MODULE *AIHelper*

EXTENDS *Integers*, *Sequences*, *TLC*, *FiniteSets*

CONSTANTS
GRID_SIZE,
ValidValue

ASSUME
$\land\ GRID\_SIZE \in Nat \setminus \{0\}$
$\land\ ValidValue = 1\ ..\ GRID\_SIZE$

VARIABLES

grid,	The current state of the *Sudoku* grid (0 for empty).
solution,	The complete, correct solution for the grid.
hint,	The strategic hint generated by the helper.
pc	The program counter to control the flow.

$vars \triangleq \langle grid,\ solution,\ hint,\ pc \rangle$

The *TypeOK* invariant defines the valid domains for all state variables.

$TypeOK \triangleq$
$\land\quad grid \in [(1\ ..\ GRID\_SIZE) \times (1\ ..\ GRID\_SIZE) \to \{0\} \cup ValidValue]$
$\land\quad solution \in [(1\ ..\ GRID\_SIZE) \times (1\ ..\ GRID\_SIZE) \to ValidValue]$
$\land\quad hint \in [$
$\quad\quad type : \{\text{“OnlyChoice”},\ \text{“LastRemaining”},\ \text{“BruteForce”},\ \text{“None”}\},$
$\quad\quad dieValue : \{0\} \cup ValidValue,$
$\quad\quad row : 0\ ..\ GRID\_SIZE,$
$\quad\quad col\ \ : 0\ ..\ GRID\_SIZE$
$\quad\quad]$
$\land\quad pc \in \{\text{“start”},\ \text{“find\_only\_choice”},\ \text{“find\_last\_remaining”},\ \text{“find\_brute\_force”},\ \text{“done”}\}$

The initial state. The grid is a valid puzzle based on the solution.

$Init \triangleq$
$\land\ solution \in [(1\ ..\ GRID\_SIZE) \times (1\ ..\ GRID\_SIZE) \to ValidValue]$ A valid *Sudoku* solution.
$\land\ grid \in \{g \in [(1\ ..\ GRID\_SIZE) \times (1\ ..\ GRID\_SIZE) \to \{0\} \cup ValidValue] :$
$\quad\quad \forall\, r,\ c \in 1\ ..\ GRID\_SIZE : g[r,\ c] \in \{0,\ solution[r,\ c]\}\}$
$\land\ hint = [type \mapsto \text{“None”},\ dieValue \mapsto 0,\ row \mapsto 0,\ col \mapsto 0]$
$\land\ pc = \text{“start”}$

Helper operators to check for conflicts on the current grid.

$IsValueInRow(g,\ r,\ v) \triangleq \exists\, c \in 1\ ..\ GRID\_SIZE : g[r,\ c] = v$

$IsValueInCol(g,\ c,\ v) \triangleq \exists\, r \in 1\,..\,GRID\_SIZE : g[r,\ c] = v$
$IsValueInBox(g,\ r,\ c,\ v) \triangleq$
LET $BoxRowStart \triangleq ((r-1) \div 2) * 2 + 1$
$BoxColStart \triangleq ((c-1) \div 3) * 3 + 1$
IN $\exists\, br \in BoxRowStart\,..\,(BoxRowStart + 1),\ bc \in BoxColStart\,..\,(BoxColStart + 2) :$
$g[br,\ bc] = v$

$IsPossible(g,\ r,\ c,\ v) \triangleq$
$\land\ g[r,\ c] = 0$
$\land\ \neg IsValueInRow(g,\ r,\ v)$
$\land\ \neg IsValueInCol(g,\ c,\ v)$
$\land\ \neg IsValueInBox(g,\ r,\ c,\ v)$

Tier 1 Action: Find an "Only Choice" (Naked Single) hint.

