Assignment 2: Verification

Maxwell Bo 43926871

April 13, 2017

1 Part A

Let

$$pre \triangleq D.len \geqslant max(\{A.len,\ B.len,\ C.len\}) \\ \wedge\ sorted(A)\ \wedge\ sorted(B)\ \wedge\ sorted(C)$$

$$post(r) \triangleq D_{[0,r)} = A \cap B \cap C$$

$$\begin{aligned} & i, j, k, r, D : [pre, post(r)] \\ \sqsubseteq\ & \{\text{Composition: middle predicate is } inv\} \\ & i, j, k, r, D : [pre, inv];\ i, j, k, r, D : [inv, post(r)] \end{aligned}$$

where

$$\begin{aligned} inv \triangleq\ & D_{[0,r)} = A_{[0,i)} \cap B_{[0,j)} \cap C_{[0,k)} \\ & \wedge\ r \in [0, D.len]\ \wedge\ i \in [0, A.len]\ \wedge\ j \in [0, B.len]\ \wedge\ k \in [0, C.len] \end{aligned}$$

$$\begin{aligned} \sqsubseteq\ & \{\text{Assignment: } pre \Rrightarrow inv[i, j, k, r \backslash 0, 0, 0, 0]\} \\ & i, j, k, r := 0, 0, 0, 0;\ i, j, k, r, D : [inv, post(r)] \end{aligned}$$

$\because$

$$\begin{aligned} inv[i, j, k, r \backslash 0, 0, 0, 0] \equiv\ & D_{[0,0)} = A_{[0,0)} \cap B_{[0,0)} \cap C_{[0,0)} \\ & \wedge\ 0 \in [0, D.len]\ \wedge\ 0 \in [0, A.len]\ \wedge\ 0 \in [0, B.len]\ \wedge\ 0 \in [0, C.len] \\ \equiv\ & \varnothing = (\varnothing \cap \varnothing \cap \varnothing)\ \wedge\ (\text{true}\ \wedge\ \text{true}\ \wedge\ \text{true}\ \wedge\ \text{true}) \\ \equiv\ & \varnothing = \varnothing\ \wedge\ \text{true} \\ \equiv\ & \text{true} \end{aligned}$$

$$\begin{aligned} \sqsubseteq\ & \{\text{Strengthen post: } inv \wedge \neg guard \Rrightarrow post(r)\} \\ & i, j, k, r := 0, 0, 0, 0;\ i, j, k, r, D : [inv, inv \wedge \neg guard] \end{aligned}$$

where $guard$ is a function that takes $i,\ j,\ k$ as implicit parameters, s.t.

$$guard(i, j, k) \triangleq (i \neq A.len\ \vee\ j \neq B.len\ \vee\ k \neq C.len)$$

$\therefore$

$$inv \wedge \neg guard \quad \equiv \quad inv \wedge (i = A.len \ \wedge \ j = B.len \ \wedge k = C.len)$$

Assuming $(i = A.len \ \wedge \ j = B.len \ \wedge k = C.len)$ holds, we can show that still

$$inv \wedge (i = A.len \ \wedge \ j = B.len \ \wedge k = C.len) \ \Rrightarrow \ post(r)$$

$\because$

$$\begin{array}{rcl} inv \wedge (i = A.len \ \wedge \ j = B.len \ \wedge k = C.len) & \equiv & inv[i, j, k \backslash A.len, B.len, C.len] \\ & \equiv & D_{[0,r)} = A_{[0,A.len)} \cap B_{[0,B.len)} \cap C_{[0,C.len)} \\ & & \wedge \ \ r \in [0, D.len] \ \ \wedge \ \ A.len \in [0, A.len] \ \ \wedge \ \ B.len \in [0, B.len] \ \ \wedge \ \ C.len \in [0 \\ & \equiv & (D_{[0,r)} = A \cap B \cap C) \ \wedge \ (r \in [0, D.len] \ \ \wedge \ \ \text{true} \ \ \wedge \ \ \text{true} \ \ \wedge \ \ \text{true}) \\ & \equiv & (D_{[0,r)} = A \cap B \cap C) \ \wedge \ (r \in [0, D.len]) \end{array}$$

$$\begin{array}{rcl} (D_{[0,r)} = A \cap B \cap C) \ \wedge \ (r \in [0, D.len]) & \Rrightarrow & post(r) \\ & \Rrightarrow & D_{[0,r)} = A \cap B \cap C \end{array}$$

