Solution notes

Specification and Verification I 2005 – Paper 8 Question 13 (MJCG)

(*b*) What is partial about partial correctness? [2 marks]

Partial correctness only considers properties of the form "if a program terminates then $\cdots$". It is partial because it does not specify termination; it deals with safety but not liveness.

(*b*) What is the difference between a variant and an invariant? [2 marks]

A variant is an expressions whose value strictly decreases on each iteration of a loop. An invariant is a statement that remains true after each iteration. Thus a variant changes value, but an invariant doesn't change its truth-value. Invariants are for partial correctness and variants are to prove termination.

(*c*) Why are annotations needed for mechanising program verification? [2 marks]

Annotations are needed because there is no algorithm that will generate invariants or variants (program correctness is undecidable, at least when arithmetic is present).

(*d*) What additional annotations are needed for total correctness? [2 marks]

For total correctness one must provide a variant as an additional annotation beyond those needed for partial correctness.

(*e*) How do refinement and *post hoc* verification differ? [2 marks]

Refinement is a correct-by-construction method of writing code; with post hoc verification one first writes code, then proves it correct.

(*f*) Give an example of a higher-order formula that is not first-order. [2 marks]

$\forall e_0\ g.\ \exists f.\ f(0) = e_0\ \wedge f(n+1) = g(f(n))$

(*g*) Why is higher-order logic typed? [2 marks]

Higher order logic is typed to avoid Russell's Paradox:

$(\lambda P.\ \neg(P(P)))(\lambda P.\ \neg(P(P)))\ =\ \neg((\lambda P.\ \neg(P(P)))(\lambda P.\ \neg(P(P))))$

(*h*) How are $\{P\}C\{Q\}$ and $\texttt{wlp}(C, Q)$ related? [2 marks]

$\{P\}C\{Q\}\ =\ P \Rightarrow \textit{wlp}(C, Q)$

(*i*) How can $[c]q$ and $<c>q$ be defined in higher-order logic? [2 marks]

$[c]q\ =\ \lambda s.\ \forall s'.\ c(s, s') \Rightarrow q(s')$ *and* $<c>q\ =\ \neg([c](\neg q))$

(*j*) Explain the difference between soundness and completeness? [2 marks]

A deductive sustem is sound if everything that can be deduced is true; it is complete if everything that is true can be deduced.

Context

This question covers the whole course.

Marking Scheme

For each section:

2 marks : complete and correct answer;

1 marks : partial answer lacking some key material or minor inaccuracies;