CSC165H1: Problem Set 1

Due October 2 before 4 p.m.

General instructions

Please read the following instructions carefully before starting the problem set. They contain important information about general problem set expectations, problem set submission instructions, and reminders of course policies.

- Your problem sets are graded on both correctness and clarity of communication. Solutions which are technically correct but poorly written will not receive full marks. Please read over your solutions carefully before submitting them. Proofs should have headers and bodies in the form described in the course note.
- Each problem set may be completed in groups of up to three. If you are working in a group for this problem set, please consult `https://github.com/MarkUsProject/Markus/wiki/Student_Groups` for a brief explanation of how to create a group on MarkUs.

  Exception: Problem Sets 0 and 1 must be completed individually.
- Solutions must be typeset electronically, and submitted as a PDF with the correct filename. **Handwritten submissions will receive a grade of ZERO.**

  The required filename for this problem set is **problem_set1.pdf**.
- Problem sets must be submitted online through MarkUs. If you haven't used MarkUs before, give yourself plenty of time to figure it out, and ask for help if you need it! If you are working with a partner, you must form a group on MarkUs, and make one submission per group. "I didn't know how to use MarkUs" is not a valid excuse for submitting late work.
- Your submitted file should not be larger than 9MB. This may happen if you are using a word processing software like Microsoft Word; if it does, you should look into PDF compression tools to make your PDF smaller, although please make sure that your PDF is still legible before submitting!
- The work you submit for credit must be your own; you may not refer to or copy from the work of other groups, or external sources like websites or textbooks. You may, however, refer to any text from the Course Notes (or posted lecture notes), except when explicitly asked not to.

Additional instructions

- Final expressions must have negation symbols ($\neg$) applied **only** to predicates or propositional variables, e.g. $\neg p$ or $\neg Prime(x)$. To express "a is not equal to b," you can write $a \neq b$.
- When rewriting logical formulas into equivalent forms (e.g., simplifying a negated formula or removing implication operators), you must **show all of the simplification steps involved**, not just the final result. We are looking for correct use of the various simplification rules here.
- You may not define your own predicates or sets for this problem set; please work with the definitions provided in the questions, or from the course notes.

1. **[6 marks] Truth tables and formulas.** Consider the following formula:

$$\neg r \Rightarrow (\neg p \Rightarrow q)$$

   (a) **[2 marks]** Write the truth table for the formula. (No need to show your calculations).

   (b) **[2 marks]** Write a logically equivalent formula that doesn't use $\Rightarrow$ or $\Leftrightarrow$, in other words it uses only $\wedge, \vee$, or $\neg$. Show how you derived the result.

   (c) **[2 marks]** Write formula that is logically equivalent to the **converse** of the given formula, and that doesn't use $\Rightarrow$ or $\Leftrightarrow$, in other words it uses only $\wedge, \vee$, or $\neg$. Show how you derived the result.

2. **[6 marks] one-to-one and onto**

Use the following definitions in the questions below.

Onto(f): $\forall n \in \mathbb{N}, \exists m \in \mathbb{N}, f(m) = n$, for $f : \mathbb{N} \to \mathbb{N}$.
OneToOne(f): $\forall m, n \in \mathbb{N}, m \neq n \Rightarrow f(m) \neq f(n)$, for $f : \mathbb{N} \to \mathbb{N}$.

   (a) **[1 mark]** Suppose $\neg$**Onto(g)**. Write this in predicate logic **without** using the predicate name **Onto**.

   (b) **[1 mark]** Suppose $\neg$**OneToOne(h)**. Write this in predicate logic **without** using the predicate name **OneToOne**.

   (c) **[1 mark]** Give an example of a function $f : \mathbb{N} \to \mathbb{N}$ where **Onto(f)** and **OneToOne(f)**.

   (d) **[1 mark]** Give an example of a function $f : \mathbb{N} \to \mathbb{N}$ where $\neg$**Onto(f)** and **OneToOne(f)**.

   (e) **[1 mark]** Give an example of a function $f : \mathbb{N} \to \mathbb{N}$ where **Onto(f)** and $\neg$**OneToOne(f)**.

   (f) **[1 mark]** Give an example of a function $f : \mathbb{N} \to \mathbb{N}$ where $\neg$**Onto(f)** and $\neg$**OneToOne(f)**.

3. **[7 marks] modular arithmetic**

   (a) **[2 marks]** Prove Example 2.19(1) from the course notes.

   (b) **[2 marks]** Prove Example 2.19(3) from the course notes.

   (c) **[3 marks]** Use Example 2.19(3) to find the units digit of 257^{256}. Use Example 2.19(3) to prove your result — we will not accept the argument that you used a calculator or programming language to compute this with brute force.

4. **[7 marks] remainders**

   (a) **[1 mark]** Prove:

$$\exists x \in [0, 34], x \equiv 3 \bmod 5 \wedge x \equiv 5 \bmod 7$$

   (b) **[1 mark]** Prove:

$$\exists m_1, m_2 \in \mathbb{Z}, (m_1 \times 7) + (m_2 \times 11) = 1$$

   ...by finding suitable values for m_1 and m_2.

   (c) **[2 marks]** Assume that m_1, m_2 are integers such that $(m_1 \times 7) + (m_2 \times 11) = 1$. Prove:

$$\forall a_1, a_2 \in \mathbb{Z}, (a_2 \times m_1 \times 7) + (a_1 \times m_2 \times 11) \equiv a_2 \bmod 11$$

   (d) **[3 marks]** Prove that if p_1, p_2 are any two distinct primes and a_1, a_2 are any two integers, then there is some integer x such that $x \equiv a_1 \bmod p_1$ and $x \equiv a_2 \bmod p_2$. **Hint**: Note that $\gcd(p_1, p_2) = 1$, read the material on gcd in the course notes, and read the previous part of this question.