CSE 390Z: Mathematics for Computation Workshop

Additional Practice Final Solutions

Name: ______________________________

NetID: ______________________________ @uw.edu

Instructions:

- This is a **simulated practice final** prepared by the CSE 390Z teaching staff for Autumn 2023.
- We make no guarantees regarding the similarity of this material to any real 311 exams, nor do we guarantee that our solutions would earn full points if marked by 311 graders.

Advice:

- You are encouraged to take this exam in an environment that mimics the test conditions of your 311 exam (i.e., limit yourself to one-hour and fifty-minutes, don't use any outside resources besides one piece of 8.5x11 inch paper with handwritten notes, and avoid electronic devices such as calculators).
- Some of the questions on this exam are significantly more difficult than the ones we've encountered in workshop. We want you to feel challenged. Try to not stress if you run out of time or get stuck.
- Remember to take deep breaths.
- Relax, you are here to learn.

Question	Max Points
Training Wheels	13
Good Ol' Proofs	14
Strong Induction	20
Structural Induction	20
Languages	20
Irregularity	10
The Other Stuff	12
Grading Morale	1
Total	**110**

1. Training Wheels [13 points]

For this problem, our domain of discourse is college football teams and college football conferences.

You are allowed to use the $\neq$ symbol to check that two objects are not equivalent.

We will use the following predicates:

- $\texttt{Team}(x) := x$ is a football team.
- $\texttt{UW}(x) := x$ is the University of Washington football team.
- $\texttt{WSU}(x) := x$ is the Washington State University football team.
- $\texttt{OSU}(x) := x$ is the Oregon State University football team.

- $\texttt{OldPac}(x) := x$ is the old Pac-12 Conference.
- $\texttt{NewPac}(x) := x$ is the new Pac-2 Conference.

- $\texttt{Member}(x, y) :=$ the football team x has been a part of the conference y.
- $\texttt{Lost}(x, y) := x$ lost to y in a football game.

(a) State whether the two statements below are equivalent. Provide a one sentence justification. **[2 points]**

$$\exists y \Big[\texttt{OldPac}(y) \land \forall x \Big(\texttt{Team}(x) \to \big(\texttt{UW}(x) \to \texttt{Member}(x, y) \big) \Big) \Big]$$

$$\exists y \Big[\texttt{OldPac}(y) \land \forall x \Big(\texttt{UW}(x) \to \texttt{Member}(x, y) \Big) \Big]$$

Solution:

Yes, they are equivalent. The UW Huskies are already a football team, so there is no need to actually state $\texttt{Team}(x)$.

(b) Translate the following sentence into predicate logic. **[3 points]**

Excluding WSU, at least one team has been a part of the new Pac-2 conference and the old Pac-12 conference.

Solution:

$$\exists x \exists y \exists z \Big[\texttt{Team}(x) \land \neg \texttt{WSU}(x) \land \texttt{OldPac}(y) \land \texttt{Member}(x, y) \land \texttt{NewPac}(z) \land \texttt{Member}(x, z) \Big]$$

(c) Translate the following statement into predicate logic. **[4 points]**

UW has won against all football teams besides itself, and WSU has lost to all fooootball teams besides itself.

Solution:

$$\exists x \exists y \Big[\texttt{UW}(x) \land \forall a \big((\texttt{Team}(a) \land (a \neq x)) \to \texttt{Lost}(a, x) \big) \land$$
$$\texttt{WSU}(y) \land \forall b \big((\texttt{Team}(b) \land (b \neq y)) \to \texttt{Lost}(y, b) \big) \Big]$$

(d) Negate the following statement. Your final answer should have zero negations. **[4 points]**

Warning: this statement makes absolutely no sense. Do **NOT** spend time thinking about its meaning. We want you to blindly follow your equivalency laws here.

$$\forall x \forall y \Big[\big(\texttt{WSU}(x) \land \texttt{OSU}(y)\big) \land \big(\neg\texttt{Lost}(x,y) \lor \neg\texttt{Lost}(y,x)\big) \Big]$$

Solution:

Students only need to write the final answer to receive full credit.

$$\exists x \exists y \Big[\big(\texttt{WSU}(x) \land \texttt{OSU}(y)\big) \to \big(\texttt{Lost}(x,y) \land \texttt{Lost}(y,x)\big) \Big]$$

The corresponding chain of equivalences is provided below for reference.

