CSE 311 : Final Exam Solutions

Instructions

- This is a take home exam. It is due at **NOON** on Thursday December 17.
- Remember to follow the collaboration policy:
  - You are permitted to work with at most 3 other students in the class; you may discuss the exam only with those students and the course staff.
  - As always, you must write up your work independently.
- This exam is open-book/open-note. You may use any resource the staff has provided, as well as external resources.
- But you may not search for the particular problems in the exam.
- Submit your responses to gradescope. You may handwrite or typeset solutions (as long as they are legible). We do not recommend trying to write on this document; we have not left enough room.
- If you have a question, please check the pinned post on Ed where we've posted common clarifications. If you still have a question, please ask a **private** question on ed.

Advice

- Remember to properly format English proofs (e.g. introduce all your variables).
- All proofs for this exam will 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**
Quantify Your Understanding	16
Set Proofs	20
Bug	6
Scorigami	26
Structural Induction	15
Some Strings Attached	18
Wait, That's Illegal	15
The Hard One at the End	7
End Matter	2
Total	**125**

First Half Fundamentals

1. Quantify your understanding [16 points]

Translate each of the following sentences into predicate logic. Let your domain of discourse be strings and languages. You may use the following notation:

- $\texttt{regular}(x)$ returns true if and only if x is a regular language. (Returns false on non-regular languages and on strings).
- $\texttt{string}(x)$ (returns true iff x is a string) and $\texttt{language}(x)$ (returns true iff x is a language) to do domain restriction
- Common set notation: $\subseteq, \in, \cup, =, \{, \}$, etc.

(a) The union of two regular languages is regular. [4 points] **Solution:**

$\forall x \forall y((regular(x) \wedge regular(y)) \rightarrow regular(x \cup y))$

(b) Every non-regular language has a regular subset. [4 points] **Solution:**

$\forall x(\neg regular(x) \rightarrow \exists y((y \subseteq x) \wedge regular(y)))$

(c) If L is regular, then for every string x, $L \cup \{x\}$ is regular. [4 points] **Solution:**

$\forall L(regular(L) \rightarrow \forall x(string(x) \rightarrow regular(L \cup \{x\})))$

Contrary to popular belief, contrapositives are fun!

(d) State the contrapositive of part (c) in English. [4 points]

2. Set Proofs [20 points]

Let R be a relation on the integers (i.e. $R \subseteq \mathbb{Z} \times \mathbb{Z}$). We will use xRy to mean $(x, y) \in R$.

For a set A, define $\text{input}(R, A) = \{x : \exists y(y \in A \wedge xRy)\}$. While it might look different on the surface, $\text{input}(R, A)$ is closely related to $\text{output}(f, A)$ from the midterm.

For a few examples: $\text{input}(|, \{3\}) = \{-3, -1, 1, 3\}$ where $|$ is the divides relation.
$\text{input}(|, \{3, 6\}) = \{-6, -3, -2, -1, 1, 2, 3, 6\}$.
$\text{input}(S, \{1, 4, 9\}) = \{-3, -2, -1, 1, 2, 3\}$, where $S = \{(x, y) : x^2 = y\}$.

(a) Calcluate $\text{input}(T, \{0, 5\})$ where $T = \{(x, y) : |x| = y\}$. (Hint, there are three elements) [3 points] **Solution:**

$\text{input}(T, \{0, 5\}) = \{-5, 0, 5\}$

Take a deep breath. There's more notation to come. If you didn't find a) to be quick, you might want to do a few more examples of input before moving on.

(b) Call a relation R "functional" if $\forall x \exists y(xRy \wedge \forall z[xRz \wedge xRy \rightarrow y = z])$, that is, every x has exactly one element it relates to. Show that the relation $\leq$ (the normal less-than-or-equal-to on $\mathbb{Z}$) is not functional. [3 points] **Solution:**

From algebra, we know that $2 \leq 3$ and $2 \leq 4$. Since $3 \neq 4$, by definition we know that $\leq$ is not functional.

(c) Show that if R is functional, then $\text{input}(R, A) \cap \text{input}(R, B) \subseteq \text{input}(R, A \cap B)$ [10 points] Before starting this proof, you might want to see that it holds on a particular example. We recommend you compute each of the sets for $R = \{(x, y) : x^2 = y\}$, $A = \{16, 25\}, B = \{25, 36\}$. **Solution:**

Goal: $functional(R) \rightarrow (\text{input}(R, A) \cap \text{input}(R, B) \subseteq \text{input}(R, A \cap B))$.

