CSE 390Z: Mathematics for Computation Workshop

Week 6 Workshop

Conceptual Review

(a) **Set Operations and Comparisons**

Set Equality: $A = B := \forall x(x \in A \leftrightarrow x \in B)$
Subset: $A \subseteq B := \forall x(x \in A \rightarrow x \in B)$
Union: $A \cup B := \{x \ : \ x \in A \vee x \in B\}$
Intersection: $A \cap B := \{x \ : \ x \in A \wedge x \in B\}$
Set Difference: $A \setminus B = A - B := \{x \ : \ x \in A \wedge x \notin B\}$
Set Complement: $\overline{A} = A^C := \{x \ : \ x \notin A\}$
Powerset: $\mathcal{P}(A) := \{B \ : B \subseteq A\}$
Cartesian Product: $A \times B := \{(a, b) \ : a \in A, \ b \in B\}$

(b) **Set Builder Notation**

Filter: $S := \{x \in U : P(x)\}$
Translation: S is all the things in U that satisfy $P(x)$.

Map: $T := \{f(x) : x \in U\}$
Translation: T is all output values from the function $f(x)$ when the input is something from U.

The : is read as "such that". It is also common to use | instead of :. When using set builder notation, the stuff before the : (or |) is the stuff in the set. The stuff after the : (or |) are requirements that stuff must fulfill to be in the set.

(c) How do we prove that for sets A and B, $A \subseteq B$?

(d) What are two ways we can prove that for sets A and B, $A = B$?

1. A Basic Subset Proof

Let A, B be sets. Consider the following claim:

$$A \cap B \subseteq A \cup B$$

(a) Write a **formal proof** that the claim holds. Use cozy-style rules for applying definitions. For example, You can replace $A \subseteq B$ by $\forall x(x \in A \rightarrow x \in B)$ with "Def of Subset" and the reverse with "Undef Subset".

(b) Translate your formal proof to an **English proof**. You may be surprised by how short your proof is!

2. Set Equality Proof

(a) Write an English proof to show that $A \cap (A \cup B) \subseteq A$ for sets A, B.

(b) Write an English proof to show that $A \subseteq A \cap (A \cup B)$ for sets A, B.

(c) Combine part (a) and (b) to conclude that $A \cap (A \cup B) = A$ for sets A, B.

(d) Re-write this proof using the Meta-Theorem template from lecture (i.e., using a chain of equivalences instead of two subset proofs).

3. Subsets

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

$$A \subseteq C \text{ follows from } A \subseteq B \text{ and } B \subseteq C$$

(a) Write a **formal proof** that the claim holds:

(b) Translate the formal proof to an **English Proof**.

4. Moderately Unsettling

Let A, B and C be the following sets:

$$A := \{x \in \mathbb{Z} : x \equiv_4 0\}$$
$$B := \{x \in \mathbb{Z} : x \equiv_4 2\}$$
$$C := \{x \in \mathbb{Z} : x \equiv_2 0\}$$

Consider the following claim:

$$C = (A \cup B)$$

(a) Write an English proof to show that $C \subseteq (A \cup B)$

(b) Write an English proof to show that $(A \cup B) \subseteq C$

(c) Combine part(a) and part(b) to show that $C = (A \cup B)$

5. $\cup \rightarrow \cap$?

Prove or disprove: for all sets A and B, $A \cup B \subseteq A \cap B$.
Recall that we can disprove a for all claim by finding a counter-example.

6. Powerful Ideas

Let A and B be sets. Consider the following claim:

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

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

7. Cartesian Product Proof

Let A, B, C, D be sets. Write an **English proof** of the follow claim:

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

8. Set Equality Proof II

Let A, B, C be sets. Consider the following claim

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

(a) Write a **formal proof** that the claim holds.

(b) Translate your proof to an **English Proof**.
Follow the Meta-Theorem template from lecture (i.e., using a chain of equivalences instead of two subset proofs).

(c) Optional: Re-write this proof as an **English Proof** that is made up of two subset proofs.

