Interactive Formal Verification (L21)

1 Replace, Reverse and Delete

▷ Define a function \texttt{replace}, such that \texttt{replace x y zs} yields \texttt{zs} with every occurrence of \texttt{x} replaced by \texttt{y}.

\begin{verbatim}
fun replace :: "'a ⇒ 'a ⇒ 'a list ⇒ 'a list" where
  "replace x y [] = []"
| "replace x y (z#zs) = (if z=x then y else z)#(replace x y zs)"
\end{verbatim}

▷ Prove or disprove (by counterexample) the following theorems. You may have to prove some lemmas first.

\textbf{lemma replace_append}: \texttt{"replace x y (xs @ ys) = replace x y xs @ replace x y ys"}

\begin{verbatim}
apply (induct "xs")
apply auto
done
\end{verbatim}

\textbf{theorem}: \texttt{"rev (replace x y zs) = replace x y (rev zs)"

\begin{verbatim}
apply (induct "zs")
apply (auto simp add: replace_append)
done
\end{verbatim}

\textbf{theorem}: \texttt{"replace x y (replace u v zs) = replace u v (replace x y zs)"

\begin{verbatim}
quickcheck
oops
\end{verbatim}

A possible counterexample: \texttt{u=0, v=1, x=-1, y=0, zs=[0]}

\textbf{theorem}: \texttt{"replace y z (replace x y zs) = replace x z zs"

\begin{verbatim}
quickcheck
oops
\end{verbatim}

A possible counterexample: \texttt{x=1, y=0, z=1, zs=[0]}
Define two functions for removing elements from a list: \( \text{del1} \ x \ xs \) deletes the first occurrence (from the left) of \( x \) in \( xs \), \( \text{delall} \ x \ xs \) all of them.

```haskell
fun \text{del1} :: 'a ⇒ 'a list ⇒ 'a list where
\quad \text{del1} \ x \ [] = []
\quad \text{del1} \ x \ (y#ys) = (\text{if} \ y=x \ \text{then} \ ys \ \text{else} \ y \ # \ \text{del1} \ x \ ys)
```

```haskell
fun \text{delall} :: 'a ⇒ 'a list ⇒ 'a list where
\quad \text{delall} \ x \ [] = []
\quad \text{delall} \ x \ (y#ys) = (\text{if} \ y=x \ \text{then} \ \text{delall} \ x \ ys \ \text{else} \ y \ # \ \text{delall} \ x \ ys)
```

Prove or disprove (by counterexample) the following theorems.

**Theorem 1:** \( \text{del1} \ x \ (\text{delall} \ x \ xs) = \text{delall} \ x \ xs \)

```proof
apply (induct "xs")
apply auto
done
```

**Theorem 2:** \( \text{delall} \ x \ (\text{delall} \ x \ xs) = \text{delall} \ x \ xs \)

```proof
apply (induct "xs")
apply auto
done
```

**Theorem 3:** \( \text{delall} \ x \ (\text{del1} \ x \ xs) = \text{delall} \ x \ xs \)

```proof
apply (induct "xs")
apply auto
done
```

**Theorem 4:** \( \text{del1} \ x \ (\text{del1} \ y \ zs) = \text{del1} \ y \ (\text{del1} \ x \ zs) \)

```proof
apply (induct "zs")
apply auto
done
```

**Theorem 5:** \( \text{delall} \ x \ (\text{del1} \ y \ zs) = \text{del1} \ y \ (\text{delall} \ x \ zs) \)

```proof
apply (auto simp add: \text{delall} \ x \ (\text{del1} \ x \ zs))
done
```

**Theorem 6:** \( \text{delall} \ x \ (\text{delall} \ y \ zs) = \text{delall} \ y \ (\text{delall} \ x \ zs) \)

```proof
apply (auto)
done
```

**Theorem 7:** \( \text{del1} \ y \ (\text{replace} \ x \ y \ xs) = \text{del1} \ x \ xs \)

```proof
quickcheck
```

2
A possible counterexample: x=1, xs=[0], y=0

```plaintext
theorem "delall y (replace x y xs) = delall x xs"
quickcheck
```

A possible counterexample: x=1, xs=[0], y=0

```plaintext
theorem "replace x y (delall x zs) = delall x zs"
  apply (induct "zs")
  apply auto
done
```

```plaintext
theorem "replace x y (delall z zs) = delall z (replace x y zs)"
quickcheck
```

A possible counterexample: x=1, y=0, z=0, zs=[1]

```plaintext
theorem "rev (del1 x xs) = del1 x (rev xs)"
quickcheck
```

A possible counterexample: x=1, xs=[1, 0, 1]

```plaintext
lemma delall_append: "delall x (xs @ ys) = delall x xs @ delall x ys"
  apply (induct "xs")
  apply auto
done
```

```plaintext
theorem "rev (delall x xs) = delall x (rev xs)"
  apply (induct "xs")
  apply (auto simp add: delall_append)
done
```