Assignment 2: Verification

Maxwell Bo   43926871

April 13, 2017

1   Part A

Let

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

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

$\quad\ i,j,k,r,D : [pre, post(r)]$

$\sqsubseteq \quad \{\text{Composition: middle predicate is } inv\}$

$\quad\ i,j,k,r,D : [pre, inv];\ i,j,k,r,D : [inv, post(r)]$

where

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

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

$\because$

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

$\sqsubseteq \quad \{\text{Strengthen post: } inv \land \neg guard \Rrightarrow post(r)\}$
$\quad\ i,j,k,r := 0,0,0,0;\ i,j,k,r,D : [inv, inv \land \neg guard]$

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

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

$\therefore$

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

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

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

$\therefore$

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

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

$\sqsubseteq$ {Repetition}

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

where

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

$\not\sqsubseteq$ {Selection: $inv \ \wedge \ guard \nRightarrow (G_1(i,j) \ \vee \ G_2(j,k) \ \vee \ G_3(k,i) \ \vee \ G_4(i,j,k))$}

$\quad i, j, k, r := 0, 0, 0, 0;$
$\quad \textbf{do} \ (i \neq A.len \ \vee \ j \neq B.len \ \vee \ k \neq C.len) \rightarrow$
$\qquad \textbf{if} \ (A_i > B_j) \rightarrow i, j, k, r, D : [(A_i > B_j) \wedge inv \wedge guard, \ inv \wedge (0 \leqslant V < V_0)]$
$\qquad [] \ (B_j > C_k) \rightarrow i, j, k, r, D : [(B_j > C_k) \wedge inv \wedge guard, \ inv \wedge (0 \leqslant V < V_0)]$
$\qquad [] \ (C_k > A_i) \rightarrow i, j, k, r, D : [(C_k > A_i) \wedge inv \wedge guard, \ inv \wedge (0 \leqslant V < V_0)]$
$\qquad \textbf{fi} \ (A_i = B_j) \wedge (B_j = C_k) \rightarrow i, j, k, r, D : [(A_i = B_j) \wedge (B_j = C_k) \wedge inv \wedge guard, \ inv \wedge (0 \leqslant V < V_0)]$
$\quad \textbf{od}$

where

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

$\therefore$

$\quad G_1(i,j) \vee G_2(j,k) \vee G_3(k,i) \vee G_4(i,j,k)$
$\equiv$ {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}
$\quad (A_i > B_j) \vee (B_j > C_k) \vee (C_k > A_i) \vee (A_i = B_j = C_k)$
$\equiv$ {Redefinition}
$\quad G_{union}(i,j,k)$

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

$$\equiv \quad \{\text{Implicit constraints}\}$$
$$((A_i > B_j) \vee (B_j > C_k) \vee (C_k > A_i) \vee (A_i = B_j = C_k)) \ \wedge \ (i \in [0, A.len) \ \wedge \ j \in [0, B.len) \ \wedge \ k \in [0, C.len))$$

With these new implicit constraints, we can conclude that

$$inv \wedge guard \quad \not\Rrightarrow \quad ((A_i > B_j) \vee (B_j > C_k) \vee (C_k > A_i) \vee (A_i = B_j = C_k))$$
$$\wedge \ (i \in [0, A.len) \ \wedge \ j \in [0, B.len) \ \wedge \ k \in [0, C.len))$$

For convenience

$$implicit\_constraints(i, j, k) \quad \triangleq \quad i \in [0, A.len) \ \wedge \ j \in [0, B.len) \ \wedge \ k \in [0, C.len)$$

By counter-example, let $i, j, k, r = A.len, 0, 0, 0$, and choosing $A = \{0\}$, $B = \{1\}$, $C = \{1\}$, s.t. $A.len = 1$, $B.len = 1$ and $C.len = 1$.

