

---

MODULE *actor*

---

EXTENDS *TLC, Integers, Sequences*

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

CONSTANTS *m, bm*

```

--fair algorithm ActorStuff {
variables actorInboxes = (0:> <<"FindPredecessor", 6, 0>>) @@ (1:> <>) @@ (3:> <>);
    triggered = FALSE;
    fingerTables = (0:> ((1:> 1) @@ (2:> 3) @@ (4:> 0)))
        @@ (1:> ((2:> 3) @@ (3:> 5) @@ (5:> 0)))
        @@ (3:> ((4:> 0) @@ (5:> 0) @@ (7:> 0)))

    <<"FindPredecessor", id, asker>

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

fair process (actor ∈ {0, 1, 3})
variables currentMessage = <"?", -1, -1>;
    kind = "?";
    content = "no_content";
    id = -1;
    asker = -1;
    i;
{
    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 = "FindPredecessor" ) {
            id := currentMessage[2];
            asker := currentMessage[3];
            if ( between01(self, id, fingerTables[self][(self + 1)%bm]) ) {

```

```

    actorInboxes[asker] := Append(actorInboxes[asker], ⟨“Predecessor”, self⟩);
  } else {
    i := m;
    FindFirstSuitableI:
    while ( i > 0 ∧ ¬((self + (2(i-1)))%bm ∈ DOMAIN fingerTables[self]) ) {
      i := i - 1;
    };
    MainLoop:
    while ( i > 0 ∧ ¬(between00(self, fingerTables[self][(self + (2(i-1)))%bm], id)) ) {
      i := i - 1;
      FindSuitableI:
      while ( i > 0 ∧ ¬((self + (2(i-1)))%bm ∈ DOMAIN fingerTables[self]) ) {
        i := i - 1;
      };
    };
    if ( i = 0 ) {
      actorInboxes[fingerTables[self][(self + (2(m-1)))%bm]] :=
        Append(actorInboxes[fingerTables[self][(self + (2(m-1)))%bm]], currentMessage);
    } else {
      actorInboxes[fingerTables[self][(self + (2(i-1)))%bm]] :=
        Append(actorInboxes[fingerTables[self][(self + (2(i-1)))%bm]], currentMessage);
    }
  }
} else {
  if ( kind = “Predecessor” ∧ currentMessage[2] = 3 ) {
    call trigger(content);
    triggered := TRUE;
  }
};
DefaultsBack:
currentMessage := ⟨“?”, -1, -1⟩;
kind := “?”;
id := -1;
asker := -1;
}
}
}

```

BEGIN TRANSLATION (*chksum*(*pcal*) = “*afae019c*” ∧ *chksum*(*tla*) = “*5415e04a*”)

CONSTANT *defaultInitValue*

VARIABLES *actorInboxes*, *triggered*, *fingerTables*, *pc*, *currentMessage*, *kind*, *id*,  
*asker*, *i*

*vars*  $\triangleq$  ⟨*actorInboxes*, *triggered*, *fingerTables*, *pc*, *currentMessage*, *kind*,  
*id*, *asker*, *i*⟩

$ProcSet \triangleq \{0, 1, 3\}$

$Init \triangleq$  Global variables  
 $\wedge actorInboxes = (0:> \langle \langle \text{"FindPredecessor"}, 6, 0 \rangle \rangle @@ (1:> \langle \rangle) @@ (3:> \langle \rangle))$   
 $\wedge triggered = \text{FALSE}$   
 $\wedge fingerTables =$   
 $\quad (0:> ((1:> 1) @@ (2:> 3) @@ (4:> 0)))$   
 $\quad @@ (1:> ((2:> 3) @@ (3:> 5) @@ (5:> 0)))$   
 $\quad @@ (3:> ((4:> 0) @@ (5:> 0) @@ (7:> 0)))$   
Process actor  
 $\wedge currentMessage = [self \in \{0, 1, 3\} \mapsto \langle \text{"?"}, -1, -1 \rangle]$   
 $\wedge kind = [self \in \{0, 1, 3\} \mapsto \text{"?"}]$   
 $\wedge id = [self \in \{0, 1, 3\} \mapsto -1]$   
 $\wedge asker = [self \in \{0, 1, 3\} \mapsto -1]$   
 $\wedge i = [self \in \{0, 1, 3\} \mapsto defaultInitValue]$   
 $\wedge pc = [self \in ProcSet \mapsto \text{"WaitForMessages"}]$

