The Use of Formal Methods for Safety-Critical Systems
Ph.D. Thesis, 1997

Paul Trafford

Addenda et Corrigenda

This document provides a proof that the unified tester is a tester for the reduction preorder in the given context. It is to be read in conjunction with the Ph.D. thesis. You may send enquiries by email (to: pt@easynet.co.uk)

The proof requires first of all a modification in Lemma 5.6.6:

LEMMA 5.6.6 Let $S, I \in Beh_{Proc}$. Suppose $\sigma \in Tr(I||T(S))$. Then $\forall I' \in Beh_{Proc}, \forall T' \in Beh_{Proc}$:

$$T' \neq \texttt{stop}.\ ((I||T(S) \overset{\sigma}{\Rightarrow} I'||T') \wedge out(T') \cap \mathcal{F}_{diag} \subseteq \{success\}) \Rightarrow \Gamma(\sigma) \overset{\varepsilon}{\Rightarrow} T'$$

Proof The proof is as given up to the instantiation in the algorithm. Then it should continue as:

"We now consider all the ways $I||T(S)$ reaches $I'||T'$ after σ' to deduce the result. This amounts to showing that extending $I_k||T_k$ by $<x>$ always gives the desired result. Through the definition of ||, I cannot do any action unilaterally, so T_k must perform x.

Suppose the tester performs the action from the first summand. Then after σ the tester can perform `fail`. This contradicts the hypothesis.

Suppose the tester performs `success` from the fourth term. Then $((I||T(S) \overset{\sigma}{\Rightarrow} I'||T')$ such that $T' = \texttt{stop}$ which contradicts the hypothesis.

The two middle terms (2nd and 3rd summands) remain to be considered. Clearly, when T_k performs x from $\Gamma(\sigma_k)$ to get T' it will satisfy either $\Gamma(\sigma)=T'$ or $\Gamma(\sigma) \overset{\tau}{\rightarrow} T'$ □

PROPOSITION 5.6.7

Let S be a finite specification and I be a non-divergent specification. Then $I \leq_{red} S$ if and only if I *must* $T(S)$.

Proof

(=>) Since I is non-divergent and $T(S)$ is finite, then owing to the definition of ||, all computations $Comp(I,T(S))$ are finite, so $I||T(S)$ eventually reaches stop. Thus it suffices to show that every termination must be a successful computation.

First we note that by construction, a successful computation must terminate in the fourth term having performed just on 'success' action.

Suppose $I||T(S) \overset{\sigma}{\Rightarrow} I'||T' : I'||T' \not\rightarrow$ Two cases arise:

1. $Tr(I||T(S)) \in L^*$

We have `fail` $\notin out(T')$ since otherwise $I'||T' \xrightarrow{fail}$. So by Lemma 5.6.6,

$\Gamma(\sigma) \overset{\varepsilon}{\Rightarrow} T'$, where there are two subcases for T':

(i) $\Gamma(\sigma)=T'$. Hence, by Lemma 5.6.4, $out(T')=L$. Thus for deadlock to occur we require $out(I')=\emptyset$. This implies $R_I(\sigma)=\mathcal{P}(L)$ and hence $Å(\sigma)=\emptyset$. Therefore T', and have $I'||T'$, can perform `success`. This is a contradiction.

(ii) $\Gamma(\sigma)\neq T'$. From the definition of the algorithm, we deduce $\Gamma(\sigma) \xrightarrow{\tau} T'$ where for some $A \in Å_S(\sigma)$, we have $T' = \sum_{a \in A} a;\Gamma(\sigma \wedge < a >)$. But from the proposition hypothesis we deduce immediately from Lemma 5.5.1 and Lemma 5.6.2 that $Å_I(\sigma) \supseteq Å_S(\sigma)$. Therefore $I' \xrightarrow{a}$ which is a contradiction.

Therefore case (1.) is not possible.

2. $Tr(I||T(S)) \notin L^*$

By construction this can only occur when a single flag action has occurred just before a stop action. There are just two choices - a `fail` from the first summand of some Γ expression or a `success` from the last term.

Suppose that the termination is from the first summand. Then $\sigma=\sigma' \wedge <b;$`fail`$>$ where $\sigma' \in Tr(I)$ (since within $||$, I and T participate in every action before `fail`). Now since `fail`$\notin out(T')$, from Lemma 5.6.6, we have: $\Gamma(\sigma') \overset{\varepsilon}{\Rightarrow} T'$.

Therefore the first term is specifically of $\Gamma(\sigma')$. Therefore $b \notin out_{\sigma'}(S)$ which implies $b \in out_{\sigma'}(I)$ since otherwise $I||T$ always deadlocks after σ'. Hence $\sigma' \wedge <b> \in Tr(I)$. But $\sigma' \wedge <b> \notin Tr(S)$ and so the proposition hypothesis is contradicted.

Thus only a successful computation is possible. □

(<=) It suffices to show the contrapositive, i.e.

$\exists \sigma \in L^*. R_I(\sigma) \not\subseteq R_S(\sigma) \Rightarrow v(I||T(S)) \neq pass$

Proof Suppose that the LHS holds. Then $\exists R \in R_I(\sigma): R \notin R_S(\sigma)$. Let $A=R$. Then from the definition of acceptance sets it follows that $A \in A_S(\sigma): A \notin A_I(\sigma)$. This implies $\exists A' \in \mathring{A}_S(\sigma): A' \notin \mathring{A}_I(\sigma)$ (consequence of Lemma 5.6.2). Therefore, by Lemma 5.5.2, $\exists I' \in Beh_{Proc}$:

$I||T(S) \stackrel{\sigma}{\Rightarrow} I'||T': \forall a \in A: I' \stackrel{a}{\not\rightarrow}$. This gives rise to deadlock in the third term, i.e. to a failed computation. □