

```

1 |----- MODULE CCTest -----|
  | Test of CC Module |
5 | EXTENDS CC |
6 |-----|
  | Test case: The following histories are from Figure 2 of the POPL'2017 paper.
  | Naming Conventions:
  |   - ha: history of Figure 2(a)
  |   - hasa: session a of history ha
  | TODO:
  | - to add more test cases
  | - to automatically generate test cases that do or do not satisfy the specs
  |   - consider Section 3.2 of POPL'2017
  |   - ref: the MonkeyDB paper
23 | hasa  $\triangleq$   $\langle W(\text{"x"}, 1, 1), R(\text{"x"}, 2, 2) \rangle$ 
24 | hasb  $\triangleq$   $\langle W(\text{"x"}, 2, 3), R(\text{"x"}, 1, 4) \rangle$ 
25 | ha  $\triangleq$   $\{hasa, hasb\}$  CM but not CCv
27 | hbsa  $\triangleq$   $\langle W(\text{"z"}, 1, 1), W(\text{"x"}, 1, 2), W(\text{"y"}, 1, 3) \rangle$ 
28 | hbsb  $\triangleq$   $\langle W(\text{"x"}, 2, 4), R(\text{"z"}, 0, 5), R(\text{"y"}, 1, 6), R(\text{"x"}, 2, 7) \rangle$ 
29 | hb  $\triangleq$   $\{hbsa, hbsb\}$  CCv but not CM
31 | hcsa  $\triangleq$   $\langle W(\text{"x"}, 1, 1) \rangle$ 
32 | hcsb  $\triangleq$   $\langle W(\text{"x"}, 2, 2), R(\text{"x"}, 1, 3), R(\text{"x"}, 2, 4) \rangle$ 
33 | hc  $\triangleq$   $\{hcsa, hcsb\}$  CC but not CM nor CCv
35 | hdsa  $\triangleq$   $\langle W(\text{"x"}, 1, 1), R(\text{"y"}, 0, 2), W(\text{"y"}, 1, 3), R(\text{"x"}, 1, 4) \rangle$ 
36 | hdsb  $\triangleq$   $\langle W(\text{"x"}, 2, 5), R(\text{"y"}, 0, 6), W(\text{"y"}, 2, 7), R(\text{"x"}, 2, 8) \rangle$ 
37 | hd  $\triangleq$   $\{hdsa, hdsb\}$  CC, CM, and CCv but no SC
39 | hesa  $\triangleq$   $\langle W(\text{"x"}, 1, 1), W(\text{"y"}, 1, 2) \rangle$ 
40 | hesb  $\triangleq$   $\langle R(\text{"y"}, 1, 3), W(\text{"x"}, 2, 4) \rangle$ 
41 | hesc  $\triangleq$   $\langle R(\text{"x"}, 2, 5), R(\text{"x"}, 1, 6) \rangle$ 
42 | he  $\triangleq$   $\{hesa, hesb, hesc\}$  not CC (nor CM, nor CCv)
44 | all  $\triangleq$   $\{ha, hb, hc, hd, he\}$ 
45 |-----|
46 | WellFormedTest  $\triangleq$ 
47 |    $\forall h \in all : WellFormed(h)$ 
48 |-----|
  | Test of utility operators for operations
52 | OpsTest  $\triangleq$ 
53 |    $\wedge PrintT(\text{"OpsTest Begin"})$ 
54 |   on history ha
55 |    $\wedge Ops(ha) = \{W(\text{"x"}, 1, 1), R(\text{"x"}, 2, 2), W(\text{"x"}, 2, 3), R(\text{"x"}, 1, 4)\}$ 
56 |    $\wedge ReadOps(ha) = \{R(\text{"x"}, 2, 2), R(\text{"x"}, 1, 4)\}$ 

```

```

57    $\wedge \text{ReadOpsOnKey}(ha, "x") = \{R("x", 2, 2), R("x", 1, 4)\}$ 
58    $\wedge \text{WriteOps}(ha) = \{W("x", 1, 1), W("x", 2, 3)\}$ 
59    $\wedge \text{WriteOpsOnKey}(ha, "x") = \{W("x", 1, 1), W("x", 2, 3)\}$ 
60   on history he
61    $\wedge \text{Ops}(he) = \{W("x", 1, 1), W("y", 1, 2), R("y", 1, 3), W("x", 2, 4), R("x", 2, 5), R("x", 1, 6)\}$ 
62    $\wedge \text{ReadOps}(he) = \{R("y", 1, 3), R("x", 2, 5), R("x", 1, 6)\}$ 
63    $\wedge \text{ReadOpsOnKey}(he, "x") = \{R("x", 2, 5), R("x", 1, 6)\}$ 