Suppose that R is a functional relation, and suppose that x be an arbitrary element of $\text{input}(R, A) \cap \text{input}(R, B)$. By definition of intersection, $x \in \text{input}(R, A)$ and $x \in \text{input}(R, B)$, which means that $y \in A$, xRy, $z \in B$, and xRz for some element y and z. Since we know that R is functional, by definition, this means that $y = z$ and combining with the previous claims, we can conclude that $y \in A$, $y \in B$, and xRy. By definition of intersection, $y \in A \cap B$ and since xRy, it follows that $x \in \text{input}(R, A \cap B)$.

Since x was arbitrary, $\text{input}(R, A) \cap \text{input}(R, B) \subseteq \text{input}(R, A \cap B)$.

(d) Show that the functional requirement is necessary.
I.e. disprove that $\forall R \forall A \forall B[\text{input}(R, A) \cap \text{input}(R, B) \subseteq \text{input}(R, A \cap B)]$. [4 points] **Solution:**

We will disprove the claim with a counter-example. Let $A = \{1\}$, $B = \{2\}$, and $R = \{(x, y) : x \leq y\}$.

Because $1 \in A$ and $0 \leq 1$, by definition, we know that $0 \in \text{input}(R, A)$. Also, because $2 \in B$ and $0 \leq 2$, by definition, we know that $0 \in B$. Therefore, by definition of intersection, we know that $0 \in \text{input}(R, A) \cap \text{input}(R, B)$.

Since $A \cap B = \emptyset$, we know that $0 \notin \text{input}(R, A \cap B)$ by definition since $y \in A \cap B$ will always be false.

We have shown that $0 \in \text{input}(R, A) \cap \text{input}(R, B)$ and $0 \notin \text{input}(R, A \cap B)$, therefore, by definition, $\text{input}(R, A) \cap \text{input}(R, B) \not\subseteq \text{input}(R, A \cap B)$.

Induction: Because it's always safe to assume I'm right.

3. Insect Puns Bug Me [6 points]

For this problem, define a "tree" as a directed graph such that:

- It has exactly one vertex ("the root") with no incoming edges.
- It has zero or more other vertices, each with exactly one incoming edge.
- There is a path from the root to every vertex.

Call the "degree" of a vertex the number of edges that touch it (whether they are coming in or leaving). You friend has written the following induction proof. It is not correct.

Let $P(n)$ be "every tree with n vertices has a vertex of degree one."
We show $P(n)$ for all $n \geq 2$ by induction on n.

Base Case: $n = 2$, There is only one tree with two vertices – one where the root connects directly to the other vertex (any other graph would be disconnected or have two many edges entering one of the vertices). In that graph, the one non-root vertex has degree one.

Inductive Hypothesis: Suppose $P(2) \wedge \cdots \wedge P(k)$ for an arbitrary $k \geq 2$.

Inductive Step:
Consider an arbitrary tree with k vertices. By Inductive Hypothesis, the tree has a vertex of degree one, call it v. Add a new vertex u. We consider the ways to connect u.
Case 1: There is an edge from v to u. Since u can have only one incoming edge, u cannot have any other edges, so it has degree one.

Case 2: There is an edge from u to v. In this graph, v would have two incoming edges, so it is not a tree and $P()$ vacuously holds.

In either case, $P()$ holds, so we have $P(k+1)$.

By the principle of induction, we have $P(n)$ for all $n \geq 2$.

(a) Give an example of a tree which does not fall into the base case, nor would be covered by any of the cases of the inductive step. [3 points] **Solution:**

A tree where the root node has 3 children fits all the definition of a tree defined in this problem, but does not fall into the base case nor any of the cases in the inductive step.

Any drawing/tree description with: a self-loop or a node with more than two children, or a non-root node with two children is also correct.
Note that a three node tree, where the root has 2 children is not correct (it falls into case 1 where v is the root).

(b) What would be the first sentence of a correct inductive step (i.e. the correct starting point)? You do not need to write the full step. Hint: what kind of claim is $P()$? [3 points] **Solution:**

The correct start should be "Consider an arbitrary tree with $k+1$ vertices". In the inductive step, it is not valid to start with an arbitrary tree with k vertices and try to build up to the tree of $k+1$ vertices.

