141 Reasoning About Program 2020 Exam Sample Solution

*Disclaimer: This is not an official answer key, so please correct any mistake if you find one, especially in **2a iii) and 2b**. There are more than one possible solution for some questions, so keep in mind that!*

1. **a.** `([1, 2], ['a', 'b'])`

   b. `([3, 6, 5, 2], ['c', 'f', 'e', 'b'])`

   c. We use structural induction on zs to prove the assertion.

   Base Case: To show $\forall \mathrm{xs} : [\mathrm{a}] \forall \mathrm{ys} : [\mathrm{b}].\ [\mathrm{splitTR}\ []\ \mathrm{xs}\ \mathrm{ys} = (\mathrm{xs}, \mathrm{ys}) <> (\mathrm{split}\ [])]$.

   Take arbitary $\mathrm{xs} : [\mathrm{a}]$ and $\mathrm{ys} : [\mathrm{b}]$.

$$\begin{aligned}\texttt{splitTR [] xs ys} &= \texttt{(xs,ys)} && \texttt{(By definition of splitTR)}\\ &= \texttt{(xs,ys)<>([],[])} && \texttt{(By Lemma A)}\\ &= \texttt{(xs,ys)<>(split [])} && \texttt{(By definition of split)}\end{aligned}$$

Hence, the base case holds.

Inductive Step: (*Alternatively, one can take arbitary variables first and then show inductive hypothesis and "to show"*)

Inductive hypothesis:
$\forall \mathrm{zs} : [(\mathrm{a}, \mathrm{b})] \forall \mathrm{xs} : [\mathrm{a}] \forall \mathrm{ys} : [\mathrm{b}].\ [\mathrm{splitTR}\ \mathrm{zs}\ \mathrm{xs}\ \mathrm{ys} = (\mathrm{xs}, \mathrm{ys}) <> (\mathrm{split}\ \mathrm{zs})]$

To show: $\forall \mathrm{m} : \mathrm{a} \forall \mathrm{n} : \mathrm{b}.\ [\mathrm{splitTR}\ ((\mathrm{m}, \mathrm{n}) : \mathrm{zs})\ \mathrm{xs}\ \mathrm{ys} = (\mathrm{xs}, \mathrm{ys}) <> (\mathrm{split}\ ((\mathrm{m}, \mathrm{n}) : \mathrm{zs}))]$

Take arbitary $\mathrm{xs} : [\mathrm{a}]$, $\mathrm{ys} : [\mathrm{b}]$, and $\mathrm{zs} : [(\mathrm{a}, \mathrm{b})]$, $\mathrm{m} : \mathrm{a}$, and $\mathrm{n} : \mathrm{b}$.

$$\begin{aligned}\texttt{splitTR ((m,n):zs) xs ys} &= \texttt{splitTR zs (xs++[m]) (ys++[n])} && \texttt{(By def. of splitTR)}\\ &= \texttt{((xs++[m]),(ys++[n]))<>(split zs)} && \texttt{(By inductive hypo.)}\\ &= \texttt{((xs,ys)<>([m],[n]))<>(split zs)} && \texttt{(By def. of <>)}\\ &= \texttt{(xs,ys)<>(([m], [n])<>(split zs))} && \texttt{(By Lemma B)}\\ &= \texttt{(xs,ys)<>(split ((m,n):zs))} && \texttt{(By def. of split)}\end{aligned}$$

Hence, the inductive cas holds. Hence, the assertion has been proved true.

d. We use structural induction again to prove this assertion.

Base Case: To show `splitTR [] [] [] = split []`.

$$\begin{aligned}\texttt{splitTR [] [] []} &= \texttt{([],[])} && \texttt{(By definition of splitTR)}\\ &= \texttt{split []} && \texttt{(By definition of split)}\end{aligned}$$

Hence, the base case holds.

Inductive Step: (*Alternatively, one can show inductive hypothesis first and then take arbitary variables*)

Take arbitary `zs:[(a,b)], m:a, n:b`.

Inductive hypothesis: `splitTR zs [] [] = split zs`

To show: `splitTR ((m,n):zs) [] [] = split ((m,n):zs)`

$$
\begin{aligned}
\texttt{splitTR ((m,n):zs) [] []} &= \texttt{splitTR zs ([]++[m]) ([]++[n])} && \texttt{(By def. of splitTR)}\\
&= \texttt{splitTR zs [m] [n]} && \texttt{(By definition of ++)}\\
&= \texttt{([m],[n])<>(split zs)} && \texttt{(By assertion from c)}\\
&= \texttt{split ((m,n):zs)} && \texttt{(By def. of split)}
\end{aligned}
$$

Hence, the inductive case holds. Hence, the assertion has been proved.

