

# A Proof of Deadlock Freedom

The proof uses the following additional definitions:

$$\begin{aligned}
 InNCS(i) &\triangleq pc[i] = \text{"ncs"} \\
 Fairness &\triangleq \forall i \in Procs : WF_{vars}((pc[i] \neq \text{"ncs"}) \wedge p(i)) \\
 SomeTrying &\triangleq \exists i \in Procs : Trying(i) \\
 NoneInCS &\triangleq \forall i \in Procs : \neg InCS(i)
 \end{aligned}$$

?

←

**Theorem**  $Spec \Rightarrow DeadlockFree$

→

1.  $Spec \Rightarrow \Box LInv$

C

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

I

2. SUFFICES ASSUME:  $\Box LInv \wedge \Box [Next]_{vars} \wedge Fairness \wedge \Box NoneInCS$

S

PROVE:  $SomeTrying \leadsto FALSE$

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

3.  $Trying(i) \Rightarrow \Box Trying(i)$  and  $\neg Trying(i) \leadsto \Box InNCS(i) \vee \Box Trying(i)$ , for all  $i \in Proc$ .

PROOF: Fairness implies  $\neg Trying(i) \leadsto InNCS(i)$ , the program implies  $InNCS(i) \leadsto Trying(i) \vee \Box InNCS(i)$ , and the program and the assumption  $\Box NoneInCS$  imply  $Trying(i) \Rightarrow \Box Trying(i)$ .

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

4.  $SomeTrying \leadsto \wedge \Box SomeTrying$   
 $\wedge \forall i \in Procs : \Box Trying(i) \vee \Box InNCS(i)$

DEFINE:  $T(i) \triangleq Trying(i)$   
 $ST \triangleq SomeTrying$

4.1.  $ST \leadsto \Box ST$

PROOF: By step 3.

4.2.  $ST \Rightarrow (\Box ST \wedge T(i)) \vee (\Box ST \wedge \neg T(i))$

PROOF: Obvious.

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

PROOF: By step 3.

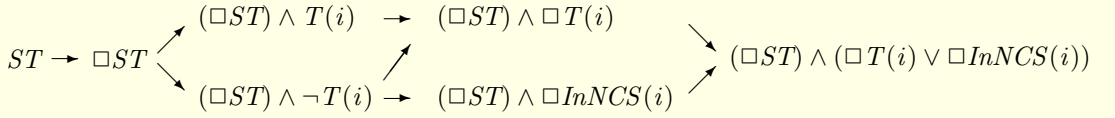
4.4.  $\Box ST \wedge \neg T(i) \leadsto \Box ST \wedge (\Box InNCS(i) \vee \Box T(i))$

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  $ST \rightsquigarrow \Box ST \wedge (\Box T(i) \vee \Box InNCS(i))$  for each  $i \in Proc$ .



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

?

←

DEFINE:  $Never(i) \triangleq \Box Trying(i) \wedge \Box \neg x[i]$

→

$Always(i) \triangleq \Box Trying(i) \wedge \Box x[i]$

C

$Blinking(i) \triangleq \Box Trying(i) \wedge \Box \Diamond x[i] \wedge \Box \Diamond \neg x[i]$

I

5.  $\Box SomeTrying \rightsquigarrow \wedge \Box SomeTrying$

S

$\wedge \forall i \in Procs :$

$\Box InNCS(i) \vee Never(i) \vee Always(i) \vee Blinking(i)$

PROOF: By step 4 and the tautology:

$$TRUE \rightsquigarrow \Box F \vee \Box \neg F \vee (\Box \Diamond F \wedge \Box \Diamond \neg F)$$

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:  $\wedge \Box SomeTrying$

$\wedge \forall i \in Procs :$

$\Box InNCS(i) \vee Never(i) \vee Always(i) \vee Blinking(i)$

PROVE: FALSE

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

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

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

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

PROOF: Let  $S$  be the set of processes  $i$  such that  $\Box Trying(i) \wedge \Box x[i]$  holds. We assume  $S$  is nonempty and obtain a contradiction. Let  $i$  be the smallest element in  $S$ . By  $\Box Trying(i) \wedge \Box x[i]$ , 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.  $\neg(\exists i \in Procs : \Box Trying(i) \wedge \Box\neg x[i])$

PROOF: We assume that there is an  $i$  such that  $\Box Trying(i) \wedge \Box\neg x[i]$  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  $\Box SomeTrying$ .

?

←

→

C

I

S