# Quiz Section 7: Set Theory and Structural Induction

## Task 1 – Squares Have Four Sides... Right?

**a)** Let $A$ and $B$ be the following sets:

$$A := \{n \in \mathbb{Z} : 2 \nmid n\}$$
$$B := \{n \in \mathbb{Z} : 4 \mid n^2 + 3\}$$

Now, consider the following claim:

$$A \subseteq B$$

Write an **English proof** that the claim holds.

Follow the structure of our template for subset proofs.

**Note**: even though we want you to write your proof directly in English, it must still look like the translation of a formal proof. In particular, you must include all steps that would be required of a formal proof, excepting only those that we have explicitly said are okay to skip in English proofs (e.g., Elim $\exists$).

**b)** Let $A$ and $B$ be the following sets:

$$A := \{n \in \mathbb{Z} : 2 \mid n\}$$
$$B := \{n \in \mathbb{Z} : \text{Odd}((n + 7)^2)\}$$

Now, consider the following claim:

$$A \subseteq B$$

Write an **English proof** that the claim holds.

## Task 2 – Cartesian Products

Let $A$, $B$, $C$, and $D$ be sets. Consider the following claim:

$$(A \cap B) \times C \subseteq A \times (C \cup D)$$

**a)** Suppose that $A = \{1, 2\}$, $B = \{1, 2, 3\}, C = \{3, 4\}, D = \{2\}$.

Calculate the values of the sets $(A \cap B) \times C$ and $A \times (C \cup D)$. Check whether the claim holds.

**b)** Suppose that $A = \{1, 2\}$, $B = \{2, 3, 4\}, C = \{1, 2, 3\}, D = \{1, 3\}$.

Calculate the values of the sets $(A \cap B) \times C$ and $A \times (C \cup D)$. Check whether the claim holds.

**c)** Write an **English proof** that the claim holds.

Follow the structure of our template for subset proofs.

**Note**: even though we want you to write your proof directly in English, it must still look like the translation of a formal proof. In particular, you must include all steps that would be required of a formal proof, excepting only those that we have explicitly said are okay to skip in English proofs.

## Task 3 – Power Sets

Let $A$ and $B$. Consider the following claim:

$$\mathcal{P}(A) \subseteq \mathcal{P}(B)$$

**a)** Suppose that $A = \{1\}$, $B = \{1, 2\}$.

Calculate the values of the sets $\mathcal{P}(A)$ and $\mathcal{P}(B)$. Check whether the claim holds.

**b)** Suppose that $A = \{1\}$, $B = \{2, 3\}$.

Calculate the values of the sets $\mathcal{P}(A)$ and $\mathcal{P}(B)$. Check whether the claim holds.

**c)** Write an **English proof** that the claim holds given that $A \subseteq B$.

(This updated claim describes the situation in part (a) but not part (b).)

Follow the structure of our template for subset proofs.

**Note**: even though we want you to write your proof directly in English, it must still look like the translation of a formal proof. In particular, you must include all steps that would be required of a formal proof, excepting only those that we have explicitly said are okay to skip in English proofs.

## Task 4 – Set Equality

Let $A$, $B$, and $C$ be sets. For each of the following claims:

1. **State** whether the the claim is true or false.
2. If the claim is true, write an **English proof** that the claim holds following the Meta Theorem *template*. (In your equivalence chain, you can skip steps showing commutativity or associativity, as long as each step is easy to follow.)
3. If it the claim false, give a **counterexample**. Provide specific finite sets for $A$, $B$, and $C$, and then calculate the value of each side of the claim, showing that they do not produce the same set. (Be sure to show the value of each intermediate expression, when calculating each side.)

**a)** $(A \setminus B) \setminus C = A \setminus (B \cap C)$

**b)** $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$

## Task 5 – Structural Induction on Lists

Recall the definition of lists of numbers from lecture:

**Basis Step**: nil $\in$ **List**
**Recursive Step**: for any $a \in \mathbb{Z}$, if $L \in$ **List**, then $a :: L \in$ **List**.

For example, the list $[1, 2, 3]$ would be created recursively from the empty list as $1 :: (2 :: (3 :: \text{nil}))$. We will consider "::" to associate to the right, so $1 :: 2 :: 3 :: \text{nil}$ means the same thing.

The parts below use two recursively-defined functions. The first is len, which calculates the length of the list. It is defined recursively as follows:

$$\begin{aligned} \text{len}(\text{nil}) &:= 0 \\ \text{len}(a :: L) &:= 1 + \text{len}(L) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

The second function, echo-pos, which duplicates each positive number in the list, is defined by:

$$\begin{aligned} \text{echo-pos}(\text{nil}) &:= \text{nil} \\ \text{echo-pos}(a :: L) &:= a :: \text{echo-pos}(L) && \text{if } a \leqslant 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \\ \text{echo-pos}(a :: L) &:= a :: a :: \text{echo-pos}(L) && \text{if } a > 0 \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

For example, with these definitions, we get $\text{echo-pos}(-1 :: 2 :: -3 :: \text{nil}) = -1 :: 2 :: 2 :: -3 :: \text{nil}$.

**a)** Write a calculation block, citing the appropriate definitions, showing that

$$\text{echo-pos}(1 :: -2 :: 3 :: \text{nil}) = 1 :: 1 :: -2 :: 3 :: 3 :: \text{nil}$$

**b)** Use structural induction to prove that

$$\forall L \in \textbf{List} \, (\text{len}(\text{echo-pos}(L)) \leqslant 2 \, \text{len}(L))$$

## Task 6 – Structural Induction on Trees

Recall the definition of rooted binary trees (with no data) from lecture:

> **Basis Step**: $\bullet \in$ **Tree**
> **Recursive Step**: if $L \in$ **Tree** and $R \in$ **Tree**, then $\text{Tree}(L, R) \in$ **Tree**.

Note that, in lecture, we drew "$\text{Tree}(L, R)$" as a picture of a tree, whereas here we are using more normal (functional) notation, which should be easier for calculations.

Next, we define the two operations on trees. The first, leaves, which returns the number of leaves in the tree, is defined by:

$$\begin{aligned} \text{leaves}(\bullet) &= 1 \\ \text{leaves}(\text{Tree}(L, R)) &= \text{leaves}(L) + \text{leaves}(R) \end{aligned}$$

The second function, size, which counts the number of nodes in the tree, is defined by:

$$\begin{aligned} \text{size}(\bullet) &= 1 \\ \text{size}(\text{Tree}(L, R)) &= 1 + \text{size}(L) + \text{size}(R) \end{aligned}$$

With tnose definitions in hand, consider the following claim:

$$\forall T \in \textbf{Tree}\, (\text{size}(T) + 1 = 2\, \text{leaves}(T))$$

Prove this claim by structural induction.