64    $\wedge \text{WriteOps}(he) = \{W("x", 1, 1), W("y", 1, 2), W("x", 2, 4)\}$ 
65    $\wedge \text{WriteOpsOnKey}(he, "y") = \{W("y", 1, 2)\}$ 
66    $\wedge \text{PrintT}(\text{"OpsTest End"})$ 
67 |-----|
    Test of the auxiliary definitions for the axioms
71  $\text{CardOfProgramOrderOfHistory}(h) \triangleq$ 
72   LET  $\text{CardOfProgramOrderOfSession}(s) \triangleq$ 
73     IF  $\text{Len}(s) \leq 1$  THEN 0 ELSE  $\text{Sum}(1 \dots \text{Len}(s) - 1)$ 
74   IN  $\text{ReduceSet}(\text{LAMBDA } s, x : \text{CardOfProgramOrderOfSession}(s) + x, h, 0)$ 
76 THEOREM  $\text{ProgramOrderCardinalityTheorem} \triangleq$  test of  $PO(h)$ 
77    $\forall h \in \{ha, hb, hc, hd, he\} :$ 
78      $\text{Cardinality}(PO(h)) = \text{CardOfProgramOrderOfHistory}(h)$ 
80  $\text{POPastTest} \triangleq$  test of  $POPast(h, o)$ 
81    $\wedge \text{PrintT}(\text{"POPastTest Begin"})$ 
82    $\wedge \text{POPast}(ha, R("x", 2, 2)) = \{W("x", 1, 1)\}$ 
83    $\wedge \text{POPast}(hb, R("y", 1, 6)) = \{W("x", 2, 4), R("z", 0, 5)\}$ 
84    $\wedge \text{POPast}(hc, W("x", 2, 2)) = \{\}$ 
85    $\wedge \text{POPast}(hd, R("x", 1, 4)) = \{W("x", 1, 1), R("y", 0, 2), W("y", 1, 3)\}$ 
86    $\wedge \text{POPast}(he, W("x", 2, 4)) = \{R("y", 1, 3)\}$ 
87    $\wedge \text{PrintT}(\text{"POPastTest End"})$ 
89  $\text{CausalPastTest} \triangleq$  TODO: test of  $\text{CausalPast}(co, o)$ 
90    $\wedge \text{PrintT}(\text{"CausalPastTest Begin"})$ 
91    $\wedge \text{FALSE}$ 
92    $\wedge \text{PrintT}(\text{"CausalPastTest End"})$ 
94  $\text{CausalHistTest} \triangleq$  TODO: test of  $\text{CausalHist}(co, o)$ 
95    $\wedge \text{PrintT}(\text{"CausalHistTest Begin"})$ 
96    $\wedge \text{FALSE}$ 
97    $\wedge \text{PrintT}(\text{"CausalHistTest End"})$ 
99  $\text{CausalArbTest} \triangleq$  TODO: test of  $\text{CausalArb}(co, ar, o)$ 
100    $\wedge \text{PrintT}(\text{"CausalArbTest Begin"})$ 
101    $\wedge \text{FALSE}$ 
102    $\wedge \text{PrintT}(\text{"CausalArbTest End"})$ 
103 |-----|
    Test of axioms

```

```

107 RWRegSemanticsTest  $\triangleq$  test of RWRegSemanticsTest(seq, o)
108    $\wedge$  PrintT("RWRegSemanticsTest Begin")
109   seq =  $\langle \rangle$ 
110    $\wedge$  RWRegSemantics( $\langle \rangle$ , R("x", InitVal, 1))
111    $\wedge$  RWRegSemantics( $\langle \rangle$ , W("x", 1, 1))
112    $\wedge$   $\neg$ RWRegSemantics( $\langle \rangle$ , R("x", 2, 1))
113   no W("x", -, -) in seq
114    $\wedge$  RWRegSemantics( $\langle$  W("y", 1, 1), W("z", 1, 2), W("y", 1, 3) $\rangle$ , R("x", InitVal, 4))
115    $\wedge$  RWRegSemantics( $\langle$  W("y", 1, 1), W("z", 1, 2), W("y", 1, 3) $\rangle$ , W("x", 1, 4))
116    $\wedge$   $\neg$ RWRegSemantics( $\langle$  W("y", 1, 1), W("z", 1, 2), W("y", 1, 3) $\rangle$ , R("x", 1, 4))
