

A Temporal Logic Proof of Deadlock Freedom

Theorem The 2-process 1-bit algorithm satisfies *DeadlockFree*

DEFINE $T0 \triangleq \text{Trying}(0)$
 $T1 \triangleq \text{Trying}(1)$
 $\text{Success} \triangleq \text{InCS}(0) \vee \text{InCS}(1)$

1 $\langle 1 \rangle 1$. SUFFICES ASSUME: $\Box \neg \text{Success}$
 PROVE: $(T0 \vee T1) \leadsto \text{FALSE}$

PROOF: By standard temporal reasoning, since *DeadlockFree* equals $(T0 \vee T1) \leadsto \text{Success}$.

2 $\langle 1 \rangle 2$. $T0 \leadsto \text{FALSE}$

2.1 $\langle 2 \rangle 1$. $T0 \leadsto \Box(pc[0] = \text{"e2"})$

PROOF: Process 0 is never at *e3* or *e4*. Therefore, from the code and fairness, we see that if *T0* is true and process 0 never reaches *cs* (which is implied by the assumption $\Box \neg \text{Success}$), then process 0 eventually reaches *e2* and stays there forever.

2.2 $\langle 2 \rangle 2$. $\Box(pc[0] = \text{"e2"}) \leadsto \Box((pc[0] = \text{"e2"}) \wedge \neg x[1])$.

2.2.1 $\langle 3 \rangle 1$. SUFFICES ASSUME: $\Box(pc[0] = \text{"e2"})$
 PROVE: $\text{TRUE} \leadsto \Box \neg x[1]$

PROOF: By the $\Box \leadsto$ Rule.

2.2.2 $\langle 3 \rangle 2$. $\text{TRUE} \leadsto (\Box(pc[1] = \text{"ncs"}) \vee \Box T1)$.

PROOF: The code and fairness imply that if process 1 never reaches *cs* (by the assumption $\Box \neg \text{Success}$), then eventually it must either reach and remain forever at *ncs*, or *T1* must become true and remain true forever.

2.2.3 $\langle 3 \rangle 3$. $\Box(pc[1] = \text{"ncs"}) \Rightarrow \Box \neg x[1]$.

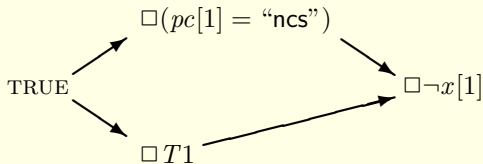
PROOF: $x[1]$ equals FALSE when process 1 is at *ncs*.

2.2.4 $\langle 3 \rangle 4$. $\Box T1 \leadsto \Box \neg x[1]$

PROOF: $(pc[0] = \text{"e2"})$ implies $x[0]$; and the code, fairness, and $\Box \neg \text{Success}$ imply that $\Box x[0]$ leads to process 1 reaching and remaining forever at *e4* with $x[1]$ equal to FALSE.

2.2.5 $\langle 3 \rangle 5$. Q.E.D.

PROOF: By $\langle 3 \rangle 1$ – $\langle 3 \rangle 4$ and Leads-To Induction, with this proof graph:



2.3 $\langle 2 \rangle 3. \Box((pc[0] = \text{"e2"}) \wedge \neg x[1]) \rightsquigarrow \text{FALSE}$

PROOF: The code and fairness imply that $(pc[0] = \text{"e2"})$ and $\Box \neg x[1]$ leads to process 0 reaching *cs*, contradicting $\Box \neg \text{Success}$.

2.4 $\langle 2 \rangle 4. \text{Q.E.D.}$

PROOF: By $\langle 2 \rangle 1$ – $\langle 2 \rangle 3$ and Leads-To Induction, with this proof graph:

$$T0 \rightarrow \Box(pc[0] = \text{"e2"}) \rightarrow \Box((pc[0] = \text{"e2"}) \wedge \neg x[1]) \rightarrow \text{FALSE}$$

3 $\langle 1 \rangle 3. T1 \rightsquigarrow \text{FALSE}$

3.1 $\langle 2 \rangle 1. T1 \Rightarrow \Box T1$

PROOF: From the code, we see that if $T1$ is true and process 1 never reaches *cs* (which is implied by the assumption $\Box \neg \text{Success}$), then $T1$ remains forever true.

3.2 $\langle 2 \rangle 2. \Box T1 \rightsquigarrow (T0 \vee \Box(T1 \wedge \neg T0))$

PROOF: By the tautologies $F \rightsquigarrow (G \vee (F \wedge \Box \neg G))$ and $\Box F \wedge \Box G \equiv \Box(F \wedge G)$.

3.3 $\langle 2 \rangle 3. \Box(T1 \wedge \neg T0) \rightsquigarrow \Box(T1 \wedge \neg x[0])$

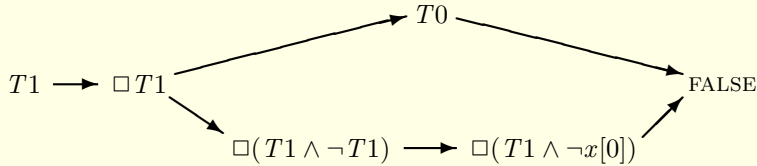
PROOF: By the code and fairness, $\Box \neg T0$ implies that eventually process 0 is always at *ncs*, which implies that $x[0]$ always equals FALSE.

3.4 $\langle 2 \rangle 4. \Box(T1 \wedge \neg x[0]) \rightsquigarrow \text{FALSE}$

PROOF: The code, fairness, and $\Box \neg x[0]$ imply that process 1 eventually reaches *e2*. Fairness and $\Box \neg x[0]$ then imply that process 1 reaches *cs*, contradicting the assumption $\Box \neg \text{Success}$.

3.5 $\langle 2 \rangle 5. \text{Q.E.D.}$

PROOF: By $\langle 2 \rangle 1$ – $\langle 2 \rangle 4$, step $\langle 1 \rangle 2$, and Leads-To Induction, with this proof graph:



4 $\langle 1 \rangle 4. \text{Q.E.D.}$

PROOF: By steps $\langle 1 \rangle 1$ – $\langle 1 \rangle 3$ and a trivial application of Leads-To Induction.