6 Hoare Logic and Model Checking (cp526)

Consider a programming language with commands $C$ consisting of the skip no-op command, sequential composition $C_1; C_2$, loops while $B$ do $C$ for boolean expressions $B$, conditionals if $B$ then $C_1$ else $C_2$, assignment $X := E$ for program variables $X$ and arithmetic expressions $E$, heap allocation $X := \text{alloc}(E_1, \ldots, E_n)$, heap assignment $[E_1] := E_2$, heap dereference $X := [E]$, and heap location disposal $\text{dispose}(E)$. Assume $\text{null} = 0$, and predicates for lists and partial lists:

\begin{align*}
\text{list}(t, []) &= (t = \text{null}) \land \text{emp} \\
\text{list}(t, h :: \alpha) &= \exists y. (t \mapsto h) \ast ((t + 1) \mapsto y) \ast \text{list}(y, \alpha) \\
\text{plist}(t_1, [], t_2) &= (t_1 = t_2) \land \text{emp} \\
\text{plist}(t_1, h :: \alpha, t_2) &= \exists y. (t_1 \mapsto h) \ast ((t_1 + 1) \mapsto y) \ast \text{plist}(y, \alpha, t_2)
\end{align*}

In the following, all triples are linear separation logic triples.

(a) Find a command $C$ satisfying the following separation logic partial correctness triple: $\{\top\} C \{X \mapsto 0 \ast X \mapsto 0\}$. [2 marks]

(b) Give a loop invariant that would serve to prove the following triple, where ‘map negate $\alpha$’ is the list of negated values in $\alpha$ (no proof outline required):

\begin{align*}
\{\text{list}(X, \alpha)\} \\
Y &= X; \text{while } Y \neq \text{null} \text{ do } (V := [Y]; [Y] = V \ast (-1); Y = [Y + 1]) \\
\{\text{list}(X, \text{map negate } \alpha)\}
\end{align*}

[4 marks]

(c) Give a loop invariant that would serve to prove the following triple, for a program that finds the last element of a list (no proof outline required):

\begin{align*}
\{\text{list}(X, \alpha \text{ ++ } [l])\} \\
\text{CUR} &= X; \text{NEXT} = [X + 1]; \\
\text{while } \text{NEXT} \neq \text{null} \text{ do } (\text{CUR} = \text{NEXT}; \text{NEXT} = [\text{NEXT} + 1]); \\
\text{LAST} &= [\text{CUR}] \\
\{\text{list}(X, \alpha \text{ ++ } [l]) \land \text{LAST} = l\}
\end{align*}

[5 marks]

(d) Explain why a proof of Part (c) would not succeed if the post-condition of the triple was replaced with $\{\text{emp} \land \text{LAST} = l\}$. [3 marks]

(e) Give a loop invariant that would serve to prove the following triple, for a program that copies a given list (no proof outline required):

\begin{align*}
\{\text{list}(X, \alpha) \land \alpha \neq []\} \\
V &= [X]; Y := \text{alloc}(V, \text{null}); \text{CUR} := [X + 1]; \text{OLD} = Y; \\
\text{while } \text{CUR} \neq \text{null} \text{ do } ( \\
&\quad V = [\text{CUR}]; N = \text{alloc}(V, \text{null}); [\text{OLD} + 1] = N; \\
&\quad \text{CUR} = [\text{CUR} + 1]; \text{OLD} = N) \\
\{\text{list}(X, \alpha) \ast \text{list}(Y, \alpha)\}
\end{align*}

[6 marks]