Appendix

A Uniqueness of the Initial State

It is inexpensive to enforce a unique initial thread state without affecting thread state reachability, provided the initial thread state set T of the given TTD $\mathcal{P}$ satisfies the following "box" property:

$$\forall (s,t) \in T,\ (s',t') \in T:\ (s,t') \in T,\ (s',t) \in T\ . \tag{5}$$

This holds if T is a singleton. More generally, it holds if all states in T have the same shared state, and it holds if all states in T have the same local state. It also holds of a set T whose elements form a complete rectangle in the graphical representation of $\mathcal{P}$.

To enforce a unique initial thread state, we build a new TTD $\mathcal{P}'$ that is identical to $\mathcal{P}$, except that it has a single initial thread state $t_I = (s_I, l_I)$ with fresh shared and local states s_I, l_I, and the following additional edges:

$$(s_I, l_I) \to (s,l) \quad \text{such that } (s,l) \in T\ , \quad \text{and} \tag{6}$$

$$(s, l_I) \to (s,l) \quad \text{such that } (s,l) \in T\ . \tag{7}$$

Suppose now some thread state $t_0 = (s_0, l_0)$ is reachable in $\mathcal{P}_n$, for some n. Then there exists a path from some global state $(s_J|l_1,\ldots,l_n)$ such that $(s_J,l_i) \in T$ for all i, to a global state with shared component s_0 and some thread in local state l_0. We can attach, to the front of this path, the prefix

$$\begin{array}{rcl} (s_I|l_I,\ldots,l_I) & \rightarrowtail & (\underline{s_J}|\underline{l_1}, l_I, l_I, \ldots, l_I) \\ & \rightarrowtail & (s_J|l_1, \underline{l_2}, l_I, \ldots, l_I) \\ & & \cdots \\ & \rightarrowtail & (s_J|l_1, l_2, l_3, \ldots, \underline{l_n})\ , \end{array}$$

with the underlined symbols changed. The new path reaches t_0 in $\mathcal{P}'_n$.

Conversely, suppose some thread state $t_0 = (s_0, l_0)$ such that $s_0 \neq s_I$, $l_0 \neq l_I$ is reachable in $\mathcal{P}'_n$, for some n. Then there exists a path p' from $\{s_I\} \times \{l_I\}^n$ to a global state with shared component s_0 and some thread in local state l_0. The very first transition of p' is by some thread executing an edge of type (6), since those are the only edges leaving the unique initial state (s_I, l_I). Let that be thread number i, and let $(s,l) \in T$ be the new state of thread i.

Consider now an arbitrary thread $j \in \{1,\ldots,n\} \setminus \{i\}$; its local state after the first transition along p' is l_I.

- If thread j is never executed along p', we build a new path p'' by inserting edge $(s,l_I) \to (s,l)$, executed by thread j, right after the first transition in p'. This is a valid edge (of type (7)) since $(s,l) \in T$. The edge moves thread j into an initial thread state $(s,l) \in T$. The modified state sequence remains a valid path in $\mathcal{P}'_n$ since no shared states have been changed, and thread j is inactive henceforth.

- If thread j is executed along p', then the first edge it executes must be of type (7), since again this is the only way to get out of local state l_I. Let $(\bar{s},\bar{l}) \in T$ be the state of thread j after executing this first edge. Then $(s,l_I) \to (s,\bar{l})$ is a valid edge (of type (7)): from $(s,l) \in T$ and $(\bar{s},\bar{l}) \in T$, we conclude $(s,\bar{l}) \in T$, by property (5). We now build a new path p'', by removing from p' thread j's first transition, and instead inserting, right behind the first transition of p', a transition where thread j executes edge $(s,l_I) \to (s,\bar{l})$:

$$p' :: (s_I,l_I) \stackrel{i}{\to} (s,t)\ , \quad \ldots \quad , (\bar{s},l_I) \stackrel{j}{\to} (\bar{s},\bar{l})$$

becomes

$$p'' :: (s_I,l_I) \stackrel{i}{\to} (s,t)\ , (s,l_I) \stackrel{j}{\to} (s,\bar{l})\ , \quad \ldots$$

  (here we add a thread index on top of an edge's arrow, to indicate the identity of the executing thread). The modified state sequence remains a valid path in $\mathcal{P}'_n$, since the shared states "match" and are not changed by any of the removed or inserted edges. Moving the local state change of thread j (from l_I to $\bar{l}$) forward leaves the path intact, since the original edge $(\bar{s},l_I) \to (\bar{s},\bar{l})$ was thread j's first activity.

