Deadlock Free and FCFS Implies Starvation Free

We first formally define FCFS.

$$\mathit{InNCS}(p) \triangleq pc[p] = \text{"ncs"}$$

$$\mathit{Waiting}(p) \triangleq pc[p] = \text{"waiting"}$$

$$\begin{array}{l} \mathit{FCFS} \triangleq \\ \quad \forall\, p,\, q \in \mathit{Procs} : \\ \qquad \Box(\mathit{Waiting}(p) \;\wedge\; \mathit{InNCS}(q) \;\wedge\; \Box\neg \mathit{InCS}(p) \;\Rightarrow\; \Box\neg \mathit{InCS}(q)) \end{array}$$

We now state and prove the theorem.

Theorem $\mathit{DeadlockFree} \wedge \mathit{FCFS} \;\Rightarrow\; \mathit{StarvationFree}$

1. SUFFICES ASSUME: $p \in \mathit{Procs}$, FCFS, $\mathit{DeadlockFree}$
   PROVE: $\mathit{Trying}(p) \wedge \Box\neg \mathit{InCS}(p) \leadsto \text{FALSE}$

   PROOF: By definition of $\mathit{StarvationFree}$ and simple temporal logic.

2. $\mathit{Trying}(p) \wedge \Box\neg \mathit{InCS}(p) \;\leadsto\; \mathit{Waiting}(p) \wedge \Box\neg \mathit{InCS}(p)$

   PROOF: By fairness for process p.

3. $\mathit{Waiting}(p) \wedge \Box\neg \mathit{InCS}(p) \;\leadsto\; \Box(\mathit{Waiting}(p) \wedge \neg \mathit{InCS}(p))$

   PROOF: The algorithm implies that $\mathit{Waiting}(p)$ can become false only by making $\mathit{InCS}(p)$ true.

4. $\forall q \in \mathit{Procs} \setminus \{p\} : \Box(\mathit{Waiting}(p) \wedge \neg \mathit{InCS}(p)) \leadsto \Box\neg \mathit{InCS}(q)$

   4.1. SUFFICES ASSUME: $q \in \mathit{Procs}$, $q \neq p$, $\Box(\mathit{Waiting}(p) \wedge \neg \mathit{InCS}(p))$
   PROVE: $\text{TRUE} \leadsto \Box\neg \mathit{InCS}(q)$

   PROOF: By simple temporal reasoning.

   4.2. $\text{TRUE} \leadsto (\Box\Diamond \mathit{InCS}(q)) \vee (\Box\neg \mathit{InCS}(q))$

   PROOF: By the temporal logic tautology $\Box\Diamond F \vee \Diamond\Box(\neg F)$.

   4.3. $\Box\Diamond \mathit{InCS}(q) \leadsto \mathit{InNCS}(q)$

   PROOF: The algorithm's code and fairness assumption imply $\mathit{InCS}(q) \leadsto \mathit{InNCS}(q)$.

   4.4. $\mathit{InNCS}(q) \Rightarrow \Box\neg \mathit{InCS}(q)$

   PROOF: By the step 4.1 assumption, which implies $\mathit{Waiting}(p) \wedge \Box\neg \mathit{InCS}(p)$ and FCFS.

   4.5. Q.E.D.

   PROOF: By 4.2–4.4 and the Leads-To Induction Rule.

5. $\mathit{Waiting}(p) \leadsto \exists q \in \mathit{Procs} : \mathit{InCS}(q)$

   PROOF: By $\mathit{DeadlockFree}$ (assumed in 1).

6. Q.E.D.

PROOF: Steps 2–5 and temporal logic yield:

$$Trying(p) \land \Box\neg InCS(p) \;\leadsto\; \begin{array}{l} \land\ \Box\neg InCS(p) \\ \land\ \forall q \in Procs \setminus \{p\} : \Box\neg InCS(q) \\ \land\ \exists q \in Procs : InCS(q) \end{array}$$

and the conjunction to the right of the $\leadsto$ equals FALSE.

CLOSE