CSE 311 : Spring 2022 Final Exam

Name:	**NetID:** @uw.edu

Instructions

- You have one-hour and fifty-minutes to complete this exam.
- You are permitted one piece of 8.5x11 inch paper with handwritten notes (notes are allowed on both sides of the paper).
- You may not use a calculator or any other electronic devices during the exam.
- We will be scanning your exams before grading them. Please write legibly, and avoid writing up to the edge of the paper.
- If you run out of room, indicate (e.g. with an arrow) that you're going to use the back of the sheet, and continue writing there.
- You may also use the last page for extra space, but tell us where to find your answer if it's not right below the problem.

Advice

- Remember to properly format English proofs (e.g. introduce all your variables).
- All proofs for this exam must be English proofs.
- We give partial credit for the beginning and end of a proof. Even if you don't know how the middle goes, you can write the start of the proof and put the "target" and conclusion at the bottom.
- Remember to take deep breaths.

Question	**Max points**
Training Wheels	14
First Proof	11
Models of Computation	12
Induction I	20
Short Answer	12
Induction II	20
Wait, That's Illegal	15
Grading Morale	1
Total	**105**

1. Training Wheels [14 points]

Let your domain of discourse be positive integers. Use the following predicates:

- $\texttt{Prime}(x)$ which is true if and only if x is prime.
- $\texttt{Divides}(x, y)$ which is true if and only if $x|y$.
- $\texttt{Greater}(x, y)$ which is true if and only if $x > y$.

(a) Translate this sentence into English. $\exists x[\texttt{Prime}(x) \wedge \texttt{Divides}(3, x)]$ [3 points]

(b) Translate this sentence into predicate logic: [3 points]

For every prime number x there is a unique positive integer y such that $y > 1$ and $y|x$.

(c) Take the contrapositive of the implication in this sentence. Your final answer must be in English. You do not need to show work for this problem. [3 points]

If x, y are prime then $x \cdot y$ is not prime.

(d) Negate the sentence from part (c). Give your answer in English. [3 points]

Fill in the correct bubble [1 point each]

(e) The original statement in (c) is
◯ True
◯ False

(f) The contrapositive of the statement from (c) has
◯ The same truth value as the original statement from (c)
◯ The opposite truth value as the original statement from (c).

2. First Proof [12 points]

(a) Prove $(A \cup B) \setminus (A \cap B) \subseteq [A \setminus (A \cap B)] \cup [B \setminus (A \cap B)]$
You must format your proof as an English proof and structure your proof by introducing arbitrary element(s) of sets as appropriate.
We recommend drawing a picture of the sets for yourself so you see why the statement is true. [7 points]

(b) From (only) what you've written above, can you conclude: $(A \cup B) \setminus (A \cap B) = [A \setminus (A \cap B)] \cup [B \setminus (A \cap B)]$ (we've changed the subset from (a) to an equals sign).

If you can conclude the new statement, briefly (1-2 sentences) explain why.
If you cannot conclude the new statement, what statement do you still need to prove to get the new conclusion? (you can give you answer in English, set notation, or predicate logic notation – whichever you find most convenient). [2 points]

(c) Disprove the following statement: $(A \cup B) \setminus (A \cap B) = [(A \cup B) \setminus A] \cap [(A \cup B) \setminus B]$ [3 points

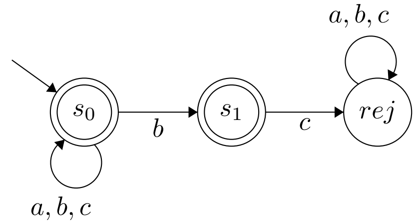
3. Models of Computation [11 points]

Let $\Sigma = \{a, b, c\}$ and let $L = \{w \in \Sigma^* : w \text{ does not contain } bc \text{ as a substring}\}$.
In English, L is the language containing all strings over $\{a, b, c\}$ that do not have a substring bc.

(a) Write a regular expression that matches L. (No explanation required). [4 points]

(b) Draw a DFA that accepts L. Include a brief description of how your machine works (You'll probably want about 2-3 sentences total, or a few words for each state about what it does). [4 points]

Your friend designs the **N**FA below, intending for it to accept L. They say that the machine, on seeing a b part of a bc will (magicially) transition right, and then on a c will transition right again to ensure if bc is a substring the machine rejects.



(c) There is at least one string that your friend's machine accepts that is not in L. [1.5 points]
◯ True
◯ False

(d) There is at least one string that your friend's machine **does not** accept that is in L. [1.5 points]
◯ True
◯ False

4. *****Induction***** [20 points]

You're trying to write code to produce a string made up of $*$'s. At your disposal, you have three methods, described below:

```
/* This method prints 5 asterisks */
void print5() {
    print("*****");
}

/* This method prints 6 asterisks */
void print6() {
    print("******");
}

/* This method prints 13 asterisks */
void print13() {
    print("*************");
}
```

Show that, for every integer $n \geq 15$, we can use some combination of the methods above to print a string with n asterisks.

You must use strong or weak induction for this proof. We **strongly** recommend strong induction.
Remember to define a predicate $P(n)$.

5. Short Answer [12 points]

(a) Suppose you know $a \equiv b \pmod{10}$. Can you conclude that $a \equiv b \pmod{5}$? If so, briefly justify (not a full proof, just 1-3 sentences); if not give a counter-example.

