
MODULE *actor*

EXTENDS *TLC, Integers, Sequences*

$between01(n1, nb, n2) \triangleq (nb \geq 0) \wedge (((n1 < n2) \wedge ((n1 < nb) \wedge (nb \leq n2))) \vee ((n1 \geq n2) \wedge ((n1 < nb) \vee (nb \leq n2))))$
 $between00(n1, nb, n2) \triangleq (nb \geq 0) \wedge (((n1 < n2) \wedge ((n1 < nb) \wedge (nb < n2))) \vee ((n1 \geq n2) \wedge ((n1 < nb) \vee (nb < n2))))$

$fingerStart(myId, k, m) \triangleq (myId + (2^{(k-1)})) \% m$

CONSTANTS $m, bm, NULL$

```

--fair algorithm ActorStuff{
variables actorInboxes = (0:> ⟨⟩) @@ (1:> ⟨⟩) @@ (3:> ⟨⟩);
    triggered = FALSE;
    make local
    fingerTables = (0:> ((1:> 0) @@ (2:> 0) @@ (4:> 0)))
    @@ (1:> ((2:> NULL) @@ (3:> NULL) @@ (5:> NULL)))
    @@ (3:> ((4:> NULL) @@ (5:> NULL) @@ (7:> NULL)));
    predecessors = (0:> 0) @@ (1:> NULL) @@ (3:> NULL);

fair process ( actor ∈ {0, 1, 3} )
variables currentMessage = ⟨"?", NULL, NULL⟩;
    kind = "?";
    id = NULL;
    i;
    joined = FALSE;
{
    Join: +
    if ( self = 0 ) {
        joined := TRUE;
    } else {
        actorInboxes[0] := Append(actorInboxes[0], ⟨"FindPredecessor", fingerStart(self, 1, bm), self⟩);
    };
    WaitForMessagesBeforeJoin: +
    if ( joined ≠ TRUE ) {
        await actorInboxes[self] ≠ ⟨⟩;
        {
            currentMessage := Head(actorInboxes[self]);
            kind := Head(currentMessage);
            actorInboxes[self] := Tail(actorInboxes[self]);
        };
        ProcessMessageBeforeJoin: +
        {
            if ( kind = "Predecessor" ) {
                predecessors[self] := currentMessage[2];
                fingerTables[self][fingerStart(self, 1, bm)] := currentMessage[3];
                joined := TRUE;
            }
        }
    }
}

```

```

    } ;
    } ;
    ReturnDefaultsBeforeJoin:+ {
        currentMessage := ⟨ "?", NULL, NULL ⟩ ;
        kind := "?" ;
        id := NULL ;
    } ;
} ;
Stabilize:+
    if ( joined ) {
        actorInboxes[self] := Append(actorInboxes[self], ⟨ "Stabilize", 1 ⟩ ) ;
    } ;
WaitForMessages:+
    if ( joined )
    {
        {
            await actorInboxes[self] ≠ ⟨ ⟩ ;
            {
                currentMessage := Head(actorInboxes[self]) ;
                kind := Head(currentMessage) ;
                actorInboxes[self] := Tail(actorInboxes[self]) ;
            } ;
        }
        ProcessMessage:+
        {
            if ( kind = "FindPredecessor" ) {
                id := currentMessage[2] ;
                await fingerTables[self][fingerStart(self, 1, bm)] ≠ NULL ;
            }
            if ( between01(self, id, fingerTables[self][fingerStart(self, 1, bm)]) ) {
                actorInboxes[currentMessage[3]] :=
                    Append(actorInboxes[currentMessage[3]], ⟨ "Predecessor", self, fingerTables[self][fingerStart(self, 1, bm)] ⟩ ) ;
            } else {
                i := m ;
                await fingerTables[self] ≠ NULL ;
                FindFirstSuitableI:
                while ( i > 0 ∧ ¬(fingerStart(self, i, bm) ∈ DOMAIN fingerTables[self]) ) {
                    i := i - 1 ;
                } ;
                await fingerTables[self][fingerStart(self, i, bm)] ≠ NULL ;
                MainLoop:
                while ( i > 0 ∧ ¬(between00(self, fingerTables[self][fingerStart(self, i, bm)], id)) ) {
                    i := i - 1 ;
                    FindSuitableI:
                    while ( i > 0 ∧ ¬(fingerStart(self, i, bm) ∈ DOMAIN fingerTables[self]) ) {
                        i := i - 1 ;
                    } ;
                } ;
            }
        }
    }

