CSE 311 : Winter 2022 Final Exam Solutions

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.
- 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	12
Set Proofs	10
Models of Computation	12
Induction I	20
Now, False or True	9
Induction II	21
Wait, That's Illegal	15
Grading Morale	1
Total	**100**

1. Training Wheels [12 points]

Let your domain of discourse be all fruits and all people.
Use the predicates IsApple, IsOrange, IsBanana, Doctor, Fruit, Person to do domain restriction.
You can also use the predicate $\texttt{Likes}(x, y)$, which is true if and only if x likes y.
You may also use quantifiers, standard logical notation (e.g., $\leftrightarrow, \wedge$, etc.), and $=$, but you may not introduce other predicates.

In addition to introducing variables, you can also use the constants Robbie, Allie.

(a) Robbie likes all apples.

Solution:

Among all fruits, Robbie likes all apples.

If f is a fruit, then if f is an apple, then Robbie likes it.

$$\forall f(\texttt{Fruit}(f) \to [\texttt{IsApple}(f) \to \texttt{Likes}(\texttt{Robbie}, f)])$$

An Apple is already a fruit, so there is no need to actually state $\texttt{Fruit}(f)$.

If a is an apple, then Robbie likes a

$$\forall a(\texttt{IsApple}(a) \to \texttt{Likes}(\texttt{Robbie}, a))$$

(b) There is a fruit that every doctor likes, and this fruit is not an orange.

Solution:

In this case you must state both $\texttt{Fruit}(f)$ and $\texttt{IsOrange}(f)$.

There exists a fruit f that is not an orange such that if d is a doctor, then d likes f.

$$\exists f(\texttt{Fruit}(f) \wedge \neg\,\texttt{IsOrange}(f) \wedge \forall d[\texttt{Doctor}(d) \to \texttt{Likes}(d, f)])$$

(c) Among all fruits, Allie likes only bananas.

Solution:

Similar approach to problem a, but predicates are switched. We must also use both $\texttt{Fruit}()$ and $\texttt{isBanana}()$. Note that the implication goes only in one direction! This statement says Allie doesn't like oranges, grapes, etc. It doesn't necessarily state that Allie likes all of the bananas.

If f is a fruit, then if Allie likes it, then it must be a banana.

$$\forall f(\texttt{Fruit}(f) \to [\texttt{Likes}(\texttt{Allie}, f) \to \texttt{IsBanana}(f)])$$

If f is a fruit and Allie likes it, then it must be a banana.

$$\forall f([\texttt{Fruit}(f) \wedge \texttt{Likes}(\texttt{Allie}, f)] \to \texttt{IsBanana}(f))$$

If f is a fruit and f is not a banana, then Allie does not like f.

$$\forall f([\texttt{Fruit}(f) \wedge \neg\,\texttt{IsBanana}(f)] \to \neg\,\texttt{Likes}(\texttt{Allie}, f))$$

Take the contrapositive of this implication (leave your answer in English, you do not need to show work). Your answer must be simplified (e.g. negations should apply to individual predicates/propositions; you should not need phrases like "it is not the case that...")

(d) "If you like apples or you like oranges, then you must also like grapes and pineapples."

Solution:

If you don't like grapes or you don't like pineapples, then you won't like apples and you won't like oranges.

2. A Set Proof [10 points]

Let $S = \{x \in \mathbb{Z} : x \equiv 1 \pmod 5\}$ and let $T = \{x \in \mathbb{Z} : x \equiv 1 \pmod{15}\}$.

(a) Prove that $T \subseteq S$. [7 points] **Solution:**

Let x be an arbitrary element of T. By definition of T, $x \equiv 1 \pmod{15}$. Applying the definition of mod, we have $15 \mid (1 - x)$. Applying the definition of divides, we know there is an integer z such that $15z = 1 - x$.

Rearranging, we have $5 \cdot (3z) = 1 - x$. Since z is an integer, $3z$ is an integer, so we can apply the definition of divides to get $5 \mid (1 - x)$ and so $1 \equiv x \pmod n$. Thus $x \in S$ by definition of S, and we can conclude $T \subseteq S$.

