

```

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  $\triangleq$  0 | we follow the convention in POPL'2017 |

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

19  Session  $\triangleq$  Seq(Operation) | A session s  $\in$  Session is a sequence of operations.
20  History  $\triangleq$  SUBSET Session | A history h  $\in$  History is a set of sessions.
21  |-----|
   | Utilities. |
25  Ops(h)  $\triangleq$  | Return the set of all operations in history h  $\in$  History.
26      UNION { Range(s) : s  $\in$  h }

28  ReadOps(h)  $\triangleq$  | Return the set of all read operations in history h  $\in$  History.
29      { op  $\in$  Ops(h) : op.type = "read" }

31  ReadOpsOnKey(h, k)  $\triangleq$  | Return the set of all read operations on key k  $\in$  Keys in history h  $\in$  History.
32      { op  $\in$  Ops(h) : op.type = "read"  $\wedge$  op.key = k }

34  WriteOps(h)  $\triangleq$  | Return the set of all write operations in history h  $\in$  History.
35      { op  $\in$  Ops(h) : op.type = "write" }

37  WriteOpsOnKey(h, k)  $\triangleq$  | Return the set of all write operations on key k  $\in$  Keys in history h  $\in$  History.
38      { op  $\in$  Ops(h) : op.type = "write"  $\wedge$  op.key = k }
39  |-----|
   | Well-formedness of history h  $\in$  History: |
   | - TODO: type invariants |
   | - uniqueness of oids |
46  WellFormed(h)  $\triangleq$ 
47       $\wedge$  h  $\in$  History
48       $\wedge$  Cardinality(Ops(h)) = ReduceSet(LAMBDA s, x : Len(s) + x, h, 0)
49  |-----|
   | Auxiliary definitions for the axioms used in the definitions of causal consistency |
53  | The program order of h  $\in$  History is a union of total orders among operations in the same session |
54  PO(h)  $\triangleq$  UNION { Seq2Rel(s) : s  $\in$  h }

56  | The set of operations that precede o  $\in$  Operation in program order in history h  $\in$  History |
57  POPast(h, o)  $\triangleq$  InverseImage(PO(h), o)

```

```

59  The set of operations that precede  $o \in \text{Operation}$  in causal order  $co$ 
60   $\text{CausalPast}(co, o) \triangleq \text{InverseImage}(co, o)$ 

62  The restriction of causal order  $co$  to the operations in the causal past of operation  $o \in \text{Operation}$ 
63   $\text{CausalHist}(co, o) \triangleq co \mid \text{CausalPast}(co, o)$ 

65  The restriction of arbitration  $arb$  to the operations in the causal past of operation  $o \in \text{Operation}$ 
66   $\text{CausalArb}(co, arb, o) \triangleq arb \mid \text{CausalPast}(co, o)$ 
67  |-----|
    Axioms used in the definitions of causal consistency

71   $\text{RWRegSemantics}(seq, o) \triangleq$  Is  $o \in \text{Operation}$  legal when it is appended to  $seq$ 
72    IF  $o.type = \text{"write"}$  THEN TRUE
73    ELSE LET  $wseq \triangleq \text{SelectSeq}(seq, \text{LAMBDA } op : op.type = \text{"write"} \wedge op.key = o.key)$ 
74        IN IF  $wseq = \langle \rangle$  THEN  $o.val = \text{InitVal}$ 
75        ELSE  $o.val = wseq[\text{Len}(wseq)].val$ 

77   $\text{AxCausalValue}(co, o) \triangleq$ 
78    LET  $seqs \triangleq \text{AllLinearExtensions}(\text{CausalHist}(co, o), \text{CausalPast}(co, o))$ 
79    IN TRUE  $\in \{\text{RWRegSemantics}(seq, o) : seq \in seqs\}$  TODO: shortcut implementation of anyTrue for efficiency

81   $\text{AxCausalArb}(co, arb, o) \triangleq$ 
82    LET  $seq \triangleq \text{AnyLinearExtension}(\text{CausalArb}(co, arb, o), \text{CausalPast}(co, o))$  it is unique
83    IN  $\text{RWRegSemantics}(seq, o)$ 
84  |-----|
    Specification of  $CC$ 

88   $CC(h) \triangleq$  Check whether  $h \in \text{History}$  satisfies  $CC$  (Causal Consistency)
89    LET  $ops \triangleq \text{Ops}(h)$ 
90    IN  $\exists co \in \text{SUBSET}(ops \times ops) :$  TODO: to generate (given a chain decomposition)
91         $\wedge \text{Respect}(co, PO(h))$  AxCausal
92         $\wedge \text{IsStrictPartialOrder}(co, ops)$ 
93         $\wedge \text{PrintT}(\text{"co: " } \circ \text{ToString}(co))$ 
94         $\wedge \forall o \in ops : \text{AxCausalValue}(co, o)$  AxCausalValue
95  |-----|
    Specification of  $CCv$ 

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

103  $CCv(h) \triangleq$  Check whether  $h \in \text{History}$  satisfies  $CCv$  (Causal Convergence)
104    LET  $ops \triangleq \text{Ops}(h)$ 
105    IN  $\exists co \in \text{SUBSET}(ops \times ops) :$  TODO: to generate (given a chain decomposition)
106         $\wedge \text{Respect}(co, PO(h))$  AxCausal
107         $\wedge \text{IsStrictPartialOrder}(co, ops)$ 
108         $\wedge \text{PrintT}(\text{"co: " } \circ \text{ToString}(co))$ 
109         $\wedge \exists arb \in \{\text{Seq2Rel}(le) : le \in \text{AllLinearExtensions}(co, ops)\} :$  AxArb
110         $\wedge \forall o \in ops : \text{AxCausalArb}(co, arb, o)$  AxCausalArb
111         $\wedge \text{PrintT}(\text{"arb: " } \circ \text{ToString}(arb))$ 

