Final Exam - Solutions

Logical Formulas and Inference Rules

1. For each candidate inference rule below, indicate if it is *sound* or *unsound* (circle the correct answer). For the rules that are unsound, provide a counter-example to show it is unsound. You do not need to provide any justification for the rules that are sound.

   a. $\dfrac{P}{Q}$

   (3) **Unsound.** Counter-example: $P =$ **True**, $Q =$ **False**.

   b. $\dfrac{(P \wedge Q) \vee (P \wedge \overline{Q})}{P}$

   (2) **Sound.** If P is **False**, there is no way to make the antecedent true since each clause is a conjunction that includes P.

   c. $\dfrac{(P \wedge Q) \wedge (P \wedge \overline{Q})}{P}$

   (2) **Sound.** If P is **False**, there is no way to make the antecedent true since each clause is a conjunction that includes P.

   d. $\dfrac{(P \implies Q) \wedge Q}{P}$

   (3) **Unsound.** Counter-example: $P =$ **False**, $Q =$ **True**.

Well Ordering

2. For each set and operator below, answer if the set is *well-ordered* or not. Support your answer with a brief, but clear and convincing, argument.

a. the even natural numbers; $<$.

(2) **Well-Ordered**. The even natural numbers is a subset of $\mathbb{N}$, which we know is well-ordered by $<$, and every subset of a well-ordered set is well-ordered.

b. the non-negative real numbers; $<$.

(3) **Not Well-Ordered.** The subset $\mathbb{S} - \{0\}$ has no least element, so the set is not well-ordered. We could prove this by contradiction. Assume (to get a contradiction) that there is a minimum element m. Since m is a non-negative real number, it is some sequence of digits, $m_1 m_2 m_3 \ldots$. The number $0 m_1 m_2 m_3 \ldots$ is a real number less than m. Hence, we have a contradiction and the set is not well-ordered. (Not necessary to prove this for full credit, but +2 bonus for a proof.)

c. the empty set; $<$.

(2) **Well-Ordered.** All subsets of the empty set have a least element, since there are no subsets of the empty set.

d. pow($\mathbb{N}$); compare$(S, T) := |S| < |T|$

(3) **Not Well-Ordered.** Here's an example of a subset of pow($\mathbb{N}$) that is not well-ordered by the comparator: $\{s_1 = \{0\}, s_2 = \{1\}\}$. Since the comparison is based on the sizes, there is no least element: neither $|s_1| < |s_2|$ or $|s_2| < |s_1|$.

Sets and Relations

3. Indicate for each statement if it is valid (always true) or invalid. For invalid statements, provide a counter-example supporting your answer.

a. For any sets A and B, $|A \cup B| \leq |A| + |B|$.

(2) **Valid.** $|A \cup B| = |A| + |B| - |A \cap B| \leq |A| + |B|$.

b. For any sets A and B, $(\forall a \in A.\, a \in B) \implies A \subseteq B$.

(2) **Valid.** Follows directly from the definition of $\subseteq$.

c. For any sets A and B, there exists a total ([≥ 1 arrow out]), injective ([≤ 1 arrow in]) relation R between A and $A - B$.

(3) **Invalid.** Counter-example: $A = B$. Then, $A - B = \emptyset$, so there is no way to have arrows out of A in R.

d. For any sets A and B, $B \in \text{pow}(A) \implies |B| < |A|$.

(3) **Invalid.** $A \in \text{pow}(A)$.

Induction

4. Prove by induction that every finite non-empty subset of the real numbers contains a *least* element, where an element $x \in S$ is defined as the least element if $\forall z \in S - \{x\}. x < z$. (Note: you should not just assume all finite sets are well ordered for this question.)

(This question was on Exam 2, Exam 1, and Problem Set 5, with minor modifications, and you were notified in the Exam 2 solutions that you would see it again on the final. So, everyone should have gotten a good proof for this!)

Our induction predicate is:

$$P(n) ::= \text{all sets of real numbers of size } n \text{ have a } \textit{least} \text{ element.}$$

We what to show this holds for all non-empty sets, $n \in \mathbb{N}^+$.

Base case: $n = 1$. Consider a set of real numbers with size one: $\{x\}$. It has only a single element, x, so that element will be the minimum.

Inductive step: $\forall m \in \mathbb{N}^+. P(m) \implies P(m+1)$. Assume all sets of real numbers of size m have a least element. Every set of size $m+1$, S', is the result of adding a new element, q, to a set of size m: $S' = S \cup \{q\}$, $q \notin S$. By the inductive hypothesis, we know S (size m) has a least element, w. There are two possibilities for the new element: (1) $q < m$. Then, minimum$(S') = q$. or (2) $q > m$. Then, minimum$(S') = m$. In both cases, the minimum of S' is well defined.