(b) Prove that $S \not\subseteq T$. [3 points] **Solution:**

Note that $11 \in S$, as $5 \mid (11 - 1)$, but $11 \notin T$, as $15 \nmid (11 - 1)$. Since there is at least one element in S but not T, $S \not\subseteq T$.

3. Models of Computation [12 points]

Let $\Sigma = \{1, 2, 3, 4\}$ and let L be the language containing all strings w where w is **nondecreasing**.

We define a string w as **nondecreasing** if for all k, the character at index k is greater than or equal to all the numbers to the left, meaning the number at index k is greater than or equal to each of the numbers at indices 0 to $k-1$.

Some example **nondecreasing** strings are:

- ε (this string vacuously meets the definition of nondecreasing)
- 13 $(3 \geq 1)$
- 11223334

An example on an invalid string (one not in the language) is 132, as $3 > 2$.

(a) Write a regular expression that matches L. (No explanation required).

Solution:

$$(1)^*(2)^*(3)^*(4)^*$$

(b) Write a CFG that generates L.
Be sure to tell us which symbol is the start symbol; also include a sentence or two of explanation of how your CFG works.

Solution:

S is the start symbol. If the string is nondecreasing then all 1's come before any 2's which come before 3's which come before 4's. So we split the string into those sections (any or all of which can be the empty string).

$$\begin{aligned}
\mathbf{S} &\rightarrow \mathbf{ABCD} \mid \varepsilon \\
\mathbf{A} &\rightarrow 1\mathbf{A} \mid \varepsilon \\
\mathbf{B} &\rightarrow 2\mathbf{B} \mid \varepsilon \\
\mathbf{C} &\rightarrow 3\mathbf{C} \mid \varepsilon \\
\mathbf{D} &\rightarrow 4\mathbf{D} \mid \varepsilon
\end{aligned}$$

4. Induction I [20 points]

Consider this function f, that takes in a natural number and outputs a natural number:

$$f(n) = \begin{cases} 2 \cdot f(n-1) + 2 \cdot f(n-2) + 3^{n-2} & \text{if } n \geq 2 \\ 3 & \text{if } n = 1 \\ 1 & \text{if } n = 0 \end{cases}$$

Use induction to show $f(n) = 3^n$ for all natural numbers n.
Remember to explicitly define a predicate $P()$ as part of your proof.

Solution:

Let $P(n)$ be "$f(n) = 3^n$". We show $P(n)$ holds for all $n \in \mathbb{N}$ by induction.

Base Case $(n = 0)$: $f(0) = 1 = 3^0$

Base Case $(n = 1)$: $f(1) = 3 = 3^1$

Inductive Hypothesis: Suppose $P(0) \wedge P(1) \wedge \ldots \wedge P(k)$ holds for an arbitrary $k \geq 1$.

Inductive Step:

$$\begin{aligned} f(k+1) &= 2 \cdot f(k) + 2 \cdot f(k-1) + 3^{k-1} && \text{definition of } f \\ &= 2 \cdot 3^k + 2 \cdot 3^{k-1} + 3^{k-1} && \text{by IH twice} \\ &= 2 \cdot 3^k + 3 \cdot 3^{k-1} && \text{combining like terms of } 3^{k-1} \\ &= 2 \cdot 3^k + 3^k = 3 \cdot 3^k = 3^{k+1} && \text{algebra} \end{aligned}$$

So $P(k+1)$ holds.

By the principle of induction, $P(n)$ holds for all $n \in \mathbb{N}$.

5. Now, False or True [9 points]

For the following questions, determine whether the statement is true or false, and write "T" or "F" on the line provided. Then provide 1-3 sentences of explanation. Your explanations do not need to be full or formal proofs, intuitive justifications are fine.

(a) _____ It is always incorrect to perform a strong induction proof with a single base case.

Solution:

False. We've seen examples of strong induction with a single base case. For example, the apples-to-apples question (homework 6, question 6) and the proof of the existence of prime factorizations in lecture.

