0: Lists without Lisp!

(a) Prove that concat is symmetric across `[]`. That is, prove that concat(L, `[]`) $=$ concat(`[]`, L)

We proceed by the structual induction. Let `P(L)` be the statement that concat(L, `[]`) $=$ concat(`[]`, L)

Base Case: `L` $=$ `[]`:

$$\begin{aligned}
\mathsf{concat}(L, \texttt{[]}) &= \mathsf{concat}(\texttt{[]}, \texttt{[]}) \\
&= \texttt{[]} && \text{[definition of } \mathsf{concat}] \\
&= \mathsf{concat}(\texttt{[]}, \texttt{[]}) && \text{[definition of } \mathsf{concat}] \\
\mathsf{concat}(L, \texttt{[]}) &= \mathsf{concat}(\texttt{[]}, L)
\end{aligned}$$

Induction Hypothesis: For some `L1` $\in$ `List`, suppose `P(L1)` is true.

Induction Step: We want to show `P(L)` is true for `L =` x`::L1` for all $x \in \mathbb{R}$.

$$\begin{aligned}
\mathsf{concat}(\texttt{L}, \texttt{[]}) &= \mathsf{concat}(x :: \texttt{L1}, \texttt{[]}) \\
&= x :: \mathsf{concat}(\texttt{L1}, \texttt{[]}) && \text{[by definition of } \mathsf{concat}] \\
&= x :: \mathsf{concat}(\texttt{[]}, \texttt{L1}) && \text{[by I.H.]} \\
&= x :: \texttt{L1} && \text{[by definition of } \mathsf{concat}] \\
&= \mathsf{concat}(\texttt{[]}, x\texttt{::L1}) && \text{[by definition of } \mathsf{concat}] \\
&= \mathsf{concat}(\texttt{[]}, \texttt{L})
\end{aligned}$$

Thus, by structual induction, we have proven our original claim that concat is symmetric across `[]`. $\square$

(b) Prove that for all lists A, B, C, concatis associative. That is: concat(concat(A, B), C) = concat(A, concat(B, C))

We proceed by the structual induction. Let P(A) be the statement that concat(concat(A, B),C) = concat(A, concat(B, C)). Let B and C be arbitrary lists. We induct on A.

Base Case: A = []

$$
\begin{aligned}
\mathsf{concat}(\mathsf{concat}(\mathtt{A},\mathtt{B}),\mathtt{C}) &= \mathsf{concat}(\mathsf{concat}(\mathtt{[]},\mathtt{B}),\mathtt{C}) && \\
&= \mathsf{concat}(\mathtt{B},\mathtt{C}) && \text{[definition of } \mathsf{concat}\text{]} \\
&= \mathsf{concat}(\mathtt{[]},\mathsf{concat}(\mathtt{B},\mathtt{C})) && \text{[definition of } \mathsf{concat}\text{]} \\
\mathsf{concat}(\mathsf{concat}(\mathtt{A},\mathtt{B}),\mathtt{C}) &= \mathsf{concat}(\mathtt{A},\mathsf{concat}(\mathtt{B},\mathtt{C}))
\end{aligned}
$$

Induction Hypothesis: For some A1 $\in$ List, let P(A1) be true.

Induction Step: We want to show P(A) where A = x::A1 for all $x \in \mathbb{R}$.

$$
\begin{aligned}
\mathsf{concat}(\mathsf{concat}(\mathtt{A},\mathtt{B}),\mathtt{C}) &= \mathsf{concat}(\mathsf{concat}(\mathtt{x::A1},\mathtt{B}),\mathtt{C}) && \\
&= \mathsf{concat}(\mathtt{x::}\mathsf{concat}(\mathtt{A1},\mathtt{B}),\mathtt{C}) && \text{[by definition of } \mathsf{concat}\text{]} \\
&= \mathtt{x} :: \mathsf{concat}(\mathsf{concat}(\mathtt{A1},\mathtt{B}),\mathtt{C}) && \text{[by definition of } \mathsf{concat}\text{]} \\
&= \mathtt{x} :: \mathsf{concat}(\mathtt{A1},\mathsf{concat}(\mathtt{B},\mathtt{C})) && \text{[by I.H.]} \\
&= \mathsf{concat}(\mathtt{x::A1},\mathsf{concat}(\mathtt{B},\mathtt{C})) && \text{[by definition of } \mathsf{concat}\text{]} \\
&= \mathsf{concat}(\mathtt{A},\mathsf{concat}(\mathtt{B},\mathtt{C}))
\end{aligned}
$$

