CSC165H1: Problem Set 1

Due January 25, 2017 before 10pm

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.
- 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 Set 0 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!
- Submissions must be made *before* the due date on MarkUs. You may use *grace tokens* to extend the deadline; please see the Problem Set page for details on using grace tokens.
- The work you submit must be that of your group; 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.

1. **[8 marks] Propositional formulas.** For each of the following propositional formulas, find the following two items:

   (i) The truth table for the formula. (You don't need to show your work for calculating the rows of the table.)

   (ii) A logically equivalent formula that only uses the $\neg$, $\wedge$, and $\vee$ operators; *no* $\Rightarrow$ or $\Leftrightarrow$. (You *should* show your work in arriving at your final result. Make sure you're reviewed the "extra instructions" for this problem set carefully.)

   (a) **[4 marks]** $(p \Leftrightarrow \neg q) \Rightarrow p$.

   (b) **[4 marks]** $p \wedge \neg r \Rightarrow q \vee r$.

2. **[3 marks] Generalizing $\Leftrightarrow$.** Suppose we have four propositional variables p, q, r, and s. Write a propositional formula that is True when all four variables have the same truth value, and False otherwise. (This is similar in spirit to the $\Leftrightarrow$ operator.) Explain in a few brief sentences how you obtained your formula – "guess and check" is not an acceptable response!

3. **[11 marks] Translating statements.** The teams in the National Hockey League (NHL) are grouped into divisions of seven or eight teams. Here are some sets and predicates used to model the NHL:

Symbol	**Definition**
T	the set of all teams (represented by city name)
D	the set of all divisions (represented by region name)
$Stanley(t)$	"team t has played in a Stanley Cup final," where $t \in T$
$Canada(t)$	"team t is in Canada," where $t \in T$
$BelongsTo(t, d)$	"team t is in division d," where $t \in T$ and $d \in D$

   Using only these sets and predicates, **the standard propositional operators and quantifiers**,[1] and the equals symbol =, translate each of the following English statements into predicate logic.

   Note: a division is just a name, and is *not* a set of teams.

   (a) **[2 marks]** Every Canadian team has played in a Stanley Cup final.

   (b) **[2 marks]** Team *toronto* is in division *atlantic*.

   (c) **[2 marks]** There is a division that contains only non-Canadian teams.

   (d) **[2 marks]** Each division has a team that has played in a Stanley Cup final.

   (e) **[3 marks]** Every team is in exactly one division. (Hint: use the equals symbol =.)

[1] Updated Jan 21.

4. **[9 marks] One-to-one functions.** So far, most of our predicates have had sets of numbers as their domains. But this is not always the case: we can define properties of any kind of object we want to study, including functions themselves!

   Let S and T be sets. We say that a function $f : S \to T$ is **one-to-one** if no two distinct inputs are mapped to the same output by f. For example, if $S = T = \mathbb{Z}$, the function $f_1(x) = x + 1$ is one-to-one, since every input x gets mapped to a distinct output. However, the function $f_2(x) = x^2$ is not one-to-one, since $f_2(1) = f_2(-1) = 1$.

   (a) **[2 marks]** Suppose we want to define a predicate $OneToOne(f)$, which expresses whether a given function $f : \mathbb{Z} \to \mathbb{Z}$ is one-to-one. (Note that we are using a concrete domain and range of $\mathbb{Z}$ here.)

   $$OneToOne(f) : \underline{\hspace{6cm}}, \text{ where } f : \mathbb{Z} \to \mathbb{Z}.$$

   Using the language of predicate logic, show how to fill in the blank to define $OneToOne$. You may use an expression like "$f(x) =$ [something]" in your formula.[2]

   (b) **[3 marks]** We say that a function $f : S \to T$ is **two-to-one** if no *three* distinct inputs are mapped to the same output by f. Now $f_2(x) = x^2$ is two-to-one (when $S = T = \mathbb{Z}$), but the function $f_3(x) = x(x-1)(x-2)$ is not. Note that $f_1(x) = x + 1$ *is* two-to-one according to this definition.

   Suppose we want to define a predicate $TwoToOne(f)$, which expresses whether a given function $f : \mathbb{Z} \to \mathbb{Z}$ is two-to-one. Using the language of predicate logic, show how to fill in the blank to define $TwoToOne$.

   $$TwoToOne(f) : \underline{\hspace{6cm}}, \text{ where } f : \mathbb{Z} \to \mathbb{Z}.$$

   (c) **[1 mark]** Consider the following rather famous statement:

   "There does not exist a one-to-one function with domain $\mathbb{R}$ and range $\mathbb{N}$."

   Let F be the set of all functions with domain $\mathbb{R}$ and range $\mathbb{N}$. First, show how to express this statement in predicate logic by filling in the blank in the following sentence:

   $$\neg\Big(\exists f \in F, \underline{\hspace{6cm}}\Big).$$

   You should reuse your work from part (a), but do not write "$OneToOne$" in your formula here. (In other words, copy over the definition of your predicate from part (a), except pay attention to the sets of numbers involved!)

   (d) **[3 marks]** Finally, use the negation rules to show how to simplify your expression for part (c) so that negations are applied only to predicates (and remember you can use the $\neq$ symbol).

[2] Recall that a legitimate formula is built up only from quantifiers (each ranging over a defined set), boolean operators ($\wedge$, $\vee$, $\neg$, etc.), and predefined functions and predicates.
Just as how when we write a^b, this is actually shorthand for an exponentiation function $exp(a, b)$, when we write $f(x)$ in a formula this is shorthand for a *function evaluation* operator $F(f, x)$. For the set A of all functions from $\mathbb{R}$ to $\mathbb{R}$, this operator is defined as $F : A \times \mathbb{R} \to \mathbb{R}$, where $F(f, x)$ is equal to $f(x)$. So while you can write a formula like $\forall x \in \mathbb{R},\ f(x) = 10$, this is actually shorthand for the more technically correct $\forall x \in \mathbb{R},\ F(f, x) = 10$.

5. **[9 marks] Working with infinity.** Sometimes when dealing with predicates over the natural numbers, we don't care so much about which numbers satisfy the given predicate, or even exactly how many numbers satisfy it. Instead, we care about whether *infinitely many* numbers satisfy the predicate. For example:[3]

"There are infinitely many primes."

We saw that for the natural numbers, we can express the idea of "infinitely many" by saying that for every natural number n_0, there is a number greater than n_0 that satisfies the predicate:

$$\forall n_0 \in \mathbb{N},\ \exists n \in \mathbb{N},\ n > n_0 \wedge Prime(n)$$

You may use the $Prime$ predicate in your solutions for all parts of this question.

(a) **[3 marks]** Use the above idea to express the following statement in predicate logic. Briefly justify why your statement is correct.

"There are infinitely many numbers that are one less than a prime number."

(b) **[4 marks]** Express the following statement in predicate logic (remember to use the negation simplification rules). Briefly justify why your statement is correct.

"There are *finitely* many numbers that cannot be written as the sum of two primes."

(c) **[2 marks]** Here is another variation of combining predicates with infinity. Let $P : \mathbb{N} \rightarrow \{\text{True}, \text{False}\}$ be a predicate defined over the natural numbers. We say that P is **eventually true** if and only if there exists a natural number n_0 such that all natural numbers greater than n_0 satisfy P.

Use this idea to express the following statement in predicate logic (no justification is required here):

"Eventually, all natural numbers are prime."

[3] Course Notes, page 26.