MODULE *hashmap*

This module describes a *hashmap* to be used for testing with *Shalev* et al.'s split-ordered list implementation of the data structure

EXTENDS *Integers*

CONSTANTS *NULL*, *PossibleKeys*, *PossibleValues*

VARIABLES *keys*, *map*

Initial state has empty map and no keys

$$HashmapInit \triangleq \begin{array}{l} \land\ keys = \{\} \\ \land\ map = [k \in PossibleKeys \mapsto NULL] \end{array}$$

Insert changes exactly one mapping of the *hashmap* and adds one key to the set of keys

$$Insert \triangleq \begin{array}{l} \exists\, k \in PossibleKeys : \\ \quad \exists\, v \in PossibleValues : \\ \qquad \land\ keys' = keys \cup \{k\} \\ \qquad \land\ map' = [map \text{ EXCEPT } ![k] = v] \end{array}$$

Remove sets exactly one mapping to *NULL*

$$Remove \triangleq \begin{array}{l} \exists\, k \in PossibleKeys : \\ \quad \land\ keys' = keys \setminus \{k\} \\ \quad \land\ map' = [map \text{ EXCEPT } ![k] = NULL] \end{array}$$

Next is either an insert or a remove

$$HashmapNext \triangleq \begin{array}{l} \lor\ Insert \\ \lor\ Remove \end{array}$$

TypeOK asserts all keys and values are of the right type

$$TypeOK \triangleq \begin{array}{l} \forall\, k \in keys : \\ \quad \land\ k \in PossibleKeys \\ \quad \land\ map[k] \in PossibleValues \end{array}$$

KeyHasValue asserts that every key is mapped to a value

$$KeyHasValue \triangleq \forall\, k \in keys : \neg(map[k] = NULL)$$

The *hashmap* specification as a temporal formula

$$HashmapSpec \triangleq HashmapInit \land \Box[HashmapNext]_{\langle keys,\, map \rangle}$$