
MODULE *actor*

EXTENDS *TLC*, *Integers*, *Sequences*

```

--fair algorithm ActorStuff{
variables actorInboxes = ("actor1" :> ⟨⟩) @@ ("actor2" :> ⟨"OthersJoin"⟩);
    triggered = FALSE;

procedure trigger( trigger_content = "?" ) {
    triggerLabel:
    triggered := TRUE;
    return;
}

fair process ( actor ∈ {"actor1", "actor2"} )
variables currentMessage = ⟨"?", "no_content"⟩;
    kind = "?";
    content = "no_content";
{
    Send:
    actorInboxes[actor] := Append(actorInboxes[actor], ⟨"trigger", "foo"⟩);
    WaitForMessages:+
    while ( TRUE ) {
        if ( actorInboxes[self] ≠ ⟨⟩ ) {
            currentMessage := Head(actorInboxes[self]);
            content := Head(Tail(currentMessage));
            kind := Head(currentMessage);
            actorInboxes[self] := Tail(actorInboxes[self]);
        };
        ProcessMessage:
        if ( kind = "OthersJoin" ) {
            call trigger(content);
        }
    }
}
}

```

BEGIN TRANSLATION (*chksum(pcal)* = "e5b874e9" \wedge *chksum(tla)* = "c9de62d2")

VARIABLES *actorInboxes*, *triggered*, *pc*, *stack*, *trigger_content*, *currentMessage*,
kind, *content*

vars \triangleq ⟨*actorInboxes*, *triggered*, *pc*, *stack*, *trigger_content*,
currentMessage, *kind*, *content*⟩

ProcSet \triangleq ({"actor1", "actor2"})

Init \triangleq Global variables
 \wedge *actorInboxes* = ("actor1" :> ⟨⟩) @@ ("actor2" :> ⟨"OthersJoin", "actor1"⟩)
 \wedge *triggered* = FALSE

Procedure trigger
 $\wedge \text{trigger_content} = [self \in ProcSet \mapsto "?"]$
 Process actor
 $\wedge \text{currentMessage} = [self \in \{\text{"actor1"}, \text{"actor2"}\} \mapsto \langle "?", \text{"no_content"} \rangle]$
 $\wedge \text{kind} = [self \in \{\text{"actor1"}, \text{"actor2"}\} \mapsto "?"]$
 $\wedge \text{content} = [self \in \{\text{"actor1"}, \text{"actor2"}\} \mapsto \text{"no_content"}]$
 $\wedge \text{stack} = [self \in ProcSet \mapsto \langle \rangle]$
 $\wedge \text{pc} = [self \in ProcSet \mapsto \text{"WaitForMessages"}]$

$\text{triggerLabel}(self) \triangleq \wedge pc[self] = \text{"triggerLabel"}$
 $\wedge \text{triggered}' = \text{TRUE}$
 $\wedge \text{pc}' = [pc \text{ EXCEPT } ![self] = \text{Head}(\text{stack}[self]).pc]$
 $\wedge \text{trigger_content}' = [\text{trigger_content} \text{ EXCEPT } ![self] = \text{Head}(\text{stack}[self]).\text{trigger_content}]$
 $\wedge \text{stack}' = [\text{stack} \text{ EXCEPT } ![self] = \text{Tail}(\text{stack}[self])]$
 $\wedge \text{UNCHANGED } \langle \text{actorInboxes}, \text{currentMessage}, \text{kind}, \text{content} \rangle$

$\text{trigger}(self) \triangleq \text{triggerLabel}(self)$

$\text{WaitForMessages}(self) \triangleq \wedge pc[self] = \text{"WaitForMessages"}$
 $\wedge \text{IF } \text{actorInboxes}[self] \neq \langle \rangle$
 $\quad \text{THEN } \wedge \text{currentMessage}' = [\text{currentMessage} \text{ EXCEPT } ![self] = \text{Head}(\text{actorInboxes}[self]).\text{currentMessage}]$
 $\quad \wedge \text{content}' = [\text{content} \text{ EXCEPT } ![self] = \text{Head}(\text{Tail}(\text{currentMessage}'[self]))]$
 $\quad \wedge \text{kind}' = [\text{kind} \text{ EXCEPT } ![self] = \text{Head}(\text{currentMessage}'[self])]$
 $\quad \wedge \text{actorInboxes}' = [\text{actorInboxes} \text{ EXCEPT } ![self] = \text{Tail}(\text{actorInboxes}[self])]$
 $\quad \text{ELSE } \wedge \text{TRUE}$
 $\quad \wedge \text{UNCHANGED } \langle \text{actorInboxes}, \text{currentMessage}, \text{kind}, \text{content} \rangle$
 $\wedge \text{pc}' = [pc \text{ EXCEPT } ![self] = \text{"ProcessMessage"}]$
 $\wedge \text{UNCHANGED } \langle \text{triggered}, \text{stack}, \text{trigger_content} \rangle$

$\text{ProcessMessage}(self) \triangleq \wedge pc[self] = \text{"ProcessMessage"}$
 $\wedge \text{IF } \text{kind}[self] = \text{"OthersJoin"}$
 $\quad \text{THEN } \wedge \wedge \text{stack}' = [\text{stack} \text{ EXCEPT } ![self] = \langle [\text{procedure} \mapsto \text{"trigger"}, \text{pc} \mapsto \text{"WaitForMessages"}, \text{trigger_content} \mapsto \text{trigger_content}[self] \circ \text{stack}[self]] \rangle]$
 $\quad \wedge \text{trigger_content}' = [\text{trigger_content} \text{ EXCEPT } ![self] = \text{content}[self]]$
 $\quad \wedge \text{pc}' = [pc \text{ EXCEPT } ![self] = \text{"triggerLabel"}]$
 $\quad \text{ELSE } \wedge \text{pc}' = [pc \text{ EXCEPT } ![self] = \text{"WaitForMessages"}]$
 $\quad \wedge \text{UNCHANGED } \langle \text{stack}, \text{trigger_content} \rangle$
 $\wedge \text{UNCHANGED } \langle \text{actorInboxes}, \text{triggered}, \text{currentMessage}, \text{kind}, \text{content} \rangle$

$\text{actor}(self) \triangleq \text{WaitForMessages}(self) \vee \text{ProcessMessage}(self)$

$$\begin{aligned}
Next &\triangleq (\exists self \in ProcSet : trigger(self)) \\
&\quad \vee (\exists self \in \{\text{"actor1"}, \text{"actor2"}\} : actor(self)) \\
Spec &\triangleq \wedge Init \wedge \Box[Next]_{vars} \\
&\quad \wedge WF_{vars}(Next) \\
&\quad \wedge \forall self \in \{\text{"actor1"}, \text{"actor2"}\} : \wedge WF_{vars}(actor(self)) \\
&\quad \quad \wedge SF_{vars}(WaitForMessages(self)) \\
&\quad \quad \wedge WF_{vars}(trigger(self))
\end{aligned}$$

END TRANSLATION

$$Triggered \triangleq triggered = \text{TRUE}$$

$$Liveness \triangleq \Diamond Triggered$$

\ * Modification History
\ * Last modified Sun Jan 30 19:33:19 YEKT 2022 by pervu
\ * Created Sun Jan 30 18:34:11 YEKT 2022 by pervu