A Proof of Deadlock Freedom

The proof uses the following additional definitions:

$$
\begin{array}{lcl}
\mathit{InNCS}(i) & \triangleq & pc[i] = \text{“ncs”} \\
\mathit{Fairness} & \triangleq & \forall\, i \in \mathit{Procs} : \mathrm{WF}_{\mathit{vars}}((pc[i] \neq \text{“ncs”}) \land p(i)) \\
\mathit{SomeTrying} & \triangleq & \exists\, i \in \mathit{Procs} : \mathit{Trying}(i) \\
\mathit{NoneInCS} & \triangleq & \forall\, i \in \mathit{Procs} : \neg \mathit{InCS}(i)
\end{array}
$$

Theorem $\mathit{Spec} \Rightarrow \mathit{DeadlockFree}$

1. $\mathit{Spec} \Rightarrow \Box \mathit{LInv}$

   Proof: This is a standard invariance proof, which is omitted.

2. Suffices Assume: $\Box \mathit{LInv} \land \Box[\mathit{Next}]_{\mathit{vars}} \land \mathit{Fairness} \land \Box \mathit{NoneInCS}$
   Prove: $\mathit{SomeTrying} \leadsto \text{FALSE}$

   Proof: By 1 and the definition of *Spec*, since *DeadlockFree* equals $\mathit{SomeTrying} \leadsto \neg \mathit{NoneInCS}$, which we prove by assuming *SomeTrying* and $\Box \mathit{NoneInCS}$ and obtaining a contradiction.

3. $\mathit{Trying}(i) \Rightarrow \Box \mathit{Trying}(i)$ and $\neg \mathit{Trying}(i) \leadsto \Box \mathit{InNCS}(i)) \lor \Box \mathit{Trying}(i)$, for all $i \in \mathit{Proc}$.

   Proof: Fairness implies $\neg \mathit{Trying}(i) \leadsto \mathit{InNCS}(i)$, the program implies $\mathit{InNCS}(i) \leadsto \mathit{Trying}(i) \lor \Box \mathit{InNCS}(i)$, and the program and the assumption $\Box \mathit{NoneInCS}$ imply $\mathit{Trying}(i) \Rightarrow \Box \mathit{Trying}(i)$.

   “The program” is an abbreviation for the assumptions $\Box \mathit{LInv}$ and $\Box[\mathit{Next}]_{\mathit{vars}}$.

4. $\mathit{SomeTrying} \leadsto \land\ \Box \mathit{SomeTrying}$
   $\qquad\qquad\qquad\quad \land\ \forall i \in \mathit{Procs} : \Box \mathit{Trying}(i) \lor \Box \mathit{InNCS}(i)$

   Define: $T(i) \triangleq \mathit{Trying}(i)$
   $\qquad\quad\ \ ST \triangleq \mathit{SomeTrying}$

   4.1. $ST \leadsto \Box ST$

   Proof: By step 3.

   4.2. $ST \Rightarrow (\Box ST \land T(i)) \lor (\Box ST \land \neg T(i))$

   Proof: Obvious.

   4.3. $\Box ST \land T(i) \leadsto \Box ST \land \Box T(i)$

   Proof: By step 3.

   4.4. $\Box ST \land \neg T(i) \leadsto \Box ST \land (\Box \mathit{InNCS}(i) \lor \Box T(i))$

   Proof: By step 3.

   4.5. Q.E.D.

   Proof: Steps 4.1–4.4 and leads-to induction with the following proof graph imply $ST \leadsto \Box ST \wedge (\Box T(i) \vee \Box \mathit{InNCS}(i))$ for each $i \in \mathit{Proc}$.

$$\begin{tikzcd}
& & (\Box ST) \wedge T(i) \arrow[r] & (\Box ST) \wedge \Box T(i) \arrow[dr] & \\
ST \arrow[r] & \Box ST \arrow[ur] \arrow[dr] & & & (\Box ST) \wedge (\Box T(i) \vee \Box \mathit{InNCS}(i)) \\
& & (\Box ST) \wedge \neg T(i) \arrow[r] \arrow[uur] & (\Box ST) \wedge \Box \mathit{InNCS}(i) \arrow[ur] &
\end{tikzcd}$$

The result follows from this, since $\forall i \in \mathit{Proc} : ST \leadsto \Box P(i)$ implies $ST \leadsto \forall i \in \mathit{Proc} : \Box P(i)$ for any $P(i)$ because Proc is a finite set.

