{Since } \log_2(n + 1) < \frac{1}{2}N \text{ and } C \geq 3 \\
&\leq CN \log_2 \left(\frac{N + 3}{2}\right) + \frac{C}{2}N && \text{Since } 2(n + 1) \leq N + 3 \\
&= CN \left(\log_2(N + 3) - \log_2 2\right) + \frac{C}{2}N \\
&= CN \log_2(N + 3) - CN + \frac{C}{2}N \\
&\leq CN \log_2(N + 3) - \frac{C}{2}N \\
&= CN \left(\log_2(N + 1) + \log_2 \left(1 + \frac{2}{N + 1}\right)\right) - \frac{C}{2}N && \text{by Log Identity} \\
&= CN \log_2(N + 1) + C\frac{1}{3} - \frac{C}{2}N && \text{Since } \log_2(1 + \frac{2}{N + 1}) < \frac{1}{3} \text{ when } N \geq 7. \\
&= CN \log_2(N + 1) - \frac{C}{6}N \\
&\leq CN \log_2(N + 1).
\end{aligned}$$

Notice that the steps above work when $C \geq 3$, so therefore $t_n \leq 3n \log_2(n + 1)$ for all $n \geq 1$. ☐

The proof above used logarithmic identities. See `http://en.wikipedia.org/wiki/List_of_logarithmic_identities`

Problem 4. [10pts] Let $S = (\mathbb{Q}, T, 0)$ be a state machine where the states are rational numbers ($\mathbb{Q}$) and the transitions are of the form $x \to x + 1$, $x \to 3x$, and $x \to \frac{x}{2}$, and the initial state is 0. Prove that no matter what transitions are used, the state $\frac{1}{3}$ will never be reached. [For 2 extra points: Prove that for any $\varepsilon > 0$, there is a sequence of transitions such that a state x is reachable where $|x - \frac{1}{3}| < \varepsilon$.]

Proof. We will prove that starting at the initial state $x = 0 = \frac{0}{1}$, the property that if $\frac{p}{q}$ is a visited rational number with p and q having no common factors, then q is a power of 2.

Initial State: $x = 0 = \frac{0}{1}$. $1 = 2^0$.

Now suppose that $\frac{p}{q}$ is a state where $q = 2^k$ for some $k \geq 0$. We will show that any state reachable from $\frac{p}{q}$ has the invariant property.

Consider $\frac{p}{q} + 1$, which equals $\frac{p+q}{q}$. Since p and q have no common factors, $p + q$ and q have no common factors (if f was a common factor, then $(p + q) = tf$ for some integer t and $q = rf$ for some integer r, but then $p = tf - rf = (t - r)f$ and hence f would be a common factor between p and q.) Therefore, the representation of $\frac{p+q}{q}$ with no common factors has the denominator a power of 2.

Consider $3\frac{p}{q}$, which equals $\frac{3p}{q}$. Since p and q have no common factors, and q is a power of 2, $3p$ and q have no common factors. Therefore, the representation of $\frac{3p}{q}$ has the denominator a power of 2.

Consider $\frac{p}{2q}$. Since q is a power of 2, $q = 2^k$ for some nonnegative integer k. If $k \geq 1$, then since p and q have no common factors, p is an odd integer. Therefore, $\frac{p}{2q} = \frac{p}{2^{k+1}}$ is the representation of this rational number with no common factors and the denominator is a power of 2. If $k = 0$, then $\frac{p}{2q} = \frac{p}{2}$. If p is odd, then p and 2 have no common factors and the denominator is a power of 2. If p is even, then $p = 2t$ for some integer t and the representation $\frac{p}{2} = \frac{t}{1}$ is the representation with no common factors, and the denominator is a power of 2.

Since the initial state satisfies the invariant property, all reachable states have the invariant property. However, the fraction $\frac{1}{3}$ does not satisfy the invariant property, so it is not a reachable state. $\square$

[Bonus]

Proof. Fix $\varepsilon > 0$. Let n be an integer large enough that $\frac{1}{2^n} < \varepsilon$. Then, let $m = \lfloor 2^{n+1}/3 \rfloor$.