Thus, by structual induction, we have proven our original claim that concat is commutative. $\square$

(c) Prove that $\mathsf{rev}(\mathsf{concat}(\texttt{A}, \texttt{B})) = \mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{rev}(\texttt{A}))$ for all lists `A` and `B`.

- Lemma 1 (proved in part a): $\mathsf{concat}(\mathrm{L}, \texttt{[]}) = \mathsf{concat}(\texttt{[]}, \mathrm{L})$
- Lemma 2 (proved in part b): $\mathsf{concat}(\mathsf{concat}(\texttt{A}, \texttt{B}), \texttt{C}) = \mathsf{concat}(\texttt{A}, \mathsf{concat}(\texttt{B}, \texttt{C}))$

We proceed by the structual induction. Let `P(A)` be the statement that $\mathsf{rev}(\mathsf{concat}(\texttt{A}, \texttt{B})) = \mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{rev}(\texttt{A}))$. Let `B` be an arbitrary list. We induct on `A`.

Base Case: $\texttt{A} = \texttt{[]}$

$$
\begin{aligned}
\mathsf{rev}(\mathsf{concat}(\texttt{A}, \texttt{B})) &= \mathsf{rev}(\mathsf{concat}(\texttt{[]}, \texttt{B})) \\
&= \mathsf{rev}(\texttt{B}) && \text{[definition of } \mathsf{concat}\text{]} \\
&= \mathsf{concat}(\texttt{[]}, \mathsf{rev}(\texttt{B})) && \text{[definition of } \mathsf{concat}\text{]} \\
&= \mathsf{concat}(\mathsf{rev}(\texttt{B}), \texttt{[]}) && \text{[by Lemma 1]} \\
&= \mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{rev}(\texttt{[]})) && \text{[definition of } \mathsf{rev}\text{]} \\
\mathsf{rev}(\mathsf{concat}(\texttt{A}, \texttt{B})) &= \mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{rev}(\texttt{A}))
\end{aligned}
$$

Induction Hypothesis: For some $\texttt{A1} \in \texttt{List}$, let `P(A1)` be true.

Induction Step: We want to show `P(A)` where `A = x::A1` for all $x \in \mathbb{R}$,

$$
\begin{aligned}
\mathsf{rev}(\mathsf{concat}(\texttt{A}, \texttt{B})) &= \mathsf{rev}(\mathsf{concat}(\texttt{x::A1}, \texttt{B})) \\
&= \mathsf{rev}(\texttt{x::}\mathsf{concat}(\texttt{A1}, \texttt{B})) && \text{[definition of } \mathsf{concat}\text{]} \\
&= \mathsf{concat}(\mathsf{rev}(\mathsf{concat}(\texttt{A1}, \texttt{B})), \texttt{x::[]}) && \text{[definition of } \mathsf{rev}\text{]} \\
&= \mathsf{concat}(\mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{rev}(\texttt{A1})), \texttt{x::[]}) && \text{[by I.H.]} \\
&= \mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{concat}(\mathsf{rev}(\texttt{A1}), \texttt{x::[]})) && \text{[by Lemma 2]} \\
&= \mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{rev}(\texttt{x::A1})) \\
\mathsf{rev}(\mathsf{concat}(\texttt{A}, \texttt{B})) &= \mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{rev}(\texttt{A}))
\end{aligned}
$$

Thus, by structual induction, we have proven our original claim that $\mathsf{rev}(\mathsf{concat}(\texttt{A}, \texttt{B})) = \mathsf{concat}(\mathsf{rev}(\texttt{B}), \mathsf{rev}(\texttt{A}))$. $\square$

1: Proving BST Insertion Works!

