8 Hoare Logic and Model Checking (jp622)

Consider commands $C$ composed from assignments $X := E$ (where $X$ is a program variable, and $E$ is an arithmetic expression), heap allocation $X := \text{alloc}(E_1, ..., E_n)$, heap assignment $[E_1] := E_2$, heap dereference $X := [E]$, disposal of heap locations $\text{dispose}(E)$, the no-op $\text{skip}$, sequencing $C_1; C_2$, conditionals $\text{if } B \text{ then } C_1 \text{ else } C_2$ (where $B$ is a boolean expression), and loops $\text{while } B \text{ do } C$. $null$ is $0$.

Recall the separation logic partial list representation predicates:

\[
\begin{align*}
\text{plist}(t, [], u) &= (t = u) \land \text{emp} \\
\text{plist}(t, h :: \alpha, u) &= \exists y. ((t \mapsto h) \ast ((t + 1) \mapsto y) \ast \text{plist}(y, \alpha, u))
\end{align*}
\]

Circular lists can be represented by $\text{clist}(t, \alpha, t)$ where $\alpha$ is a boolean expression), heap allocation $E \ast \alpha$, heap dereference $X \ast \alpha$, and loops $\text{while } B \text{ do } C$.

(a) Assuming $\vdash \{P_1\} C_1 \{Q_1\}$ and $\vdash \{P_2\} C_2 \{Q_2\}$:

(i) explain precisely why $\vdash \{P_1 \ast P_2\} C_1; C_2 \{Q_1 \ast Q_2\}$ [2 marks]

(ii) give a counterexample to $\vdash \{P_1 \land P_2\} C_1; C_2 \{Q_1 \land Q_2\}$. [1 mark]

(b) Give a proof outline for the following circular list ‘next’ triple:

\[
\{\text{clist}(X, t :: \alpha)\} X := [X + 1] \{\text{clist}(X, \alpha + [t])\}
\]

[3 marks]

(c) Give a loop invariant (no need for a proof outline) for the following circular list ‘length’ triple:

\[
\{\text{clist}(X, \alpha)\}
\]

if $X = \text{null}$ then $Y := 0$

else $(Z := [X + 1]; Y := 1; \text{while } Z \neq X \text{ do } (Z := [Z + 1]; Y := Y + 1))$

\[
\{\text{clist}(X, \alpha) \ast Y = \text{length}(\alpha)\}
\]

[3 marks]

(d) Give a loop invariant (no need for a proof outline) for the following triple for a ‘previous’ operation on non-empty circular lists:

\[
\{\text{clist}(X, \alpha + [t])\}
\]

$Z := X; Y := [X + 1]; (\text{while } Y \neq X \text{ do } (Z := Y; Y := [Y + 1])); X := Z$

\[
\{\text{clist}(X, t :: \alpha)\}
\]

[4 marks]

(e) Give a loop invariant (no need for a proof outline) for the following triple for a ‘dial to minimum’ operation on non-empty circular lists:

\[
\{\text{clist}(X, \alpha_1 + [t] + \alpha_2) \land \text{sorted}(t :: \text{merge(sort(\alpha_1), sort(\alpha_2)))}\}
\]

$Z := X; M := [X]; Y := [X + 1];$

(while $Y \neq Z$ do

$N := [Y]; (\text{if } N < M \text{ then } X := Y \text{ else skip}); Y := [Y + 1]);$

\[
\{\text{clist}(X, [t] + \alpha_2 + \text{reverse(\alpha_1)})\}
\]

[5 marks]

(f) Describe precisely all pairs of a stack and a heap that satisfy

\[
\exists y, z. ((X \mapsto y \ast y \mapsto z \ast z \mapsto X) \land Y = 0)
\]

[2 marks]