

# Programming Languages: Imperative Program Construction

## Practicals 9: Array Manipulation

Shin-Cheng Mu

Autumn Term, 2021

1. Given  $a : \mathbf{array} [0..10] \text{ of } Int$ , compute  $wp(a[i] := 0) (a[2] \neq 0)$ .

**Solution:**

$$\begin{aligned}
 & wp(a[i] := 0) (a[2] \neq 0) \\
 \equiv & 0 \leq i < 10 \wedge (a[i \mapsto 0])[2] \neq 0 \\
 \equiv & \{ \text{function alteration} \} \\
 & 0 \leq i < 10 \wedge (i = 2 \Rightarrow 0 \neq 0) \wedge (i \neq 2 \Rightarrow a[2] \neq 0) \\
 \equiv & \{ 0 \neq 0 \equiv False \} \\
 & 0 \leq i < 10 \wedge (i = 2 \Rightarrow False) \wedge (i \neq 2 \Rightarrow a[2] \neq 0) \\
 \equiv & \{ P \Rightarrow False \equiv \neg P \} \\
 & 0 \leq i < 10 \wedge i \neq 2 \wedge (i \neq 2 \Rightarrow a[2] \neq 0) \\
 \equiv & \{ \text{proposition logic} \} \\
 & 0 \leq i < 10 \wedge i \neq 2 \wedge a[2] \neq 0 .
 \end{aligned}$$

2. Given constant  $N, Y : Int$  with  $0 \leq N$ , and variables  $b : \mathbf{array} [0..N] \text{ of } Int, x, i : Int$ ,  
 (a) compute  $wp(b[i-1] := x+1) (\forall j : i \leq j < N : b[j] = Y)$ .

**Solution:**

$$\begin{aligned}
 & wp(b[i-1] := x+1) (\forall j : i \leq j < N : b[j] = Y) \\
 \equiv & 0 \leq i-1 < N \wedge (\forall j : i \leq j < N : (b[i-1 \mapsto x+1])[j] = Y) \\
 \equiv & \{ \text{since } i-1 < j, \text{ function alteration} \} \\
 & 1 \leq i \leq N \wedge (\forall j : i \leq j < N : b[j] = Y) .
 \end{aligned}$$

- (b) Compute  $wp(b[i-1] := x+1; i := i-1) (\forall j : i \leq j < N : b[j] = Y)$ .

**Solution:**

$$\begin{aligned}
 & wp(b[i-1] := x+1; i := i-1) (\forall j : i \leq j < N : b[j] = Y) \\
 \equiv & wp(b[i-1] := x+1) (\forall j : i-1 \leq j < N : b[j] = Y) \\
 \equiv & 0 \leq i-1 < N \wedge (\forall j : i-1 \leq j < N : (b[i-1 \mapsto x+1])[j] = Y) \\
 \equiv & \{ \text{split off } j = i-1 \} \\
 & 1 \leq i \leq N \wedge (b[i-1 \mapsto x+1])[i-1] = Y \wedge \\
 & (\forall j : i \leq j < N : (b[i-1 \mapsto x+1])[j] = Y) \\
 \equiv & \{ \text{function alteration} \} \\
 & 1 \leq i \leq N \wedge x+1 = Y \wedge (\forall j : i \leq j < N : b[j] = Y) .
 \end{aligned}$$

### 3. Derive

```

con  $N : \text{Int } \{1 \leq N\}$ 
con  $F : \text{array } [0..N) \text{ of } \text{Int}$ 
var  $h : \text{array } [0..N) \text{ of } \text{Int}$ 
running_sum
 $\{ \langle \forall k : 0 \leq k < N : h[k] = \langle \sum i : 0 \leq i \leq k : F[i] \rangle \rangle \}$  .

```

**Solution:** This problem can be seen as a slightly varied instance of Simple Array Assignment mentioned in the handouts. We could have utilised the results. For practice, however, let's start from the basics.

Let  $P \ n \equiv \langle \forall k : 0 \leq k < n : h[k] = \langle \sum i : 0 \leq i \leq k : F[i] \rangle \rangle$ . Conjecture the following skeleton:

```

con  $N : \text{Int } \{1 \leq N\}$ 
con  $F : \text{array } [0..N) \text{ of } \text{Int}$ 
var  $h : \text{array } [0..N) \text{ of } \text{Int}$ 
var  $n : \text{Int}$ 
initialise
 $\{ P \ 1 \}$ 
 $n := 1$ 
 $\{ P \ n \wedge 1 \leq n \leq N, \text{bnd} : N - n \}$ 
do  $n \neq N \rightarrow \text{step}$ 
        $n := n + 1$ 
od
 $\{ \langle \forall k : 0 \leq k < N : h[k] = \langle \sum i : 0 \leq i \leq k : F[i] \rangle \rangle \}$  .