```

```

    } ;
    if ( i = 0 ) {
        actorInboxes[fingerTables[self][fingerStart(self, m, bm))] :=
            Append(actorInboxes[fingerTables[self][fingerStart(self, m, bm)]], currentMessage) ;
    } else {
        actorInboxes[fingerTables[self][fingerStart(self, i, bm))] :=
            Append(actorInboxes[fingerTables[self][fingerStart(self, i, bm)]], currentMessage) ;
    } ;
} ;
} else {
    {
        if ( kind = "Stabilize" ) {
            actorInboxes[fingerTables[self][fingerStart(self, currentMessage[2], bm))] :=
                Append(actorInboxes[fingerTables[self][fingerStart(self, currentMessage[2], bm)]], <"GetPredecessor", self>) ;
        } else {
            if ( kind = "GetPredecessor" ) {
                actorInboxes[currentMessage[2]] :=
                    Append(actorInboxes[currentMessage[2]], <"GotPredecessor", predecessors[self]>) ;
            } else {
                if ( kind = "GotPredecessor" ) {
                    if ( between00(currentMessage[2], fingerTables[self][fingerStart(self, 1, bm)], self,
                        fingerTables[self][fingerStart(self, 1, bm)]) := currentMessage[2] ;
                    } ;
                    actorInboxes[fingerTables[self][fingerStart(self, 1, bm)]] :=
                        Append(actorInboxes[fingerTables[self][fingerStart(self, 1, bm)]], <"Notify", self>) ;
                } else {
                    if ( kind = "Notify" ) {
                        if ( predecessors[self] ≠ NULL ) {
                            if ( between00(currentMessage[2], self, predecessors[self]) ) {
                                predecessors[self] := currentMessage[2] ;
                            } ;
                        } ;
                    } ;
                } ;
            } ;
        } ;
    } ;
} ;
} ;
ReturnDefaults:+ {
    currentMessage := <"?", NULL, NULL> ;
    kind := "?" ;
    id := NULL ;
} ;
} ;

```

```

    } ;
  } ;
} ;

BEGIN TRANSLATION (chksum(pcal) = "e76f72bb" ∧ chksum(tla) = "6a8eed9d")
CONSTANT defaultInitValue
VARIABLES actorInboxes, triggered, fingerTables, predecessors, pc,
           currentMessage, kind, id, i, joined

vars ≜ ⟨actorInboxes, triggered, fingerTables, predecessors, pc,
        currentMessage, kind, id, i, joined⟩

ProcSet ≜ ({0, 1, 3})

Init ≜ Global variables
    ∧ actorInboxes = (0:> ⟨⟩) @@ (1:> ⟨⟩) @@ (3:> ⟨⟩)
    ∧ triggered = FALSE
    ∧ fingerTables =
        (0:> ((1:> 0) @@ (2:> 0) @@ (4:> 0)))
        @@ (1:> ((2:> NULL) @@ (3:> NULL) @@ (5:> NULL)))
        @@ (3:> ((4:> NULL) @@ (5:> NULL) @@ (7:> NULL)))
    ∧ predecessors = (0:> 0) @@ (1:> NULL) @@ (3:> NULL)
    Process actor
    ∧ currentMessage = [self ∈ {0, 1, 3} ↦ ⟨"?", NULL, NULL⟩]
    ∧ kind = [self ∈ {0, 1, 3} ↦ "?"]
    ∧ id = [self ∈ {0, 1, 3} ↦ NULL]
    ∧ i = [self ∈ {0, 1, 3} ↦ defaultInitValue]
    ∧ joined = [self ∈ {0, 1, 3} ↦ FALSE]
    ∧ pc = [self ∈ ProcSet ↦ "Join"]

Join(self) ≜ ∧ pc[self] = "Join"
    ∧ IF self = 0
        THEN ∧ joined' = [joined EXCEPT ![self] = TRUE]
            ∧ UNCHANGED actorInboxes
        ELSE ∧ actorInboxes' = [actorInboxes EXCEPT ![0] = Append(actorInboxes[0], ⟨"FindPredecessors", self⟩)]
            ∧ UNCHANGED joined
    ∧ pc' = [pc EXCEPT ![self] = "WaitForMessagesBeforeJoin"]
    ∧ UNCHANGED ⟨triggered, fingerTables, predecessors,
                  currentMessage, kind, id, i⟩

WaitForMessagesBeforeJoin(self) ≜ ∧ pc[self] = "WaitForMessagesBeforeJoin"
    ∧ IF joined[self] ≠ TRUE
        THEN ∧ actorInboxes[self] ≠ ⟨⟩
            ∧ currentMessage' = [currentMessage EXCEPT ![self] = Head(currentMessage[self])]
            ∧ kind' = [kind EXCEPT ![self] = Head(currentMessage'[self])]
            ∧ actorInboxes' = [actorInboxes EXCEPT ![self] = Tail(actorInboxes[self])]
            ∧ pc' = [pc EXCEPT ![self] = "ProcessMessageBeforeJoin"]
        ELSE ∧ pc' = [pc EXCEPT ![self] = "Stabilize"]