$$\begin{aligned}
&\neg\forall x \forall y \Big[\big(\texttt{WSU}(x) \land \texttt{OSU}(y)\big) \land \big(\neg\texttt{Lost}(x,y) \lor \neg\texttt{Lost}(y,x)\big) \Big] && \\
\equiv\ &\neg\forall x \forall y \Big[\big(\texttt{WSU}(x) \land \texttt{OSU}(y)\big) \land \neg\big(\texttt{Lost}(x,y) \land \texttt{Lost}(y,x)\big) \Big] && \text{DeMorgans} \\
\equiv\ &\exists x \exists y \neg\Big[\big(\texttt{WSU}(x) \land \texttt{OSU}(y)\big) \land \neg\big(\texttt{Lost}(x,y) \land \texttt{Lost}(y,x)\big) \Big] && \text{DeMorgans for Quantifiers} \\
\equiv\ &\exists x \exists y \Big[\neg\big(\texttt{WSU}(x) \land \texttt{OSU}(y)\big) \lor \neg\neg\big(\texttt{Lost}(x,y) \land \texttt{Lost}(y,x)\big) \Big] && \text{DeMorgans} \\
\equiv\ &\exists x \exists y \Big[\neg\big(\texttt{WSU}(x) \land \texttt{OSU}(y)\big) \lor \big(\texttt{Lost}(x,y) \land \texttt{Lost}(y,x)\big) \Big] && \text{Double Negation} \\
\equiv\ &\exists x \exists y \Big[\big(\texttt{WSU}(x) \land \texttt{OSU}(y)\big) \to \big(\texttt{Lost}(x,y) \land \texttt{Lost}(y,x)\big) \Big] && \text{Law of Implication}
\end{aligned}$$

2. Good Ol' Proofs [14 points]

(a) Prove for some predefined sets A, B, C that $(A \setminus B) \cup (C \setminus B) = (A \cup C) \setminus B$. **[8 points]**

Hint: You will need to use proof by cases.

Solution:

$(\subseteq)$

Let $x \in (A \setminus B) \cup (C \setminus B)$ be arbitrary. We will do a proof by cases on whether $x \in A \setminus B$ or $x \in C \setminus B$.

Case 1: Suppose $x \in A \setminus B$. By definition, $x \in A$ and $x \notin B$. Since $x \in A$, surely $x \in A \cup C$. Since $x \in A \cup C$ and $x \notin B$, we have $x \in (A \cup C) \setminus B$.

Case 2: Suppose $x \in C \setminus B$. By definition, $x \in C$ and $x \notin B$. Since $x \in C$, surely $x \in C \cup A$, or equivalently $x \in A \cup C$. Since $x \in A \cup C$ and $x \notin B$, we have $x \in (A \cup C) \setminus B$.

Since x was an arbitrary, we have shown that $(A \setminus B) \cup (C \setminus B) \subseteq (A \cup C) \setminus B$.

$(\supseteq)$

Let $x \in (A \cup C) \setminus B$ be arbitary. By definition, $x \in A \cup C$ and $x \notin B$. We will do a proof by cases on whether $x \in A$ or $x \in C$.

Case 1: Suppose $x \in A$. Since $x \in A$ and $x \notin B$, we have $x \in A \setminus B$. Then, we can also say $x \in (A \setminus B) \cup (C \setminus B)$.

Case 2: Suppose $x \in C$. Since $x \in C$ and $x \notin B$, we have $x \in C \setminus B$. Then, we can also say $x \in (C \setminus B) \cup (A \setminus B)$, or equivalently $x \in (A \setminus B) \cup (C \setminus B)$.

Since x was an arbitrary, we have shown that $(A \cup C) \setminus B \subseteq (A \setminus B) \cup (C \setminus B)$.

Therefore, since we have proven that $(A \setminus B) \cup (C \setminus B) \subseteq (A \cup C) \setminus B$ and $(A \cup C) \setminus B \subseteq (A \setminus B) \cup (C \setminus B)$, we have shown that $(A \setminus B) \cup (C \setminus B) = (A \cup C) \setminus B$.

(b) Prove true or false: if $a \equiv 1 \pmod 5$ and $b \equiv 1 \pmod 5$, then $\gcd(a, b) \equiv 1 \pmod 5$. **[3 points]**

Solution:

False. For example, when $a = 6$ and $b = 21$, their $\gcd$ is 3 and clearly $3 \not\equiv 1 \pmod 5$.

(c) Consider the following statement: if $a, b \in \mathbb{Z}$ and $a \geq 2$, then $a \nmid b$ or $a \nmid b + 1$.

Your goal is to disprove this statement using proof by contradiction.

Write out **JUST THE FIRST SENTENCE** of the proof.

Your answer should look something like "Suppose, for the sake of contradiction, ..." **[3 points]**

Solution:

Suppose, for the sake of contradiction, that there exist $a, b \in \mathbb{Z}$ with $a \geq 2$ where $a \mid b$ and $a \mid b + 1$.

3. Strong Induction [20 points]

Consider the function f, which takes a natural number as input and outputs a natural number.