4. Scorigami [26 points]

Scorigami[1] is the art of tracking the possible final scores in your favorite game. In this problem, we'll play scorigami for a game Robbie just made up.

Robbie's game is played over rounds between two players. At the start of the game, each player has 0 points. In each round, exactly one of the following things occurs:

1. One player gets 3 points, and the other gets 1 point.
2. One player gets 4 points, and the other gets 0 points
3. One player gets 6 points, and the other gets 3 points.

A game may end after any number of rounds, as long as the players have a different number of total points. When the game is over, the player that has more points is the winner.

(a) Prove by induction that the losing player could end the game with any natural number of points. [8 points] **Solution:**

Define $P(n)$ is "a losing player may end the game with n points." We show $P(n)$ for all natural numbers by induction on n.

Base Case $n = 0$: A losing player may end the game with 0 points if the first round uses rule 2, and the game ends after that round.

Induction Hypothesis: Suppose that $P(k)$ holds for an arbitrary integer k.

Induction Step: We show $P(k+1)$. By inductive hypothesis, the game could end with the loser scoring k points. Call the winner a and the loser b. Suppose our game plays out identically, but does not end after the (previously) final round. One more round is played with a getting 3 points, and b getting 1. b now has $k+1$ points, and a has more than b (It already did prior to the round starting, since it won), thus a is still ahead. End the game after this round, and b will lose with $k+1$ points.

Thus $P(n)$ holds for all natrual numbers by induction.

(b) Prove by induction that the winning player could end the game with n points for every $n \geq 3$ (prove by induction on n – be careful, make sure the player is the winner!) [16 points] **Solution:**

Let $Q(n)$ be the predicate "The winner may end the game with n points"

Base Cases
$n = 3$ A single round is played, ending according rule 1, has the winner winning 3 to 1.
$n = 4$ A single round is played, ending according to rule 2, has the winning player winning 4 to 0.
$n = 5$ Two rounds are played. In the first a gets 4 points, b gets 0. In the second a gets 1 point, and b gets 3. a leads 5 to 3 when the game ends.

Inductive Hypothesis
Suppose $Q(n)$ holds for $n = 3, \ldots, k$ for an arbitrary integer k, $k \geq 5$.

Induction Step We show $Q(k+1)$. By the inductive hypothesis applied to $k-2$, we have that the winner can end the game with $k-2$ points (observe that since $k \geq 5$, $k-2 \geq 3$, so the IH does indeed apply).

Let a be the winner of the game guaranteed by the inductive hypothesis. And suppose one more round was played with a getting 3 points and b getting 1 (and have the game end). a is still the winner (their lead having only increased in the last round), and they end with $k-2+3 = k+1$ points, giving $Q(k+1)$.

Conclusion $Q(n)$ holds for all integers $n \geq 3$ by indcution.

[1] As defined by Jon Bois in this video; if you enjoy football, you may enjoy this video. If you don't, you won't miss anything by not watching it.

Solution:

REMARK: You cannot do an inductive step applying rule 1 with the winner getting one point. There is no guarantee that the winner remains the winner after that application.

(c) Can you combine the two claims to get: for every $n \geq 3$ and every $m \geq 0$, if $n > m$ then a game can end with the winning player having n points, and the losing player having m points?
If yes, describe in 1-3 sentences how you could prove it (e.g. how you would modify or combine your proofs above); if not, give 1-3 sentences explaining why the statement is false.
You do not need to give a formal proof in either case. [2 points] **Solution:**

The claim is false. There is no way for a game to end 3 to 0 (every round awards a total of at least 4 points, and at least one round must be played to break the tie).

5. Structural Induction [15 points]

Consider the following context-free-grammar, which generates the language L:

$$S \rightarrow 00S1|S010010|010S|\varepsilon$$

We could rewrite this CFG using the following recursive definition:

Basis: $\varepsilon \in L$
Recursive: if $x \in L$ then $00x1 \in L$, $x010010 \in L$ and $010x \in L$.

Prove that $\forall x \in L$, the number of 0s in x is twice the number of 1's in x.

You must use structural induction for this problem.

Solution:

(1) Define 2 new functions '$numOfZeros(x)$' and '$numOfOnes(x)$' where the input x is a string and it returns the number of zeros and ones in the string respectively. Let $P(x)$ be "$numOfZeros(x) = 2 \cdot numOfOnes(x)$". We will show $P(x)$ is true for all $x \in L$ by structural induction.

