```
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 Naturals, Sequences, FiniteSets, Functions, FiniteSetsExt,
9          RelationUtils, TLC

11 CONSTANTS Keys, Vals
12 InitVal ≜ 0  we follow the convention in POPL'2017

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

19 Session ≜ Seq(Operation)  A session s ∈ Session is a sequence of operations.
20 History ≜ SUBSET Session  A history h ∈ History is a set of sessions.
21 ----------
   Utilities.
25 Ops(h) ≜  Return the set of all operations in history h ∈ History.
26   UNION {Range(s) : s ∈ h}
27 ----------
   Well-formedness of history h ∈ History:
   – TODO: type invariants
   - uniqueness of oids
34 WellFormed(h) ≜
35  ∧ h ∈ History
36   ∧ Cardinality(Ops(h)) = ReduceSet(LAMBDA s, x : Len(s) + x, h, 0)
37 ----------
   Sequential semantics of read-write registers.
41 ----------
   Auxiliary definitions for the axioms used in the definitions of causal consistency
45  The program order of h ∈ History is a union of total orders among operations in the same session
46 ProgramOrder(h) ≜ UNION {Seq2Rel(s) : s ∈ h}

48  The set of operations that preceed o ∈ Operation in program order in history h ∈ History
49 POPast(h, o) ≜ InverseImage(ProgramOrder(h), o)

51  The set of operations that preceed o ∈ Operation in causal order co
52 CausalPast(co, o) ≜ InverseImage(co, o)

54  The restriction of arbitration arb to the operations in the causal past of operation o ∈ Operation
55 CausalArb(co, arb, o) ≜ arb | CausalPast(co, o)
56 ----------
   Axioms used in the defintions of causal consistency
60 AxCausalArb(co, arb, o) ≜
```

```
61      LET seq ≜ AnyLinearExtension(CausalArb(co, arb, o), CausalPast(co, o))  it is unique
62          wseq ≜ SelectSeq(seq, LAMBDA op : op.type = "write" ∧ op.key = o.key)
63      IN    IF wseq = ⟨⟩ THEN o.val = InitVal
64            ELSE  o.val = wseq[Len(wseq)].val
65 ├──────────────────────────────────────────────────────────────────────────┤
   Specification of Causal Consistency: CC, CCv, and CM

   To generate possible ordering relations, not to enumerate and test them
73 CCv(h) ≜  Check whether h ∈ History satisfies CCv (Causal Convergence)
74      LET ops ≜ Ops(h)
75      IN  ∃ co ∈ SUBSET (ops × ops) :  TODO: to generate (given a chain decomposition)
76            ∧ Respect(co, ProgramOrder(h))                    AxCausal
77            ∧ IsStrictPartialOrder(co, ops)
78            ∧ PrintT("co: " ∘ ToString(co))
79            ∧ ∃ arb ∈ {Seq2Rel(le) : le ∈ AllLinearExtensions(co, ops)} :  AxArb
80                  ∧ ∀ o ∈ ops : AxCausalArb(co, arb, o)  AxCausalArb
81                  ∧ PrintT("arb: " ∘ ToString(arb))
   Version 2: re-arrange clauses
85 CCv2(h) ≜  Check whether h ∈ History satisfies CCv (Causal Convergence)
86      LET ops ≜ Ops(h)
87      IN  ∃ co ∈ SUBSET (ops × ops) :  FIXME: efficiency!!!
88            ∧ Respect(co, ProgramOrder(h))  AxCausal
89            ∧ IsStrictPartialOrder(co, ops)
90            ∧ PrintT("co: " ∘ ToString(co))
91            ∧ ∃ arb ∈ SUBSET (ops × ops) :  to generate; not to test
92                  ∧ Respect(arb, co)                          AxArb
93                  ∧ IsStrictTotalOrder(arb, ops)
94                  ∧ ∀ o ∈ ops : AxCausalArb(co, arb, o)  AxCausalArb
95                  ∧ PrintT("arb: " ∘ ToString(arb))
   Version 1: Following the definition of POPL2017
99 CCv1(h) ≜  Check whether h ∈ History satisfies CCv (Causal Convergence)
100     LET ops ≜ Ops(h)
101     IN  ∃ co ∈ SUBSET (ops × ops) :  FIXME: efficiency!!!
102           ∧ ∃ arb ∈ SUBSET (ops × ops) :
103               ∧ PrintT("co: " ∘ ToString(co))
104               ∧ PrintT("arb: " ∘ ToString(arb))
105               ∧ IsStrictPartialOrder(co, ops)
106               ∧ IsStrictTotalOrder(arb, ops)
107               ∧ Respect(co, ProgramOrder(h))              AxCausal
108               ∧ Respect(arb, co)                          AxArb
109               ∧ ∀ o ∈ ops : AxCausalArb(co, arb, o)       AxCausalArb
110 └──────────────────────────────────────────────────────────────────────────┘
   \* Modification History
   \* Last modified Tue Apr 13 09:07:52 CST 2021 by hengxin
```

\ * Created *Tue Apr* 01 10:24:07 *CST* 2021 by *hengxin*