By induction, we have proven $P(n)$ holds for all $n \in \mathbb{N}^+$.

State Machines

5. Consider the state machine, $M_1 = (S, G, q_0)$ below:

$$\begin{aligned} S &= \{(a,b) \mid a, b \in \mathbb{N}\} \\ G &= \{(a,b) \rightarrow (a',b') \mid a, a', b, b' \in \mathbb{N} \wedge a + b = a' + b'\} \\ q_0 &= (0,0) \end{aligned}$$

a. Describe the reachable states for M_1.

(4) There is only one reachable state: $(0,0)$. Since the transitions rule requires $a' + b' = a + b$, if $a + b = 0$ (as it does for state $q_0 = (0,0)$), there are no other natural numbers that some to 0.

(If you misread the $\mathbb{N}$ as $\mathbb{Z}$, which is a more sensible state machine, then the reachable states are the states $(a, -a)$ for all $a \in \mathbb{Z}$, which was also worth full credit.)

For each of the following predicates, answer whether or not it is a *preserved invariant* for M_1 as defined above, and provide a brief justification supporting your answer.

b. $P(q = (a,b)) := a > b$

(3) **Not Preserved**. There is a transition from $(3,2) \rightarrow (1,4)$, but the property is satisfied by $(3,2)$ but not by $(1,4)$.

c. $P(q = (a,b)) := a + b$ is odd

(3) **Preserved**. The transitition rule ensures $a' + b' = a + b$, so if $a + b$ is odd, so is $a' + b'$.

Cardinality

6. For each set defined below, answer is the set if *countable* or *uncountable* and support your answer with a convincing and concise proof. (Recall that $\mathbb{N}$ is the set of natural numbers, $\mathbb{R}$ is the set of real numbers.)

a. set of all subsets of students in cs2102

(2) **Countable.** The number of students in cs2102 is finite, so the size of its powerset of also finite. All finite sets are countable.

b. $\{(a, b)|a, b \in \mathbb{N}\}$

(3) **Countable.** This set is the same as $\mathbb{N} \times \mathbb{N}$, which we already proved is countable. Alternately, you should show a bijection between $\mathbb{N}$ and the set:

$$0 \leftrightarrow (0, 0), 1 \leftrightarrow (0, 1), 2 \leftrightarrow (1, 0), 3 \leftrightarrow (1, 1), 4 \leftrightarrow (1, 2), \ldots$$

dove-tailing through the pairs similarly to the proof that the rationals are countable.

c. $\{(a, b)|a \in \mathbb{N}, b \in \mathbb{R}\}$

(2) **Uncountable.** Since $\mathbb{R}$ is uncountable, any sequence that includes an element from $\mathbb{R}$ must also be uncountable.

d. the set of all Turing Machines that accept no strings

(3) **Countable.** This is a subset of the set of all Turing Machines, which we know is countable. Any subset of a countable set must be countable.

Proofs

Consider the Take-Away game: start with n sticks; at each turn, a player must remove 1 or 2 sticks. The player who takes the last stick wins.

7. Prove that Player 1 has a winning strategy for a two-player game of Take-Away where Player 1 moves first if the initial number of sticks is n is not divisible by 3.

We prove using strong induction. The predicate is:

$P(n)$ ::= Player 1 has a winning strategy for a Take-Away game starting with n sticks, if n is not divisible by 3.

We want to show $P(n)$ holds for all positive natural numbers.

Base cases: $P(1)$ and $P(2)$. Player 1 can win by removing 1 (for $n = 1$) stick or 2 (for $n = 2$) sticks on her first move.

Inductive case: $\forall m \in \mathbb{N}, m > 2. \forall k \leq m. P(k) \implies P(m+1)$.

We have three cases to consider:

(1) $m = 3n$. To show $P(3n+1)$, Player 1 can remove 2 sticks, leaving $3n - 1 = 3(n-1) + 2$ sticks. Since this is not divisible by three, by the strong induction hypothesis, we know $P(3(n-1)+2)$ is true, so this is a winning strategy, showing $P(m+1)$ for this case.

(2) $m = 3n + 1$. To shows $P(3n+2)$, Player 1 can remove 1 stick, leaving $3n + 1$ sticks. Since this is not divisible by 3, by the strong induction hypothesis, we know $P(3n+1)$ is true, so this is a winning strategy.

(3) $m = 3n + 2$. To show $P(3n+3)$, we observe that $3n + 3$ is divisible by 3. Hence, $P(3(n+1))$ holds, since the predicate only says there needs to be a winning strategy then n is divisible by 3.