```

Note that  $1 \leq N$ , and we decided to start the loop with  $n = 1$ . The *initialise* statement thus has to be  $h[0] := F[0]$ . (Proof omitted – do it if it is not yet familiar to you!) The reason we start with  $n = 1$  will be evident later.

We conjecture that *step* can be performed by a single array assignment  $h[l] := E$ . We then have to find  $l$  and  $E$  such that

$$P \ n \wedge 1 \leq n < N \Rightarrow (P \ (n + 1)) [h \setminus (h : l \mapsto E)] \ .$$

Let us inspect  $P \ (n + 1)$ , assuming  $1 \leq n < N$ :

$$\begin{aligned}
& \langle \forall k : 0 \leq k < n + 1 : h[k] = \langle \sum i : 0 \leq i \leq k : F[i] \rangle \rangle \\
& \equiv \{ 1 \leq n < N, \text{split off } k = n \} \\
& \langle \forall k : 0 \leq k < n : h[k] = \langle \sum i : 0 \leq i \leq k : F[i] \rangle \rangle \wedge \\
& h[n] = \langle \sum i : 0 \leq i \leq n : F[i] \rangle \ .
\end{aligned}$$

(One could start with expanding  $(P \ (n + 1)) [h \setminus (h : l \mapsto E)]$  directly. I find it easier to take it slower.)

Now consider  $(P \ (n + 1)) [h \setminus (h : l \mapsto E)]$ , assuming  $P \ n \wedge 1 \leq n < N$

$$\begin{aligned}
& \langle \forall k : 0 \leq k < n : (h : l \mapsto E)[k] = \langle \sum i : 0 \leq i \leq k : F[i] \rangle \rangle \wedge \\
& (h : l \mapsto E)[n] = \langle \sum i : 0 \leq i \leq n : F[i] \rangle \\
& \equiv \{ 1 \leq n < N, \text{split off } i = n. \text{ (*) see the **Think** remark in the end} \} \\
& \langle \forall k : 0 \leq k < n : (h : l \mapsto E)[k] = \langle \sum i : 0 \leq i \leq k : F[i] \rangle \rangle \wedge \\
& (h : l \mapsto E)[n] = \langle \sum i : 0 \leq i \leq n - 1 : F[i] \rangle + F[n] \\
& \equiv \{ P \ n \} \\
& \langle \forall k : 0 \leq k < n : (h : l \mapsto E)[k] = h[k] \rangle \wedge \\
& (h : l \mapsto E)[n] = h[n - 1] + F[n] \\
& \equiv \{ \text{choose } l = n, E = h[n - 1] + F[n] \} \\
& \langle \forall k : 0 \leq k < n : h[k] = h[k] \rangle \wedge \\
& h[n - 1] + F[n] = h[n - 1] + F[n] \\
& \equiv \text{True} \ .
\end{aligned}$$

```

con  $N : \text{Int} \{1 \leq N\}$ 
con  $F : \text{array } [0..N) \text{ of } \text{Int}$ 
var  $h : \text{array } [0..N) \text{ of } \text{Int}$ 
var  $n : \text{Int}$ 
 $h[0] := F[0]$ 
 $\{P\ 1\}$ 
 $n := 1$ 
 $\{P\ n \wedge 1 \leq n \leq N, \text{bnd} : N - n\}$ 
do  $n \neq N \rightarrow h[n] := h[n - 1] + F[n]$ 
     $n := n + 1$ 
od
 $\{\langle \forall k : 0 \leq k < N : h[k] = \langle \sum i : 0 \leq i \leq k : F[i] \rangle \rangle\}$  .

```

In retrospect, we need  $1 \leq n < N$  to guarantee *def*  $(h[n - 1] + F[n])$  (that is, both array accesses are within bound). Therefore we have to start the loop with  $n = 1$ . Fortunately we can do so because  $1 \leq N$ .

**Think:** in the step labelled (\*) above, why could we not do the following instead of the splitting?

$$\begin{aligned}
 & \dots \wedge (h : I \rightarrow E)[n] = \langle \sum i : 0 \leq i \leq n : F[i] \rangle \\
 & = \{P\ n\} \\
 & \dots \wedge (h : I \rightarrow E)[n] = h[n] \ .
 \end{aligned}$$

**Practice:** try solving this problem using Simple Array Assignment.

4. Derive

```

con  $N : \text{Int} \{1 \leq N\}$ 
var  $f : \text{array } [0..N) \text{ of } \text{Int}$ 
con  $H : \text{array } [0..N) \text{ of } \text{Int}$ 
decompose
 $\{\langle \forall k : 0 \leq k < N : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle\}$  .