```

$$\begin{aligned}
& \wedge \text{UNCHANGED } \langle \text{actorInboxes}, \\
& \quad \text{currentMessage}, \\
& \quad \text{kind} \rangle \\
& \wedge \text{UNCHANGED } \langle \text{triggered}, \text{fingerTables}, \\
& \quad \text{predecessors}, \text{id}, i, \text{joined} \rangle \\
\text{ProcessMessageBeforeJoin}(self) & \triangleq \wedge pc[self] = \text{"ProcessMessageBeforeJoin"} \\
& \wedge \text{IF } \text{kind}[self] = \text{"Predecessor"} \\
& \quad \text{THEN } \wedge \text{predecessors}' = [\text{predecessors} \text{ EXCEPT } ![self] = \text{currentMessage}] \\
& \quad \wedge \text{fingerTables}' = [\text{fingerTables} \text{ EXCEPT } ![self][\text{fingerStart}(\text{self})]] \\
& \quad \wedge \text{joined}' = [\text{joined} \text{ EXCEPT } ![self] = \text{TRUE}] \\
& \quad \text{ELSE } \wedge \text{TRUE} \\
& \quad \wedge \text{UNCHANGED } \langle \text{fingerTables}, \\
& \quad \quad \text{predecessors}, \\
& \quad \quad \text{joined} \rangle \\
& \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"ReturnDefaultsBeforeJoin"}] \\
& \wedge \text{UNCHANGED } \langle \text{actorInboxes}, \text{triggered}, \\
& \quad \text{currentMessage}, \text{kind}, \text{id}, i, \text{joined} \rangle \\
\text{ReturnDefaultsBeforeJoin}(self) & \triangleq \wedge pc[self] = \text{"ReturnDefaultsBeforeJoin"} \\
& \wedge \text{currentMessage}' = [\text{currentMessage} \text{ EXCEPT } ![self] = \langle "?", \text{NULL}, \text{NULL} \rangle] \\
& \wedge \text{kind}' = [\text{kind} \text{ EXCEPT } ![self] = "?"] \\
& \wedge \text{id}' = [\text{id} \text{ EXCEPT } ![self] = \text{NULL}] \\
& \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Stabilize"}] \\
& \wedge \text{UNCHANGED } \langle \text{actorInboxes}, \text{triggered}, \\
& \quad \text{fingerTables}, \text{predecessors}, \\
& \quad i, \text{joined} \rangle \\
\text{Stabilize}(self) & \triangleq \wedge pc[self] = \text{"Stabilize"} \\
& \wedge \text{IF } \text{joined}[self] \\
& \quad \text{THEN } \wedge \text{actorInboxes}' = [\text{actorInboxes} \text{ EXCEPT } ![self] = \text{Append}(\text{actorInboxes}[self], \langle \text{currentMessage}, \text{kind}, \text{id}, i, \text{joined} \rangle)] \\
& \quad \text{ELSE } \wedge \text{TRUE} \\
& \quad \wedge \text{UNCHANGED } \text{actorInboxes} \\
& \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"WaitForMessages"}] \\
& \wedge \text{UNCHANGED } \langle \text{triggered}, \text{fingerTables}, \text{predecessors}, \\
& \quad \text{currentMessage}, \text{kind}, \text{id}, i, \text{joined} \rangle \\
\text{WaitForMessages}(self) & \triangleq \wedge pc[self] = \text{"WaitForMessages"} \\
& \wedge \text{IF } \text{joined}[self] \\
& \quad \text{THEN } \wedge \text{actorInboxes}[self] \neq \langle \rangle \\
& \quad \wedge \text{currentMessage}' = [\text{currentMessage} \text{ EXCEPT } ![self] = \text{Head}(\text{actorInboxes}[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 \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"ProcessMessage"}] \\
& \quad \text{ELSE } \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Done"}] \\
& \quad \wedge \text{UNCHANGED } \langle \text{actorInboxes},
\end{aligned}$$