(2) **Base Case** $(x = \varepsilon)$**:**

By definition, ε does not contain any characters. In other words, $numOfZeros(\varepsilon) = 0$ and $numOfOnes(\varepsilon) = 0$.

$$\begin{aligned} numOfZeros(\varepsilon) &= 0 \\ &= 2(0) \\ &= 2 \cdot numOfOnes(\varepsilon) \end{aligned}$$

Therefore, $P(\varepsilon)$ holds.

(3) **Inductive Hypothesis:** Suppose that $P(a)$ is true for some arbitrary a in L.

(4) **Inductive Step:**

We must show that P holds for all of our recursive cases.

Goal: Show $P(00a1)$.

$$\begin{aligned} numOfZeros(00a1) &= 2 + numOfZeros(a) && \text{since “00” adds 2 zeros to } a \\ &= 2 + 2 \cdot numOfOnes(a) && \text{By IH} \\ &= 2\,(1 + numOfOnes(a)) \\ &= 2 \cdot numOfOnes(00a1) && \text{since “1” adds 1 one to } a \end{aligned}$$

Therefore, $P(00a1)$ holds.

Goal: Show $P(a010010)$.

$$\begin{aligned} numOfZeros(a010010) &= 4 + numOfZeros(a) && \text{since “0”, “00”, “0” adds 4 zeros to } a \\ &= 4 + 2 \cdot numOfOnes(a) && \text{By IH} \\ &= 2\,(2 + numOfOnes(a)) \\ &= 2 \cdot numOfOnes(a010010) && \text{since “1”, “1” adds 2 ones to } a \end{aligned}$$

Therefore, $P(a010010)$ holds.

Goal: Show $P(010a)$.

$$\begin{aligned} numOfZeros(010a) &= 2 + numOfZeros(a) && \text{since “0” and “0” adds 2 zeros to } a \\ &= 2 + 2 \cdot numOfOnes(a) && \text{By IH} \\ &= 2\,(1 + numOfOnes(a)) \\ &= 2 \cdot numOfOnes(010a) && \text{since “1” adds 1 one to } a \end{aligned}$$

Therefore, $P(010a)$ holds.

(a) **Conclusion:** Therefore, P(x) is true for all $x \in L$ by structural induction.

The Last Few Weeks

6. Some strings attached [18 points]

(a) Write a regular expression for the language $\{xy : x$ starts with 1 and ends with 0, y starts with 0 and ends with $1\}$
Solution:

$1(0 \cup 1)^*00(0 \cup 1)^*1$.

(b) Write a context free grammar for the language $\{0^m1^n : m \not\equiv n \pmod 3\}$
If your grammar has more than one symbol: tell us which non-terminal is the start symobl and for every other non-terminal tell us what strings it generates. **Solution:**

$S \to 000S|S111|0S1|00|11|0|1$

OR

$$S \to 0S1|ABC$$
$$A \to 000A|\varepsilon$$
$$C \to C111|\varepsilon$$
$$B \to 00|11|0|1$$

