```
MODULE CC
```

TLA+ specification of Causal Consistency variants, including *CC*, *CM*, and *CCv*.

See the paper "On Verifying Causal Consistency" (*POPL*'2017).

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

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

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

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

Utilities.

```
Ops(h) ≜   Return the set of all operations in history h ∈ History.
  UNION {Range(s) : s ∈ h}
```

Well-formedness of history $h \in History$:

- *TODO*: type invariants
- uniqueness of oids

```
WellFormed(h) ≜
 ∧ h ∈ History
 ∧ Cardinality(Ops(h)) = ReduceSet(LAMBDA s, x : Len(s) + x, h, 0)
```

Sequential semantics of read-write registers.

Auxiliary definitions for the axioms used in the definitions of causal consistency

```
 The program order of h ∈ History is a union of total orders among operations in the same session
ProgramOrder(h) ≜ UNION {Seq2Rel(s) : s ∈ h}

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

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

 The restriction of causal order co to the operations in the causal past of operation o ∈ Operation
CausalHist(co, o) ≜ co | CausalPast(co, o)

 The restriction of arbitration arb to the operations in the causal past of operation o ∈ Operation
CausalArb(co, arb, o) ≜ arb | CausalPast(co, o)
```

Axioms used in the defintions of causal consistency

$AxCausalValue(co,\ o) \triangleq$
  LET $seqs \triangleq AllLinearExtensions(CausalHist(co,\ o),\ CausalPast(co,\ o))$
  IN TRUE
$AxCausalArb(co,\ arb,\ o) \triangleq$
  LET $seq \triangleq AnyLinearExtension(CausalArb(co,\ arb,\ o),\ CausalPast(co,\ o))$ it is unique
    $wseq \triangleq SelectSeq(seq,$ LAMBDA $op : op.type = \text{“write”} \land op.key = o.key)$
  IN IF $wseq = \langle\rangle$ THEN $o.val = InitVal$
    ELSE $o.val = wseq[Len(wseq)].val$

Specification of *CC*

$CC(h) \triangleq$ Check whether $h \in History$ satisfies *CC* (Causal Consistency)
  LET $ops \triangleq Ops(h)$
  IN $\exists\, co \in$ SUBSET $(ops \times ops) :$ *TODO*: to generate (given a chain decomposition)
    $\land\ Respect(co,\ ProgramOrder(h))$ *AxCausal*
    $\land\ IsStrictPartialOrder(co,\ ops)$
    $\land\ PrintT(\text{“co: ”} \circ ToString(co))$
    $\land\ \forall\, o \in ops : AxCausalValue(co,\ o)$ *AxCausalValue*

Specification of *CCv*

To generate possible ordering relations, not to enumerate and test them

$CCv(h) \triangleq$ Check whether $h \in History$ satisfies *CCv* (Causal Convergence)
  LET $ops \triangleq Ops(h)$
  IN $\exists\, co \in$ SUBSET $(ops \times ops) :$ *TODO*: to generate (given a chain decomposition)
    $\land\ Respect(co,\ ProgramOrder(h))$ *AxCausal*
    $\land\ IsStrictPartialOrder(co,\ ops)$
    $\land\ PrintT(\text{“co: ”} \circ ToString(co))$
    $\land\ \exists\, arb \in \{Seq2Rel(le) : le \in AllLinearExtensions(co,\ ops)\} :$ *AxArb*
      $\land\ \forall\, o \in ops : AxCausalArb(co,\ arb,\ o)$ *AxCausalArb*
      $\land\ PrintT(\text{“arb: ”} \circ ToString(arb))$

Version 2: re-arrange clauses

$CCv2(h) \triangleq$ Check whether $h \in History$ satisfies *CCv* (Causal Convergence)
  LET $ops \triangleq Ops(h)$
  IN $\exists\, co \in$ SUBSET $(ops \times ops) :$ *FIXME*: efficiency!!!
    $\land\ Respect(co,\ ProgramOrder(h))$ *AxCausal*
    $\land\ IsStrictPartialOrder(co,\ ops)$
    $\land\ PrintT(\text{“co: ”} \circ ToString(co))$
    $\land\ \exists\, arb \in$ SUBSET $(ops \times ops) :$ to generate; not to test
      $\land\ Respect(arb,\ co)$ *AxArb*
      $\land\ IsStrictTotalOrder(arb,\ ops)$
      $\land\ \forall\, o \in ops : AxCausalArb(co,\ arb,\ o)$ *AxCausalArb*
      $\land\ PrintT(\text{“arb: ”} \circ ToString(arb))$

Version 1: Following the definition of *POPL*2017

```
116 CCv1(h) ≜  Check whether h ∈ History satisfies CCv (Causal Convergence)
117         LET ops ≜ Ops(h)
118         IN  ∃ co ∈ SUBSET (ops × ops) :  FIXME: efficiency!!!
119                 ∧ ∃ arb ∈ SUBSET (ops × ops) :
120                     ∧ PrintT("co: " ∘ ToString(co))
121                     ∧ PrintT("arb: " ∘ ToString(arb))
122                     ∧ IsStrictPartialOrder(co, ops)
123                     ∧ IsStrictTotalOrder(arb, ops)
124                     ∧ Respect(co, ProgramOrder(h))          AxCausal
125                     ∧ Respect(arb, co)                      AxArb
126                     ∧ ∀ o ∈ ops : AxCausalArb(co, arb, o)   AxCausalArb
127 ╚═══════════════════════════════════════════════════════════════════════
```

* Modification *History*
* Last modified Sat *Apr* 17 19:30:51 *CST* 2021 by *hengxin*
* Created *Tue Apr* 01 10:24:07 *CST* 2021 by *hengxin*