Name: ________________________________

UW ID: ______________________________

Instructions:

- This is a simulated practice final. You will not be graded on your performance on this exam.
- This final was written to take 50 minutes. The real final will be an hour and 50 minutes.
- Nevertheless, please treat this as if it is a real exam. That means that you may not discuss with your neighbors, reference outside material, or use your devices during the next 50 minute period.
- If you get stuck on a problem, consider moving on and coming back later. In the actual exam, there will likely be opportunity for partial credit.
- There are 5 problems on this exam.
1. All the Machines! [15 points]
Let the alphabet be \( \Sigma = \{a, b\} \). Consider the language \( L = \{w \in \Sigma^* : \text{every } a \text{ has a } b \text{ two characters later}\} \).
In other words, \( L \) is the language of all strings in the alphabet \( a, b \) where after any \( a \), the character after the \( a \) can be anything, but the character after that one must be a \( b \).

Some strings in \( L \) include \( \varepsilon, abb, aabb, bbbabb \). Some strings not in \( L \) include \( a, ab, aab, ababb \). Notice that the last two characters of the string cannot be an \( a \).

(a) (5 points) Give a regular expression that represents \( L \).

Solution:
\[(b \cup abb \cup aabb)^*\]

(b) (5 points) Give a CFG that represents \( L \).

Solution:
\[S \rightarrow bS \mid aabbS \mid abbS \mid \varepsilon\]

(c) (5 points) Give a DFA that represents \( L \).

Solution:
2. Induction 1 [20 points]
Recall the recursive definition of a list of integers:

- \([\ ]\) is the empty list
- If \(L\) is a list and \(a\) is an integer, then \(a :: L\) is a list whose first element is \(a\), followed by the elements of \(L\).

Consider the following functions defined on lists:

\[
\begin{align*}
\text{len}([\ ] &= 0 \\
\text{len}(x :: L) &= 1 + \text{len}(L) \\
\text{inc}([\ ]) &= [] \\
\text{inc}(x :: L) &= (x + 1) :: \text{inc}(L) \\
\text{sum}([\ ]) &= 0 \\
\text{sum}(x :: L) &= x + \text{sum}(L)
\end{align*}
\]

Prove that for all lists \(L\), \(\text{sum(inc}(L)) = \text{sum}(L) + \text{len}(L)\).

**Solution:**
Let \(P(L)\) be "\(\text{sum(inc}(L)) = \text{sum}(L) + \text{len}(L)\)". We prove that \(P(L)\) is true for all lists \(L\) by structural induction.

**Base Case:** \(L = [\ ]\). Then:

\[
\begin{align*}
\text{sum}(\text{inc}([\ ])) &= \text{sum}([\ ]) & \text{Definition of inc} \\
&= 0 & \text{Definition of sum} \\
&= 0 + 0 & \text{Algebra} \\
&= \text{sum}([\ ] + \text{len}([\ ])) & \text{Definition of sum, len}
\end{align*}
\]

**Inductive Hypothesis:** Suppose that \(P(L)\) is true for an arbitrary list \(L\).

**Inductive Step:** We aim to show that \(P(x :: L)\) holds.

\[
\begin{align*}
\text{sum}(\text{inc}(x :: L)) &= \text{sum}((x + 1) :: \text{inc}(L)) & \text{Definition of inc} \\
&= (x + 1) + \text{sum}(\text{inc}(L)) & \text{Definition of sum} \\
&= (x + 1) + \text{sum}(L) + \text{len}(L) & \text{Inductive Hypothesis} \\
&= x + \text{sum}(L) + 1 + \text{len}(L) & \text{Algebra} \\
&= \text{sum}(x :: L) + \text{len}(x :: L) & \text{Definition of sum, len}
\end{align*}
\]

So \(P(x :: L)\) holds.