(b) Suppose you know $a \equiv b \pmod{10}$. Can you conclude that $a \equiv b \pmod{20}$? If so, briefly justify (not a full proof, just 1-3 sentences); if not give a counter-example.

(c) Your goal is to show "if there is a context free grammar for L, then there is a DFA for L." using proof by contradiction.

Finish **just** the first sentence of the proof:

Proof. Suppose, for the sake of contradiction, there is a language L, such that

$\square$

(d) Let

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

Imagine you define $P(n)$ to be "$2^{n-1} + 2 \cdot 2^{n-2}$." with the intent of proving $P(n)$ by induction. Briefly justify (1-2 sentences) why this is not a proper definition of $P(n)$.

6. II noitcudnI [20 points]

In this problem we define **CharTree**s as follows:

Basis Step: `Null` is a **CharTree**.
Recursive Step: If L, R are **CharTree**s and $c \in \Sigma$, then $\texttt{CharTree}(L, c, R)$ is also a **CharTree**.

Intuitively, a **CharTree** is a tree where the non-null nodes store a `char` data element.

We also define the following operations on **CharTree**s:

- The preorder function returns the preorder traversal of all elements in a **CharTree**.

$$\begin{aligned} \mathsf{preorder}(\texttt{Null}) &= \varepsilon \\ \mathsf{preorder}(\texttt{CharTree}(L, c, R)) &= c \cdot \mathsf{preorder}(L) \cdot \mathsf{preorder}(R) \end{aligned}$$

- The postorder function returns the postorder traversal of all elements in a **CharTree**.

$$\begin{aligned} \mathsf{postorder}(\texttt{Null}) &= \varepsilon \\ \mathsf{postorder}(\texttt{CharTree}(L, c, R)) &= \mathsf{postorder}(L) \cdot \mathsf{postorder}(R) \cdot c \end{aligned}$$

- The mirror function produces the mirror image of a **CharTree**.

$$\begin{aligned} \mathsf{mirror}(\texttt{Null}) &= \texttt{Null} \\ \mathsf{mirror}(\texttt{CharTree}(L, c, R)) &= \texttt{CharTree}(\mathsf{mirror}(R), c, \mathsf{mirror}(L)) \end{aligned}$$

Finally, for all strings x, let the "reversal" of x (in symbols x^R) produce the string in reverse order.

We have an example of all of these operations on the next page.

Now, let's actually get to the problem. Show, via structural induction, that for every **CharTree** T, the reversal of the preorder traversal of T is the same as the postorder traversal of the mirror of T.

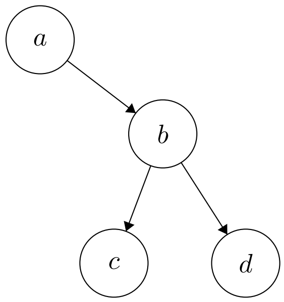
In notation, you are to prove for every **CharTree**, T: $[\mathsf{preorder}(T)]^R = \mathsf{postorder}(\mathsf{mirror}(T))$.

You may use the following facts without proving them:
Fact 1: For all strings $x_1, x_2, \ldots, x_k$: $(x_1 \cdot x_2 \cdot \cdots \cdot x_k)^R = x_k^R \cdot x_{k-1}^R \cdot \cdots \cdot x_1^R$
Fact 2: For every character c, $c^R = c$.
When applying these facts, you may treat characters as strings (and length-one strings as characters).

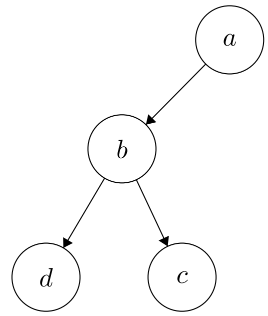
In your proof, apply at most one of the definitions/facts above in every step.

If you run out of room, continue on the next page.

For more intuition on the rules, we have included an example below. **There are no new questions on this page.** But you may use the space below if you run out of room on the previous page.



Let T_i be the tree above.
$\mathsf{preorder}(T_i) =$"abcd".
T_i is built as $(\texttt{null}, a, U)$
Where U is (V, b, W),
$V = (\texttt{null}, c, \texttt{null}), W = (\texttt{null}, d, \texttt{null})$.



This tree is $\mathsf{mirror}(T_i)$.
$\mathsf{postorder}(\mathsf{mirror}(T_i)) =$"dcba",
"dcba" is the reversal of "abcd" so
$[\mathsf{preorder}(T_i)]^R = \mathsf{postorder}(\mathsf{mirror}(T_i))$ holds for T_i

7. Wait that's illegal [15 points]

Choose to do exactly one of these two problems.

(a) Let $L = \{a^{311}b^nc^{n+311} : n \geq 0\}$.

Using the proof technique from class, show that L is irregular.

(b) Let $\mathcal{F}$ be the set of all functions that take a binary string as input and produce either 'a' or 'b' as output.

In notation, $\mathcal{F} = \{f | f : \{0,1\}^* \to \{a, b\}\}$

Using the proof technique from class, show that $\mathcal{F}$ is uncountable.

8. Grading Morale [1 point]

Put something on this page. Depending on your mood, this might be a poem, the statement of your favorite theorem, the statement of your least favorite theorem, or a piece of art. Or you can tell us what you believe Oob's story may be (and Oob's rise to representing CSE 311).

As long as you make some mark on this page, you will get the point.