This procedure is applied to every thread $j \neq i$, with the result that, after the first n transitions, all threads are in a state belonging to T. The suffix of p'' following these transitions reaches t_0 in $\mathcal{P}_n$. □

B Proof of Lemma 2

Lem. 2 *If thread state t_F is reachable in $\mathcal{P}_\infty$, then t_F is also reachable in $\overline{\mathcal{P}}$.*

Proof: We show that t_F is reachable in $\mathcal{P}^+$; the fact that t_F is reachable in $\overline{\mathcal{P}}$ then follows from standard properties of the SCC quotient graph.

Let $t_F = (s_F, l_F)$, and $t_I = (s_I, l_I)$ be the initial state. Since t_F is reachable in $\mathcal{P}_\infty = \cup_{n=1}^{\infty} \mathcal{P}_n$, let n be such that t_F is reachable in $\mathcal{P}_n$ via a witness path p:

$$p \ :: \ (s_I|\underbrace{l_I,\ldots,l_I}_{n}) \quad \rightarrowtail \quad \cdots \quad \rightarrowtail \quad (s_F|l_1,\ldots,l_{i-1},l_F,l_{i+1},\ldots,l_n). \tag{8}$$

Let further $(e_i) := (e_1,\ldots,e_z)$ be the sequence of TTD edges executed along p. We drop all "horizontal" edges from (e_i), i.e. edges of the form $(s,\cdot) \to (s,\cdot)$, to obtain a subsequence $(g_i) := (g_1,\ldots,g_{z'})$ $(z' \leq z)$. Given (g_i), we construct a path σ from t_I to t_F in $\mathcal{P}^+$, by *processing* the edges g_i, defined recursively as follows:

(1) Edge g_1 is processed by copying it to σ.
(2) Suppose edge g_{k-1} has been processed, and suppose its target state is (s,l_i). Edge g_k's source state has shared component s as well, since edges g_{k-1} and g_k are consecutive in p, except for some horizontal edges in between that

may have been dropped, but these do not change the shared state. So let g_k's source state be (s, l_j).
Edge g_k is now processed as follows. If $l_i = l_j$, append g_k to σ. Otherwise, first append $(s, l_i) \dashrightarrow (s, l_j)$ to σ, then g_k. Note that $(s, l_i) \dashrightarrow (s, l_j)$ is a valid expansion edge in R^+, since there exist two non-horizontal edges, g_{k-1} and g_k, adjacent to the expansion edge's source and target, respectively.

Step (2) is repeated until all edges g_i have been processed. It is clear by construction that σ is a valid path in $\mathcal{P}^+$, and that it starts in $t_I = (s_I, l_I)$. We finally have to show that it ends in $t_F = (s_F, l_F)$. It may in fact not: let (s_F, l_f) be the target state of the final edge $g_{z'}$; l_f may or may not be equal to l_F. If it is not, we append an edge $(s_F, l_f) \dashrightarrow (s_F, l_F)$ to σ. This is a valid expansion edge by Def. 1, and σ now ends in t_F, which is hence reachable in $\mathcal{P}^+$. □

C Proof of Theorem 3

Before we turn to this proof, we establish a lemma that uses the δ_l's defined in Sect. 5 to compactly determine local state l's summary along σ^+.

Lem. 3 *Let $b_l = \Sigma_l(1)$ if $l_k = l$ (path σ^+ ends in local state l), and $b_l = \Sigma_l(0)$ otherwise. Then $\Sigma_l(n_l) = n_l \oplus_{b_l} \delta_l$.*

The lemma suggests: in order to determine local state l's summary function in compact form, first compute the constant $\Sigma_l(1)$ (or $\Sigma_l(0)$) using Alg. 2. $\Sigma_l(n_l)$ is then the formula as specified in the lemma.

Proof of Lem. 3: by induction on the number k of vertices of $\sigma^+ = t_1, \ldots, t_k$.

$\boxed{k = 1:}$ then σ^+ has no edges, so $\Sigma_l(n_l) = n_l$, $b_l = 0$, and $\delta_l = 0$. Thus, $\Sigma_l(n_l) = n_l = n_l \oplus_{b_l} 0 = n_l \oplus_{b_l} \delta_l$.