$\sqsubseteq$ {Repetition}
$\quad i, j, k, r := 0, 0, 0, 0;$
$\quad$ **do** $(i \neq A.len \ \vee \ j \neq B.len \ \vee \ k \neq C.len) \rightarrow$
$\quad\quad i, j, k, r, D : [inv \ \wedge \ guard, \ inv \ \wedge \ (0 \leqslant V < V_0)]$
$\quad$ **od**

where

$$\begin{array}{rcl} V & \triangleq & (A.len - i) + (B.len - j) + (C.len - k) \\ & \triangleq & (A.len + B.len + C.len) - (i + j + k) \end{array}$$

$\not\sqsubseteq$ {Selection: $inv \ \wedge \ guard \not\Rrightarrow (G_1(i,j) \vee G_2(j,k) \vee G_3(k,i) \vee G_4(i,j,k))$}
$\quad i, j, k, r := 0, 0, 0, 0;$
$\quad$ **do** $(i \neq A.len \ \vee \ j \neq B.len \ \vee \ k \neq C.len) \rightarrow$
$\quad\quad$ **if** $(A_i > B_j) \rightarrow i, j, k, r, D : [(A_i > B_j) \wedge inv \wedge guard, \ inv \wedge (0 \leqslant V < V_0)]$
$\quad\quad [\!]$ $(B_j > C_k) \rightarrow i, j, k, r, D : [(B_j > C_k) \wedge inv \wedge guard, \ inv \wedge (0 \leqslant V < V_0)]$
$\quad\quad [\!]$ $(C_k > A_i) \rightarrow i, j, k, r, D : [(C_k > A_i) \wedge inv \wedge guard, \ inv \wedge (0 \leqslant V < V_0)]$
$\quad\quad$ **fi** $(A_i = B_j) \wedge (B_j = C_k) \rightarrow i, j, k, r, D : [(A_i = B_j) \wedge (B_j = C_k) \wedge inv \wedge guard, \ inv \wedge (0 \leqslant V < V_0)]$
$\quad$ **od**

where

$$\begin{array}{rcl} G_1(i,j) & \triangleq & A_i > B_j \\ G_2(j,k) & \triangleq & B_j > C_k \\ G_3(k,i) & \triangleq & C_k > A_i \\ G_4(i,j,k) & \triangleq & (A_i = B_j) \wedge (B_j = C_k) \end{array}$$

$\therefore$

$\quad G_1(i,j) \vee G_2(j,k) \vee G_3(k,i) \vee G_4(i,j,k)$
$\equiv$ {Expansion of the guard definitions}
$\quad (A_i > B_j) \vee (B_j > C_k) \vee (C_k > A_i) \vee ((A_i = B_j) \wedge (B_j = C_k))$
$\equiv$ {Transitivity}
$\quad (A_i > B_j) \vee (B_j > C_k) \vee (C_k > A_i) \vee (A_i = B_j = C_k)$

Actions that have undefined behaviour, such as out-of-bounds array indexing, are inexpressible in the Guarded Query Language. Therefore, for any array-index pairing A_i, there is an implicit constraint that $i \in [0, A.len)$.

Thus

$$\begin{aligned}(A_i > B_j) \vee (B_j > C_k) \vee (C_k > A_i) \vee (A_i = B_j = C_k) \;\; &\equiv \;\; ((A_i > B_j) \vee (B_j > C_k) \vee (C_k > A_i) \vee (A_i = B_j = C_k)) \\ &\qquad \wedge (i \in [0, A.len) \;\wedge\; j \in [0, B.len) \;\wedge\; k \in [0, C.len))\end{aligned}$$

$$\begin{aligned}\mathit{inv} \wedge \mathit{guard} \;\; &\not\Rightarrow \;\; ((A_i > B_j) \vee (B_j > C_k) \vee (C_k > A_i) \vee (A_i = B_j = C_k)) \\ &\qquad \wedge (i \in [0, A.len) \;\wedge\; j \in [0, B.len) \;\wedge\; k \in [0, C.len))\end{aligned}$$

By counter-example, let $r, i, j, k = 0, A.len, 0, 0$.

$$\begin{aligned}\mathit{inv}[i, j, k, r \backslash 0, A.len, 0, 0] \;\; &\equiv \;\; D_{[0,0)} = A_{[0,A.len)} \cap B_{[0,0)} \cap C_{[0,0)} \\ &\qquad \wedge 0 \in [0, D.len] \;\wedge\; A.len \in [0, A.len] \;\wedge\; 0 \in [0, B.len] \;\wedge\; 0 \in [0, C.len] \\ &\equiv \;\; \varnothing = (A \cap \varnothing \cap \varnothing) \;\wedge\; (\text{true} \;\wedge\; \text{true} \;\wedge\; \text{true} \;\wedge\; \text{true}) \\ &\equiv \;\; \varnothing = \varnothing \;\wedge\; \text{true} \\ &\equiv \;\; \text{true}\end{aligned}$$

The specification does not refine to the provided program.