```

```

Version 2: re-arrange clauses
115  $CCv2(h) \triangleq$  Check whether  $h \in History$  satisfies  $CCv$  (Causal Convergence)
116   LET  $ops \triangleq Ops(h)$ 
117   IN  $\exists co \in SUBSET (ops \times ops) :$  FIXME: efficiency!!!
118      $\wedge Respect(co, PO(h))$  AxCausal
119      $\wedge IsStrictPartialOrder(co, ops)$ 
120      $\wedge PrintT("co: " \circ ToString(co))$ 
121      $\wedge \exists arb \in SUBSET (ops \times ops) :$  to generate; not to test
122        $\wedge Respect(arb, co)$  AxArb
123        $\wedge IsStrictTotalOrder(arb, ops)$ 
124        $\wedge \forall o \in ops : AxCausalArb(co, arb, o)$  AxCausalArb
125        $\wedge PrintT("arb: " \circ ToString(arb))$ 

Version 1: Following the definition of POPL2017
129  $CCv1(h) \triangleq$  Check whether  $h \in History$  satisfies  $CCv$  (Causal Convergence)
130   LET  $ops \triangleq Ops(h)$ 
131   IN  $\exists co \in SUBSET (ops \times ops) :$  FIXME: efficiency!!!
132      $\wedge \exists arb \in SUBSET (ops \times ops) :$ 
133        $\wedge PrintT("co: " \circ ToString(co))$ 
134        $\wedge PrintT("arb: " \circ ToString(arb))$ 
135        $\wedge IsStrictPartialOrder(co, ops)$ 
136        $\wedge IsStrictTotalOrder(arb, ops)$ 
137        $\wedge Respect(co, PO(h))$  AxCausal
138        $\wedge Respect(arb, co)$  AxArb
139        $\wedge \forall o \in ops : AxCausalArb(co, arb, o)$  AxCausalArb
140 |-----|

Specification of CM
144  $CM(h) \triangleq$  Check whether  $h \in History$  satisfies  $CM$  (Causal Memory)
145   FALSE TODO
146 |-----|

The checking algorithms in POPL'2017.
150  $IsDifferentiated(h) \triangleq$  Is  $h \in History$  differentiated?
151    $\forall k \in Keys :$ 
152     LET  $writes \triangleq WriteOpsOnKey(h, k)$ 
153     IN  $\forall w1 \in writes, w2 \in writes :$ 
154        $\wedge w1.val \neq w2.val$ 
155        $\wedge w1.val \neq InitVal$ 

157  $RF(h) \triangleq$  the read-from relation
158    $\{\langle w, r \rangle \in WriteOps(h) \times ReadOps(h) : w.key = r.key \wedge w.val = r.val\}$ 

160  $CO(h) \triangleq$  the  $CO$  order defined as the transitive closure of the union of  $PO(h)$  and  $RF(h)$ 
161    $TC(PO(h) \cup RF(h))$ 

All bad patterns defined in POPL'2017

```

```

165  $CyclicCO(h) \triangleq Cyclic(PO(h) \cup RF(h))$ 
167  $WriteCOInitRead(h) \triangleq$ 
168    $\exists k \in Keys :$ 
169      $\exists r \in ReadOpsOnKey(h, k), w \in WriteOpsOnKey(h, k) :$ 
170        $\wedge \langle w, r \rangle \in CO(h)$  TODO: for efficiency
171        $\wedge r.val = InitVal$ 
173  $ThinAirRead(h) \triangleq$ 
174    $\exists k \in Keys :$ 
175      $\exists r \in ReadOpsOnKey(h, k) :$ 
176        $\wedge r.val \neq InitVal$ 
177        $\wedge \forall w \in WriteOpsOnKey(h, k) : \langle w, r \rangle \notin RF(h)$ 
179  $WriteCORead(h) \triangleq$ 
180    $\exists k \in Keys :$ 
181      $\exists w1, w2 \in WriteOpsOnKey(h, k), r1 \in ReadOpsOnKey(h, k) :$ 
182        $\wedge \langle w1, w2 \rangle \in CO(h)$ 
183        $\wedge \langle w2, r1 \rangle \in CO(h)$  TODO: efficiency
184        $\wedge \langle w1, r1 \rangle \in RF(h)$ 
186  $CyclicHB(h) \triangleq$  TODO:
187   FALSE
189  $CyclicCF(h) \triangleq$  TODO:
190   FALSE
191    $Cyclic(CF(h) \cup CO(h))$ 
192
```

\\* Modification *History*

\\* Last modified *Mon Apr 19 19:49:35 CST 2021* by *hengxin*

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