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

$ProcessMessage(self) \triangleq$   $\wedge pc[self] = \text{"ProcessMessage"}$   
 $\wedge \text{IF } kind[self] = \text{"FindPredecessor"}$   
 $\quad \text{THEN } \wedge id' = [id \text{ EXCEPT } ![self] = currentMessage[self][2]]$   
 $\quad \wedge asker' = [asker \text{ EXCEPT } ![self] = currentMessage[self][3]]$   
 $\quad \wedge \text{IF } between01(self, id'[self], fingerTables[self][(self + 1) \% bm])$   
 $\quad \quad \text{THEN } \wedge actorInboxes' = [actorInboxes \text{ EXCEPT } ![asker'[self]] = A]$   
 $\quad \quad \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"DefaultsBack"}]$   
 $\quad \quad \wedge i' = i$   
 $\quad \quad \text{ELSE } \wedge i' = [i \text{ EXCEPT } ![self] = m]$   
 $\quad \quad \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"FindFirstSuitable"}]$   
 $\quad \quad \wedge \text{UNCHANGED } actorInboxes$   
 $\quad \wedge \text{UNCHANGED } triggered$   
 $\quad \text{ELSE } \wedge \text{IF } kind[self] = \text{"Predecessor"} \wedge currentMessage[self][2] = 3$   
 $\quad \quad \text{THEN } \wedge triggered' = \text{TRUE}$   
 $\quad \quad \text{ELSE } \wedge \text{TRUE}$   
 $\quad \quad \wedge \text{UNCHANGED } triggered$   
 $\quad \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"DefaultsBack"}]$   
 $\quad \wedge \text{UNCHANGED } \langle actorInboxes, id, asker, i \rangle$   
 $\wedge \text{UNCHANGED } \langle fingerTables, currentMessage, kind \rangle$

$$\begin{aligned}
FindFirstSuitableI(self) &\triangleq \wedge pc[self] = \text{"FindFirstSuitableI"} \\
&\wedge \text{IF } i[self] > 0 \wedge \neg((self + (2^{(i[self]-1)})) \% 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, currentMessage, kind, \\
&\quad \quad \quad id, asker \rangle \\
MainLoop(self) &\triangleq \wedge pc[self] = \text{"MainLoop"} \\
&\wedge \text{IF } i[self] > 0 \wedge \neg(between00(self, fingerTables[self][(self + (2^{(i[self]-1)})) \% bm], id[self])) \\
&\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][(self + (2^{(i[self]-1)})) \% bm], id[self])]] \\
&\quad \quad \text{ELSE } \wedge actorInboxes' = [actorInboxes \text{ EXCEPT } ![fingerTables[self][(self + (2^{(i[self]-1)})) \% bm], id[self])]] \\
&\quad \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"DefaultsBack"}] \\
&\quad \wedge i' = i \\
&\wedge \text{UNCHANGED } \langle triggered, fingerTables, currentMessage, \\
&\quad \quad \quad kind, id, asker \rangle \\
FindSuitableI(self) &\triangleq \wedge pc[self] = \text{"FindSuitableI"} \\
&\wedge \text{IF } i[self] > 0 \wedge \neg((self + (2^{(i[self]-1)})) \% 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 currentMessage, kind, id, asker \rangle \\
DefaultsBack(self) &\triangleq \wedge pc[self] = \text{"DefaultsBack"} \\
&\wedge currentMessage' = [currentMessage \text{ EXCEPT } ![self] = \langle "?", -1, -1 \rangle] \\
&\wedge kind' = [kind \text{ EXCEPT } ![self] = "?"] \\
&\wedge id' = [id \text{ EXCEPT } ![self] = -1] \\
&\wedge asker' = [asker \text{ EXCEPT } ![self] = -1] \\
&\wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"WaitForMessages"}] \\
&\wedge \text{UNCHANGED } \langle actorInboxes, triggered, fingerTables, i \rangle \\
actor(self) &\triangleq WaitForMessages(self) \vee ProcessMessage(self) \\
&\quad \vee FindFirstSuitableI(self) \vee MainLoop(self) \\
&\quad \vee FindSuitableI(self) \vee DefaultsBack(self) \\
Next &\triangleq (\exists self \in \{0, 1, 3\} : actor(self)) \\
Spec &\triangleq \wedge Init \wedge \Box[Next]_{vars}
\end{aligned}$$

$$\begin{aligned} & \wedge \text{WF}_{vars}(Next) \\ & \wedge \forall self \in \{0, 1, 3\} : \text{WF}_{vars}(actor(self)) \wedge \text{SF}_{vars}(WaitForMessages(self)) \end{aligned}$$

END TRANSLATION

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

$$Liveness \triangleq \Diamond \Box Triggered$$

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

---

\ \* Modification History  
\ \* Last modified Sun Feb 20 18:05:45 YEKT 2022 by pervu  
\ \* Created Sun Jan 30 18:34:11 YEKT 2022 by pervu