$\boxed{k \to k+1:}$ Suppose $\sigma^+ = t_1, \ldots, t_{k+1}$ has $k+1$ vertices, and Lem. 3 holds for all paths of k vertices. One such path is the **suffix** $\tau^+ = t_2, \ldots, t_{k+1}$ of σ^+. By the induction hypothesis, τ^+'s summary function $\mathcal{T}_l$ satisfies $\mathcal{T}_l(n_l) = n_l \oplus_{c_l} \gamma_l$ for the real edge summary γ_l along τ^+, and $c_l = \mathcal{T}_l(1)$ if $l_{k+1} = l$; otherwise $c_l = \mathcal{T}_l(0)$. Note that τ^+ and σ^+ have the same final state $t_{k+1} = (s_{k+1}, l_{k+1})$.

We now distinguish what Alg. 2 does to the first edge $e_1 = (t_1, t_2) = ((s_1, l_1), (s_2, l_2))$ of σ^+ (which is traversed last):

Case 1: $e_1 \in R$ and $l_1 = l$: Then $\Sigma_l(n_l) = \mathcal{T}_l(n_l)+1$, $\delta_l = \gamma_l + 1$, and $b_l = c_l + 1$. Using the induction hypothesis (IH), we get $\Sigma_l(n_l) = n_l \oplus_{c_l} (\delta_l - 1) + 1$.
- If $n_l + \delta_l - 1 \geq c_l$, then $n_l \oplus_{c_l} (\delta_l - 1) + 1 = n_l + \delta_l = n_l \oplus_{b_l} \delta_l$ since $n_l + \delta_l \geq c_l + 1 = b_l$.
- If $n_l + \delta_l - 1 < c_l$, then $n_l \oplus_{c_l} (\delta_l - 1) + 1 = c_l + 1 = b_l = n_l \oplus_{b_l} \delta_l$ since $n_l + \delta_l < c_l + 1 = b_l$.

Case 2: $e_1 \in R$ and $l_2 = l$: This case is analogous to Case 1; for completeness, we spell it out. We have $\Sigma_l(n_l) = \mathcal{T}_l(n_l) - 1$, $\delta_l = \gamma_l - 1$, and $b_l = c_l - 1$. Using the IH, we get $\Sigma_l(n_l) = n_l \oplus_{c_l} (\delta_l + 1) - 1$.

- If $n_l + \delta_l + 1 \geq c_l$, then $n_l \oplus_{c_l} (\delta_l + 1) - 1 = n_l + \delta_l = n_l \oplus_{b_l} \delta_l$ since $n_l + \delta_l \geq c_l - 1 = b_l$.
- If $n_l + \delta_l + 1 < c_l$, then $n_l \oplus_{c_l} (\delta_l + 1) - 1 = c_l - 1 = b_l = n_l \oplus_{b_l} \delta_l$ since $n_l + \delta_l < c_l - 1 = b_l$.

Case 3: $e_1 \in R^+ \setminus R$ and $l_1 = l$: Then $\Sigma_l(n_l) = \mathcal{T}_l(n_l) \ominus 1 + 1$, $\delta_l = \gamma_l$, and $b_l = c_l \ominus 1 + 1$. Using the IH, we get $\Sigma_l(n_l) = n_l \oplus_{c_l} \delta_l \ominus 1 + 1$.

- If $c_l \geq 1$, then $b_l = c_l$, so $n_l \oplus_{c_l} \delta_l \geq c_l \geq 1$, hence $n_l \oplus_{c_l} \delta_l \ominus 1 + 1 = n_l \oplus_{c_l} \delta_l = n_l \oplus_{b_l} \delta_l$.
- If $c_l = 0$, then $b_l = 1$.
  - If $n_l + \delta_l \geq 1$, then $n_l \oplus_{c_l} \delta_l \ominus 1 + 1 = n_l + \delta_l \ominus 1 + 1 = n_l + \delta_l = n_l \oplus_{b_l} \delta_l$.
  - If $n_l + \delta_l \leq 0$, then $n_l \oplus_{c_l} \delta_l \ominus 1 + 1 = c_l \ominus 1 + 1 = 1 = n_l \oplus_{b_l} \delta_l$.

Case 4: none of the above. In this case e_1 has no impact on the path summary generated by Alg. 2. Thus, $\Sigma_l(n_l) = \mathcal{T}_l(n_l)$; in particular we have $b_l = c_l$ and $\delta_l = \gamma_l$. Further, $\Sigma_l(n_l) = \mathcal{T}_l(n_l) \overset{\text{(IH)}}{=} n_l \oplus_{c_l} \gamma_l = n_l \oplus_{b_l} \delta_l$. □

We now turn to the main goal of this section, the proof of Thm. 3. We repeat it here for convenience, **except** that, applying Lem. 3, we replace term $n_l \oplus_{b_l} \delta_l$ in the original theorem formulation by $\Sigma_l(n_l)$, which simplifies the proof.

