7 Hoare Logic and Model Checking (jp622)

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

(*a*) Explain informally what it means for a separation logic partial correctness triple $\{P\}\ C\ \{Q\}$ to be valid. [3 marks]

(*b*) Explain informally what it means in terms of the executions of C for the separation logic partial correctness triple $\{\top\}\ C\ \{\bot\}$ to be valid. [2 marks]

(*c*) Recall the list representation predicate *list*:

$$list(t, []) = (t = \texttt{null}) \quad list(t, h :: \alpha) = \exists y.\, ((t \mapsto h) * ((t+1) \mapsto y) * list(y, \alpha))$$

We write $[]$ for the empty mathematical list; $h :: \alpha$ for the mathematical list the head of which is h, and the tail of which is α; $\alpha +\!\!+\ \beta$ for the concatenation of mathematical lists α and β; $\alpha[i]$ for the i-th element of the list α, starting at 0; and $[k, \ldots, n]$ for the ascending list of integers from k to n, including k and n. Give a proof outline, including a loop invariant, for the following triple:

$\{N = n \wedge N \geq 0\}$
$X := \texttt{null}; \texttt{while}\ N > 0\ \texttt{do}\ (X := \texttt{alloc}(N, X); N := N - 1)$
$\{list(X, [1, \ldots, n])\}$ [4 marks]

(*d*) Also recall the partial list representation predicate *plist*:

$$plist(t, [], u) = (t = u)$$
$$plist(t, h :: \alpha, u) = \exists y.\, ((t \mapsto h) * ((t+1) \mapsto y) * plist(y, \alpha, u))$$

Give a loop invariant for the following list sum triple:

$\{list(X, \alpha)\}$
$Y := X; N := 0; \texttt{while}\ Y \neq \texttt{null}\ \texttt{do}\ (M := [Y]; N := N + M; Y := [Y + 1])$
$\left\{list(X, \alpha) \wedge N = \sum_{i=0}^{length(\alpha)-1} \alpha[i]\right\}$ [4 marks]

(*e*) Give a loop invariant for the following list concatenation triple:

$\{list(X, \alpha) * list(Y, \beta)\}$
$\texttt{if}\ X = \texttt{null}\ \texttt{then}\ Z := Y\ \texttt{else}$
$$\left(\begin{array}{l} Z := X; U := Z; V := [Z+1]; \\ \texttt{while}\ V \neq \texttt{null}\ \texttt{do}\ (U := V; V := [V+1]); \\ [U+1] := Y \end{array}\right)$$
$\{list(Z, \alpha +\!\!+\ \beta)\}$ [5 marks]

(*f*) Describe precisely a stack and a heap that satisfy $list(X, [1, \ldots, 3])$. [2 marks]