$$\begin{aligned}
& \text{currentMessage}, \text{kind} \rangle \\
& \wedge \text{UNCHANGED } \langle \text{triggered}, \text{fingerTables}, \text{predecessors}, \\
& \quad \text{id}, i, \text{joined} \rangle \\
\text{ProcessMessage}(\text{self}) \triangleq & \wedge \text{pc}[\text{self}] = \text{"ProcessMessage"} \\
& \wedge \text{IF } \text{kind}[\text{self}] = \text{"FindPredecessor"} \\
& \quad \text{THEN } \wedge \text{id}' = [\text{id} \text{ EXCEPT } ![\text{self}] = \text{currentMessage}[\text{self}][2]] \\
& \quad \wedge \text{IF } \text{between01}(\text{self}, \text{id}'[\text{self}], \text{fingerTables}[\text{self}][\text{fingerStart}(\text{self}, 1, \text{bm})]) \\
& \quad \quad \text{THEN } \wedge \text{actorInboxes}' = [\text{actorInboxes} \text{ EXCEPT } ![\text{currentMessage}[\text{self}]]] \\
& \quad \quad \wedge \text{pc}' = [\text{pc} \text{ EXCEPT } ![\text{self}] = \text{"ReturnDefaults"}] \\
& \quad \quad \wedge \text{i}' = i \\
& \quad \quad \text{ELSE } \wedge \text{i}' = [i \text{ EXCEPT } ![\text{self}] = m] \\
& \quad \quad \wedge \text{pc}' = [\text{pc} \text{ EXCEPT } ![\text{self}] = \text{"FindFirstSuitable"}] \\
& \quad \quad \wedge \text{UNCHANGED } \text{actorInboxes} \\
& \quad \wedge \text{UNCHANGED } \langle \text{fingerTables}, \text{predecessors} \rangle \\
& \text{ELSE } \wedge \text{IF } \text{kind}[\text{self}] = \text{"Stabilize"} \\
& \quad \text{THEN } \wedge \text{actorInboxes}' = [\text{actorInboxes} \text{ EXCEPT } ![\text{fingerTables}[\text{self}]]] \\
& \quad \wedge \text{UNCHANGED } \langle \text{fingerTables}, \\
& \quad \quad \text{predecessors} \rangle \\
& \quad \text{ELSE } \wedge \text{IF } \text{kind}[\text{self}] = \text{"GetPredecessor"} \\
& \quad \quad \text{THEN } \wedge \text{actorInboxes}' = [\text{actorInboxes} \text{ EXCEPT } ![\text{currentMessage}[\text{self}]]] \\
& \quad \quad \wedge \text{UNCHANGED } \langle \text{fingerTables}, \\
& \quad \quad \quad \text{predecessors} \rangle \\
& \quad \quad \text{ELSE } \wedge \text{IF } \text{kind}[\text{self}] = \text{"GotPredecessor"} \\
& \quad \quad \quad \text{THEN } \wedge \text{IF } \text{between00}(\text{currentMessage}[\text{self}], \text{id}'[\text{self}], \text{fingerTables}[\text{self}][\text{fingerStart}(\text{self}, 1, \text{bm})]) \\
& \quad \quad \quad \quad \text{THEN } \wedge \text{fingerTables}' = [\text{fingerTables} \text{ EXCEPT } ![\text{id}'[\text{self}]]] \\
& \quad \quad \quad \quad \text{ELSE } \wedge \text{TRUE} \\
& \quad \quad \quad \quad \wedge \text{UNCHANGED } \text{fingerTables} \\
& \quad \quad \quad \wedge \text{actorInboxes}' = [\text{actorInboxes} \text{ EXCEPT } ![\text{fingerTables}[\text{self}]]] \\
& \quad \quad \quad \wedge \text{UNCHANGED } \text{predecessors} \\
& \quad \quad \text{ELSE } \wedge \text{IF } \text{kind}[\text{self}] = \text{"Notify"} \\
& \quad \quad \quad \text{THEN } \wedge \text{IF } \text{predecessors}[\text{self}][\text{fingerStart}(\text{self}, 1, \text{bm})] \\
& \quad \quad \quad \quad \text{THEN } \wedge \text{IF } \text{between00}(\text{currentMessage}[\text{self}], \text{id}'[\text{self}], \text{fingerTables}[\text{self}][\text{fingerStart}(\text{self}, 1, \text{bm})]) \\
& \quad \quad \quad \quad \quad \text{THEN } \wedge \text{fingerTables}' = [\text{fingerTables} \text{ EXCEPT } ![\text{id}'[\text{self}]]] \\
& \quad \quad \quad \quad \quad \text{ELSE } \wedge \text{TRUE} \\
& \quad \quad \quad \quad \quad \wedge \text{UNCHANGED } \text{fingerTables} \\
& \quad \quad \quad \quad \text{ELSE } \wedge \text{TRUE} \\
& \quad \quad \quad \quad \wedge \text{UNCHANGED } \text{fingerTables} \\
& \quad \quad \quad \text{ELSE } \wedge \text{TRUE} \\
& \quad \quad \quad \wedge \text{UNCHANGED } \text{predecessors} \\
& \quad \quad \wedge \text{UNCHANGED } \langle \text{actorInboxes}, \\
& \quad \quad \quad \text{fingerTables} \rangle \\
& \quad \wedge \text{pc}' = [\text{pc} \text{ EXCEPT } ![\text{self}] = \text{"ReturnDefaults"}] \\
& \quad \wedge \text{UNCHANGED } \langle \text{id}, i \rangle \\
& \wedge \text{UNCHANGED } \langle \text{triggered}, \text{currentMessage}, \text{kind},
\end{aligned}$$