We also accepted statements like "You can always turn a weak IH into a strong IH and the IS is still correct." and "As long as the IS uses something assumed by the IH, the proof will be correct" and "You need multiple base cases when your IS would need to go back multiple steps, which doesn't always happen with strong hypotheses."

(b) _____ "p is sufficient for q" and "q is necessary for p", are both best translated as $p \to q$.

Solution:

True. Both are $p \to q$. The first says that p is enough for you to guarantee q must follow (i.e., $p \to q$) the second says that in order for p to occur, q must occur. So when p happens q must have happened to (thus $p \to q$).

(c) _____ Suppose you wish to prove the implication $p \to q$. One way to prove the claim is to show the converse $q \to p$ is false. Since converses are different from each other, you can conclude $p \to q$ must be true.

Solution:

False. There are implications where both the converse and the original statement are true, so this is not a valid proof technique.

6. Induction II [21 points]

You'll see "**up-trees**" in your data structures class. Every **up-tree**, T, has a number of nodes (which we call $\texttt{Nodes}(T)$) and a "rank" (think of it as a measure of size), which we denote by $\textsf{Rank}(T)$.

Up-trees are defined with the following recursive definition:

Basis Step: A single node is an **up-tree**.
$\texttt{Nodes}(T) = 1$ and $\textsf{Rank}(T) = 0$ when T is a single node.

Recursive Steps: There are three ways to make new **up-tree**s. You should both read the rules on this page and look at the examples on the next page.

- **Combine 1** Given two **up-tree**s S and T where $\textsf{Rank}(S) = \textsf{Rank}(T)$, you can make a new combined up-tree, N, by making the root of T be the root of N, and making the root of S a child of the root of T.

  In this case: $\texttt{Nodes}(N) = \texttt{Nodes}(S) + \texttt{Nodes}(T)$ and $\textsf{Rank}(N) = 1 + \textsf{Rank}(T)$.

- **Combine 2** Given two **up-trees**, S, T, where $\textsf{Rank}(S) < \textsf{Rank}(T)$, you can make a new combined **up-tree**, N, by making the root of T be the root of N, and making the root of S a child of the root of T.

  In this case: $\texttt{Nodes}(N) = \texttt{Nodes}(S) + \texttt{Nodes}(T)$ and $\textsf{Rank}(N) = \textsf{Rank}(T)$.

- **Compress** Given one **up-tree**, S, you can "compress" it into a new **up-tree**, N, by keeping the root the same and making all other nodes direct children of the root.

  In this case: $\texttt{Nodes}(N) = \texttt{Nodes}(S)$ and $\textsf{Rank}(N) = \textsf{Rank}(S)$.

Prove using structural induction that $\texttt{Nodes}(T) \geq 2^{\textsf{Rank}(T)}$ holds for all **up-tree**s T.

Solution:

Let $P(T)$ be "$\texttt{Nodes}(T) \geq 2^{\textsf{Rank}(T)}$". We prove $P(T)$ holds for all **up-tree**s T by structural induction on T.

Base Case ($T = \textsf{Node}$): By basis definitions, $\texttt{Nodes}(T) = 1 \geq 1 = 2^0 = 2^{\textsf{Rank}(T)}$, so $P(\textsf{Node})$ holds.

Inductive Hypothesis: Suppose $P(S)$ and $P(T)$ hold for arbitrary **up-tree**s S and T.

Inductive Step: There are three inductive steps, one for each recursive rule.

- **Combine 1:** Suppose that $\textsf{Rank}(S) = \textsf{Rank}(T)$, and we use **Combine 1** to generate the next tree N.