(a) Prove that for all $b \in \mathbb{Z}, v \in \mathbb{Z}$ and all trees $\texttt{T}$, if $\mathsf{less}(b,\ \texttt{T})$, and $b > v$, then $\mathsf{less}(b,\ \mathsf{insert}(v,\ \texttt{T}))$

Let $\texttt{P(T)}$ be the statement that for all $b \in \mathbb{Z}, v \in \mathbb{Z}$, if $\mathsf{less}(\text{b},\ \texttt{T})$ and $b > v$, then $\mathsf{less}(\text{b},\ \mathsf{insert}(\text{v},\ \texttt{T}))$ i.e. for all $b \in \mathbb{Z}, v \in \mathbb{Z}$, $\mathsf{less}(b, \texttt{T}) \land b > v \Rightarrow \mathsf{less}(b, \mathsf{insert}(v, \texttt{T}))$.

We prove $\text{P}(\texttt{T})$ for all $\texttt{T} \in \texttt{Trees}$ by structural induction on $\texttt{T}$.

<u>Base Case:</u> $\texttt{T} = \texttt{Nil}$

$$\begin{aligned}
\mathsf{less}(b, \texttt{T}) \land b > v &= \mathsf{less}(b, \texttt{Nil}) \land b > v \\
&= \texttt{true} \land b > v && \text{[definition of } \mathsf{less}] \\
&\Rightarrow \texttt{true} \land \texttt{true} \land b > v \\
&= \mathsf{less}(v, \texttt{Nil}) \land \mathsf{less}(v, \texttt{Nil}) \land b > v && \text{[definition of } \mathsf{less}] \\
&= \mathsf{less}(b, \texttt{Tree}(v, \texttt{Nil}, \texttt{Nil})) && \text{[definition of } \mathsf{less}] \\
&= \mathsf{less}(b, \mathsf{insert}(v, \texttt{Nil})) && \text{[definition of } \mathsf{insert}] \\
&= \mathsf{less}(b, \mathsf{insert}(v, \texttt{T}))
\end{aligned}$$

Therefore, $\mathsf{less}(b, \texttt{T}) \land b > v \Rightarrow \mathsf{less}(b, \mathsf{insert}(v, \texttt{T}))$ for $\texttt{T} = \texttt{Nil}$ and our base case holds.

<u>Induction Hypothesis:</u> For some $\texttt{L},\ \texttt{R} \in \texttt{Trees}$, suppose $\texttt{P(L)}$ and $\texttt{P(R)}$ are true.

<u>Induction Step:</u> We want to show $\texttt{P(T)}$ is true for $\texttt{T = Tree(}x\texttt{, L, R)}$ for all $x \in \mathbb{Z}$. We start by looking at two cases: $v < x$ and $v \geq x$.

<u>Case: $v < x$</u>

$$\begin{aligned}
\mathsf{less}(b, \texttt{T}) \land b > v &= \mathsf{less}(b, \texttt{Tree(}x\texttt{, L, R)}) \land b > v \\
&= x < b \land \mathsf{less}(b, \texttt{L}) \land \mathsf{less}(b, \texttt{R}) \land b > v && \text{[definition of } \mathsf{less}] \\
&\Rightarrow x < b \land \mathsf{less}(b, \mathsf{insert}(v, \texttt{L})) \land \mathsf{less}(b, \texttt{R}) && \text{[by } \texttt{P(L)} \text{ in I.H.]} \\
&= \mathsf{less}(b, \texttt{Tree(}x\texttt{, }\mathsf{insert}\texttt{(}v\texttt{, L), R)}) && \text{[definition of } \mathsf{less}]
\end{aligned}$$

<u>Case: $v \geq x$</u>

$$\begin{aligned}
\mathsf{less}(b, \texttt{T}) \land b > v &= \mathsf{less}(b, \texttt{Tree(}x\texttt{, L, R)}) \land b > v \\
&= x < b \land \mathsf{less}(b, \texttt{L}) \land \mathsf{less}(b, \texttt{R}) \land b > v && \text{[definition of } \mathsf{less}] \\
&\Rightarrow x < b \land \mathsf{less}(b, \texttt{L}) \land \mathsf{less}(b, \mathsf{insert}(v, \texttt{R})) && \text{[by } \texttt{P(R)} \text{ in I.H.]} \\
&= \mathsf{less}(b, \texttt{Tree(}x\texttt{, L, }\mathsf{insert}\texttt{(}v\texttt{, R))}) && \text{[definition of } \mathsf{less}]
\end{aligned}$$