joined

$$\begin{aligned}
FindFirstSuitableI(self) \triangleq & \wedge pc[self] = \text{"FindFirstSuitableI"} \\
& \wedge \text{IF } i[self] > 0 \wedge \neg(fingerStart(self, i[self], bm) \in \text{DOMAIN } fingerTables[self]) \\
& \quad \text{THEN } \wedge i' = [i \text{ EXCEPT } ![self] = i[self] - 1] \\
& \quad \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"FindFirstSuitableI"}] \\
& \quad \text{ELSE } \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"MainLoop"}] \\
& \quad \wedge i' = i \\
& \wedge \text{UNCHANGED } \langle actorInboxes, triggered, \\
& \quad \quad \quad fingerTables, predecessors, \\
& \quad \quad \quad currentMessage, kind, id, joined \rangle
\end{aligned}$$

$$\begin{aligned}
MainLoop(self) \triangleq & \wedge pc[self] = \text{"MainLoop"} \\
& \wedge \text{IF } i[self] > 0 \wedge \neg(between00(self, fingerTables[self][fingerStart(self, i[self], bm)], id[self] - 1)) \\
& \quad \text{THEN } \wedge i' = [i \text{ EXCEPT } ![self] = i[self] - 1] \\
& \quad \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"FindSuitableI"}] \\
& \quad \wedge \text{UNCHANGED } actorInboxes \\
& \quad \text{ELSE } \wedge \text{IF } i[self] = 0 \\
& \quad \quad \text{THEN } \wedge actorInboxes' = [actorInboxes \text{ EXCEPT } ![fingerTables[self][fingerStart(self, i[self], bm)]] \\
& \quad \quad \text{ELSE } \wedge actorInboxes' = [actorInboxes \text{ EXCEPT } ![fingerTables[self][fingerStart(self, i[self], bm)]] \\
& \quad \quad \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"ReturnDefaults"}] \\
& \quad \quad \wedge i' = i \\
& \wedge \text{UNCHANGED } \langle triggered, fingerTables, predecessors, \\
& \quad \quad \quad currentMessage, kind, id, joined \rangle
\end{aligned}$$

