Final Exam

Logical Formulas and Inference Rules

1. (Average: 9.6, Median: 10) 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. $\dfrac{P \wedge Q}{P}$

   Circle one: **Sound** *Unsound*
   Counter-example (if unsound):

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

   Circle one: *Sound* **Unsound**
   Counter-example (if unsound): P = True, Q = False.

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

   Circle one: **Sound** *Unsound*
   Counter-example (if unsound):

   d. $\dfrac{(P \wedge Q) \wedge (P \vee Q)}{P \text{ XOR } Q}$

   Circle one: *Sound* **Unsound**
   Counter-example (if unsound): P = True, Q = True.

Well Ordering

2. (Average 8.7 / Median 10) 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 set of integers between -100 and 100; $<$.

   Circle one: **Well-Ordered** *Not Well-Ordered*
   Justification: This is a finite set, so all of its subsets would also be finite. The minimum by $<$ is the least integer in the set.

b. The non-negative rational numbers; $<$.

Circle one: *Well-Ordered* **Not Well-Ordered**
Justification: This set has no minimum, since you can always find a smaller non-negative rational number by dividing it by 2.

c. The set $P_S = \text{pow}(S)$ (i.e., the set of all subsets of S) where $S = \{1, 2, 3\}$; under the comparator: $\subseteq$.

Circle one: *Well-Ordered* **Not Well-Ordered**
Justification: Consider the subset $T = \{\{1\}, \{2\}\}$, the elements $\{1\}, \{2\}$. There is no element x, such that for all other elements $z \in T$ since $\{1\} \not\subseteq \{2\}$ and $\{2\} \not\subseteq \{1\}$.

Revisiting Revisiting Exam 1

3. (Average 7.2, Median 7) Recall this question that appeared on Exam 1 and Exam 2: *Explain why the set of real numbers x where $-1 \leq x \leq 1$ is not well ordered by "$<$".* For each of the candidate answers below, indicate if the answer is *Good*, *Fixable*, or *Hopeless*. If the answer is *Fixable*, explain concisely how to fix it. If the answer is *Hopeless*, explain why the approach used in the answer cannot reasonably lead to a good answer.

a. We prove using the well-ordering principle. Define the set of counterexamples,

$$C = \{z \mid \text{the set of real numbers } z \leq 1 \text{ is not well-ordered by } <\}.$$

If C is non-empty, there exists a minimum $m \in C$. By the definition of C, the set of real numbers $m \leq 1$ is not well-ordered by $<$. But, m is the minimum element of that set. So, we have a contradition. Thus, by the well-ordering principle the proposition must be true.

Circle one: *Good* or *Fixable* or **Hopeless**

Explanation (nothing required if *Good*; if *Fixable*, explain how to fix; if *Hopeless* explain why):
We cannot use the well-ordering principle on a set that is not well ordered, like the real numbers here. This also isn't proving the given proposition. There's no sensible way to fix this, other than erasing it all and starting over.

b. We prove by induction on the size of the set, S.

Base case: $|S| = 1$. The set has one element, so that element must be its minimum. *Induction case*: by the induction hypothesis, we assume that all sets R where $|R| = |S| - 1$ have a minimum element, m_r, and show that S has a minimum element. So, there is some new element, x, such that $S = R \cup \{x\}$. The minimum element of S is either x, if $x < m_r$, or m_r otherwise. Thus, S has a minimum.

Circle one: *Good* or *Fixable* or **Hopeless**

Explanation: (nothing required if *Good*; if *Fixable*, explain how to fix; if *Hopeless* explain why)
This is a proof that all finite sets are well ordered!

c. Assume by contradiction that, R_1, the set of real numbers between -1 and 1 is well ordered. By the definition of a well-ordered set, all non-empty subsets of R_1 have a minimum element. Consider $S = R_1 - \{-1\}$. We know $S \subset R_1$ since every element of S is an element of R_1. We show that

R_1 is not well-ordered by contradiction. By the assumption that R_1 is well ordered, since S is a non-empty subset of R_1 it must have a minimum, m. Define $m^* = (-1+m)/2$. Then $m^* < m$ since $m > -1$ and the average of -1 and m must be greater than -1. But $m^* \in S$ since it is a real number between -1 and 1. This contradicts the assumption that m is the minimum of S. This contradicts the assumption that R_1 is well-ordered, since we have shown a nonempty subset of R_1 that has no minimum.

Circle one: **Good** or *Fixable* or *Hopeless*

Explanation: This is verbatim the answer from Exam 2.

Sets and Relations

4. (Average 8.3, Median 10) 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 finite sets A and B, it holds that $|A| - |B| \leq |A \cap B|$.

Circle one: *Valid* **Invalid**
Counter-example (if invalid): $A = \{1,2,3,4,5\}, B = \{3\}. |A| - |B| = 4, |A \cap B| = 1$.

b. For any sets A, B, and C, if there is an injective ([≤ 1 arrow in]) function ([≤ 1 arrow out]) f from A to B and an injective function g from B to C, then there exists an injective function from A to C.

