MODULE *squirrelcount*

EXTENDS *Naturals*, *FiniteSets*

A concurrent counter with a broken lock. "num" below is
the number of counters.
Each counter performs a READ of a global count variable,
performs a local addition, and then WRITES the updated value
to the global count variable. As this is non-atomic, the
operation breaks when two counters perform a READ concurrently
and then later write. This results in a lost update.

CONSTANT $BLUE$, RED, $ADDING$, $READY$, $STATUS$, $VALUE$, N
VARIABLES $cameras$, $truth$, $datastore$

$$
\begin{array}{l}
TypeInfo \triangleq \\
\quad \land cameras \in [\{BLUE,\ RED\} \to \{READY,\ ADDING\} \times (0\ ..\ N)] \\
\quad \land truth \in (0\ ..\ N) \\
\quad \land datastore \in (0\ ..\ N)
\end{array}
$$

$$
\begin{array}{l}
Init \triangleq \\
\quad \land cameras = [col \in \{BLUE,\ RED\} \mapsto \\
\qquad\qquad \langle READY,\ 0\rangle] \\
\quad \land truth = 0 \\
\quad \land datastore = 0
\end{array}
$$

$$
\begin{array}{l}
Read \triangleq \\
\quad \exists\, cam \in \{BLUE,\ RED\} : \\
\qquad \land\ cameras[cam][STATUS] = READY \\
\qquad \land\ cameras' = [col \in \{BLUE,\ RED\} \mapsto \\
\qquad\quad \text{IF } col = cam \text{ THEN } \langle ADDING,\ datastore\rangle \text{ ELSE } cameras[col]] \\
\qquad \land\ \text{UNCHANGED } truth \\
\qquad \land\ \text{UNCHANGED } datastore
\end{array}
$$

$$
\begin{array}{l}
Store \triangleq \\
\quad \exists\, cam \in \{BLUE,\ RED\} : \\
\qquad \land\ cameras[cam][STATUS] = ADDING \\
\qquad \land\ datastore' = cameras[cam][VALUE] + 1 \\
\qquad \land\ truth' = truth + 1 \\
\qquad \land\ cameras' = [col \in \{BLUE,\ RED\} \mapsto \\
\qquad\quad \text{IF } col = cam \text{ THEN } \langle READY,\ 0\rangle \text{ ELSE } cameras[col]]
\end{array}
$$

$$
\begin{array}{l}
Next \triangleq \\
\quad \lor Read \\
\quad \lor Store
\end{array}
$$

$$
\begin{array}{l}
InvariantTruthMatchesDatastore \triangleq \\
\quad \lor \exists\, cam \in \{BLUE,\ RED\} : \\
\qquad cameras[cam][STATUS] = ADDING \\
\quad \lor truth = datastore
\end{array}
$$