e. Inductive principle:

$$
\begin{gathered}
\forall \texttt{m,i,j:Int.Q(m,m+i,i,j,i)} \land \forall \texttt{n,i,j:Int.Q(n+j,n,i,j,j)} \land \\
\forall \texttt{m,n,i,j,r:Int.}\,[\texttt{m+i}\neq\texttt{n} \land \texttt{n+j}\neq\texttt{m} \land \texttt{r=F(m,n,i+2,j+3)} \land \texttt{Q(m,n,i+2,j+3,r)} \to \texttt{Q(m,n,i,j,5*r)}] \\
\to \forall \texttt{m,n,i,j,r:Int.}\,[\texttt{r=F(m,n,i,j)} \to \texttt{Q(m,n,i,j,r)}]
\end{gathered}
$$

2. **a. i)** The invariant is as shown below:

$$
\begin{gathered}
I \triangleq \texttt{str[..)} \approx \texttt{str[..)}_{pre} \land 0 < \texttt{i} \leq \texttt{str.length} \land 0 \leq \texttt{j} \leq \texttt{str.length} \land \\
\texttt{str.length} - \texttt{j} \leq \texttt{i} \land \mathrm{Wb}(\texttt{str[..i-1]}) \land \forall \texttt{n} \in \texttt{[0..stack.length-j)}.\mathrm{Cb}(\texttt{stack[n]})
\end{gathered}
$$

ii) $V \triangleq \texttt{str.length} - \texttt{i}$

iii) $M_1 \triangleq I \land \texttt{i<str.length} \land \texttt{c=str[i]}$

$M_2 \triangleq I \land \texttt{i<str.length} \land \mathrm{Ob}(\texttt{c})$

$M_3 \triangleq I \land \texttt{i<str.length} \land \mathrm{Cb}(\texttt{c}) \land (\texttt{j=stack.length} \leftrightarrow \forall \texttt{n} \in [\texttt{..i}).\,\mathrm{Nb}(\texttt{str[n]}))$

b. Given:

`1)`	$\texttt{str[..)} \approx \texttt{str[..)}_{pre}$	(From loop invariant)
`2)`	$\forall \texttt{n} \in \texttt{[0..stack.length-j)}.\texttt{Cb(stack[n])}$	(From loop invariant)
`3)`	$\texttt{j = stack.length} \leftrightarrow \forall \texttt{n} \in [\texttt{..i}).\,\mathrm{Nb}(\texttt{str[n]})$	(From M3)
`4)`	`Cb(c)`	(From line 17)
`5)`	$\texttt{j = stack.length} \lor \texttt{c}\neq\texttt{stack[j]}$	(From line 19)
`6)`	`r = false`	(From line 20)

To show:

`7)` $\texttt{str[..)} \approx \texttt{str[..)}_{pre}$

`8)` $\texttt{r} \leftrightarrow \mathrm{Wb}(\texttt{str[..)})$

`7)` follows directly from `1)`.

Case 1: When `j = stack.length` in `5)` holds true.

Then from `3)` we know that `9)` $\forall n \in [\,..i).\,\texttt{Nb(str[n])}$

From the definition of $Wb()$ and `9)` we know that `10)` `Wb(str[..i))`

From `9)` and `Lemma 1` we know that `11)` $\neg$`Wb(str[..i])`

From `6)` and `11)` we know that $\texttt{false} \leftrightarrow \texttt{Wb(str[..i])}$, which is equivalent to `8)`

Case 2: When `5.1)` $\mathtt{j} \neq \mathtt{stack.length}$ and `5.2)` $\mathtt{c} \neq \mathtt{stack[j]}$ holds.

From `5.1)` and 3 we know that $\neg \forall \mathrm{n} \in [\,..\,\mathrm{i}).\, \mathrm{Nb}(\mathtt{str[n]})$, which is equivalent to $\exists \mathrm{n} \in [\,..\,\mathrm{i}).\, \neg \mathrm{Nb}(\mathtt{str[n]})$, which is equivalent to $\exists \mathrm{n} \in [\,..\,\mathrm{i}).\, \mathrm{Ob}(\mathtt{str[n]}) \vee \mathrm{Cb}(\mathtt{str[n]})$.

From `2)`, `5.2)` and `Lemma 2`, we know that $\neg \mathrm{Wb}(\mathrm{str}[\,..\,\mathrm{i}])$, which would also prove `8)` following from case 1.

We can create an array with length of only half of the length of the string. If the string is well-bracketed, then for each opening bracket, there should be a closing bracket at the proper location. This means that if there are more opening bracket than closing bracket (or the other way around), the string is definitely NOT well-bracketed. Hence, we only need an array with length half of the string to determine whether the stirng is well-bracketed or note. A possible worse case scenario: `str="(([[]]))"`. (*Any string with the first half being opening bracket and the second half being closing brackets properly paired will do*).