Thm. 3 *Let superscript* $^{(\kappa)}$ *denote* κ *function applications. Then, for* $\kappa \geq 1$,

$$\Sigma_l^{(\kappa)}(n_l) = \Sigma_l(n_l) \oplus_{b_l} (\kappa - 1) \cdot \delta_l \ . \tag{9}$$

Proof: by induction on κ. For $\kappa = 1$, the right-hand side (rhs) of (9) equals $\Sigma_l(n_l) \oplus_{b_l} 0 = \Sigma_l(n_l)$ since $\Sigma_l(n_l) + 0 = \Sigma_l(n_l) \geq b_l$ by Lem. 3.

Now suppose (9) holds. For the inductive step we obtain:

$$\begin{aligned} \Sigma_l^{(\kappa+1)}(n_l) &= \Sigma_l(\Sigma_l^{(\kappa)}(n_l)) \\ &\overset{\text{(IH)}}{=} \Sigma_l(\Sigma_l(n_l) \oplus_{b_l} (\kappa - 1) \cdot \delta_l) \\ &\overset{\text{(Lem. 3)}}{=} (\Sigma_l(n_l) \oplus_{b_l} (\kappa - 1) \cdot \delta_l) \oplus_{b_l} \delta_l \ . \end{aligned} \tag{10}$$

We now distinguish three cases ($\langle \ \dots \ \rangle$ below contains proof step justifications):

(1) If $\delta_l \geq 0$:

$$\begin{aligned} & (10) \\ = \quad & \langle\ (\kappa - 1) \cdot \delta_l \geq 0,\ \Sigma_l(n_l) \geq b_l, \text{ hence } \Sigma_l(n_l) + (\kappa - 1) \cdot \delta_l \geq b_l\ \rangle \\ & (\Sigma_l(n_l) + (\kappa - 1) \cdot \delta_l) \oplus_{b_l} \delta_l \\ = \quad & \langle\ \delta_l \geq 0\ \rangle \\ & (\Sigma_l(n_l) + (\kappa - 1) \cdot \delta_l) + \delta_l \\ = \quad & \\ & \Sigma_l(n_l) + \kappa \cdot \delta_l \\ = \quad & \langle\ \Sigma_l(n_l) + \kappa \cdot \delta_l \geq b_l\ \rangle \\ & \Sigma_l(n_l) \oplus_{b_l} \kappa \cdot \delta_l \ , \end{aligned}$$

the final expression being the rhs of (9), for κ replaced by $\kappa + 1$.

(2) If $\delta_l < 0$ and $\Sigma_l(n_l) + (\kappa - 1) \cdot \delta_l < b_l$, then also $\Sigma_l(n_l) + \kappa \cdot \delta_l < b_l$, and:

$$\begin{aligned}
& (10) \\
= \quad & \langle\ \Sigma_l(n_l) + (\kappa - 1) \cdot \delta_l < b_l\ \rangle \\
& b_l \oplus_{b_l} \delta_l \\
= \quad & \langle\ \delta_l < 0\ \rangle \\
& b_l \\
= \quad & \langle\ \Sigma_l(n_l) + \kappa \cdot \delta_l < b_l\ \rangle \\
& \Sigma_l(n_l) \oplus_{b_l} \kappa \cdot \delta_l\ .
\end{aligned}$$

(3) If finally $\delta_l < 0$ and $\Sigma_l(n_l) + (\kappa - 1) \cdot \delta_l \geq b_l$, then (10) reduces to $(\Sigma_l(n_l) + (\kappa - 1) \cdot \delta_l) \oplus_{b_l} \delta_l$. To get an overview of what we need to prove, let

$$\begin{aligned}
X &= \Sigma_l(n_l) + (\kappa - 1) \cdot \delta_l\ , & X' &= \Sigma_l(n_l)\ , \\
Y &= \delta_l\ , & Y' &= \kappa \cdot \delta_l\ .
\end{aligned}$$

Then (the reduced) (10) equals $X \oplus_{b_l} Y$, and the rhs of (9) equals $X' \oplus_{b_l} Y'$. Further, observe that $X + Y = X' + Y'$. This implies that $X \oplus_{b_l} Y = X' \oplus_{b_l} Y'$, which follows immediately by distinguishing whether $X + Y \geq b_l$ or not. The equality $X \oplus_{b_l} Y = X' \oplus_{b_l} Y'$ is what we needed to prove. $\square$