**Conclusion:** Thus \(P(L)\) holds for all lists \(L\) by structural induction.
3. Induction 2 [20 points]
Consider the following recursive definition of $a_n$:

\[
a_1 = 1 \\
a_2 = 1 \\
a_n = \frac{1}{2}(a_{n-1} + \frac{2}{a_{n-2}}) \quad \text{for } n > 2
\]

Prove that $1 \leq a_n \leq 2$ for all integers $n \geq 1$.

**Solution:**
Define $P(n)$ to be $1 \leq a_n \leq 2$. We prove $P(n)$ holds for all integers $n \geq 1$ by strong induction.

**Base Case** $P(1), P(2)$ Observe that $a_1 = a_2 = 1$, and $1 \leq 1 \leq 2$. So $P(1)$ and $P(2)$ hold.

**Inductive Hypothesis:** Suppose that $P(j)$ is true for all $1 \leq j \leq k$ for some arbitrary integer $k \geq 2$.

**Inductive Step:**

\[
a_{k+1} = \frac{1}{2}(a_k + \frac{2}{a_{k-1}}) \\
= \frac{a_k}{2} + \frac{1}{a_{k-1}} \\
\leq \frac{2}{2} + \frac{1}{a_{k-1}} \quad \text{By IH, since } a_k \leq 2 \\
\leq 1 + \frac{1}{1} \quad \text{By IH, since } a_{k-1} \geq 1, \text{ so } \frac{1}{a_{k-1}} \leq \frac{1}{1} \\
= 2
\]

\[
a_{k+1} = \frac{1}{2}(a_k + \frac{2}{a_{k-1}}) \\
= \frac{a_k}{2} + \frac{1}{a_{k-1}} \\
\geq \frac{1}{2} + \frac{1}{a_{k-1}} \quad \text{By IH, since } a_k \geq 1 \\
\geq \frac{1}{2} + \frac{1}{2} \quad \text{By IH, since } a_{k-1} \leq 2, \text{ so } \frac{1}{a_{k-1}} \geq \frac{1}{2} \\
= 1
\]

So $1 \leq a_{k+1} \leq 2$.

**Conclusion:** Thus we have proven $P(n)$ for all integers $n \geq 1$ by strong induction.
4. Modular Arithmetic [10 points]
(a) Prove or disprove: If \( a \equiv b \pmod{10} \), then \( a \equiv b \pmod{5} \). [5 points]

Solution:
True. Suppose that \( a \equiv b \pmod{10} \). Then \( 10 \mid (a - b) \). Then there exists some integer \( k \) such that \( a - b = 10k \) for some integer \( k \). In particular, \( a - b = 5(2k) \). Then \( 5 \mid (a - b) \). So \( a \equiv b \pmod{5} \).

(b) Prove or disprove: If \( a \equiv b \pmod{10} \), then \( a \equiv b \pmod{20} \). [5 points]

Solution:
False. For example, for \( a = 1 \) and \( b = 11 \). Then \( a \equiv b \pmod{10} \), but \( a \not\equiv b \pmod{20} \).
5. Irregularity [20 points]
Prove that the set of strings \( \{0^n10^n : n \geq 0\} \) is not regular.

Solution:
\( L = \{0^n10^n : n \geq 0\} \). Let \( D \) be an arbitrary DFA, and suppose for contradiction that \( D \) accepts \( L \). Consider \( S = \{0^n : n \geq 0\} \). Since \( S \) contains infinitely many strings and \( D \) has a finite number of states, two strings in \( S \) must end up in the same state. Say these strings are \( 0^i \) and \( 0^j \) for some \( i, j \geq 0 \) such that \( i \neq j \). Append the string \( 10^i \) to both of these strings. The two resulting strings are:

- \( a = 0^i10^i \) Note that \( a \in L \).
- \( b = 0^j10^i \) Note that \( b \notin L \), since \( i \neq j \).

Since \( a \) and \( b \) end up in the same state, but \( a \in L \) and \( b \notin L \), that state must be both an accept and reject state, which is a contradiction. Since \( D \) was arbitrary, there is no DFA that recognizes \( L \), so \( L \) is not regular.