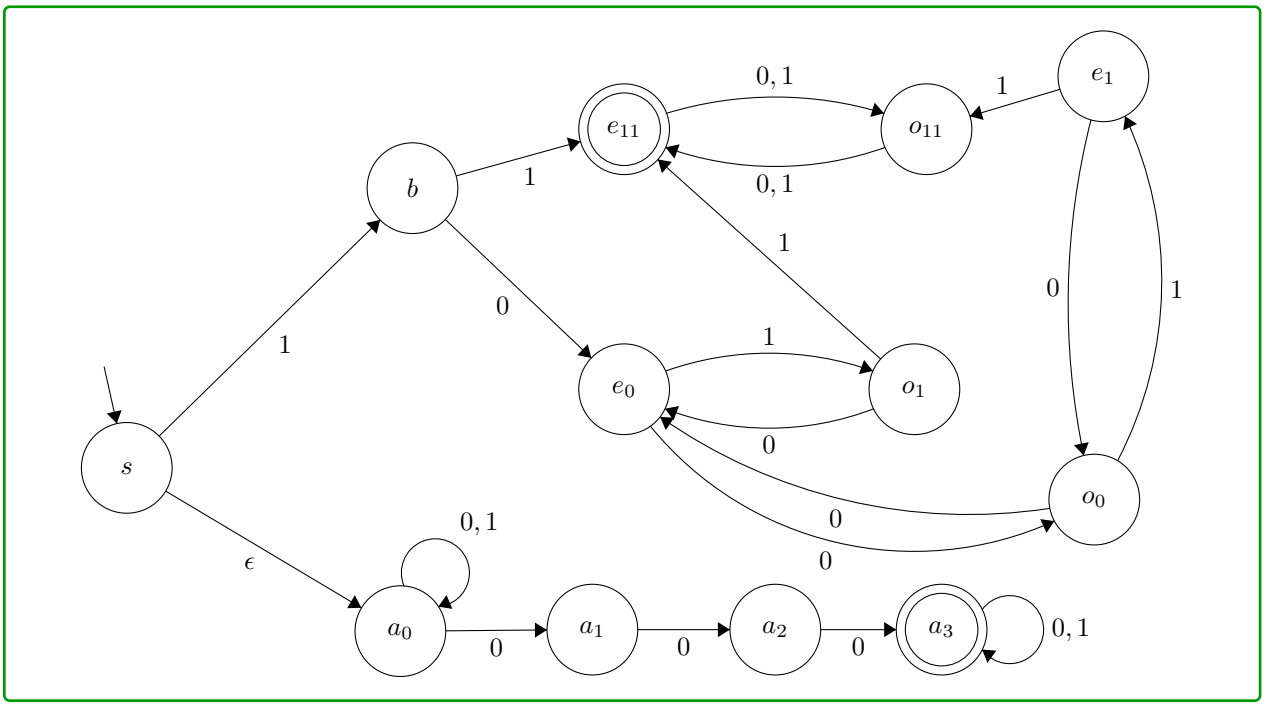
OR

$$S \to 0S1|A|B$$
$$A \to 000A|N$$
$$B \to B111|N$$
$$N \to 00|11|0|1$$

OR anything equivalent.

(c) Draw an NFA for the language $\{x : [\text{len}(x) \equiv 0 \pmod 2) \wedge x \text{ starts with a } 1 \wedge x \text{ contains } 11] \vee [x \text{ contains } 000]\}$. Please name your states, and briefly explain how your machine works (2-3 sentences total).

Solution:



For drawing the NFA, you may want to use an interface like http://madebyevan.com/fsm/

7. Wait, That's Illegal [15 points]

Choose **one** of the two problems to do. Do not submit attempts at both.

7.1. This is Highly Irregular

Use the method from class (i.e. the one in lecture 27) to prove that $\{w\#w|w \in \{0,1,2\}^*\}$ is not regular. (# is just a character here, i.e. for this problem $\Sigma = \{0,1,2,\#\}$) **Solution:**

(1) Suppose for contradiction that an arbitrary DFA, M, recognizes $L = \{w\#w|w \in \{0,1,2\}^*\}$.

(2) Consider the infinite set $S = \{0^n : n \geq 0\}$.

(3) Since S is infinite and M contains a finite number of states, there must be 2 strings:

$$\text{String } s_1 = 0^a$$
$$\text{String } s_2 = 0^b$$

in S for some $a, b \geq 0$ and $a \neq b$ that end up in the same state M.

(4) Consider appending the string $t = \#0^a$ to both strings s_1 and s_2.

(5) Since s_1 and s_2 both end up at the same state in M and we appended t to both s_1 and s_2, s_1t and s_2t must also end up in the same final state in M. Since, $s_1t \in L$ and $s_2t \notin L$, M does not recognize L.

(6) Since M was arbitrary, there is no DFA that recognizes L, and therefore L is irregular.

7.2. 1,2,3,...∞

A "Turing Machine" is a common abstraction for normal computers. The memory of a Turing Maching is an infinite "tape." The tape is divided into cells (think: bits) that are either 0 1 or $\sqcup$ ($\sqcup$ corresponds to "not initialized yet"). The tape is infinite in the following sense: for every integer i, there is a cell i of the tape (i.e., the nubmer of cells is countable).

Show that the number of possible tapes is uncountable by adapting Cantor's Diagonalization argument.

Solution:

Suppose for the sake of contradiction, that the number of possible tapes is countable. And let g be a bijection from the natural numbers to the set of possible tapes.

We know from class that $\mathbb{Z}$ is countable, so let $f()$ be a bijection from the natrual numbers to the integers.