9. Structural Induction: Divisible by 4

Define a set T of numbers by:

- 4 and 12 are in T
- If $x \in T$ and $y \in T$, then $x + y \in T$ and $x - y \in T$

Prove by structural induction that every number in T is divisible by 4.

10. More Induction...Literally

Define a set S as follows:
Basis: $6 \in S$; $15 \in S$
Recursive: if $x, y \in S$ then $x + y \in S$

Define a set T as follows:
Basis: $6 \in T$; $15 \in T$
Recursive: if $x \in T$ then $x+6 \in T$ and $x+15 \in T$

In lecture you proved that every element of T is an element of S.
Now we're going to prove that every element of S is an element of T.

(a) First, use structural induction to prove the following lemma:
The sum of any two elements in T is also in T. Formally this is: $\forall a, b \in T(a + b \in T)$

(b) Now, use structural induction to prove the main claim: Every element of S is also in T.
You can use the Lemma from part (a) by citing "part (a) lemma".

11. We'll do this next week, but you can try it after Wednesday's lecture. Structural Induction: CharTrees

Recursive Definition of CharTrees:

- Basis Step: `Null` is a **CharTree**
- Recursive Step: If L, R are **CharTree**s and $c \in \Sigma$, then $\texttt{CharTree}(L, c, R)$ is also a **CharTree**

Intuitively, a **CharTree** is a tree where the non-null nodes store a `char` data element.

Recursive functions on CharTrees:

- The preorder function returns the preorder traversal of all elements in a **CharTree**.

$$\begin{aligned} \text{preorder}(\texttt{Null}) &= \varepsilon \\ \text{preorder}(\texttt{CharTree}(L, c, R)) &= c \cdot \text{preorder}(L) \cdot \text{preorder}(R) \end{aligned}$$

- The postorder function returns the postorder traversal of all elements in a **CharTree**.

$$\begin{aligned} \text{postorder}(\texttt{Null}) &= \varepsilon \\ \text{postorder}(\texttt{CharTree}(L, c, R)) &= \text{postorder}(L) \cdot \text{postorder}(R) \cdot c \end{aligned}$$

- The mirror function produces the mirror image of a **CharTree**.

$$\begin{aligned} \text{mirror}(\texttt{Null}) &= \texttt{Null} \\ \text{mirror}(\texttt{CharTree}(L, c, R)) &= \texttt{CharTree}(\text{mirror}(R), c, \text{mirror}(L)) \end{aligned}$$

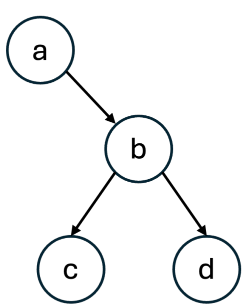
- Finally, for all strings x, x^R, the "reversal" of x, produces the string in reverse order.

Additional Facts:
You may use the following facts:

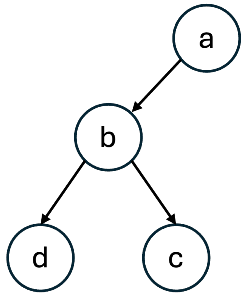
- **Fact 1:** For any strings $x_1, ..., x_k$: $(x_1 \cdot ... \cdot x_k)^R = x_k^R \cdot ... \cdot x_1^R$
- **Fact 2:** For any character c, $c^R = c$

It turns out that for any CharTree T, the reversal of the preorder traversal of T is the same as the postorder traversal of the mirror of T.

Example for Intuition:



Let T be the tree above.
preorder(T) ="abcd".
T is built as $(\texttt{Null}, a, U)$
Where U is (V, b, W),
$V = (\texttt{Null}, c, \texttt{Null}), W = (\texttt{Null}, d, \texttt{Null})$.



This tree is mirror(T).
postorder(mirror(T)) =“dcba”,
“dcba” is the reversal of “abcd” so
$[\text{preorder}(T)]^R = \text{postorder}(\text{mirror}(T))$ holds for T

Use structural induction to prove the following claim:

For every **CharTree**, T: $[\text{preorder}(T)]^R = \text{postorder}(\text{mirror}(T))$