```
1  ---------- MODULE CC ----------
   TLA+ specification of Causal Consistency variants, including CC, CM, and CCv.
   See the paper "On Verifying Causal Consistency" (POPL'2017).
8  EXTENDS TLC, Naturals, Sequences, FiniteSets, FiniteSetsExt, FunctionUtils

10 CONSTANTS Keys, Vals
11 InitVal ≜ CHOOSE v : v ∉ (Keys ∪ Vals)

13  oid: unique operation identifier
14 Operation ≜ [op : {"read", "write"}, key : Keys, val : Vals, oid : Nat]
15 R(k, v, oid) ≜ [op ↦ "read", key ↦ k, val ↦ v, oid ↦ oid]
16 W(k, v, oid) ≜ [op ↦ "write", key ↦ k, val ↦ v, oid ↦ oid]

18 Session ≜ Seq(Operation)  A session s ∈ Session is a sequence of operations.
19 History ≜ SUBSET Session  A history h ∈ History is a set of sessions.
20 ----------

   Utilities.
24 Ops(h) ≜  Return the set of all operations in history h ∈ History.
25   UNION {Range(s) : s ∈ h}
26 ----------

   Well-formedness of history h ∈ History:
   - type invariants
   - uniqueness of oids
33 WellFormed(h) ≜
34  ∧ h ∈ History
35   ∧ Cardinality(Ops(h)) = Sum({Len(s) : s ∈ h})
36 ----------

   Test case: The following histories are from Figure 2 of the POPL'2017 paper.
   Naming:
   – ha: history of Figure 2(a)
   – hasa: session a of history ha
45 hasa ≜ ⟨W("x", 1, 1), R("x", 2, 2)⟩
46 hasb ≜ ⟨W("x", 2, 3), R("x", 1, 4)⟩
47 ha ≜ {hasa, hasb}  CM but not CCv

49 hbsa ≜ ⟨W("z", 1, 1), W("x", 1, 2), W("y", 1, 3)⟩
50 hbsb ≜ ⟨W("x", 2, 4), R("z", 0, 5), R("y", 1, 6), R("x", 2, 7)⟩
51 hb ≜ {hbsa, hbsb}  CCv but not CM

53 hcsa ≜ ⟨W("x", 1, 1)⟩
54 hcsb ≜ ⟨W("x", 2, 2), R("x", 1, 3), R("x", 2, 4)⟩
55 hc ≜ {hcsa, hcsb}  CC but not CM nor CCv

57 hdsa ≜ ⟨W("x", 1, 1), R("y", 0, 2), W("y", 1, 3), R("x", 1, 4)⟩
58 hdsb ≜ ⟨W("x", 2, 5), R("y", 0, 6), W("y", 2, 7), R("x", 2, 8)⟩
```

```
59  hd ≜ {hdsa, hdsb}  CC, CM, and CCv but no SC

61  hesa ≜ ⟨W("x", 1, 1), W("y", 1, 2)⟩
62  hesb ≜ ⟨R("y", 1, 3), W("x", 2, 4)⟩
63  hesc ≜ ⟨R("x", 2, 5), R("x", 1, 6)⟩
64  he ≜ {hesa, hesb, hesc}  not CC (nor CM, nor CCv)
65 ├──────────────────────────────────────────────────────────────────────┤
     Program order: a union of total orders among operations in the same session.
69  ProgramOrder(h) ≜
70    LET RECURSIVE SessionProgramOrder(_)
71        SessionProgramOrder(s) ≜
72          IF s = ⟨⟩ THEN {}
73           ELSE LET sh ≜ Head(s)
74                    st ≜ Tail(s)
75                IN  {⟨sh, t⟩ : t ∈ Range(st)} ∪ SessionProgramOrder(st)
76    IN  UNION {SessionProgramOrder(s) : s ∈ h}

     Test case: TODO: Cardinality testing
81   CardOfProgramOrderOf(h) ≜
82   THEOREM CardOfProgramOrderTheorem ≜
83     ∀ h ∈ {ha, hb, hc, hd, he}:
84       Cardinality(ProgramOrder(h)) = CardOfProgramOrderOf(h)
85 ├──────────────────────────────────────────────────────────────────────┤
     Sequential semantics of read-write registers.
89 ├──────────────────────────────────────────────────────────────────────┤
     Specification of Causal Consistency: CC, CCv, and CM
93  CCv(h) ≜  Check whether h ∈ History satisfies CCv (Causal Convergence)
94    ∧ LET ops ≜ Ops(h)
95      IN  ∧ ∃ co ∈ SUBSET (ops × ops) :
96              ∃ arb ∈ SUBSET (ops × ops) :
97                ∀ op ∈ ops : TRUE
98    ∧ FALSE
99 └──────────────────────────────────────────────────────────────────────┘
```