

Study questions for Lecture 2 – Answers

a) We gave an "axiom" (assertion oriented definition) for a "one sided" conditional, i.e. if C then S. Construct an axiom for a "two sided" conditional., i.e. if C then S1 else S2

{P and C} S1 {Q}

and

{P and not C} S2 {Q}

implies

{P} if C then S1 else S2 {Q}

b) Construct an intermediate assertion for the following algorithm, identify all pairs of assertions A1,A2 which have an assertion free path between them, and for each argue that if A1 is true and the path followed, then A2 will be true

```
present(element, a[], n) /{integer: element, a[], n; n > 0}
j = 0;
result = False;
while (j < n)
    {if (a[j] == element)
        {result = True;
         exit;}}
    /{ for all i <= j, a[i] != element); j < n; result == False}
    else j = j+1;
}
/{ if element == a[k] for some k, 0 <= k < n then result = True, else False}
```

i) /{integer: element, a[], n; n > 0} -> /{ for all i <= j, a[i] != element); j < n; result == False}

If the first assertion is true, and we follow the path to the second, it will be true because the value of result gets set to False along this path, because j will be 0, and we the branch condition along this path ensure that the only element a[i] for which k <= j, namely i = 0, is not equal to element.

ii) /{ for all i <= j, a[i] != element); j < n; result == False}

-> /{ for all i <= j, a[i] != element); j < n; result == False}

This is the loop assertion where we go around the loop and get back to the same assertion.

If it is true before it will be true again, because, we will have checked the updated value of j at the top of the loop and it will still be < n. Also, because of the branch structure, result will still be False, Finally, the extra value of a[i] for the incremented value of j, will still not be equal to element because of the loop exit branch right before the assertion.

iii) $\{ \text{for all } i \leq j, a[i] \neq \text{element}; j < n; \text{ result} == \text{False} \}$

$\rightarrow \{ \text{if element} == a[k] \text{ for some } k, 0 \leq k < n \text{ then result} = \text{True, else False} \}$

This is the loop termination assertion. There are two ways to get here, one from the loop condition and the other from the mid-loop exit. If the first assertion is true and we leave the loop at the looping statement at the top of the loop, we will know that there is no k , $0 \leq k < n$ for which $a[k] = \text{element}$, and result will still be False. Hence, this will be consistent with the loop termination assertion. If we exit from the mid-loop loop exit, then result will be changed to True and we know that the first part of the loop termination assertion will be false, so that this will be consistent with the loop termination assertion.