Tutotial 6

Example 1

It is NOT possible to prove

Pre: $l = \langle a_1, a_2, \ldots, a_n \rangle$ s.t. VALUE_TYPE=int

$x := sum(l)$

$y := sum(l)$

Post: $x + y = 2 \cdot \sum_{i=1\ldots n} a_i.value$

because it is not in general guaranteed that the list l is not modified after the execution of $x := sum(l)$. However, by changing the specification for the `sum` function as follows

Pre: $l = \langle a_1, a_2, \ldots, a_n \rangle$ s.t. VALUE_TYPE=int

$x := sum(l)$

Post: $x = \sum_{i=1\ldots n} a_i.value$ and $\underline{l' = l}$

we can now prove the original claim:

Pre: $l = \langle a_1, a_2, \ldots, a_n \rangle$ s.t. VALUE_TYPE=int

$\{l = \langle a_1, a_2, \ldots, a_n \rangle$ and VALUE_TYPE=int $\}$

$x := sum(l)$

$\{x = \sum_{i=1\ldots n} a_i.value$ and $l = \langle a_1, a_2, \ldots, a_n \rangle$ and VALUE_TYPE=int $\}$

$y := sum(l)$

$\{y = x = \sum_{i=1\ldots n} a_i.value\}$

Post: $x + y = 2 \cdot \sum_{i=1\ldots n} a_i.value$

Example 2

- **find**(i, l) =
  $j := 1$
  $x := l.first$
  while (NOT $l.nil\_entry(x)$) and $j < i$ **do**
  $x := l.next(x)$
  $j := j + 1$
  end while
  return x

- $O(|l|)$

Example 3

Pre: $l = \langle a_1, a_2, \ldots, a_n \rangle$

$res := l.first$

Post: $res = a_1$ if $l \neq \langle\rangle$ and $res =$NIL otherwise ($l = \langle\rangle$), and $l' = l$

Pre: $l = \langle a_1, a_2, \ldots, x, \ldots, a_n \rangle$

$res := l.next(x)$

Post: $res = a_i$ if $l = \langle a_1, a_2, \ldots, x, a_i, \ldots, a_n \rangle$ (i.e. x is not the last entry) and $res =$NIL if $x = a_n$ (i.e. x is the last entry), and $l' = l$