Quiz Section 8: Arrays – Solutions

Task 1 – Better Get Proving [14 pts]

The function replace : $(\text{List}, y, z) \to \text{List}$ is defined by

$$\begin{aligned} \text{replace}(\text{nil}, y, z) &:= \text{nil} \\ \text{replace}(x :: L, y, z) &:= z :: \text{replace}(L, y, z) \quad \text{if } x = y \\ \text{replace}(x :: L, y, z) &:= x :: \text{replace}(L, y, z) \quad \text{if } x \neq y \end{aligned}$$

In this problem, we will prove that replace works the same way at the end of the list that it does at the front of the list, i.e.:

$$\begin{aligned} \text{replace}(L +\!\!+ [x], y, z) &= \text{replace}(L, y, z) +\!\!+ [z] \quad \text{if } x = y \\ \text{replace}(L +\!\!+ [x], y, z) &= \text{replace}(L, y, z) +\!\!+ [x] \quad \text{if } x \neq y \end{aligned}$$

a) Explain, in your own words, why the following statement, if proven, would imply the ones above:

$$\text{replace}(L +\!\!+ [x], y, z) = \text{replace}(L, y, z) +\!\!+ \text{replace}([x], y, z)$$

If $x = y$, then $\text{replace}([x], y, z) = [z]$ by the definition of replace, and if $x \neq y$, then $\text{replace}([x], y, z) = [x]$ by the definition of replace.

b) Explain, in your own words, why the following claim, if proven, would imply the one from part (a):

$$\text{replace}(L +\!\!+ R, y, z) = \text{replace}(L, y, z) +\!\!+ \text{replace}(R, y, z)$$

Setting $R = [x]$ gives the claim from part (a).

c) Prove the claim from part (b) by induction on L.

Define $P(L)$ to be the claim $\text{replace}(L +\!\!+ R, y, z) = \text{replace}(L, y, z) +\!\!+ \text{replace}(R, y, z)$. We will prove that this holds for all values of L by induction.

Base Case. We can see that $P(\text{nil})$ holds as follows:

$$\begin{aligned} \text{replace}(\text{nil} +\!\!+ R, y, z) &= \text{replace}(R, y, z) \\ &= \text{nil} +\!\!+ \text{replace}(R, y, z) \\ &= \text{replace}(\text{nil}, y, z) +\!\!+ \text{replace}(R, y, z) \quad \text{def of replace} \end{aligned}$$

Inductive Hypothesis. Suppose that $P(L)$ holds for some arbitrary L.

Inductive Step. We must show that $P(w :: L)$ holds for any w so we will continue by cases:

Suppose that $w = y$. Then, we can see that

$$\begin{aligned}
&\text{replace}((w :: L) +\!\!+ R, y, z) \\
&= z :: \text{replace}(L +\!\!+ R, y, z) && \text{def of replace (since } w = y) \\
&= z :: (\text{replace}(L, y, z) +\!\!+ \text{replace}(R, y, z)) && \text{Ind. Hyp.} \\
&= (z :: \text{replace}(L, y, z)) +\!\!+ \text{replace}(R, y, z) \\
&= \text{replace}(w :: L, y, z) +\!\!+ \text{replace}(R, y, z) && \text{def of replace (since } w = y)
\end{aligned}$$

Now, suppose that $w \neq y$. Then, we can see that

$$\begin{aligned}
&\text{replace}((w :: L) +\!\!+ R, y, z) \\
&= w :: \text{replace}(L +\!\!+ R, y, z) && \text{def of replace (since } w \neq y) \\
&= w :: (\text{replace}(L, y, z) +\!\!+ \text{replace}(R, y, z)) && \text{Ind. Hyp.} \\
&= (w :: \text{replace}(L, y, z)) +\!\!+ \text{replace}(R, y, z) \\
&= \text{replace}(w :: L, y, z) +\!\!+ \text{replace}(R, y, z) && \text{def of replace (since } w \neq y)
\end{aligned}$$

Since these two cases are exhaustive, we have proven $P(w :: L)$ holds in general.