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 2 & \text{if } n = 1 \\ f(n-1) + 2 \cdot f(n-2) & \text{if } n \geq 2 \end{cases}$$

Prove that $f(n) = 2^n$ for all $n \in \mathbb{N}$.

Solution:

Let $P(n)$ be the claim that $f(n) = 2^n$. We will prove $P(n)$ true for all $n \in \mathbb{N}$ by strong induction.

Base Case:

- $f(0) = 1 = 1 = 2^0$ so $P(0)$ holds.
- $f(1) = 2 = 2 = 2^1$ so $P(1)$ holds.

Inductive Hypothesis: Suppose that for some arbitrary integer $k \geq 1$, that $P(j)$ holds for all $j \in \mathbb{N}$ such that $j \leq k$.

Inductive Step:

Goal: Show $P(k+1)$, i.e. $f(k+1) = 2^{k+1}$

$$\begin{aligned} f(k+1) &= f(k+1-1) + 2 \cdot f(k+1-2) && \text{Definition of } f \\ &= f(k) + 2 \cdot f(k-1) \\ &= 2^k + 2 \cdot 2^{k-1} && \text{I.H. twice} \\ &= 2^k + 2^{k-1+1} \\ &= 2^k + 2^k \\ &= 2 \cdot 2^k \\ &= 2^{k+1} \end{aligned}$$

Clearly, $P(k+1)$ holds.

Conclusion: Therefore, we have proven the claim true by strong induction.

4. Structural Induction [20 points]

We define **List** as such:

- Basis step: $\texttt{nil} \in$ **List**
- Recursive step: if $L \in$ **List** and $a \in \mathbb{Z}$, then $a :: L \in$ **List**

We define the function len as such:

- $\texttt{len}(\texttt{nil}) \coloneqq 0$
- $\texttt{len}(a :: L) \coloneqq \texttt{len}(L) + 1$ for any $L \in$ **List** and $a \in \mathbb{Z}$

We define the function append as such:

- $\texttt{append}(\texttt{nil}, L) \coloneqq L$ for any $L \in$ **List**
- $\texttt{append}(a :: L, R) \coloneqq a :: \texttt{append}(L, R)$ for any $L, R \in$ **List** and $a \in \mathbb{Z}$

Prove that $\texttt{append}(L, \texttt{append}(R, M)) = \texttt{append}(\texttt{append}(L, R), M)$ for all $L, R, M \in$ **List**. In other words, we are asking you to prove the associative property for append.

Solution:

Let $P(L)$ be "$\texttt{append}(L, \texttt{append}(R, M)) = \texttt{append}(\texttt{append}(L, R), M)$ for all $R, M \in$ **List**". We will prove $P(L)$ true for all $L \in$ **List** by structural induction.

Base Case: Consider when $L = \texttt{nil}$. Let $R, M \in$ **List** be arbitrary. Then, $\texttt{append}(\texttt{nil}, \texttt{append}(R, M)) = \texttt{append}(R, M) = \texttt{append}(\texttt{append}(\texttt{nil}, R), M)$. Clearly, $P(\texttt{nil})$ holds.

Let D be an arbitrary list not covered by the base case. By the exclusion rule, $D = x :: C$ for some $C \in$ **List** and some $x \in \mathbb{Z}$.

Inductive Hypothesis: Suppose that $P(C)$ holds.

Inductive Step:

Goal: Prove $P(D)$ holds, i.e. that $P(x :: C)$ holds

Let $R, M \in$ **List** be arbitrary.

$$\begin{aligned}
\texttt{append}(D, \texttt{append}(M, R)) &= \texttt{append}(x :: C, \texttt{append}(M, R)) && \text{Definition of } D \\
&= x :: \texttt{append}(C, \texttt{append}(M, R)) && \text{Definition of append} \\
&= x :: \texttt{append}(\texttt{append}(C, M), R) && \text{I.H.} \\
&= \texttt{append}(x :: \texttt{append}(C, M), R) && \text{Definition of append} \\
&= \texttt{append}(\texttt{append}(x :: C, M), R) && \text{Definition of append} \\
&= \texttt{append}(\texttt{append}(D, M), R) && \text{Definition of } D
\end{aligned}$$

Since R and M were arbitrary, we have shown that $P(D)$ holds.

Conclusion: Therefore, per the principles of structural induction, the claim holds for all $L \in$ **List**.

5. Languages [20 points]

(a) Describe in English the language matched by the regular expression $(0 \cup 1)^*10101(0 \cup 1)^*$. **[2 points]**

Solution:

All binary strings containing the substring 10101.

