───────────────── MODULE *JupiterInterface* ─────────────────

This module declares the parameters and defines the operators that describe the interface of a family of *Jupiter* specs.

6 EXTENDS *Integers*, *SequenceUtils*, *OT*

7 ├──

8 CONSTANTS
9   *Char*,   the set of characters
10   *Client*,   the set of client replicas
11   *Server*,   the (unique) server replica
12   *InitState*   the initial state of each replica

14 ASSUME   We assume that all inserted elements are unique.
15   ∧ $Range(InitState) \cap Char = \{\}$ due to the uniqueness requirement

16 ├──

17 VARIABLES
18   *state*, $state[r]$: state (the list content) of replica $r \in Replica$
19   *cincoming*, $cincoming[c]$: incoming channel at the client $c \in Client$
20   *sincoming*,   incoming channel at the *Server*
21   *chins*   a set of chars allowed to insert; this is for model checking

23 $intVars \triangleq \langle state,\ cincoming,\ sincoming,\ chins \rangle$

24 ├──

25 $Comm(Msg) \triangleq$ INSTANCE *CSComm*

27 $Replica \triangleq Client \cup \{Server\}$

29 $List \triangleq Seq(Char \cup Range(InitState))$ all possible lists
30 $MaxLen \triangleq Cardinality(Char) + Len(InitState)$ the max length of lists in any state

32 $ClientNum \triangleq Cardinality(Client)$
33 $Priority \triangleq$ CHOOSE $f \in [Client \rightarrow 1 .. ClientNum] : Injective(f)$

34 ├──

The set of all operations. Note: The positions are indexed from 1.

38 $Rd \triangleq [type : \{ \text{“Rd”} \}]$
39 $Del \triangleq [type : \{ \text{“Del”} \}, pos : 1 .. MaxLen]$
40 $Ins \triangleq [type : \{ \text{“Ins”} \}, pos : 1 .. (MaxLen + 1), ch : Char, pr : 1 .. ClientNum]$ *pr*: priority

42 $Op \triangleq Ins \cup Del$ Now we don't consider *Rd* operations

43 ├──

44 $TypeOKInt \triangleq$
45      $\wedge state \in [Replica \rightarrow List]$
46      $\wedge chins \subseteq Char$

48 $InitInt \triangleq$
49      $\wedge state = [r \in Replica \mapsto InitState]$
50      $\wedge chins = Char$

1

52   $DoIns(DoOp(\_, \_),\ c) \ \stackrel{\Delta}{=}$ Client $c \in Client$ generates an "Ins" operation.

53     $\exists\, ins \in \{\, op \in Ins :$

54                $\wedge\ op.pos \in 1\ ..\ (Len(state[c]) + 1)$

55                $\wedge\ op.ch \in chins \wedge op.pr = Priority[c]\,\} :$

56        $\wedge\ DoOp(c,\ ins)$

57        $\wedge\ chins' = chins \setminus \{ins.ch\}$ We assume that all inserted elements are unique.

59   $DoDel(DoOp(\_, \_),\ c)\ \stackrel{\Delta}{=}$ Client $c \in Client$ generates a "Del" operation.

60     $\exists\, del \in \{\, op \in Del : op.pos \in 1\ ..\ Len(state[c])\} :$

61        $\wedge\ DoOp(c,\ del)$

62        $\wedge\ \textsc{unchanged}\ chins$

64   $DoInt(DoOp(\_, \_),\ c)\ \stackrel{\Delta}{=}$ Client $c \in Client$ issues an operation.

65        $\wedge\ DoIns(DoOp,\ c)$

66        $\wedge\ DoDel(DoOp,\ c)$

68   $CRevInt(c)\ \stackrel{\Delta}{=}$ Client $c \in Client$ receives a message from the $Server$.

69        $\wedge\ \textsc{unchanged}\ chins$

71   $SRevInt\ \stackrel{\Delta}{=}$ The $Server$ receives a message.

72        $\wedge\ \textsc{unchanged}\ chins$

73

\ ∗ Modification History
\ ∗ *Last* modified *Mon Dec* 31 20:19:23 *CST* 2018 by *hengxin*
\ ∗ Created *Tue Dec* 04 19:01:01 *CST* 2018 by *hengxin*