CSC165H1: Problem Set 1

Due Thursday January 24 before 4pm

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 that 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(s) should not be larger than 9MB. You might exceed this limit if you use a word processor like Microsoft Word to create a PDF; 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 Homework 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 part of the Course Notes (or posted lecture notes), except when explicitly asked not to.

Additional instructions

- Final expressions in predicate logic 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 $a \neq b$.
- You may not define your own propositional operators, predicates, or sets for this problem set. Please work with the symbols we have introduced in lecture, and any additional definitions provided in the questions.

1. **[9 marks] Pets**. Let P be the set of all pets. We define the following predicates over P:

   - $Cat(p)$: "p is a cat"
   - $Dog(p)$: "p is a dog"
   - $House(p)$: "p is allowed in the house"
   - $Behaved(p)$: "p is well-behaved"
   - $Attacks(p_1, p_2)$: "p_1 will attack p_2" (note: $Attacks(p_1, p_2)$ is *not* the same as $Attacks(p_2, p_1)$)

   For each of the following statements, if it is English then translate it into predicate logic, and if it is in symbolic form translate it into English.

   (a) Any pet that will attack itself is not allowed in the house.

   (b) Pets who will attack at least one other pet are not well-behaved.

   (c) $\exists p \in P,\ Dog(p) \wedge \Big(\forall c \in P,\ Cat(c) \Rightarrow \neg Attacks(p, c)\Big)$

   (d) For any two distinct pets, if they will attack each other, then at most one of them is allowed in the house.

   You can use $=$ and $\neq$ to check whether two pets are the same or not.

2. **[9 marks] Working with strings**. Consider the following definitions on strings.

   - Let U be the set of all strings.
   - Let s be a string. The **length/size** of a string, denoted $|s|$, is the number of characters in s.
   - Let s be a string, and $i \in \mathbb{N}$ such that $0 \leq i < |s|$. We write $s[i]$ to represent the character of s at index i, where indexing starts at 0 (so $s[0]$ is the first character, and $s[|s| - 1]$ is the last character).

   Using these definitions, we can represent interesting properties of strings in formal predicate logic. For example, here is how we could define a predicate over U to check whether a string contains the letter A:

   $$\mathit{ContainsAnA}(s): \quad \exists i \in \mathbb{N},\ 0 \leq i < |s| \wedge s[i] = A$$

   For the following translations, you may use the notation for the size and indexing of a string, and the equals symbol = to compare *single letters*. Do not use the = symbol to compare entire strings.

   (a) Let s_1 and s_2 be strings. We say that s_1 is a **substring** of s_2 if s_1 appears somewhere in s_2 (in a contiguous block). For example, the string abc is a substring of $zabcdef$, but it is not a substring of $abzdefc$. Every string is a substring of itself, and the empty string is a substring of every string.

   Define a predicate $\mathit{HasCSC}(s)$, which is true if and only if the string CSC is a substring of s.

   (b) Define a predicate $\mathit{Substring}(s_1, s_2)$, which is true if and only if s_1 is a substring of s_2.

   (c) Let s be a string. We say that s is a **palindrome** if and only if its reversal (the string obtained by taking the letters of s in reverse order) equals the original string. For example, the string $abbcbba$ is a palindrome. The empty string and all strings of length 1 are palindromes.

   Define a predicate $\mathit{Palindrome}(s)$, which is true if and only if s is a palindrome.

   (d) Consider the following statement in predicate logic:

   $$\forall s_1, s_2 \in U,\ \mathit{Substring}(s_1, s_2) \Leftrightarrow |s_1| \leq |s_2|$$

   Is this statement True or False? Briefly explain your answer; no formal proof is necessary.

3. **[9 marks] Properties of functions**. Let $f : \mathbb{R} \to \mathbb{R}$ be a function. Consider the following definitions.

   - f is **strictly increasing** if and only if for every pair of real numbers x and y, if $x < y$ then $f(x) < f(y)$.
   - f is **strictly decreasing** if and only if for every pair of real numbers x and y, if $x < y$ then $f(x) > f(y)$.
   - f has a **global maximum at** z if and only if $f(x)$ is always less than or equal to $f(z)$, for every real number x.

   Translate each of the following statements into predicate logic.

   (a) The function f_1 is *not* strictly increasing. (You may use the variable f_1 in your translation.)

   (b) The function f_2 is neither strictly increasing nor strictly decreasing. (You may use the variable f_2 in your translation.)

   (c) The function f_3 has a global maximum at two *different* numbers. (You may use the variable f_3 in your translation.)

   (d) For every function $f : \mathbb{R} \to \mathbb{R}$, if f is strictly increasing then f does not have a global maximum at any real number.

4. **[7 marks] Choosing a universe and predicates.**

   (a) Consider the following statement:

   $$\forall x \in \mathbb{N},\ P(x, 165) \Rightarrow P(x, 1)$$

   Provide one definition of a binary predicate P over $\mathbb{N} \times \mathbb{N}$ that makes the above statement True, and another definition of P that makes the statement False. Briefly justify your answers, but no formal proofs are necessary.

   (b) Consider the following statement:

   $$\forall x \in U,\ \Big(\big(P(x) \Rightarrow Q(x)\big) \Rightarrow R(x)\Big) \Leftrightarrow \Big(P(x) \Rightarrow \big(Q(x) \Rightarrow R(x)\big)\Big)$$

   Provide one definition of a *non-empty* set U, and predicates P, Q, and R over U, that makes the above statement True, and another definition definition of a non-empty set U, and predicates P, Q, and R that makes the statement False. Briefly justify your answers, but no formal proofs are necessary.