Circle one: **Valid** *Invalid*
Counter-example (if invalid):

c. If $A \in \text{pow}(B)$ and $B \in \text{pow}(C)$, then $A \in \text{pow}(C)$.

Circle one: **Valid** *Invalid*
Counter-example (if invalid):

Induction

5. (Average 7.6, Median 8) Suppose A and B are finite sets. Prove by induction that $|A \times B| = |A| \cdot |B|$ where $A \times B$ is the cartesian product of A and B. (Hint: you can apply the induction over the size of A. A good answer must clearly define the induction predicate.)

Induction predicate: $P(n) =$ for any set A where $|A| = n$, $|A \times B| = |A| \cdot |B|$.

Base case: $n = 0$.

$|A \times B| = 0$. $|A| \cdot |B| = 0$. Thus, $P(0)$ holds.

Induction case: Prove $\forall n > 0. P(n) \implies P(n+1)$.

So, $|A| = n$. We define m as the size of B, $|B| = m$. Since we assume $P(n)$, we know $|A \times B| = |A| \cdot |B| = n \times m = k$. When we add one element to A, $A' = A \cup \{z\}$ where $z \notin A$. Then, $|A'| = n+1$. By definition of cartesian product, $A' \times B = (A \times B) \cup \{(z,b) \mid \forall b \in B\}$. The size of the set of the new elements is m. So, $|A' \times B| = |A \times B| + m = (n \times m) + m = (n+1) \times m$. This shows that $P(n+1)$ holds.

Cardinality

6. (Average 8.9, Median 10) For each set defined below, answer if the set is *"finite"*, *"countably infinite"*, 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. pow(pow(pow(S))) where S is the set of all students in cs2102 Fall 2017.

Circle one: **Finite** *Countably Infinite* *Uncountable*

Proof: Since the set S is finite, pow(S) is also finite since the power set of a finite set is finite. We can repeat this argument any number of times, so pow(pow(pow(S))) is finite.

b. $\{r \mid r = y/2^x, x \in \mathbb{N}, y \in \mathbb{N}\}$

Circle one: *Finite* **Countably Infinite** *Uncountable*

Proof: There is one set element for each pair of natural numbers, so this is countably infinite. We can provide a binjection between this set an $\mathbb{N}$, using the same argument we used to show the set of rationals is countably infinite.

c. $\{(a,b) \mid a \in \mathbb{N}, b \in \text{pow}(\mathbb{N})\}$

Circle one: *Finite* *Countably Infinite* **Uncountable**

Proof: Since pow($\mathbb{N}$) is uncountable, and this set includes all pairs of a natural number and an element from that set, it is uncountable.

d. $\{(a,b,c) \mid a \in \mathbb{N}, b \in \mathbb{N}, c \in \mathbb{N}\}$

Circle one: *Finite* **Countably Infinite** *Uncountable*

Proof: This is $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$. We can provide a bijection to $\mathbb{N}$ by using the rational number bijection twice, or by doing a three-dimensional version of this. For example, $(0,0,0), (0,0,1), (0,1,0), (1,0,0), (0,1,1), (1,0,1), \ldots$.

Recursive Data Types

Consider the recursive data types, *XBF* (short for Xor-Based Formula) defined by:

– **Base:** *true* is an *XBF*.
– **Base:** *false* is an *XBF*.
– **Constructor:** for all *XBF* objects f_1, f_2, *xor*(f_1, f_2) is a *XBF*.

The Value of an *XBF*, f, should be the Boolean value of it when it is evaluated as a logical formula based on XOR gates. To distinguish the syntactic symbols used in defining *XBF* s from the logical Booleans, we use **True** and **False** to represent the Boolean true and false values, and $\oplus$ to represent the Boolean operation xor.

So, for example,

$$\text{Value}(\textit{xor}(\textit{xor}(\textit{true}, \textit{true}), \textit{false})) = \textbf{False}.$$

7. (Average 8.3, Median 9) Provide a precise, sensible, and complete definition of Value for all *XBF* objects.

   Value(*true*) = **True**
   Value(*false*) = **False**
   Value(*xor*(f_1, f_2)) = Value(f_1) $\oplus$ Value(f_2)

8. [Excluded from exam]

Number Theory

9. (Average 8.4, Median 9) Indicate for each statement if it is True or False. Either way, provide a short reason supporting your answer.

a. For all positive natural numbers a, b, c, it holds that $\gcd(a, \gcd(b, c)) = \gcd(\gcd(a, b), c)$.

   Circle one: **True** *False*
   Short justification: $\gcd(a, \gcd(b, c)) = \gcd(a, b, c) = \gcd(\gcd(a, b), c)$.

b. $\mathbb{N}_5$ is a field if we use ($+$ mod 5) for addition and ($\times$ mod 5) for the multiplication operations.