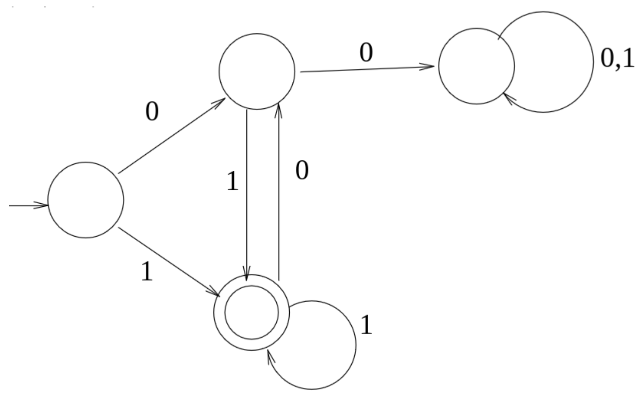
(b) Write a regular expression that recognizes all binary strings which do **NOT** contain the substring 110. **[6 points]**

Solution:

$(0 \cup 10)^*1^*$

(c) Design a DFA that recognizes all binary strings which end with a 1 and do **NOT** contain the substring 00. **[6 points]**

Solution:



(d) Design a CFG for the language consisting of all binary non-palindromes. **[6 points]**

Solution:

$$
\begin{aligned}
S &\to 0S0 \mid 1S1 \mid D \\
D &\to 1A0 \mid 0A1 \\
A &\to \epsilon \mid 0A \mid 1A
\end{aligned}
$$

6. Irregularity [10 points]

Show that the language $L = \{w \in \{0,1\} : w$ contains exactly two more 1s than it does 0s$\}$ is irregular.

Solution:

Let L be as defined above. Let D be an arbitrary DFA. Suppose for the sake of contradiction that D accepts L. Consider $S = \{1^n 11 : n \geq 0\}$. Since S has infinitely many strings and D has a finite amount of states, two different strings from S are bound to end up in the same state in D. Say these strings are $1^a 11$ and $1^b 11$ for some $a, b \geq 0$ such that $a \neq b$. Now, append 0^a to both strings. The resulting strings are:

- $x = 1^a 110^a$. Note that $x \in L$.
- $y = 1^b 110^a$. Note that $y \notin L$, because the $\#$ of 1s is $b+2$ and the $\#$ of 0s is a. Since $b \neq a$, we also know $b + 2 \neq a + 2$, i.e. it is impossible for the number of 1s to be exactly two greater than the number of 0s.

Both x and y end up in the same state, but $x \in L$ while $y \notin L$. Thus, this state must both accept and reject, which is a contradiction. Since D was arbitrary, no DFA accepts L. Therefore, L is irregular.

7. The Other Stuff [12 points]

(a) Let Q be the relation $\{(1,1),(1,3),(1,4),(2,2),(3,1),(3,4),(4,1),(4,3)\}$ on the set $A = \{1,2,3,4\}$.

Consider the following properties: reflexivity, symmetry, anti-symmetry, transitivity. If the property holds, simply state so. If the property does not hold, provide a counterexample. **[2 points]**

Solution:

- not reflexive, since $(3,3) \notin Q$
- symmetric
- not antisymmetric, since $(1,3) \in Q$ and $(3,1) \in Q$ but clearly $1 \neq 3$
- not transitive, since $(4,1) \in Q$ and $(1,4) \in Q$ but $(4,4) \notin Q$

(b) Suppose that R and S are symmetric relations on some non-empty set.

Prove or disprove the claim that $R - S$ is symmetric. **[6 points]**

Solution:

Let $(a,b) \in R - S$ be arbitrary. By definition, $(a,b) \in R$ and $(a,b) \notin S$. Since R is symmetric and $(a,b) \in R$, we also know $(b,a) \in R$. Suppose, for the sake of contradiction, that $(b,a) \in S$. Since S is symmetric, that would entail $(a,b) \in S$, but that is impossible since $(a,b) \notin S$. Thus, $(b,a) \notin S$. Since $(b,a) \in R$ and $(b,a) \notin S$, by definition $(b,a) \in R - S$. Since (a,b) was arbitrary, we have proven that $R - S$ is symmetric.

(c) Consider the following claim: "for any set of strings, if it cannot be matched by a regular expression, then we cannot write a Java program that recognizes it."

Is the claim valid? If it's true, write a brief explanation (2-3 sentences) as to why. If it's false, provide a counterexample, explain why your counterexample is irregular, and explain how to design a Java program that recognizes it. **[4 points]**

Solution:

False. Recall from lecture that the set of binary palindromes cannot be matched by a regular expression. In Java, we could take a string as input, store a copy of that string reversed, and then check for equality.

8. Grading Morale [1 point]

What's one of your go-to jokes?

Solution:

How many 311 students does it take to change a lightbulb?

None. They prove that it's possible and leave the implementation as an exercise for the reader.