CS 578 Programming Language Semantics Review Quiz – Spring 2024
Sample Solutions

1. Proof by induction over n.

Base case: $n = 0$. Left and right hand sides are both 0, so equation holds.

Induction case. Suppose as induction hypothesis (IH) that the equation is true for some n, i.e. $\sum_{i=0}^{n} i(i+1) = \frac{n(n+1)(n+2)}{3}$.

We must show that $\sum_{i=0}^{n+1} i(i+1) = \frac{(n+1)(n+2)(n+3)}{3}$.

We have

$\sum_{i=0}^{n+1} i(i+1) =$ (splitting off last term)

$\sum_{i=0}^{n} i(i+1) + (n+1)(n+2) =$ (by IH)

$\frac{n(n+1)(n+2)}{3} + (n+1)(n+2) =$ (forming common denominator)

$\frac{n(n+1)(n+2)+3(n+1)(n+2)}{3} =$ (regrouping)

$\frac{(n+3)(n+1)(n+2)}{3} =$ (reordering)

$\frac{(n+1)(n+2)(n+3)}{3}$

as desired.

2. We need to make (implicit or explicit) use of the definition of $\subseteq$, namely that if $T \subseteq U$, then $x \in T$ implies $x \in U$.

The three defining properties of a partial order follow easily.

- Reflexivity. For any set, it is immediate that $T \subseteq T$.
- Transitivity. If $T \subseteq U$ and $U \subseteq V$, then any element of T must be in U and hence also in V, so $T \subseteq V$.
- Antisymmetry. If $T \subseteq U$ and $U \subseteq T$, then every element of T must be in U and vice-versa, so indeed $T = U$.

3. Here's an OCaml solution:

```
type exp =
  True
| False
| And of exp * exp
| Or of exp * exp
| Not of exp

let rec eval e = match e with
  True -> true
| False -> false
| And(e1,e2) -> eval e1 && eval e2
| Or(e1,e2) -> eval e1 || eqval e2
| Not e -> not (eval e)
```