The three cases show $P(m+1)$ holds in all cases, proving $P(n)$ holds for all $n \in \mathbb{N}^+$.

Program Correctness

Consider the Python program below, that returns `True` if and only if none of the elements of the input list are below 5. You may assume `p` is a non-empty list of natural numbers.

```
def all_good(p):
   i = 0
   good = True
   while i < len(p):
      if p[i] < 5:
         good = False
      i = i + 1
   return good
```

(Note: this was from the program used to evaluate grades, giving a data point if all the problem sets were at least at the "green star" level. The actual Python code I used is `all([ps >= 5 for ps in stud.ps])` which is more concise, but should be equivalent to the code above.)

8. Complete the definition of the state machine, $M_g = (S, G, q_0)$, below that models `all_good`.

$$
\begin{aligned}
S &= \{(i, g) \,|\, i \in \mathbb{N}, g \in \{\textbf{True}, \textbf{False}\}\} \\
G &= \{(i, g) \to (i', g') \mid \\
&\qquad i, i' \in \mathbb{N}, g, g' \in \{\textbf{True}, \textbf{False}\} \\
&\qquad \wedge \underline{i < \text{len(p)}} \\
&\qquad \wedge i' = \underline{i + 1} \\
&\qquad \wedge g' = \begin{cases} \underline{\textbf{False}} & \text{if p[i]} < 5 \\ \underline{g} & \text{otherwise} \end{cases} \\
&\qquad \} \\
q_0 &= \underline{(0, \textbf{True})}
\end{aligned}
$$

9. Prove that for any input that is a finite list of natural numbers, the state machine M_g always terminates, and the final state is a state where the value of g is **True** if and only if the input list contains no elements with value below 5.

First, we prove termination. All the transitions must satisfy $i <$ len(p), so there are no transitions from any state where $i \geq$ len(p). Since len(p) is finite, the initial state has $i = 0$, and all transitions include $i' = i + 1$ which increases i by 1, the machine must eventually reach a state where $i =$ len(p). Since that state has no outgoing edges, the machine terminates.

To prove correctness, we use the invariant principle. The preserved invariant is:

$$P(q = (i, g)) := g = \bigwedge_{k=0}^{i-1} \text{p[k]} \geq 5.$$

That is, the value of g is **True** if and only fi all p[i] values below i are not less than 5.

To prove correctness using the invariant principle, we need to show (1) $P(q_0)$ is true, (2) $P(\cdot)$ is a preserved invariant, and (3) $P(q_t) \implies$ desired correctness property for all terminating states q_t.

(1) $P(q_0)$ is true. $q_0 = (0, \textbf{True})$. This is true since there are no clauses in the $\bigwedge$ when $i = 0$, so it means $g = \textbf{True}$, satisfying $P(q_0)$.

(2) $P(\cdot)$ is a preserved invariant. We need to show $P(i, g) \implies P(i', g')$ for all states. By the transition rules we have $i' = i + 1$ and two cases for g':

– Case 1. When p[i] < 5, we have $g' =$ **False**. The predicate for (i', g') is

$$g' = \bigwedge_{k=0}^{i'-1} \text{p[k]} \geq 5 = \bigwedge_{k=0}^{i} \text{p[k]} \geq 5.$$

Since p[i] < 5, the $\bigwedge$ is **False**, and so is g'.

– Case 2. When p[i] ≥ 5, we have $g' = g$. The predicate for (i', g') is

$$g' = \bigwedge_{k=0}^{i'-1} \text{p[k]} \geq 5 = \bigwedge_{k=0}^{i-1} \text{p[k]} \geq 5 \wedge \text{p[i]} \geq 5.$$

Since the new clause is true, the value of the $\bigwedge$ is whatever it was before, which, by the inductive hypothesis, is g.

(3) $P(q_f) \implies g$ is **True** if and only if the input list contains no elements with value below 5. In the termination sub-proof, we showed the machine terminates in a state where $i =$ len(p). So, $q_f =$ (len(p), g_f).

Recursive Data Types

Define a *BalancedTree* as:

– **Base case: null** $\in$ *BalancedTree*.

– **Constructor case:** if $t_1, t_2 \in$ *BalancedTree* and $\text{count}(t_1) = \text{count}(t_2)$ then $\text{node}(t_1, t_2) \in$ *BalancedTree*.

where count is defined for all *BalancedTree* objects as:

$$\text{count}(t) := \begin{cases} 0 & t = \textbf{null} \\ 1 + \text{count}(t_1) + \text{count}(t_2) & t = \text{node}(t_1, t_2) \end{cases}$$

The left and right operations are defined by:

$$\text{left}(\text{node}(t_1, t_2)) := t_1$$
$$\text{right}(\text{node}(t_1, t_2)) := t_2$$

10. Explain why left is not a *total function* for *BalancedTree* objects domain.

To be a total function, it must have an arrow out for every domain element. But, left is not defined for **null**, so is not a total function.