Starting at the $x = 0$ state, use the $x \to x + 1$ transition m times to reach the state $\frac{m}{1}$. Then, use the transition $x \to \frac{x}{2}$ $n + 1$ times to reach the state $\frac{m}{2^{n+1}}$. Since $m = \lfloor 2^{n+1}/3 \rfloor$, the difference $\frac{m}{2^{n+1}} - \frac{1}{3}$ has absolute value strictly less than $\frac{1}{2^n}$, which is less than ε. $\square$

Problem 5. [15pts] Let $\Sigma = \{0, 1\}$ (here we say Σ is the name of a set, not the summation notation). For $k \geq 0$, the set Σ^k is the set of k-tuples where every entry comes from Σ. The set Σ^* is equal to $\cup_{k=0}^{\infty}\Sigma^k$, the set of all finite binary strings (note that for every finite binary string $\mathbf{x}$ of length k, $\mathbf{x} \in \Sigma^k \subset \Sigma^*$). We will denote a string $\mathbf{x} = (x_1, x_2, \dots, x_k)$ as $x_1x_2\dots x_k$. Define a state machine $(\Sigma^*, T, 0)$ where the initial state is the string 0, and the transitions are of the form

$$x_1x_2\dots x_{k-1}1 \to x_1x_2\dots x_{k-1}10 \tag{1}$$
$$x_1x_2\dots x_{k-1}0 \to x_1x_2\dots x_{k-1}00 \tag{2}$$
$$x_1x_2\dots x_{k-1}0 \to x_1x_2\dots x_{k-1}01 \tag{3}$$
$$x_1x_2\dots x_{k-1}x_k \to x_kx_{k-1}\dots x_2x_1x_1x_2\dots x_{k-1}x_k. \tag{4}$$

Note that there are four types of transitions: (1) Take a string ending in 1 and add a 0, (2) take a string ending in 0 and add a 0, (3) take a string ending in 0 and add a 1, (4) take a string and add its reversal to the beginning (turning it into a palindrome).

a. Prove that the string 10011001 is reachable from the initial string 0.

Proof. Starting with the initial state 0, we use the following transition types:

$$0 \overset{(3)}{\to} 01 \overset{(4)}{\to} 1001 \overset{(4)}{\to} 10011001.$$ □

b. Prove that the string 0110110 is not reachable from the initial string 0.

Proof. Note that every transition increases the length of a string by at least one, so there are a finite number of cases to check, as we can ask: "What strings of length at most 7 are reachable (in at most 6 steps)?" Notice that if $\mathbf{x}$ is a state that is visited after the initial state, then $\mathbf{x}$ appears as a substring in all of the remaining states.

In one step, we could transition $0 \to 00$, $0 \to 01$, or $0 \to 00$ (so 00 appears twice here). However, the string 00 does not appear in the goal string 0110110, so we will not consider that state. Thus, the first step to take must have been $0 \to 01$.

In the second step, we could transition $01 \to 010$ or $01 \to 1001$. However, neither string 010 or 1001 appears as a substring of 0110110, so this state is not reachable. □

c. Prove that a reachable string can never contain three consecutive 1's.

Proof. We will prove the following invariant property: If the string 11 appears in a reachable state, then it appears within a substring 0110. (Observe that this invariant property will imply that 111 is not a substring, as it contains a substring 11 that is not within a substring 0110.)

Initial State: the string 0 does not contain a string 11, so the property holds vacuously.

Suppose that $x_1 \dots x_k$ is a state where every 11 substring is contained in a 0110 substring. Consider the transitions from this state.

If we apply a type 1 transition, then $x_k = 1$ and we add $x_{k+1} = 0$ to the end of the string. Thus, we did not create a new 11 substring, so all 11 substrings in $x_1 \dots x_kx_{k+1}$ exist within the string $x_1 \dots x_k$ and hence the 11 substrings are contained in 0110 substrings.

If we apply a type 2 or 3 transition, then $x_k = 0$ and we add x_{k+1} to the end of the string. Thus, we did not create a new 11 substring, so all 11 substrings in $x_1 \dots x_k x_{k+1}$ exist within the string $x_1 \dots x_k$ and hence the 11 substrings are contained in 0110 substrings.

If we apply a type 4 transition, then we create the string $x_k x_{k-1} \dots x_2 x_1 x_1 x_2 \dots x_{k-1} x_k$. Suppose there is a 11 substring of this string. If it is of the form $x_i x_{i+1}$ for $i \geq 1$ (i.e. in the right half of the string), then this string appeared in the string $x_1 \dots x_k$ and is still contained in a 0110 substring. If it is of the form $x_{i+1} x_i$ for $i \geq 1$ (i.e. in the left half of the string), then this string appeared also in the string $x_1 \dots x_k$ (in the form $x_i x_{i+1}$) and is still contained in a 0110 substring. Now suppose that a 11 substring appears as $x_1 x_1$. This string appears as a substring of $x_2 x_1 x_1 x_2$. If $x_2 = 0$, then this is a 0110 substring as we want. Thus, if $x_2 = 1$, then $x_1 x_2$ is a 11 substring of $x_1 \dots x_k$ that is not in a substring of the form 0110, contradicting the property on the string $x_1 \dots x_k$.

Thus, our property is an invariant property and hence all reachable strings have this property. Any string containing 111 as a substring does not have this property. $\square$