117   contains W("x", -, -) in seq
118    $\wedge$  RWRegSemantics( $\langle$  W("x", 1, 1), W("y", 1, 2), W("x", 2, 3), W("z", 1, 4) $\rangle$ , R("x", 2, 5))
119    $\wedge$   $\neg$ RWRegSemantics( $\langle$  W("x", 1, 1), W("y", 1, 2), W("x", 2, 3), W("z", 1, 4) $\rangle$ , R("x", 1, 5))
120    $\wedge$  PrintT("RWRegSemanticsTest End")

122 AxCausalValueTest  $\triangleq$  TODO: test of AxCausalValue()
123    $\wedge$  FALSE

125 AxCausalArbTest  $\triangleq$  TODO: test of AxCausalArb()
126    $\wedge$  FALSE

127 |-----|
    | Test of the definitions of causal consistency |
    | ha: 4; hb: 7; hc: 4; hd: 8; he: 6 |
133 CCDefTest  $\triangleq$ 
134    $\wedge$  PrintT("CCDefTest Begin")
135    $\wedge$  PrintT(CC(ha))
136    $\wedge$  PrintT(CC(hc))
137    $\wedge$  PrintT( $\neg$ CC(he)) too slow
138    $\wedge$  LET sat  $\triangleq$  {ha, hb, hc, hd}
139   IN    $\wedge$   $\forall h \in$  sat: CC(h)
140        $\wedge$   $\forall h \in$  all \ sat:  $\neg$ CC(h)
141    $\wedge$  PrintT("CCDefTest End")

143 CCvDefTest  $\triangleq$ 
144    $\wedge$  PrintT("CCvDefTest Begin")
145    $\wedge$  PrintT( $\neg$ CCv(ha))
146    $\wedge$  CCv(hb)
147    $\wedge$  PrintT( $\neg$ CCv(hc))
148    $\wedge$  CCv(hd)
149    $\wedge$  PrintT( $\neg$ CCv(he))

151   LET sat  $\triangleq$  {hb, hd}
152   IN    $\wedge$   $\forall h \in$  sat: CCv(h)
153        $\wedge$   $\forall h \in$  all \ sat:  $\neg$ CCv(h)
154    $\wedge$  PrintT("CCvDefTest End")
155 |-----|

```

```

Test of the checking algorithms for causal consistency
ha: 4; hb: 7; hc: 4; hd: 8; he: 6

161 CCAlgTest  $\triangleq$  Test of the checking algorithm CCAlg for CC (Causal Consistency)
162   LET sat  $\triangleq$  {ha, hb, hc, hd}
163   IN    $\wedge \forall h \in sat :$ 
164          $\wedge PrintT(ToString(h) \circ " \text{ is differentiated: } " \circ ToString(IsDifferentiated(h)))$ 
165          $\wedge CCAlg(h)$ 
166    $\wedge \forall h \in all \setminus sat :$ 
167          $\wedge PrintT(ToString(h) \circ " \text{ is differentiated: } " \circ ToString(IsDifferentiated(h)))$ 
168          $\wedge \neg CCAlg(h)$ 

170 CCvAlgTest  $\triangleq$  Test of the checking algorithm CCvAlg for CCv (Causal Convergence)
171   LET sat  $\triangleq$  {hb, hd}
172   IN    $\wedge \forall h \in sat :$ 
173          $\wedge PrintT(ToString(h) \circ " \text{ is differentiated: } " \circ ToString(IsDifferentiated(h)))$ 
174          $\wedge CCvAlg(h)$ 
175    $\wedge \forall h \in all \setminus sat :$ 
176          $\wedge PrintT(ToString(h) \circ " \text{ is differentiated: } " \circ ToString(IsDifferentiated(h)))$ 
177          $\wedge \neg CCvAlg(h)$ 

178 |-----|
179 VARIABLES x keep it so that the model can be run
180 |-----|

\ * Modification History
\ * Last modified Wed Apr 21 21:41:21 CST 2021 by hengxin
\ * Created Fri Apr 09 11:53:33 CST 2021 by hengxin

```