  Goal: Show $\texttt{Nodes}(N) \geq 2^{\textsf{Rank}(N)}$, i.e. $\texttt{Nodes}(S) + \texttt{Nodes}(T) \geq 2^{1+\textsf{Rank}(T)}$

$$\begin{aligned}
\texttt{Nodes}(N) &= \texttt{Nodes}(S) + \texttt{Nodes}(T) && \text{since } \texttt{Nodes}(N) = \texttt{Nodes}(S) + \texttt{Nodes}(T) \\
&\geq 2^{\textsf{Rank}(S)} + 2^{\textsf{Rank}(T)} && \text{IH} \\
&= 2^{\textsf{Rank}(T)} + 2^{\textsf{Rank}(T)} && \text{since } \textsf{Rank}(S) = \textsf{Rank}(T) \\
&= 2 \cdot 2^{\textsf{Rank}(T)} \\
&= 2^{1+\textsf{Rank}(T)} \\
&= 2^{\textsf{Rank}(N)} && \text{since } \textsf{Rank}(N) = 1 + \textsf{Rank}(T)
\end{aligned}$$

- **Combine 2:** Suppose $\textsf{Rank}(S) < \textsf{Rank}(T)$, and we use **Combine 2** to generate the next tree N.

  Goal: Show $\texttt{Nodes}(N) \geq 2^{\textsf{Rank}(N)}$, i.e. $\texttt{Nodes}(S) + \texttt{Nodes}(T) \geq 2^{\textsf{Rank}(T)}$

$$\begin{aligned}
\texttt{Nodes}(N) &= \texttt{Nodes}(S) + \texttt{Nodes}(T) && \text{since } \texttt{Nodes}(N) = \texttt{Nodes}(S) + \texttt{Nodes}(T) \\
&\geq 2^{\textsf{Rank}(S)} + 2^{\textsf{Rank}(T)} && \text{IH} \\
&\geq 2^{\textsf{Rank}(T)} && \text{since } 2^{\textsf{Rank}(S)} + 2^{\textsf{Rank}(T)} \geq 2^{\textsf{Rank}(T)} \\
&= 2^{\textsf{Rank}(N)} && \text{since } \textsf{Rank}(N) = \textsf{Rank}(T)
\end{aligned}$$

- **Compress:** Suppose we **Compress** the **up-tree** S to create the next tree N.

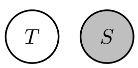
Goal: Show $\text{Nodes}(N) \geq 2^{\text{Rank}(N)}$, i.e. $\text{Nodes}(S) \geq 2^{\text{Rank}(S)}$

$$\begin{aligned} \text{Nodes}(N) &= \text{Nodes}(S) && \text{since } \text{Nodes}(N) = \text{Nodes}(S) \\ &\geq 2^{\text{Rank}(S)} && \text{IH} \\ &= 2^{\text{Rank}(N)} && \text{since } \text{Rank}(N) = \text{Rank}(S) \end{aligned}$$

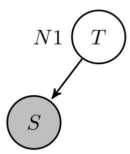
So $P()$ holds for all recursive steps.

Therefore $P(T)$ holds for all **up-tree**s T by structural induction.

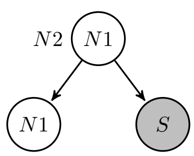
For more intuition on the rules, we have included some examples below. **There are no new questions on this page.** But you may use the space below if you run out of room on the previous page.



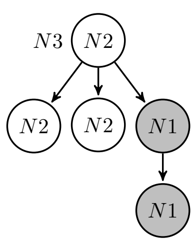
Here, T and S are two **up-tree**s as defined by our basis step. Thus, we have the following:
$\text{Node}(T) = \text{Node}(S) = 1$
$\text{Rank}(T) = \text{Rank}(S) = 0$



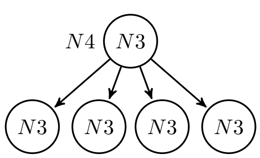
$N1$ is created by combining T and S with **Combine 1**. We take the root of T and make it the root of $N1$, then append the root of S as a child of T. Thus, we have the following:
$\text{Node}(N1) = \text{Node}(S) + \text{Node}(T) = 2$
$\text{Rank}(N1) = 1 + \text{Rank}(T) = 1$



