# CSC236 Tutorial Exercises, May 31, 2017
# Sample Solutions

Start the tutorial by having different students write a basis and begin the inductive step (write the hypothesis) for question 1. Discuss with the full group until you are satisfied. Then complete the inductive step (derive the conclusion).

Same exercise for question 2, if time permits.

1. Define the set of expressions $\mathcal{E}$ as the smallest set such that:

   (a) $x, y, z \in \mathcal{E}$.

   (b) If $e_1, e_2 \in \mathcal{E}$, then so are $(e_1 + e_2)$ and $(e_1 \times e_2)$.

   Define $p(e)$ : Number of parentheses in $e$.

   Define $s(e)$ : Number of symbols from $\{x, y, z, +, \times\}$ in $e$, counting duplicates.

   Use structural induction to prove that for all $e \in \mathcal{E}$, $p(e) = s(e) - 1$.

   **sample solution:** Proof by structural induction. For convenience, I define the predicate $P(e) : p(e) = s(e) - 1$.

   **verify basis:** The basis elements are three symbols $x, y, z$. For $e \in \{x, y, z\}$ the solitary symbol means $s(e) = 1$. There are no parentheses, so $p(e) = 0$, and $p(e) = 0 = 1 - 1 = s(e) - 1$. So, for any $e$ in the basis, $P(e)$.

   **inductive step:** Let $e_1, e_2$ be arbitrary elements of $\mathcal{E}$. Assume $H(\{e_1, e_2\})$: $P(e_1)$ and $P(e_2)$, that is $p(e_1) = s(e_1) - 1$ and $p(e_2) = s(e_2) - 1$.

   **derive $C(\{e_1, e_2\})$: $P((e_1 + e_2))$ and $P((e_1 \times e_2))$.** In other words, I must show $p((e_1 + e_2)) = s((e_1 + e_2)) - 1$ and $p((e_1 \times e_2)) = s((e_1 - e_2)) - 1$.
   *Let*$\odot \in \{+, \times\}$. $p((e_1 \odot e_2)) = p(e_1) + p(e_2) + 2$, since the new expression adds 2 parentheses to those contained in $e_1$ or $e_2$. $s((e_1) \odot e_2)) = s(e_1) + s(e_2) + 1$, since the new expression has all the symbols of $e_1$ and $e_2$, plus $\odot$. Putting these ideas together:

$$
\begin{aligned}
p((e_1 \odot e_2)) &= p(e_1) + p(e_2) + 2 \\
&= s(e_1) - 1 + s(e_2) - 1 + 2 \qquad \# \text{ by } H(\{e_1, e_2\}) \\
&= (s(e_1) + s(e_2) + 1) - 1 \qquad \# \text{ regrouping} \\
&= s((e_1 \odot e_2)) - 1
\end{aligned}
$$

   So, $P((e_1 \odot e_2))$. Since $\odot$ is an arbitrary element of $\{+, \times\}$, this establishes $C(\{e_1, e_2\})$.■

2. Define the set of non-empty full binary trees, $\mathcal{T}$, as the smallest set such that:

(a) Any single node is an element of $\mathcal{T}$.

(b) If $t_1, t_2 \in \mathcal{T}$, then so is any root node with edges to $t_1$ and $t_2$.[1]

Use structural induction to prove that any non-empty full binary tree has an odd number of nodes.

**sample solution:** Proof by structural induction. For convenience I define the predicate $P(t) : t$ has an odd number of nodes.

**basis:** The basis consists of single-node FBTs, hence every element of the basis has an odd, i.e. 1, number of nodes.

**inductive step:** Let $t_1, t_2$ be arbitrary elements of $\mathcal{T}$. Assume $H(\{t_1, t_2\})$: $P(t_1)$ and $P(t_2)$, that is, $t_1$ and $t_2$ each have an odd number of nodes.

**derive** $C(\{t_1, t_2\})$: Let $t$ be a tree formed by an arbitrary root node with edges to $t_1$ and $t_2$. Then $P(t)$, i.e. $t$ has an odd number of nodes.
Let $k_1, k_2 \in \mathbb{N}$ such that $t_1$ has $2k_1 + 1$ nodes and $t_2$ has $2k_2 + 1$ nodes. # By $P(t_1)$ and $P(t_2)$ each tree has an odd number of nodes.
The number of nodes in $t$ is the sum of the nodes in $t_1$ and $t_2$, that or $1 + 2k_1 + 1 + 2k_2 + 1 = 2(k_1 + k_2 + 1) + 1$. This is an odd number since $(k_1 + k_2 + 1) \in \mathbb{N}$, due to $k_1, k_2, 1, 2 \in \mathbb{N}$ and $\mathbb{N}$ being closed under $+, \times$.
So $P(t)$, which establishes $C(\{t_1, t_2\}$.■

3. Define the set $S = \{n \in \mathbb{N} \mid n \geq 2$ and $n$does not have a prime factorization$\}$. Assume that $S$ is not empty. Therefore, by well ordering principle it has a least element, denote it $l$. The number $l$ cannot be a prime number, then there are numbers $l_1$ and $l_2$ such that $l = l_1 \cdot l_2$, and $2 \leq l_1, l_2 < l$. Then, $l$ has a prime factorization, contradiction.

[1] An extremely precise definition would insist that $t_1$, $t_2$ and the root node must have no nodes in common, see the Course Notes, page 105.