We see from our cases that if $v < x$ then $\mathsf{less}(b, \texttt{T}) \land b > v \Rightarrow \mathsf{less}(b, \texttt{Tree(}x\texttt{, }\mathsf{insert}\texttt{(}v\texttt{, L), R)})$ else $\mathsf{less}(b, \texttt{T}) \land b > v \Rightarrow \mathsf{less}(b, \texttt{Tree(}x\texttt{, L, }\mathsf{insert}\texttt{(}v\texttt{, R))})$.

By the definition of insert, we can roll both implications into one and state that $\mathsf{less}(b, \texttt{T}) \land b > v \Rightarrow \mathsf{less}(b, \mathsf{insert}(v, \texttt{Tree(}x\texttt{, L, R)}))$ which is equivalent to $\mathsf{less}(b, \texttt{T}) \land b > v \Rightarrow \mathsf{less}(b, \mathsf{insert}(v, \texttt{T}))$

Thus, by structual induction, we have shown that $\texttt{P(T)}$ is true for all $\texttt{T} \in \texttt{Trees}$. $\square$

(b) Prove that for all trees $\texttt{T}$ and all $v \in \mathbb{Z}$, if $\mathsf{isBST}(\texttt{T})$, then $\mathsf{isBST}(\mathsf{insert}(v, \texttt{T}))$.

Let $\texttt{P(T)}$ be the statement that for all $v \in \mathbb{Z}$, if $\mathsf{isBST}(\texttt{T})$, then $\mathsf{isBST}(\mathsf{insert}(\text{v}, \texttt{T}))$ i.e. for all $v \in \mathbb{Z}, \mathsf{isBST}(\texttt{T}) \Rightarrow \mathsf{isBST}(\mathsf{insert}(v, \texttt{T}))$.

We prove $\text{P}(\texttt{T})$ for all $\texttt{T} \in \texttt{Trees}$ by structural induction on $\texttt{T}$.

Base Case: $\texttt{T} = \texttt{Nil}$

$$
\begin{aligned}
\mathsf{isBST}(\texttt{T}) &= \mathsf{isBST}(\texttt{Nil}) \\
&= \texttt{true} && \text{[definition of } \mathsf{isBST}] \\
&\Rightarrow \texttt{true} \land \texttt{true} \land \texttt{true} \\
&= \mathsf{less}(v, \texttt{Nil}) \land \mathsf{isBST}(\texttt{Nil}) \land \mathsf{greater}(v, \texttt{Nil}) \land \mathsf{isBST}(\texttt{Nil}) && \text{[definition of } \mathsf{less}, \mathsf{isBST}, \text{ and } \mathsf{greater}] \\
&= \mathsf{isBST}(\texttt{Tree}(v, \texttt{Nil}, \texttt{Nil})) && \text{[definition of } \mathsf{isBST}] \\
&= \mathsf{isBST}(\mathsf{insert}(v, \texttt{Nil})) && \text{[definition of } \mathsf{insert}] \\
&= \mathsf{isBST}(\mathsf{insert}(v, \texttt{T}))
\end{aligned}
$$

Therefore, $\mathsf{isBST}(\texttt{T}) \Rightarrow \mathsf{isBST}(\mathsf{insert}(v, \texttt{T}))$ for $\texttt{T} = \texttt{Nil}$ and our base case holds.

Induction Hypothesis: For some $\texttt{L, R} \in \texttt{Trees}$, suppose $\texttt{P(L)}$ and $\texttt{P(R)}$ are true.

Induction Step: We want to show $\texttt{P(T)}$ is true for $\texttt{T = Tree(}x\texttt{, L, R)}$ for all $x \in \mathbb{Z}$. We start by looking at two cases: $v < x$ and $v \geq x$.