$N2$ is created by combining $N1$ and S with **Combine 2.** $\text{Rank}(S) < \text{Rank}(N1)$, so we take the root of $N1$ and make it the root of $N2$, then append the root of S as a child of $N1$. Thus, we have the following:
$\text{Node}(N2) = \text{Node}(S) + \text{Node}(N1) = 3$
$\text{Rank}(N2) = \text{Rank}(N1) = 1$



$N3$ is created by combining $N1$ and $N2$ with **Combine 1**. We take the root of $N2$ and make it the root of $N3$, then append the root of $N1$ as a child of $N2$. Thus, we have the following:
$\text{Node}(N3) = \text{Node}(N1) + \text{Node}(N2) = 5$
$\text{Rank}(N3) = 1 + \text{Rank}(N2) = 2$



$N4$ is created by **compressing** $N3$. We take the root of $N3$ and make all other nodes in $N3$ direct children of the root. Thus, we have the following:
$\text{Node}(N4) = \text{Node}(N3) = 5$
$\text{Rank}(N4) = \text{Rank}(N3) = 2$

7. Wait, that's illegal [15 points]

Choose to do exactly one of these two problems.

(a) Let $L = \{0^k 1^j 0^k \mid j, k \in \mathbb{N}, j \neq k\}$. Prove that L is not regular.

Solution:

This is essentially the proof that binary palindromes can't be recognized by any DFA.

Let $L = \{0^k 1^j 0^k \mid j, k \in \mathbb{N}, j \neq k\}$. Let D be an arbitrary DFA, and suppose for the sake of contradiction that D accepts L. Consider $S = \{0^k 1 \mid k \geq 0, k \neq 1\}$. 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. These strings are in the form $0^i 1$ and $0^j 1$ for some $i, j \geq 0$ where $i, j \neq 1$, and we have $i \neq j$ since these are two different strings.

Now, we append the string 0^i to both of these strings. The two resulting strings are:

$a = 0^i 1 0^i$ Note that $a \in L$.

$b = 0^j 1 0^i$ Note that $b \notin L$, since $i \neq j$.

Since a and b end up in the same state, D must either accept both strings or reject both strings. However, since $a \in L$ and $b \notin L$, it must give the wrong answer for one string, which contradicts that it accepts L. Thus, our only remaining assumption, that D exists, must be false.

Since D was arbitrary, there is no DFA that recognizes L, so L is not regular.

(b) Let S be the set of all real numbers in $[0, 1)$ such that the even-indexed digits after the decimal point in its decimal expansion are all 0. Prove that S is uncountably infinite.

For example $\frac{1}{100} = 0.010000....$ has a 0 at indices $0, 2, 3, 4, 5, ...$ and a 1 at index 1. Since it has 0's at indices $0, 2, 4, ...$, it is in S.

$\pi - 3 = 0.14159... \notin S$, since the number at index 0 is 1.

Solution:

Suppose S is countable. Then there exists a surjection $f : \mathbb{N} \to S$. Then, for each natural number i we have some decimal sequence that i maps to.
We construct an element x, x starts with $0.$(0 before decimal point), and the $(2i + 1)th$ digit after the decimal point equals to 3 if the $(2i + 1)th$ digit of $f(i)$ doesn't equal to 3(or doesn't have $(2i + 1)th$ digit), and equals to 4 otherwise, for all $i \in \mathbb{N}$. And the $(2i)th$ digit after decimal point be 0.
For any $i \in \mathbb{N}$, by our construction, $f(i)$ differs from x on the $(2i + 1)th$ digit. And by our construction the even-indexed digits after the decimal point of x is always 0, and $x \in [0, 1)$, therefore $x \in S$. Since x is not in the range of f, f is not a surjection, and therefore is not a bijection, thus a contradiction. Therefore S is uncountable.

8. Grading Morale [1 point]

Put something on this page. Depending on your mood, this might be a poem, description of favorite NFT[1], or a piece of art. Or you can tell us what you believe Oob's story may be.

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

Solution:

Gumball is proud of you for making it through the quarter! :D

[1]Please, don't actually buy NFTs. All the NFT problems this quarter were jokes. Seriously, don't get NFTs.