NAME (PRINT): ______________________________

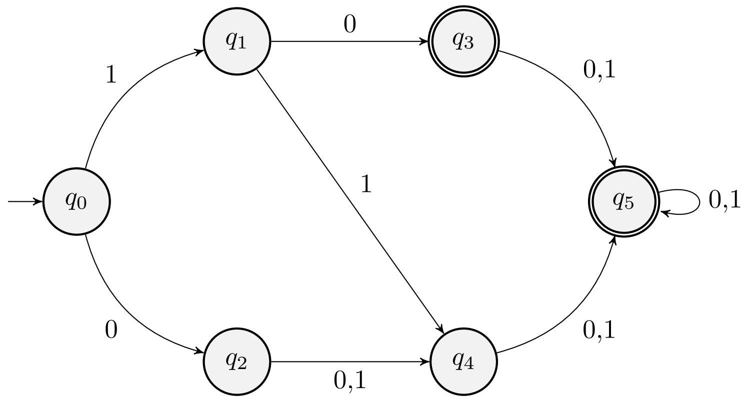
STUDENT NUMBER (PRINT): ______________________

**University of Toronto Mississauga**
**FALL 2024 MOCK FINAL EXAMINATION**
**Introduction to Theory Computation**
**Macho Man (m^2)**
**Duration - ~~3 hours~~ 10 minutes**
**Aids : お前のお母さん:**
https://youtu.be/3UHR_XWrGhc?si=kO03tfL7xwKDE3-o

This is a mock exam designed for studying CSC236. Any and all similarities with the Fall 2024 CSC236 final examination are purely coincidence.

This examination is meant to be done in 3 hours, but it is a little bit long, so do not worry about finishing. Part marks are very easy to get!

**Q1. (9 points)** Consider the DFA below:




a) (1 point) Describe the language accepted by the following DFA.

b) (3 points) Convert this DFA into a minimal NFA (i.e., there is no smaller NFA that accepts this language). Give a brief justification.

c) (2 points) Provide a DFA that accepts the language matched by $(a + ab)^*$.

d) (3 points) Prove the correctness of the DFA you provided.

**Q2. (10 points)** For each of the statements below, decide whether it is **true or false** and provide a proof justifying your answer.

a) (2 points) Let $\Sigma = \{0, 1\}$. The language $L = \{w \in \Sigma^* : |w| = 3\}$ is regular.

b) (2 points) Let $\Sigma = \{0, 1\}$. $L = \{1^{n^2} : n \in \mathbb{N}\}$ is a regular language.

c) (2 points) Let $L, M$ be regular languages. The language $L \cap M$ is regular.

d) (2 points) Let $\Sigma = \{a, b, c\}$. The language of all strings whose characters are alphabetically ordered is regular.

e) (2 points) Let $L, M$ be languages. If $L$ is not regular, then $L \cup M$ is not regular.

**Q3. (6 points)**

a) (2 points) State the CLRS version of master theorem. Define all variables and state their conditions.

b) (4 points) Let $f : \mathbb{N} \to \mathbb{R}$ be a nonnegative function. Prove that if $f \in \Theta(n^k)$ for some $k > 0$, then the regularity condition holds true.

**Q4. (7 points)** Consider the program below:

```
def binary_search(x: int, lst: list[int]):
    l ← 0, r ← len(lst) − 1
    while(r − l > 0):
        mid ← (l+r) // 2
        if(lst[mid] = x):
            return mid
        elif(lst[mid] < x):
            r ← mid
        else:
            l ← mid + 1
    return l
```

a) (2 points) State the preconditions and postconditions of this program.

b) (5 points) Prove that this program is correct.

**Q5. (6 points)** Let $A$ be a set of functions defined recursively as follows:

- $\sqrt{x} \in A$
- If $f \in A$, then $\frac{1}{f} - f \in A$

a) (4 points) Let $P(f)$ be a predicate on $A$ and suppose you have managed to prove that

  - $P(\sqrt{x})$ is true,
  - $P(f) \implies P(\frac{1}{f} - f)$ for all $f \in A$.

  Prove that $\forall f \in A, P(f)$. You may only assume that the principle of induction holds for the natural numbers, NOT the set $A$.

b) (2 points) Prove that $\forall f \in A,\ f\left(\frac{1}{2}\right) = \frac{1}{\sqrt{2}}$.

**Q6. (7 points)** Let $\Sigma = \{0, 1\}$. For any language $L \in \Sigma^*$, define

$$S = \{w \in \Sigma^* : 0w1 \in L\}$$

Given that $L$ is a regular language, prove that $S$ is a regular language.

Now, this is where a normal exam would end. However, this is not a normal exam.

**Q7. (9 points)** Recall that a segment tree is a data structure effective for querying and updating information about a range of values in a list, such as the minimum element in a provided range. In fact, we will examine an implementation of a segment tree that does this.

Suppose that we have a list of numbers `lst`. In this implementation, the segment tree can be thought of as a binary tree, with each node storing a range of indices `[l, r]`, a value which represents the minimum number in `lst[l:r]` (inclusive of `l`, exclusive of `r`), and pointers to a left and right child. **Note that a node does not have a left and right child if and only if it is true that** `l = r`. Otherwise, the left child will keep track of the range $[\mathtt{l},\ \lfloor \frac{\mathtt{l + r}}{2} \rfloor]$ and the right child will track $[\lfloor \frac{\mathtt{l + r}}{2} \rfloor \mathtt{\ + 1,\ r}]$

As an example, consider the following pseudocode for `update`, which updates the segment tree to correctly return queries about `lst` after setting `lst[i] = x`:

```
def update(root: Node, x: int, i: int):
    if i not in [root.l, root.r] then return
    else if root.l = i = root.r
        then root.val ← x
    update(root.left_child, x, i)
    update(root.right_child, x, i)
    root.val ← min(root.left_child.val, root.right_child.val)
```

a) (4 points) Prove that this method is correct.

b) (3 points) Find the recursive worst-case runtime of `update`.

c) (2 points) Find the tight asymptotic bound for the runtime of `update`.

**Q8. (15 points)** Recall the statement of the *pumping lemma:*

Let $L$ be a regular language associated with some alphabet $\Sigma$. Then $\exists p \in \mathbb{N}^+, \forall w \in L$, where $|w| \geq p$, $\exists x, y, z \in \Sigma^*$ so that

- $|y| \geq 1$
- $|xy| \leq p$
- $\forall n \in \mathbb{N}, xy^n z \in L$

a) (2 points) Show why the statement must have the condition $|xy| \leq p$. (Hint: Consider the language $\{w \in \{a, b\}^* : w = a^i b^j, i \leq j\}$)

b) (2 points) Find with proof the value of $p$ for the language $\mathcal{L}(((ab)^* a)^*)$

c) (1 point) Explain why every finite language can be pumped.

Now, assume that $L, M$ are regular languages that satisfy the pumping lemma, that is, they each have a value $p_L$ and $p_M$ that make the statement of the pumping lemma true.

d) (2 points) Prove that $L \cup M$ can be pumped.

e) (2 points) Prove that $M^*$ can be pumped (You would want to consider when $M$ is finite or when $M$ is infinite).

f) (3 points) Prove that $LM$ can be pumped.

g) (3 points) Prove the pumping lemma.