Define:
$$\begin{aligned}
\mathit{Never}(i) &\triangleq \Box \mathit{Trying}(i) \wedge \Box \neg x[i] \\
\mathit{Always}(i) &\triangleq \Box \mathit{Trying}(i) \wedge \Box x[i] \\
\mathit{Blinking}(i) &\triangleq \Box \mathit{Trying}(i) \wedge \Box\Diamond x[i] \wedge \Box\Diamond \neg x[i]
\end{aligned}$$

5. $\Box \mathit{SomeTrying} \leadsto \wedge\ \Box \mathit{SomeTrying}$
$\wedge\ \forall i \in \mathit{Procs} :$
$\Box \mathit{InNCS}(i) \vee \mathit{Never}(i) \vee \mathit{Always}(i) \vee \mathit{Blinking}(i)$

Proof: By step 4 and the tautology:

$$\text{TRUE} \leadsto \Box F \vee \Box \neg F \vee (\Box\Diamond F \wedge \Box\Diamond \neg F)$$

which asserts that either F is eventually forever true or forever false, or else it is infinitely often true and infinitely often false.

6. Suffices Assume: $\wedge\ \Box \mathit{SomeTrying}$
$\wedge\ \forall i \in \mathit{Procs} :$
$\Box \mathit{InNCS}(i) \vee \mathit{Never}(i) \vee \mathit{Always}(i) \vee \mathit{Blinking}(i)$
Prove: FALSE

Proof: By step 5, this provides the desired contradiction.

7. $\forall i \in \mathit{Proc} : \neg \mathit{Blinking}(i)$

Proof: We assume $\mathit{Blinking}(j)$ is true for some j and obtain a contradiction. Let i be the smallest such j. By $\Box \mathit{Trying}(i) \wedge \Box\Diamond \neg x[i]$, process i must eventually execute $e3$, find $x[\mathit{other}] = \text{TRUE}$, and reach $e5$, which by LInv implies $i > \mathit{other}$. Hence $\mathit{Blinking}(\mathit{other})$ is false (because i is the smallest j with $\mathit{Blinking}(j)$ true) and $x[\mathit{other}] = \text{TRUE}$ implies $\mathit{Never}(\mathit{other})$ is false. Therefore, the step 6 assumption implies that $\mathit{Always}(\mathit{other})$ is true, which implies $\Box x[\mathit{other}]$. This implies that i must stay forever at $e5$, making $\Box \neg x[i]$ true. This is a contradiction because $\mathit{Blinking}(i)$ implies $\Box\Diamond x[i]$.

8. $\neg (\exists i \in \mathit{Procs} : \Box \mathit{Trying}(i) \wedge \Box x[i])$

Proof: Let S be the set of processes i such that $\Box \mathit{Trying}(i) \wedge \Box x[i]$ holds. We assume S is nonempty and obtain a contradiction. Let i be the smallest element in S. By $\Box \mathit{Trying}(i) \wedge \Box x[i]$, process i must eventually reach $e6$ and remain there forever, with $other > i$, so $other$ is not in S. By step 7 and the step 6 assumption, this implies $\Box\neg x[other]$, so i must eventually execute $e6$ and reach $e2$, which is a contradiction.

9. $\neg(\exists i \in Procs : \Box Trying(i) \land \Box\neg x[i])$

   PROOF: We assume that there is an i such that $\Box Trying(i) \land \Box\neg x[i]$ holds and obtain a contradiction. The assumption implies that i eventually reaches and remains forever at $e5$. However, steps 7 and 8 and the step 6 assumption imply that $\Box\neg x[j]$ holds for all processes j, so fairness implies that process i cannot remain forever at $e5$, which is the required contradiction.

10. Q.E.D.

    PROOF: Steps 7–9 and the second conjunct of the step 6 assumption imply $\forall i \in Procs : \Box InNCS(i)$, which is a contradiction because the step 6 assumption also implies $\Box SomeTrying$.