Homework 1 – MATH 2510 Spring 2018

1. Let $X = \{1, 2, 3\}$, $Y = \{2, 3, 4\}$, and $Z = \{1, 2, 3, 5, 7\}$.

   (a) Is $2 \in X$? Is $2 \in Y$? Is $2 \in Z$?
   Solution: Yes, $2 \in X$. Yes, $2 \in Y$. Yes, $2 \in Z$.

   (b) Is $X \subseteq Y$? Is $X \subseteq Z$? Is $Y \subseteq Z$? Is $Z \subseteq X$? Is $Z \subseteq Y$?
   Solution: No, $X \not\subseteq Y$, because $1 \in X$ while $1 \notin Y$. Yes, $Y \subseteq Z$. Not, $Z \not\subseteq X$ because $7 \in Z$ while $7 \notin X$. No, $Z \not\subseteq Y$ because $7 \in Z$ while $7 \notin Y$.

   (c) What is $X \cup Y$? What is $X \cup Z$? What is $Y \cup Z$? What is $X \cup Y \cup Z$?
   Solution: Compute

   $$X \cup Y = \{1, 2, 3\} \cup \{2, 3, 4\} = \{1, 2, 3, 4\},$$

   $$X \cup Z = \{1, 2, 3\} \cup \{1, 2, 3, 5, 7\} = \{1, 2, 3, 5, 7\},$$

   $$Y \cup Z = \{2, 3, 4\} \cup \{1, 2, 3, 5, 7\} = \{1, 2, 3, 4, 5, 7\},$$

   and

   $$X \cup Y \cup Z = \{1, 2, 3\} \cup \{2, 3, 4\} \cup \{1, 2, 3, 5, 7\} = \{1, 2, 3, 4, 5, 7\}.$$

   (d) What is $X \cap Y$? What is $X \cap Z$? What is $Y \cap Z$? What is $X \cap Y \cap Z$?
   Solution: Compute

   $$X \cap Y = \{1, 2, 3\} \cap \{2, 3, 4\} = \{2, 3\},$$

   $$X \cap Z = \{1, 2, 3\} \cap \{1, 2, 3, 5, 7\} = \{1, 2, 3\},$$

   $$Y \cap Z = \{2, 3, 4\} \cap \{1, 2, 3, 5, 7\} = \{2, 3\},$$

   and

   $$X \cap Y \cap Z = \{1, 2, 3\} \cap \{2, 3, 4\} \cap \{1, 2, 3, 5, 7\} = \{2, 3\}.$$

   (e) What is $X \times Y$?
   Solution: Compute

   $$\begin{aligned} X \times Y &= \{(a, b) \colon a \in X, b \in Y\} \\ &= \{(1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}. \end{aligned}$$

2. Let $X = \{1, 2, 3\}$ and $Y = \{4, 5, 6\}$. Let $R \subset X \times Y$ be a relation given by $R = \{(1, 5), (1, 6), (2, 6), (3, 6)\}$. Is R a function? Why or why not?
   Solution: No, R is not a function. This is because both $(1, 5) \in R$ and $(1, 6) \in R$, but a function is a special kind of a relation with the property that if $(a, b) \in R$ and $(a, c) \in R$, then $b = c$, which does not happen here!

3. Is the following string of symbols a formula of propositional logic?

$$(A \wedge B) \wedge ((\neg A) \wedge B)$$

Solution: Yes. We can build it up using the appropriate rules (rules ($R1$) and ($R2$)). We know that both A is a formula and B is a formula because they are atomic formulas. It follows that $\neg A$ is a formula. Thus both $A \wedge B$ and $(\neg A) \wedge B$ are formulas. From this we may conclude that $(A \wedge B) \wedge ((\neg A) \wedge B)$ is a formula.

4. Is the following string of symbols a formula of propositional logic?

$$(A \wedge \neg) \wedge (A \wedge (B \wedge \neg A)).$$

Solution: This is not a formula because if it were, then both $(A \wedge (B \wedge \neg A))$ would have to be a formula (it is) and $A \wedge \neg$ would have to be a formula. But $A \wedge \neg$ is not a formula because $\neg$ is not an atomic formula.

5. Prolog exercise. Consider the Prolog code `family.pl` at https://github.com/tomcuchta/math2510spring2018/blob/master/family.pl. Copy this code to SWISH.

   (a) What code do you run to check if Bob is a sibling of Joe? What is the result when it is run?
   Solution: You run

   ```
   sibling(bob,joe).
   ```

   The result is `true`.

   (b) What code do you run to check if Tim is a sibling of Alice? What is the result when it is run?
   Solution: You run

   ```
   sibling(tim,alice).
   ```

   The result is `false`.

   (c) What code can you use to define the "brother of" relation?
   Solution: Define the "brother of" relation with

   ```
   brother(X,Y) :-
      sibling(X,Y),
      male(X).
   ```

   You can check to see that this works by running `brother(bob,sue).` which returns `true` while `brother(sue,bob).` which returns `false`.

   (d) What code can you use to define the "uncle of" relation?
   Solution: Define the "uncle of" relation by

```
uncle(X,Y) :-
   parent(Z,Y),
   brother(X,Z).
```

Test to see that it works: run `uncle(joe,tim).` which returns `true` while `uncle(tim,joe).` returns `false`.

NOTE: the code defined this way behaves strangely with the query `uncle(jerry,tim).` because it returns `true` although which you expect to return `false`. The root of this is the sibling relation being reflexive, i.e. running `sibling(jerry,jerry).` returns `true`. This could be fixed by changing the definition of `sibling` to be the following:

```
sibling(X,Y) :-
    parent(Z,X),
    parent(Z,Y),
    not(X=Y).
```

This change will cause the query `uncle(jery,tim).` to resolve as expected.