```

**Solution:** Similar to the previous exercise, we let  $P\ n \equiv \langle \forall k : 0 \leq k < n : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle$ , and conjecture the following skeleton:

```

con  $N : \text{Int} \{1 \leq N\}$ 
con  $H : \text{array } [0..N) \text{ of } \text{Int}$ 
var  $f : \text{array } [0..N) \text{ of } \text{Int}$ 
var  $n : \text{Int}$ 
 $f[0] := H[0]$ 
 $\{P\ 1\}$ 
 $n := 1$ 
 $\{P\ n \wedge 1 \leq n \leq N, \text{bnd} : N - n\}$ 
do  $n \neq N \rightarrow \text{step}$ 
     $n := n + 1$ 
od
 $\{\langle \forall k : 0 \leq k < N : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle\}$  .

```

Conjecture that *step* can be performed by a single array assignment  $f[I] := E$ . We then have to find  $I$  and  $E$  such that

$$P\ n \wedge 1 \leq n < N \Rightarrow (P\ (n+1))[f \setminus (f:l \mapsto E)] \ .$$

Inspect  $P\ (n+1)$ , assuming  $1 \leq n < N$ :

$$\begin{aligned} & \langle \forall k : 0 \leq k < n+1 : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle \\ \equiv & \{ 1 \leq n < N, \text{ split off } k = n \} \\ & \langle \forall k : 0 \leq k < n : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle \wedge \\ & H[n] = \langle \sum i : 0 \leq i \leq n : f[i] \rangle \ . \end{aligned}$$

Now consider  $(P\ (n+1))[f \setminus (f:l \mapsto E)]$ , assuming  $P\ n \wedge 1 \leq n < N$

$$\begin{aligned} & \langle \forall k : 0 \leq k < n : H[k] = \langle \sum i : 0 \leq i \leq k : (f:l \mapsto E)[i] \rangle \rangle \wedge \\ & H[n] = \langle \sum i : 0 \leq i \leq n : (f:l \mapsto E)[i] \rangle \\ \equiv & \{ 1 \leq n < N, \text{ split off } i = n \} \\ & \langle \forall k : 0 \leq k < n : H[k] = \langle \sum i : 0 \leq i \leq k : (f:l \mapsto E)[i] \rangle \rangle \wedge \\ & H[n] = \langle \sum i : 0 \leq i \leq n-1 : (f:l \mapsto E)[i] \rangle + \langle f:l \mapsto E \rangle[n] \\ \equiv & \{ \text{choose } l = n, \text{ see below } (*) \} \\ & \langle \forall k : 0 \leq k < n : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle \wedge \\ & H[n] = \langle \sum i : 0 \leq i \leq n-1 : f[i] \rangle + E \\ \equiv & \{ P\ n \} \\ & \langle \forall k : 0 \leq k < n : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle \wedge \\ & H[n] = H[n-1] + E \\ \equiv & \{ \text{choose } E = H[n] - H[n-1] \} \\ & \langle \forall k : 0 \leq k < n : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle \wedge \\ & H[n] = H[n-1] + (H[n] - H[n-1]) \\ \equiv & \{ P\ n \} \\ & \text{True} \ . \end{aligned}$$

In the step marked (\*), since  $0 \leq i \leq n-1$ , by choosing  $l = n$  both occurrences of  $(f:l \mapsto E)[i]$  reduce to  $f[i]$ . Meanwhile,  $(f:l \mapsto E)[n]$  reduces to  $E$ .

The derived program is

```

con  $N : \text{Int}$   $\{ 1 \leq N \}$ 
con  $H : \text{array } [0..N] \text{ of } \text{Int}$ 
var  $f : \text{array } [0..N] \text{ of } \text{Int}$ 
var  $n : \text{Int}$ 
 $f[0] := H[0]$ 
 $\{ P\ 1 \}$ 
 $n := 1$ 
 $\{ P\ n \wedge 1 \leq n \leq N, \text{ bnd} : N - n \}$ 
do  $n \neq N \rightarrow f[n] := H[n] - H[n-1]$ 
     $n := n + 1$ 
od
 $\{ \langle \forall k : 0 \leq k < N : H[k] = \langle \sum i : 0 \leq i \leq k : f[i] \rangle \rangle \}$  .

```