```
1  ──────────────────── MODULE Fischer1 ────────────────────
2  EXTENDS FischerPreface
3  ──────────────────────────────────────────────────────────
4  SetTimer(t, timer, tau) ≜
5      timer' = [timer EXCEPT ![t] = tau]
6
7  ResetUBTimer(t, timer) ≜
8      SetTimer(t, timer, Infinity)
9  ──────────────────────────────────────────────────────────
10 NCS(t) ≜
11       ∧ GoFromTo(t, "ncs", "a")
12       ∧ UNCHANGED ⟨x, now, lbTimer, ubTimer, counter⟩
13
14 StmtA(t) ≜
15     ∧ x = NotAThread
16     ∧ GoFromTo(t, "a", "b")
17     ∧ SetTimer(t, ubTimer, Delta)
18     ∧ UNCHANGED ⟨x, now, lbTimer, counter⟩
19
20 StmtB(t) ≜
21     ∧ x' = t
22     ∧ GoFromTo(t, "b", "c")
23     ∧ ResetUBTimer(t, ubTimer)
24     ∧ SetTimer(t, lbTimer, Epsilon)
25     ∧ UNCHANGED ⟨now, counter⟩
26
27 StmtC(t) ≜
28     ∧ At(t, "c")
29     ∧ TimedOut(t, lbTimer)
30     ∧ IF x ≠ t THEN GoTo(t, "a") ELSE GoTo(t, "cs")
31     ∧ UNCHANGED ⟨x, now, lbTimer, ubTimer, counter⟩
32
33 CS(t) ≜
34       ∧ GoFromTo(t, "cs", "d")
35       ∧ counter' = [counter EXCEPT ![t] = @ + 1]
36       ∧ UNCHANGED ⟨x, now, lbTimer, ubTimer⟩
37
38 StmtD(t) ≜
39     ∧ x' = NotAThread
40     ∧ GoFromTo(t, "d", "ncs")
41     ∧ UNCHANGED ⟨now, lbTimer, ubTimer, counter⟩
42
43 Tick ≜
44     ∃ d ∈ Real :
45         ∧ d > 0
46         ∧ ∀ t ∈ Thread :
47             ubTimer[t] ≠ Infinity ⇒ ubTimer[t] > d
```

```
        ∧ now′ = now + d    Where is now used in the spec?
        ∧ ubTimer′ = [t ∈ Thread ↦
                        IF ubTimer[t] = Infinity THEN Infinity
                                                 ELSE ubTimer[t] − d]
        ∧ lbTimer′ = [t ∈ Thread ↦ Max(0, lbTimer[t] − d)]
        ∧ UNCHANGED ⟨x, pc, counter⟩
------------------------------------------------------------------------
Next ≜
    ∨ Tick
    ∨ ∃ t ∈ Thread :
        ∨ NCS(t)
        ∨ StmtA(t) ∨ StmtB(t) ∨ StmtC(t)
        ∨ CS(t)
        ∨ StmtD(t)
------------------------------------------------------------------------
SafetySpec ≜ Init ∧     □[Next]_vars

THEOREM SafetySpec ⇒ □MutualExclusion
------------------------------------------------------------------------
Liveness ≜
    ∧ ∀ t ∈ Thread : WF_vars(StmtA(t) ∨ StmtC(t) ∨ StmtD(t))
    ∧ ∀ r ∈ Real : ◇(now > r)

FSpec1 ≜ SafetySpec ∧ Liveness

Progress ≜
    (∃ t ∈ Thread : At(t, "a") ∨ At(t, "b") ∨ At(t, "c")) ⇝
      (∃ t ∈ Thread : At(t, "cs"))

THEOREM FSpec1 ⇒ Progress
========================================================================
\* Modification History
\* Last modified Fri Aug 06 10:26:36 CST 2021 by hengxin
\* Created Wed Aug 04 16:13:33 CST 2021 by hengxin
```