

```

1  ┌────────────────────────── MODULE CCAlg ───────────────────────────┐
    │ TLA+ Checking Algorithm of Causal Consistency variants, including CC, CM, and CCv.  

    │ See the paper “On Verifying Causal Consistency” (POPL’2017).  

8  EXTENDS CC
    │ Auxiliary operators used in the checking algorithms: We consider only differentiated histories.
14 KeyOf(h)  $\triangleq$  the set of keys read or written in h  $\in$  History
15   {op.key : op  $\in$  Ops(h)}
17 IsDifferentiated(h)  $\triangleq$  Is h  $\in$  History differentiated?
18    $\forall k \in \text{KeyOf}(h) :$ 
19     LET writes  $\triangleq$  WriteOpsOnKey(h, k)
20     IN  $\forall w1 \in \text{writes}, w2 \in \text{writes} :$ 
21        $\wedge w1.val \neq w2.val$ 
22        $\wedge w1.val \neq \text{InitVal}$ 
    │ Auxiliary relations used in the checking algorithms
26 RF(h)  $\triangleq$  the read-from relation TODO: using infix symbolic operator???
27   { $\langle w, r \rangle \in \text{WriteOps}(h) \times \text{ReadOps}(h) : w.key = r.key \wedge w.val = r.val$ }
29 CO(h)  $\triangleq$  the CO order defined as the transitive closure of the union of PO(h) and RF(h)
30   TC(PO(h)  $\cup$  RF(h))
32 CF(h)  $\triangleq$  the conflict relation
33   LET co  $\triangleq$  CO(h)
34   rf  $\triangleq$  RF(h)
35   reads  $\triangleq$  ReadOps(h)
36   writes  $\triangleq$  WriteOps(h)
37   IN { $\langle w1, w2 \rangle \in \text{writes} \times \text{writes} :$ 
38      $\wedge w1.key = w2.key$ 
39      $\wedge w1.val \neq w2.val$ 
40      $\wedge \exists r \in \text{reads} : \langle w1, r \rangle \in co \wedge \langle w2, r \rangle \in rf$ }
42 HB(h)  $\triangleq$  \ * All of the happened-before relation of operation o in history h
44 BaseHB(h, o)  $\triangleq$  CO | CasualPast(o)
45   LET co  $\triangleq$  CO(h)
46   IN co | CausalPast(co, o)
48 HBo(h, o)  $\triangleq$  Happened-before relation for o, denoted HBo  $\subseteq O \times O$ , to be the smallest relation such that
49   LET po  $\triangleq$  PO(h)
50   writes  $\triangleq$  WriteOps(h)
51   base  $\triangleq$  BaseHB(h, o) CO | CasualPast(o)  $\subseteq$  HBo
52   RECURSIVE HBoRE(-)
53   HBoRE(hbo)  $\triangleq$ 
54     LET update  $\triangleq$  {
55        $\langle w1, w2 \rangle \in \text{writes} \times \text{writes} :$ 

```

```

56       $\wedge w1.key = w2.key$ 
57       $\wedge w1.val \neq w2.val$ 
58       $\wedge \exists r2 \in ReadOpsOnKey(h, w2.key) :$ 
59           $\wedge r2.val = w2.val$ 
60           $\wedge \langle w1, r2 \rangle \in hbo$ 
61           $\wedge \vee r2 = o$ 
62           $\vee \langle r2, o \rangle \in po$ 
63       $\}$ 
64       $hbo2 \triangleq update \cup hbo$ 
65      IN IF  $hbo2 = hbo$ 
66          THEN  $hbo$ 
67          ELSE  $HBoRE(TC(hbo2))$ 
68      IN  $TC(HBoRE(base))$ 

```

```

70  $HB(h) \triangleq$  All happened-before relation for  $o \in$  history  $h$ 
71  $\{\langle o, HBo(h, o) \rangle : o \in Ops(h)\}$ 