Case: $v < x$

$$
\begin{aligned}
\mathsf{isBST}(\texttt{T}) &= \mathsf{isBST}(\texttt{Tree(}x\texttt{, L, R)}) \\
&= \mathsf{less}(x, \texttt{L}) \land \mathsf{isBST}(\texttt{L}) \land \mathsf{greater}(x, \texttt{R}) \land \mathsf{isBST}(\texttt{R}) && \text{[definition of } \mathsf{isBST}] \\
&\Rightarrow \mathsf{less}(x, \texttt{L}) \land x > v \land \mathsf{isBST}(\texttt{L}) \land \mathsf{greater}(x, \texttt{R}) \land \mathsf{isBST}(\texttt{R}) && \text{[by our case]} \\
&\Rightarrow \mathsf{less}(x, \mathsf{insert}(v, \texttt{L})) \land \mathsf{isBST}(\texttt{L}) \land \mathsf{greater}(x, \texttt{R}) \land \mathsf{isBST}(\texttt{R}) && \text{[by theorem proven in part a]} \\
&\Rightarrow \mathsf{less}(x, \mathsf{insert}(v, \texttt{L})) \land \mathsf{isBST}(\mathsf{insert}(v, \texttt{L})) \land \mathsf{greater}(x, \texttt{R}) \land \mathsf{isBST}(\texttt{R}) && \text{[by } \texttt{P(L)} \text{ in I.H.]} \\
&= \mathsf{isBST}(\texttt{Tree(}x\texttt{, }\mathsf{insert}(v\texttt{, L})\texttt{, R)}) && \text{[definition of } \mathsf{isBST}]
\end{aligned}
$$

Case: $v \geq x$

$$
\begin{aligned}
\mathsf{isBST}(\texttt{T}) &= \mathsf{isBST}(\texttt{Tree(}x\texttt{, L, R)}) \\
&= \mathsf{less}(x, \texttt{L}) \land \mathsf{isBST}(\texttt{L}) \land \mathsf{greater}(x, \texttt{R}) \land \mathsf{isBST}(\texttt{R}) && \text{[definition of } \mathsf{isBST}] \\
&\Rightarrow \mathsf{less}(x, \texttt{L}) \land \mathsf{isBST}(\texttt{L}) \land \mathsf{greater}(x, \texttt{R}) \land v \geq x \land \mathsf{isBST}(\texttt{R}) && \text{[by our case]} \\
&\Rightarrow \mathsf{less}(x, \texttt{L}) \land \mathsf{isBST}(\texttt{L}) \land \mathsf{greater}(x, \mathsf{insert}(v, \texttt{R})) \land \mathsf{isBST}(\texttt{R}) && \text{[symmetric theorem for greater]} \\
&\Rightarrow \mathsf{less}(x, \texttt{L}) \land \mathsf{isBST}(\texttt{L}) \land \mathsf{greater}(x, \mathsf{insert}(v, \texttt{R})) \land \mathsf{isBST}(\mathsf{insert}(v, \texttt{R})) && \text{[by } \texttt{P(R)} \text{ in I.H.]} \\
&= \mathsf{isBST}(\texttt{Tree(}x\texttt{, L, }\mathsf{insert}(v\texttt{, R)})) && \text{[definition of } \mathsf{isBST}]
\end{aligned}
$$

We see from our cases that if $v < x$ then $\mathsf{isBST}(\texttt{T}) \Rightarrow \mathsf{isBST}(\texttt{Tree(}x\texttt{, }\mathsf{insert}(v\texttt{, L), R)}))$ else $\mathsf{isBST}(\texttt{T}) \Rightarrow \mathsf{isBST}(\texttt{Tree(}x\texttt{, L, }\mathsf{insert}(v\texttt{, R)})))$.

By the definition of insert, we can roll both implications into one and state that $\mathsf{isBST}(\texttt{T}) \Rightarrow \mathsf{isBST}(\mathsf{insert}(v, \texttt{Tree(}x\texttt{, L, R)}))$ which is equivalent to $\mathsf{isBST}(\texttt{T}) \Rightarrow \mathsf{isBST}(\mathsf{insert}(v, \texttt{T}))$

Thus, by structual induction, we have shown that $\texttt{P(T)}$ is true for all $\texttt{T} \in \texttt{Trees}$. $\square$