COMPUTER SCIENCE TRIPOS Part II – 2021 – Paper 8

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 dereference $X := [E]$, heap assignment $[E_1] := E_2$, the no-op `skip`, sequencing $C_1; C_2$, conditionals `if` B `then` C_1 `else` C_2 (where B is a boolean expression), and loops `while` B `do` C. `null` is 0. We write $\texttt{align}(t, s)$ for the smallest multiple of s larger than t.
Let $\mathsf{block}(t, 0) = \mathsf{emp}, \quad \mathsf{block}(t, n+1) = (\exists t'.\, t \mapsto t') * \mathsf{block}(t+1, n)$.

(*a*) Explain why the following postcondition for an allocator that returns aligned blocks is incorrect, and propose a fix.

$$\{\mathsf{block}(B, E-B) * 1 \le S\}$$
$$\texttt{if align}(B, 2^S) + 2^S < E$$
$$\texttt{then } (R := \texttt{align}(B, 2^S); B := B + 2^S) \texttt{ else } R := 0$$
$$\left\{ \begin{array}{l} \mathsf{block}(B, E-B) * \\ (R \neq 0 \implies (\mathsf{block}(R, 2^S) * R = \texttt{align}(R, S))) \end{array} \right\}$$

[3 marks]

(*b*) With this specification, allocations cannot be chained, as in $C_{\text{alloc}}; Y := X; C_{\text{alloc}}$. Explain why, and propose a fix. [2 marks]

(*c*) Strengthen the precondition just enough to guarantee the success of allocation (so that $R \neq 0 \implies$ is not needed anymore). [2 marks]

(*d*) Consider the following representation predicate for lists of free blocks of size 2^S:

$$\texttt{freelist}(t, S) = (t = \texttt{null} * \texttt{emp}) \vee \left(\exists t'. \left(\begin{array}{l} t = \texttt{align}(t, 2^S) \wedge \\ \left(\begin{array}{l} t \mapsto t' * \\ \texttt{block}(t+1, 2^S - 1) * \\ \texttt{freelist}(t', S) \end{array} \right) \end{array} \right) \right)$$

Give a loop invariant, and precisely but informally explain why it is preserved, for this "add the contents of a block into a free list" triple:

$$\{B = \texttt{align}(B, 2^S) * \mathsf{block}(B, E-B) * 1 \le S * L = \texttt{null}\}$$
$$\texttt{while } B + 2^S < E \texttt{ do}$$
$$\quad \big([B] := L; L := B; B := B + 2^S\big)$$
$$\{\mathsf{block}(B, E-B) * \texttt{freelist}(L, S)\}$$

[7 marks]

(*e*) Give a loop invariant, and precisely but informally explain why it is preserved, for this "coalesce blocks of a size S free list into a size $S+1$ free list" triple:

$$\{\texttt{freelist}(L1, S) * L2 = \texttt{null} * D = 0\}$$
$$\texttt{while } D = 0 \texttt{ do} \left(\begin{array}{l} \texttt{if } L1 = \texttt{null then } D := 1 \\ \texttt{else} \left(\begin{array}{l} X := [L1]; \\ \texttt{if } X = \texttt{null or } X \texttt{ mod } 2^{S+1} \neq 0 \texttt{ then } D := 1 \\ \texttt{else} \left(\begin{array}{l} Y := [X]; \\ \texttt{if } X + 2^S \neq Y \texttt{ then } D := 1 \\ \texttt{else } \big([X] := L2; L2 := X\big) \end{array} \right) \end{array} \right) \end{array} \right)$$
$$\{\texttt{freelist}(L1, S) * \texttt{freelist}(L2, S+1)\}$$

[6 marks]