Conclusion. $P(L)$ holds for all L by induction.

Task 2 – Jumping Through Loops [16 pts]

In this problem, you will implement the following function:

```
/**
 * Writes over each copy of y in A with the value z.
 * @param A .. y .. z ..
 * @modifies A
 * @effects A = replace(A_0, y, z)
 */
public void replace(int[] A, int y, int z) { .. }
```

With each loop invariant below, fill in the missing parts of the code to make it correct with the **given invariant**. (If the code works correctly with some other invariant, it is not correct.)

a)

```
int i = __________

// Inv: A[.. i] = A_0[.. i] and A[i+1 ..] = replace(A_0[i+1 ..], y, z)
while (__________) {



}
```

```
int i = A.length - 1;

// Inv: A[.. i] = A_0[.. i] and A[i+1 ..] = replace(A_0[i+1 ..], y, z)
while (i >= 0) {
     if (A[i] == y)
         A[i] = z;
     i = i - 1;
}
```

b)

```
int i = __________

// Inv: A[.. i-1] = replace(A_0[.. i-1], y, z) and A[i ..] = A_0[i ..]
while (__________) {



}
```

```
int i = 0;

// Inv: A[.. i-1] = replace(A_0[.. i-1], y, z) and A[i ..] = A_0[i ..]
while (i < A.length) {
```

```
    if (A[i] == y)
        A[i] = z;
    i = i + 1;
}
```

Task 3 – Rally the Loops **[15 pts]**

Recall the function remove : $(\text{List}, y) \to \text{List}$ defined as follows:

$$\begin{aligned} \text{remove}(\text{nil}, y) &:= \text{nil} \\ \text{remove}(x :: L, y) &:= \text{remove}(L, y) && \text{if } x = y \\ \text{remove}(x :: L, y) &:= x :: \text{remove}(L, y) && \text{if } x \neq y \end{aligned}$$

It is possible to prove, as the same manner as we did in Task 1, that the following holds:

$$\begin{aligned} \text{remove}(L +\!\!+ [x], y) &= \text{remove}(L, y) && \text{if } x = y \\ \text{remove}(L +\!\!+ [x], y) &= \text{remove}(L, y) +\!\!+ [x] && \text{if } x \neq y \end{aligned}$$

You can use the above facts below without proof. Refer to them as "Lemma".

In this problem, we will check the correctness of the following code that implements remove. Specifically, it writes $\text{remove}(A, y)$ into some prefix of the array, $A[..\, j-1]$, and returns j.

```
int i = 0;
int j = 0;
{{ P1: ________________________________________ }}
{{ Inv : A[.. j − 1] = remove(A0[.. i − 1], y) and A[i ..] = A0[i ..] and len(A) = len(A0) }}
while (i != A.length) {
  if (A[i] == y) {
      {{ P2: ____________________________________ }}
      {{ Q2: ____________________________________ }}
  } else {
      {{ P3: ____________________________________ }}
      {{ Q3: ____________________________________ }}
      A[j] = A[i];
      j = j + 1;
  }
  i = i + 1;
}
{{ P4: ________________________________________ }}
{{ Post: A[.. j − 1] = remove(A0, y) }}
return j;
```

a) Fill in P_1 using forward reasoning and then prove that P_1 implies Inv.

P_1 should say that $i = 0$ and $j = 0$ (and $A = A_0$ if you want).
The first fact of Inv holds since

$$\begin{aligned} A[..j-1] &= A[..0-1] && \text{since } j = 0 \\ &= \text{nil} \\ &= \text{remove}(\text{nil}, y) && \text{def of remove} \\ &= \text{remove}(A_0[..-1], y) \\ &= \text{remove}(A_0[..i-1], y) && \text{since } i = 0 \end{aligned}$$

The second fact of Inv holds since

$$\begin{aligned} A[i..] &= A[0..] && \text{since } i = 0 \\ &= A \\ &= A_0 && \text{since } A = A_0 \\ &= A_0[0..] \\ &= A_0[i..] && \text{since } i = 0 \end{aligned}$$

The third fact of Inv holds since

$$\text{len}(A) = \text{len}(A_0) \quad \text{since } A = A_0$$

b) Fill in P_4 using forward reasoning and then prove that P_4 implies the postcondition.