   Circle one: **True** *False*
   Short justification:
   5 is a prime, the additive identity is 0, the multiplicative identity is 1. Every element except 0 has an multiplicative inverse. Other properties also hold.

c. There exist (distinct) primes $p < q < r$ such that $pr = q^2$.

   Circle one: *True* **False**
   Short justification:
   If $pr = q^2 = n$ and p, q, r are all prime, then n wouhd have two factoring into primes which is impossible.

Program Correctness

We call a list $p_0, \ldots, p_n$ a non-decreasing list, if $p_0 \leq p_1 \leq \ldots p_n$. Consider the Python program below, that returns `True` if and only if the elements in the input list are non-decreasing. We assume that `p` is a non-empty list of natural numbers. (Note: this is corrected to include the `-1` that was missing in the original exam, and the $i <$ `len(p) - 1` condition that was missing from G.)

```
def non_dec(p):
   i = 0
   q = p[0]
   g = True
   while i < len(p) - 1:
      i = i + 1
```

```
    if p[i] < q:
       g = False
    q = p[i]
 return g
```

10. (Average 9.0, Median 9) Complete the definition of the state machine, $M_g = (S, G, q_0)$, below that models `non_dec`.

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

11. (Average 7.5, Median 8) Prove that for any input that is a finite non-empty 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 is non-decreasing. Note that you need to do the following: (a) Prove that the program ends. (b) Formally define a property P and show that P for states of the machine above and prove that P is a preserved invariant. (c) Show that P holds for the initial state. (d) Show that if P holds for a final state, then that state will have the correct answer.

(a) *Termination*: In q_0, $i = 0$. Each transition increases the value of the i part of the state by one, because all transistion rules include $i' = i + 1$. There are no transitions from states where $i \geq$ `len(p)` $- 1$. So, after `len(p)` steps, the meachine must terminate. Since the input `p` must have finite length, this proves termination.

(b) *Preserved Invariant*: $P(q = (i,q,g)) ::= g =$ the sequence `p[0], p[1], ..., p[i]` is non-decreasing, $q =$ `p[i]`.

P is a preserved invariant: When $P(m)$ holds, if `p[0:m+1]` is non-decreasing, g is **True**. (Case 1) If `p[m+1] >= p[m]`, `p[0: m+1]` is non-decreasing. By the transition graph $g' = g =$ **True**. Since $q =$ `p[i]` (by P), `p[m+1]` $\geq$ `q`, `p[0: m+1]` is non-decreasing and $g' = g =$ **True**. (Case 2) If `p[m+1]` $<$ `p[m]`, then `p[0: m+1]` is not non-decreasing. Since `p[m+1]` $<$ `q`, $g' =$ **False**, and the invariant is preserved.

(Case 3) If `p[0:m]` is not non-decreasing, g is **False**. Then, `p[0:m+1]` is also not non-decreasing. By the transition graph, g' must be **False** in the next state, so P is preserved.

For all cases, $q =$ `p[i]` is preserved, since $q' = p[i']$ and $i' = i + 1$.

(c) *Initially True*: $P(0)$ says the sequence of just one element is non-decreasing, which must be true. In q_0, $g =$ **True**, so the invariant is preserved.

(d) *Final state*: In the final state, $i =$ `len(p) - 1`. Hence, $P(f)$ says that $g =$ the sequence `p[0]`, `p[1]`, ..., `p[len(p) - 1]` is non-decreasing. Thus, the preserved invariant implies the desired correctness property for the full input array.

Statistics

Average: 82.9
Median: 85.5

Anything else you want us to know?

Here's a few favorites:

"the funny video on well ordering still in my head"

"The TAs for this class are amazing."

"Also, I have watched the PSO video from Devin Kim too many times. It's a great video."

"Throughout this semester, I grew tremendously from this class. cs2102 has taught me how to communicate well in teams, especially for problem sets, and even, in my own ROMANTIC BIJECTIVE relationship!"

"I admit that it was hard for me to see the relevance of discrete math in the first half of the semester. But as I learned about trees and recursion in CS and I started taking on a few internship interviews, I realized that the logic of discrete math is really really helpful. I've been appreciating it more and more as the semester goes on."

"I appreciate your outlook on grading as it allows me to just focus on absorbing the material. That is, unless I do poorly, in which case I hate your outlook on grading."

"I didn't eat breakfast before this exam, and now I'm rally mahMOODY."

"I loved the last lecture when Prof. Evans said a CS education is meant to take the "magic" out of how computers work"

"This class has been surprisingly enjoyable and I love when you wear Hawaiian shirts with your red shoes."

"I realized I had never faced a challenge like this in my academic life. But I decided I was not going to let Discrete win this challenge nor win this fight."

"This class could easily be the most abstract thing in the world, but it felt at least somewhat grounded in reality."

"Thank you for an amazing semester and your exquisite, foolproof guide to not getting bamboozled while trick-or-treating!"

"This class in a nutshell: the empty set is well ordered."

