Tail invariants

Maarten Fokkinga

Version of June 23, 1994

Tail invariants are weaker than normal invariants, and thus allow for more program manipulations. This is shown formally in a very general setting.

Head and tail invariant. Without loss of generality we way assume that a repetitive construct has the following form:

$$x := x_0;\ \textbf{do}\ B \to x := fx\ \textbf{od} \quad .$$

Here, x may be a collection of variables, possibly including a counter, and f may be correspondingly complex. Let N be the number of iterations, and let x_N be the final value of x. Then, the *head invariant* (my naming!) and the *tail invariant* (standard naming) are, by definition, the following two:

$$\begin{array}{lll} H: & \exists i :: & x = f^i\, x_0 \\ T: & \exists i :: & f^{N-i} x = x_N \quad . \end{array}$$

Here i ranges over $0, .., N$. It is right to say that the head invariant expresses *all* "what has been computed so far since the initial state of the repetition", and that the tail invariant expresses *only* "what has to be computed in the future up to the final state of the repetition". But notice that these interpretations only make sense if the repetition is known; during program construction, when an invariant is derived from a given post-condition and the repetition doesn't yet exist, the previous interpretation doesn't make sense.

Correctness proofs. It might be interesting to compare the two correctness proofs that use invariants H and T, respectively. These proofs will use the following definitions:

$$\begin{array}{lcll} f^0 & = & id & \text{(basis)} \\ f \circ f^i & = & f^{i+1} \quad . & \text{(step)} \end{array}$$

Following Dijkstra (EWD1159) we observe:

- using invariant H, property (basis) is used in the initialisation, and (step) is used in the step, and only pure predicate logic is used in the finalisation;

- using invariant T, only pure predicate logic is used in the initialisation, (step) is used in the step, and (basis) is used in the finalisation.

We are not interested in the use of the definitions "$x_N = f^N\, x_0$" and "$N =$ the smallest number n such that $f^n\, x_0$ satisfies B", since these can be eliminated by substituting them into the invariants.

Advantage of T over H. Clearly, the tail invariant is weaker than the head invariant: $H \Rightarrow T$. Therefore it is likely that more properties of f can be exploited when it is only T that is to be kept invariant. We stress that 'likely' doesn't mean 'certainly'. Let us try to be precise.

Suppose we add another alternative to the repetition:

$$x := x_0;\ \textbf{do}\ B \rightarrow x := fx\ [\!]\ C \rightarrow x := g\, x\ \textbf{od} \quad .$$

We'll investigate invariance of the H and T defined above. Using the semantics of assignment, we find that the new alternative maintains H iff property $(invH)$ below holds true, and similarly it maintains T iff $(invT)$ holds:

$$\begin{array}{lcll} C \wedge H[x := gx] & \Rightarrow & H & (invH) \\ C \wedge T[x := gx] & \Rightarrow & T \quad . & (invT) \end{array}$$

Since $H \Rightarrow T$ (for all x), we have:

$$\begin{array}{lll} (invH) \Rightarrow (invT) & \text{follows from} & C \wedge T[x := gx] \Rightarrow H[x := gx] \qquad (0) \\ (invT) \Rightarrow (invH) & \text{follows from} & T \Rightarrow H \quad . \qquad (1) \end{array}$$

Neither of the two right-hand sides is true in general. But the former one is weaker than the latter, meaning that it is more likely that maintenance of H by the new alternative also implies maintenance of T, rather than the other way around. So, tail invariants are to be preferred, in general; they allow for more additional alternatives.

We can be slightly more precise. Let us derive sufficient conditions on f that guarantee the truth of the left-hand side of (0) and (1), respectively. For (1) we calculate:

$$\begin{array}{ll} & (invT) \Rightarrow (invH) \\ \Leftarrow & \quad \text{observed above in (1)} \\ & T \Rightarrow H \\ \equiv & \quad \text{definition } T \text{ and } H \\ & (\exists i :: f^{N-i}\, x = f^N\, x_0) \quad \Rightarrow \quad (\exists i :: x = f^i\, x_0) \\ \Leftarrow & \quad \text{predicate logic} \\ & \forall i :: \quad f^{N-i}\, x = f^N\, x_0 \ \Rightarrow\ x = f^i\, x_0 \\ \Leftarrow & \quad \text{function composition} \end{array}$$

$$
\begin{array}{ll}
 & \forall i :: \quad f^{N-i}\, x = f^{N-i}\,(f^i\, x_0) \;\Rightarrow\; x = f^i\, x_0 \\
\Leftarrow & \quad \text{definition injectivity} \\
 & \forall i :: \quad f^{N-i} \text{ is injective} \\
\Leftarrow & \quad \text{Leibniz} \\
 & f \text{ is injective} \quad .
\end{array}
$$

On the other hand, for (0) we get a sufficient condition that is slightly weaker, as expected:

$$
\begin{array}{ll}
 & (invH) \Rightarrow (invT) \\
\Leftarrow & \quad \text{observed above in (0)} \\
 & C \wedge T[x := gx] \Rightarrow H[x := gx] \\
\Leftarrow & \quad \text{proposition logic} \\
 & T[x := gx] \Rightarrow H[x := gx] \\
\equiv & \quad \text{definition } T \text{ and } H \\
 & (\exists i :: f^{N-i}\, gx = f^N\, x_0) \quad \Rightarrow \quad (\exists i :: gx = f^i\, x_0) \\
\Leftarrow & \quad \text{as above, writing } gx \text{ for } x \\
 & \forall i :: \quad f^{N-i} \text{ is injective on } \mathit{Range}(g) \cup \{i ::\ f^i\, x_0\} \\
\Leftarrow & \quad \text{Leibniz} \\
 & f \text{ is injective on } \mathit{Range}(g) \cup \{i ::\ f^i\, x_0\} \quad .
\end{array}
$$

Although this latter condition is weaker than the former, the two conditions resemble each other very much. The reader should keep in mind, however, that the conditions are only sufficient conditions, not necessary ones. In practical program construction, the situation may be much more specific. The discussion by Dijkstra in EWD1159 gives a nice example.

Remark. (*For specialists only.*) In the relational framework one works with relations, so that both programs and pre- and postconditions are relations, and the statement $\{P\}\ S\ \{Q\}$ translates to $P'; S \subseteq Q'$.

Now consider an unguarded repetition $S^* = I \cup (S; S^*)$, with specification $P; S^* \subseteq Q$. Define the head and tail invariant to be:

$$
\begin{array}{lcl}
H & = & P; S^* \\
T & = & Q/S^* \quad .
\end{array}
$$

In the relational calculus (including the definition: $X \subseteq R/S \;\equiv\; X; S \subseteq R$), it is now immediate that:

$$
\begin{array}{lll}
P \subseteq H & P \subseteq T & \text{precondition implies invariants} \\
H; S \subseteq H & T; S \subseteq T & \text{invariance of invariants} \\
H \subseteq Q & T \subseteq Q & \text{invariants imply postcondition} \quad ,
\end{array}
$$

and, moreover, H is the strongest invariant with respect to P, and T the weakest one with respect to Q:

$$\begin{array}{lcll} P \subseteq X & \wedge & X;S \subseteq X & \Rightarrow \quad H \subseteq X \\ X \subseteq Q & \wedge & X;S \subseteq X & \Rightarrow \quad X \subseteq T \quad . \end{array}$$

In particular it follows that $H \subseteq T$.

Similarly for a guarded repetition $(B;S)^*;\neg B$.