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

$$
\begin{aligned}
guard[i, j, k, r \backslash A.len, 0, 0, 0] \quad &\equiv \quad (i \neq A.len \ \vee \ j \neq B.len \ \vee \ k \neq C.len)[i, j, k, r \backslash A.len, 0, 0, 0] \\
&\equiv \quad (A.len \neq A.len \ \vee \ 0 \neq B.len \ \vee \ 0 \neq C.len) \\
&\equiv \quad \text{false} \ \vee \ \text{true} \ \vee \ \text{true} \\
&\equiv \quad \text{true}
\end{aligned}
$$

$$
\begin{aligned}
&\quad (G_{union}(i, j, k) \ \wedge \ implicit\_constraints(i, j, k))[i, j, k, r \backslash A.len, 0, 0, 0] \\
\equiv \quad & \\
&\quad G_{union}(A.len, 0, 0) \ \wedge \ implicit\_constraints(A.len, 0, 0) \\
\equiv \quad & \\
&\quad G_{union}(A.len, 0, 0) \ \wedge \ (A.len \in [0, A.len) \ \wedge \ 0 \in [0, B.len) \ \wedge \ 0 \in [0, C.len)) \\
\equiv \quad & \\
&\quad G_{union}(A.len, 0, 0) \ \wedge \ (\text{false} \ \wedge \ \text{true} \ \wedge \ \text{true}) \\
\equiv \quad & \\
&\quad G_{union}(A.len, 0, 0) \ \wedge \ \text{false} \\
\equiv \quad & \\
&\quad \text{false}
\end{aligned}
$$

$\therefore$

$$
\begin{aligned}
(inv \ \wedge \ guard)[i, j, k, r \backslash A.len, 0, 0, 0] \quad &\not\Rrightarrow \quad (G_{union}(i, j, k) \\
&\qquad \wedge \ implicit\_constraints(i, j, k))[i, j, k, r \backslash A.len, 0, 0, 0] \\
inv[i, j, k, r \backslash A.len, 0, 0, 0] \ \wedge \ guard[i, j, k, r \backslash A.len, 0, 0, 0] \quad &\not\Rrightarrow \quad \text{false} \\
\text{true} \ \wedge \ \text{true} \quad &\not\Rrightarrow \quad \text{false} \\
\text{true} \quad &\not\Rrightarrow \quad \text{false}
\end{aligned}
$$

The specification does not refine to the provided program.

Instead of $(i \neq A.len \ \vee \ j \neq B.len \ \vee \ k \neq C.len)$ as the **do** guard, the choice of $(i \neq A.len \ \wedge \ j \neq B.len \ \wedge \ k \neq C.len)$ prevents out-of-bounds array indexing in the guards of the **if** statement, and could be refined to by the specification.

We can quickly show that this new guard, $guard'$, doesn't fall to the counter-example we used before.

$$
\begin{aligned}
guard'[i,j,k,r \backslash A.len, 0, 0, 0] \quad &\equiv \quad (i \neq A.len \,\wedge\, j \neq B.len \,\wedge\, k \neq C.len)[i,j,k,r \backslash A.len, 0, 0, 0] \\
&\equiv \quad (A.len \neq A.len \,\wedge\, 0 \neq B.len \,\wedge\, 0 \neq C.len) \\
&\equiv \quad \text{false} \,\wedge\, \text{true} \,\wedge\, \text{true} \\
&\equiv \quad \text{false}
\end{aligned}
$$

s.t.

$$
\begin{aligned}
(inv \,\wedge\, guard')[i,j,k,r \backslash A.len, 0, 0, 0] \quad &\Rightarrow \quad (G_{union}(i,j,k) \\
&\qquad \wedge\, implicit\_constraints(i,j,k))[i,j,k,r \backslash A.len, 0, 0, 0] \\
inv[i,j,k,r \backslash A.len, 0, 0, 0] \,\wedge\, guard'[i,j,k,r \backslash A.len, 0, 0, 0] \quad &\Rightarrow \quad \text{false} \\
\text{true} \,\wedge\, \text{false} \quad &\Rightarrow \quad \text{false} \\
\text{false} \quad &\Rightarrow \quad \text{false}
\end{aligned}
$$