We now build a table – Row i will correspond to the i^{th} tape (as ordered by g), and column j will correspond to the j^{th} cell of the tape (as ordered by $f()$).

More formally, for $i, j \in \mathbb{N}$, let $T_{i,j}$ be the contents of cell $f(j)$ on tape $g(i)$.

Consider the diagonal entries of the table $T_{i,i}$. We use those entries to build a new tape.

Specifically, the $f(i)^{\text{th}}$ entry of the tape will be:

$$\begin{cases} 1 & \text{if } T_{i,i} = 0 \text{ or } \sqcup \\ \sqcup & \text{if } T_{i,i} = 1 \end{cases}$$

Observe that since $f()$ is a bijection between cell lables and natural numbers, by considering every natural number i we have defined a full tape. Call this new tape N. We claim that N is not on the list – indeed it cannot be $g(k)$ for any k, as N differs from $g(k)$ in cell $f(k)$, by construction.

Thus N is not on the list, but that contradicts g being a bijection!

So we have that the number of possible tapes is uncountable.

8. The really hard one at the end [7 points]

This is (in Robbie's opinion) the hardest problem on the exam. We've intentionally made it worth a small amount of points so that you do not feel bad about giving up if you get stuck.

A **Tourist Machine** is a finite automaton: it has a finite set of states, a start state, and transitions between states labeled with an $x \in \Sigma$. For every state q and every character x, there is at least one transition labeled with x leaving q (so computation can't "die").
A Tourist Machine processes a string much like an NFA does – it can read a character from the string and follow a transition (of its choice if more than one) labeled with that character. But a Tourist Machine does not have any final states. Instead, a Tourist machine accepts input x if and only if the machine can vist all of its states while processing x.

The language of a Tourist Machine is the set of all strings that it accepts.

In this problem you'll show Tourist Machines are less powerful than DFAs.

(a) Suppose L is the language of a Tourist Machine. Give a construction that shows L must be regular (i.e. tell us how to make an NFA/DFA/regex). Include all the details we would need to build your regular expression, NFA, or DFA for any particular Tourist Machine. You do not have to formally prove your construction is correct, but you should write 2-3 sentences briefly explaining why it works. [4 points]

Solution:

Intuition: The key idea is to adapt the idea of the powerset construction (to keep track of which states you have visited, instead of which states you could be in) and combine that with the idea of the cross-product construction (i.e. simulate the Tourist Machine)

Let L be an arbitrary language of a Tourist Machine, and let M be a machine whose language is L.

Let S be the set of states in M. We build an NFA, N, as follows:

- N's set of states is $\mathcal{P}(S) \times S$.
- For every state (T, q) of N, have a transition to $(T \cup \{q'\}, q')$ labeled with character c for every q' such that M has a transition from q to q' on character c.
- (S, q) will be a final state for every q (and no other states are final states).
- The start state is $(\{q_0\}, q_0)$ where q_0 is the start state of M.

(b) Prove that there is a regular language L that is not the language of a Tourist Machine.
Hint: think about languages where our DFAs definitely wouldn't visit every state in an accepting computation. It is not enough to just state a language here. You must give a regular language, **and** prove that no Tourist Machine can accept that language. [3 points]

Solution:

Consider the regular language $\{\varepsilon\}$.

Suppose, for the sake of contradiction, that there is a Tourist Machine M that accepts L. Since ε is accepted by the Tourist Machine, it must have only one state. But then consider how the Tourist Machine behaves on the string 1. By the definition of a Tourist Machine, the computation cannot die, and we've already visited every state! So 1 must be accepted. But then the language of M isn't really L. That's a contradiciton! So $\{\varepsilon\}$ is not the language of any Tourist Machine.

Feedback [1 point]

So we can help 311 instructors teaching remotely this year:

(a) How long did you spend studying for the final **before** working on the final (estimate to the nearest hour)

(b) How long did you work on the final overall (estimate to nearest hour)

(c) Of the time in (b), how long was "studying" (e.g. looking up old lectures or old problems)

(d) Of the time in (b), how long was making the pretty final submission (typing or writing/uploading/finding the best wording/checking your answers)

For grading morale [1 point]

Produce a piece of art to reflect how you will relax over Winter break, now that you won't have 311 homework. Suggested media include drawing, poetry, or short story.
The TAs will use these for morale boosts during grading breaks.

Any non-empty submission will receive full credit.

Gumball is proud of you. He says you deserve a nap.