```

All bad patterns defined in *POPL'2017* (see Table 2 of *POPL'2017*)

```

77  $CyclicCO(h) \triangleq Cyclic(PO(h) \cup RF(h))$ 

79  $WriteCOInitRead(h) \triangleq$ 
80    $\exists k \in KeyOf(h) :$ 
81      $\exists r \in ReadOpsOnKey(h, k), w \in WriteOpsOnKey(h, k) :$ 
82        $\wedge \langle w, r \rangle \in CO(h)$  TODO: for efficiency
83        $\wedge r.val = InitVal$ 

85  $ThinAirRead(h) \triangleq$ 
86    $\exists k \in KeyOf(h) :$ 
87      $\exists r \in ReadOpsOnKey(h, k) :$ 
88        $\wedge r.val \neq InitVal$ 
89        $\wedge \forall w \in WriteOpsOnKey(h, k) : \langle w, r \rangle \notin RF(h)$ 

91  $WriteCOWrite(h) \triangleq$ 
92    $\exists k \in KeyOf(h) :$ 
93      $\exists w1, w2 \in WriteOpsOnKey(h, k), r1 \in ReadOpsOnKey(h, k) :$ 
94        $\wedge \langle w1, w2 \rangle \in CO(h)$ 
95        $\wedge \langle w2, r1 \rangle \in CO(h)$  TODO: efficiency
96        $\wedge \langle w1, r1 \rangle \in RF(h)$ 

98  $CyclicCF(h) \triangleq$ 
99    $Cyclic(CF(h) \cup CO(h))$ 

101  $WriteHBInitRead(h) \triangleq$ 
102    $\exists o \in Ops(h) :$ 
103     LET  $hbo \triangleq HBo(h, o)$ 
104      $popast \triangleq POPast(h, o)$ 

```

```

105      IN    $\exists r \in \text{popast} :$ 
106           $\wedge r.\text{val} = \text{InitVal}$ 
107           $\wedge \text{LET } \text{writes} \triangleq \text{WriteOpsOnKey}(h, r.\text{key})$ 
108              IN    $\exists w \in \text{writes} :$ 
109                   $\langle w, r \rangle \in \text{hbo}$ 

111  $\text{CyclicHB}(h) \triangleq$ 
112      $\exists o \in \text{Ops}(h) :$ 
113          $\text{Cyclic}(\text{HBo}(h, o))$ 

```

Checking algorithms of *POPL'2017* (see Table 3 of *POPL'2017*)

```

120  $\text{CCAlg}(h) \triangleq$  Checking algorithm for CC (Causal Consistency)
121      $\wedge \neg \text{CyclicCO}(h)$ 
122      $\wedge \neg \text{WriteCOInitRead}(h)$ 
123      $\wedge \neg \text{ThinAirRead}(h)$ 
124      $\wedge \neg \text{WriteCOWrite}(h)$ 

126  $\text{CCvAlg}(h) \triangleq$  Checking algorithm for CCv (Causal Convergence)
127      $\wedge \neg \text{CyclicCO}(h)$ 
128      $\wedge \neg \text{WriteCOInitRead}(h)$ 
129      $\wedge \neg \text{ThinAirRead}(h)$ 
130      $\wedge \neg \text{WriteCOWrite}(h)$ 
131      $\wedge \neg \text{CyclicCF}(h)$ 

133  $\text{CMAlg}(h) \triangleq$  TODO: Checking algorithm for CM (Causal Memory)
134      $\wedge \neg \text{CyclicCO}(h)$ 
135      $\wedge \neg \text{WriteCOInitRead}(h)$ 
136      $\wedge \neg \text{ThinAirRead}(h)$ 
137      $\wedge \neg \text{WriteCOWrite}(h)$ 
138      $\wedge \neg \text{WriteHBInitRead}(h)$ 
139      $\wedge \neg \text{CyclicHB}(h)$ 

```

141 |