11. Prove that for all non-null *BalancedTree* objects, t, $\text{count}(\text{left}(t)) = \text{count}(\text{right}(t))$.

To prove this, we use structural induction. The induction predicate is:

$$P(t) := \text{count}(\text{left}(t)) = \text{count}(\text{right}(t)).$$

We need to show $P(t)$ holds for all non-null *BalancedTree* objects.

Base objects: Since the base object is **null**, there is nothing to do for the base objects since $P(t)$ only needs to hold for non-null trees.

Constructor case: $P(t)$ for $t = \text{node}(t_1, t_2)$. By the inductive hypothesis, we can assume $P(t_1)$ and $P(t_2)$. The constructor rule requires $\text{count}(t_1) = \text{count}(t_2)$, so this follows directly.

Turing Machines

12. Prove that there exists a Turing Machine that (1) accepts an infinite number of inputs and (2) does not terminate on an infinite number of inputs.

We prove this by constructing a Turing Machine that satisfies the property. Here's one (of infinitely many possible) example:

$$\begin{aligned}
M = (S &= \{0, 1\}, \\
T &= \{(0, \mathbf{0}) \rightarrow (1, \mathbf{0}, \mathbf{R}), \\
&\quad (1, \mathbf{0}) \rightarrow (1, \mathbf{0}, \mathbf{R}), \\
&\quad (1, \mathbf{1}) \rightarrow (1, \mathbf{1}, \mathbf{R}), \\
&\quad (1, \textbf{-}) \rightarrow (1, \textbf{-}, \mathbf{Halt}), \\
&\quad (0, \mathbf{1}) \rightarrow (0, \mathbf{1}, \mathbf{L}), \\
q_0 &= 0, q_{Accept} = \{1\})
\end{aligned}$$

The machine accepts all strings that start with a **0**, moving right through the input in state 1. For any string that starts with a **1**, the machine runs forever without terminating (recall that if the machine would move left past the left edge it remains in the same position). There are infinitely many strings of both type, so this satisfies the property.

Countable Sets are Well-Ordered

13. Prove that all countable sets are well-ordered.

As we showed in Problem Set 9, if a non-empty set C is countable, there is a total surjective function from $\mathbb{N}$ to C. This provides a simple way to order C according to the first element of $\mathbb{N}$ that maps to C. This covers all the non-empty sets; to cover all countable sets, we also need to cover the empty set (which is indeed countable, since it is finite). The empty set is also well-ordered, since the definition of being well-ordered is that all non-empty subsets of the set have a least element. Since the empty set has no non-empty subsets, this is vacuously true.

Optional Guidance

Anything else you want me to know?

A few favorite answers, shared anonymously:

"I apologize for never waking up my friend when he slept during class sometimes. He was in 21 credits, so I felt bad for him. Didn't mean any disrespect."

"also, your proof of why the egg came before the chicken has changed my outlook on life as a whole. I think."

"My English teacher told me in high school that I wasn't comfortable enough with facing a challenge and struggling to stick with engineering, but after this class I think I have developed a new kind of resiliency to problem solving that will help me succeed at UVA and throughout my life."

"the class wasn't actually as bad as everyone says it is"

"The few times I understood what was happening, I was awestruck"

"I very much enjoyed your grading outlook. At first I found it stressful but over time it helped me focus less on grades and more on learning. I quite like the material we've learned now. I don't think I would have if we have followed the traditional way of grading. [9 gold stars]"

"PS. RIP Harambe"

"I'm really sick of the Harambe jokes."

Grading Summary

Everyone got 20 points for the cover page, and the other 13 questions were graded where a good answer was worth 10 points (excellent answers were occasionally worth more). So, there are 150 points available for all good answers.

Several of the questions were quite difficult, and very few people did well on the Turing Machine questions (which is understandable since we didn't have a problem set on these), so the average score was 116.5.

The pass rates by question are:

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13
Great (≥ 10)	31%	11%	34%	45%	25%	27%	11%	49%	7%	15%	18%	4%	20%
Passable (≥ 8)	46%	22%	50%	53%	35%	51%	30%	53%	37%	15%	29%	6%	27%
Okay (≥ 6)	53%	34%	53%	57%	46%	57%	51%	56%	50%	18%	37%	19%	47%