$$\begin{aligned}
FindSuitableI(self) \triangleq & \wedge pc[self] = \text{"FindSuitableI"} \\
& \wedge \text{IF } i[self] > 0 \wedge \neg(fingerStart(self, i[self], bm) \in \text{DOMAIN } fingerTables[self]) \\
& \quad \text{THEN } \wedge i' = [i \text{ EXCEPT } ![self] = i[self] - 1] \\
& \quad \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"FindSuitableI"}] \\
& \quad \text{ELSE } \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"MainLoop"}] \\
& \quad \wedge i' = i \\
& \wedge \text{UNCHANGED } \langle actorInboxes, triggered, fingerTables, \\
& \quad \quad \quad predecessors, currentMessage, kind, id, \\
& \quad \quad \quad joined \rangle
\end{aligned}$$

$$\begin{aligned}
ReturnDefaults(self) \triangleq & \wedge pc[self] = \text{"ReturnDefaults"} \\
& \wedge currentMessage' = [currentMessage \text{ EXCEPT } ![self] = \langle "?", NULL, NULL \rangle] \\
& \wedge kind' = [kind \text{ EXCEPT } ![self] = "?"] \\
& \wedge id' = [id \text{ EXCEPT } ![self] = NULL] \\
& \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Done"}] \\
& \wedge \text{UNCHANGED } \langle actorInboxes, triggered, fingerTables, \\
& \quad \quad \quad predecessors, i, joined \rangle
\end{aligned}$$

$$\begin{aligned}
actor(self) \triangleq & Join(self) \vee WaitForMessagesBeforeJoin(self) \\
& \vee ProcessMessageBeforeJoin(self) \\
& \vee ReturnDefaultsBeforeJoin(self) \vee Stabilize(self)
\end{aligned}$$

$$\begin{aligned}
& \vee \textit{WaitForMessages}(\textit{self}) \vee \textit{ProcessMessage}(\textit{self}) \\
& \vee \textit{FindFirstSuitableI}(\textit{self}) \vee \textit{MainLoop}(\textit{self}) \\
& \vee \textit{FindSuitableI}(\textit{self}) \vee \textit{ReturnDefaults}(\textit{self})
\end{aligned}$$

Allow infinite stuttering to prevent deadlock on termination.

$$\begin{aligned}
\textit{Terminating} & \triangleq \wedge \forall \textit{self} \in \textit{ProcSet} : \textit{pc}[\textit{self}] = \text{"Done"} \\
& \wedge \text{UNCHANGED } \textit{vars}
\end{aligned}$$

$$\begin{aligned}
\textit{Next} & \triangleq (\exists \textit{self} \in \{0, 1, 3\} : \textit{actor}(\textit{self})) \\
& \vee \textit{Terminating}
\end{aligned}$$

$$\begin{aligned}
\textit{Spec} & \triangleq \wedge \textit{Init} \wedge \Box[\textit{Next}]_{\textit{vars}} \\
& \wedge \text{WF}_{\textit{vars}}(\textit{Next}) \\
& \wedge \forall \textit{self} \in \{0, 1, 3\} : \wedge \text{WF}_{\textit{vars}}(\textit{actor}(\textit{self})) \\
& \quad \wedge \text{SF}_{\textit{vars}}(\textit{Join}(\textit{self})) \wedge \text{SF}_{\textit{vars}}(\textit{WaitForMessagesBeforeJoin}(\textit{self})) \wedge \text{SF}_{\textit{vars}}(\textit{Proc})
\end{aligned}$$

$$\textit{Termination} \triangleq \Diamond(\forall \textit{self} \in \textit{ProcSet} : \textit{pc}[\textit{self}] = \text{"Done"})$$

END TRANSLATION

$$\textit{Triggered} \triangleq \textit{predecessors} = (0:>0) @@ (1:>0) @@ (3:>0)$$

$$\textit{Liveness} \triangleq \Diamond \textit{Triggered}$$

$$\textit{LenStateConstraint} \triangleq \textit{Len}(\textit{actorInboxes}[0]) \leq 0 \wedge \textit{Len}(\textit{actorInboxes}[1]) \leq 0 \wedge \textit{Len}(\textit{actorInboxes}[3]) \leq 0$$

\ * Modification History
\ * Last modified Sun May 01 22:55:46 YEKT 2022 by pervu
\ * Created Sun Jan 30 18:34:11 YEKT 2022 by pervu