P_4 should say $A[..j-1] = \text{remove}(A_0[..i-1], y)$, $A[i..] = A_0[i..]$, $\text{len}(A) = \text{len}(A_0)$ and $i = \text{len}(A)$. This gives us the postcondition since

$$\begin{aligned} A[..j-1] &= \text{remove}(A_0[..i-1], y) && \text{since } A[..j-1] = \text{remove}(A_0[..i-1], y) \\ &= \text{remove}(A_0[..\text{len}(A)-1], y) && \text{since } i = \text{len}(A) \\ &= \text{remove}(A_0[..\text{len}(A_0)-1], y) && \text{since len}(A_0) = \text{len}(A) \\ &= \text{remove}(A_0, y) \end{aligned}$$

c) Fill in P_2 using forward reasoning and Q_2 using backward. Then, prove that P_2 implies Q_2.

P_2 should say:
$A[..j-1] = \text{remove}(A_0[..i-1], y)$, $A[i..] = A_0[i..]$, $\text{len}(A) = \text{len}(A_0)$, $i \neq \text{len}(A)$, and $A[i] = y$.

Q_2 should say:
$A[..j-1] = \text{remove}(A_0[..i], y)$, $A[i+1..] = A_0[i+1..]$, and $\text{len}(A) = \text{len}(A_0)$.

The second part of Q_2 is implied by the second fact from P_2 (since $A[i+1..]$ is a sublist of $A[i..]$). The third fact is directly implied. The first fact follows since

$$\begin{aligned} A[..j-1] &= \text{remove}(A_0[..i-1], y) \\ &= \text{remove}(A_0[..i-1] \mathbin{+\!\!+} [A[i]], y) && \text{Lemma (since } y = A[i]) \\ &= \text{remove}(A_0[..i-1] \mathbin{+\!\!+} [A_0[i]], y) && \text{since } A[i..] = A_0[i..] \text{ implies } A[i] = A_0[i] \\ &= \text{remove}(A_0[..i], y) \end{aligned}$$

d) Fill in P_3 using forward reasoning and Q_3 using backward. Then, prove that P_3 implies Q_3.

P_3 should say:
$A[..\, j-1] = \text{remove}(A_0[..\, i-1], y)$, $A[i\, ..] = A_0[i\, ..]$, $\text{len}(A) = \text{len}(A_0)$, $i \neq \text{len}(A)$ and $A[i] \neq y$.

Q_3 should say:
$A[..\, j-1] \mathbin{+\!\!+} [A[i]] = \text{remove}(A_0[..\, i], y)$, $A[i+1\, ..] = A_0[i+1\, ..]$, and $\text{len}(A) = \text{len}(A_0)$.

The second part of Q_3 is implied by the second fact from P_3 (since $A[i+1\, ..]$ is a sublist of $A[i\, ..]$). The third fact is directly implied. The first fact follows since

$$\begin{aligned}
& A[..\, j-1] \mathbin{+\!\!+} [A[i]] \\
&= \text{remove}(A_0[..\, i-1], y) \mathbin{+\!\!+} [A[i]] && \text{since } A[..\, j-1] = \text{remove}(A_0[..\, i-1], y) \\
&= \text{remove}(A_0[..\, i-1] \mathbin{+\!\!+} [A[i]], y) && \text{by Lemma since } y \neq A[i] \\
&= \text{remove}(A_0[..\, i-1] \mathbin{+\!\!+} [A_0[i]], y) && \text{since } A[i..] = A_0[i..] \text{ implies } A[i] = A_0[i] \\
&= \text{remove}(A_0[..\, i], y)
\end{aligned}$$