```
1  Ins ≜ [type : {"Ins"}, pos : 1 .. (MaxLen + 1)]
2
3  ---------- MODULE XJupiterImpl ----------
4  We show that XJupiter (XJupiterExtended) implements CJupiter.
5  EXTENDS XJupiterExtended
6
7  VARIABLES
8      op2ss,   a function from an operation (represented by its Oid)
9               to the part of 2D state space produced while the operation is transformed
10     c2ssX    c2ssX[c]: redundant (eXtra) 2D state space maintained for client c ∈ Client
11
12 varsImpl ≜ ⟨varsEx, op2ss, c2ssX⟩
13 ----------
14 TypeOKImpl ≜
15     ∧ TypeOKEx
16     ∧ ∀ oid ∈ DOMAIN op2ss : oid ∈ Oid ∧ IsSS(op2ss[oid])
17     ∧ ∀ c ∈ Client : IsSS(c2ssX[c])
18 ----------
19 InitImpl ≜
20     ∧ InitEx
21     ∧ op2ss = ⟨⟩
22     ∧ c2ssX = [c ∈ Client ↦ [node ↦ {{}}, edge ↦ {}]]
23 ----------
24 Take union of 2D state spaces ss1 and ss2.
27 ss1 ⊕ ss2 ≜
28     [ss1 EXCEPT !.node = @ ∪ ss2.node,
29                 !.edge = @ ∪ ss2.edge]
32 Ignore the lr field in edges of 2D state space ss.
33 IgnoreDir(ss) ≜
34     [ss EXCEPT !.edge =
35         {[field ∈ (DOMAIN e \ {"lr"}) ↦ e.field] : e ∈ @}]
36       {[from ↦ e.from, to ↦ e.to, cop ↦ e.cop] : e ∈ @}]
37 ----------
38 DoImpl(c) ≜
39     ∧ DoEx(c)
40     ∧ UNCHANGED ⟨op2ss, c2ssX⟩
41
42 RevImpl(c) ≜
43     ∧   RevEx(c)
44     ∧   LET cop ≜ Head(cincoming[c])
45         IN  c2ssX′ = [c2ssX EXCEPT ![c] = @ ⊕ op2ss[cop.oid]]
46     ∧   UNCHANGED ⟨op2ss⟩
47
48 SRevImpl ≜
49     ∧ SRevEx
50     ∧ LET cop ≜ Head(sincoming)
51           c ≜ cop.oid.c
```

```
52              ss ≜ xForm(cop, s2ss[c], cur[Server], Remote)   TODO: performance!!!
53          IN  op2ss′ = op2ss @@ (cop.oid :> [node ↦ Range(ss.node), edge ↦ Range(ss.edge)])
54      ∧ UNCHANGED ⟨c2ssX⟩
55 ├──────────────────────────────────────────────────────────────────────────┤
56  NextImpl ≜
57      ∨ ∃ c ∈ Client : DoImpl(c) ∨ RevImpl(c)
58      ∨ SRevImpl

60  SpecImpl ≜ InitImpl ∧ □[NextImpl]_varsImpl
61      ∧ WF_varsImpl(SRevImpl ∨ ∃ c ∈ Client : RevImpl(c))

63  CJ ≜ INSTANCE CJupiter
64              WITH cincoming ← cincomingCJ,   sincoming needs no substitution
65                   css ← [r ∈ Replica ↦
66                              IF r = Server
67                               THEN IgnoreDir(SetReduce( ⊕ , Range(s2ss),
68                                                 [node ↦ {{}}, edge ↦ {}]))
69                               ELSE IgnoreDir(c2ss[r] ⊕ c2ssX[r])]

71  THEOREM SpecImpl ⇒ CJ!Spec
72 └──────────────────────────────────────────────────────────────────────────┘
    \* Modification History
    \* Last modified Sat Nov 10 22:33:20 CST 2018 by hengxin
    \* Created Fri Oct 26 15:00:19 CST 2018 by hengxin
```