$FindOnlyChoice \triangleq$
$\land\ pc =$ "find_only_choice"
$\land\ \exists\, r,\ c \in 1\,..\,GRID\_SIZE :$
$\land\ grid[r,\ c] = 0$
$\land$ LET
$val \triangleq solution[r,\ c]$
$PossibleInRow \triangleq \{col \in 1\,..\,GRID\_SIZE : IsPossible(grid,\ r,\ col,\ val)\}$
$PossibleInCol \triangleq \{row \in 1\,..\,GRID\_SIZE : IsPossible(grid,\ row,\ c,\ val)\}$
$PossibleInBox \triangleq \{\langle br,\ bc\rangle \in$ DOMAIN $grid :$
LET $BoxRowStart \triangleq ((r-1) \div 2) * 2 + 1$
$BoxColStart \triangleq ((c-1) \div 3) * 3 + 1$
IN $br \in BoxRowStart\,..\,(BoxRowStart + 1) \land$
$bc \in BoxColStart\,..\,(BoxColStart + 2) \land$
$IsPossible(grid,\ br,\ bc,\ val)\}$
IN $\lor\ Cardinality(PossibleInRow) = 1$
$\lor\ Cardinality(PossibleInCol) = 1$
$\lor\ Cardinality(PossibleInBox) = 1$
$\land$ LET $CellToHint \triangleq$ CHOOSE $cell \in \{\langle r,\ c\rangle \in$ DOMAIN $grid :$
LET $val \triangleq solution[r,\ c]$
$PossibleInRow \triangleq \{col \in 1\,..\,GRID\_SIZE : IsPossible(grid,\ r,\ col,\ val)\}$
$PossibleInCol \triangleq \{row \in 1\,..\,GRID\_SIZE : IsPossible(grid,\ row,\ c,\ val)\}$
$PossibleInBox \triangleq \{\langle br,\ bc\rangle \in$ DOMAIN $grid :$
LET $BoxRowStart \triangleq ((r-1) \div 2) * 2 + 1$
$BoxColStart \triangleq ((c-1) \div 3) * 3 + 1$
IN $br \in BoxRowStart\,..\,(BoxRowStart + 1) \land$
$bc \in BoxColStart\,..\,(BoxColStart + 2) \land$
$IsPossible(grid,\ br,\ bc,\ val)\}$
IN $grid[r,\ c] = 0 \land (Cardinality(PossibleInRow) = 1 \lor Cardinality(PossibleInCol) = 1 \lor Cardinal$
IN $hint' = [type \mapsto$ "OnlyChoice", $dieValue \mapsto solution[CellToHint[1],\ CellToHint[2]],\ row \mapsto CellToHi$

$\land\ pc' = \text{“done”}$
$\land$ UNCHANGED $\langle grid,\ solution \rangle$

Tier 2 Action: Find a “Last Remaining Cell” (Hidden Single) hint.

$FindLastRemaining\ \triangleq$
$\land\ pc = \text{“find\_last\_remaining”}$
$\land\ \lor$ Search in a row
$(\exists\, r \in 1\,..\,GRID\_SIZE,\ v \in ValidValue :$
LET $PossibleCells\ \triangleq\ \{c \in 1\,..\,GRID\_SIZE : IsPossible(grid,\ r,\ c,\ v)\}$
IN $Cardinality(PossibleCells) = 1)$
$\lor$ Search in a column
$(\exists\, c \in 1\,..\,GRID\_SIZE,\ v \in ValidValue :$
LET $PossibleCells\ \triangleq\ \{r \in 1\,..\,GRID\_SIZE : IsPossible(grid,\ r,\ c,\ v)\}$
IN $Cardinality(PossibleCells) = 1)$
$\lor$ Search in a box
$(\exists\, b\_row \in 0\,..\,2,\ b\_col \in 0\,..\,1,\ v \in ValidValue :$
LET $BoxRowStart\ \triangleq\ b\_row * 2 + 1$
$BoxColStart\ \triangleq\ b\_col\ * 3 + 1$
$PossibleCells\ \triangleq\ \{\langle br,\ bc \rangle \in \text{DOMAIN } grid :$
$br \in BoxRowStart\,..\,(BoxRowStart + 1) \land$
$bc \in BoxColStart\,..\,(BoxColStart\ \ + 2) \land$
$IsPossible(grid,\ br,\ bc,\ v)\}$
IN $Cardinality(PossibleCells) = 1)$
$\land$ LET
$CellToHint\ \triangleq$
CHOOSE $cell \in \{\langle r,\ c,\ v \rangle \in (1\,..\,GRID\_SIZE) \times (1\,..\,GRID\_SIZE) \times ValidValue :$
$\lor$ LET $PossibleCs\ \triangleq\ \{cc \in 1\,..\,GRID\_SIZE : IsPossible(grid,\ r,\ cc,\ v)\}$
IN $Cardinality(PossibleCs) = 1 \land c \in PossibleCs$
$\lor$ LET $PossibleRs\ \triangleq\ \{rr \in 1\,..\,GRID\_SIZE : IsPossible(grid,\ rr,\ c,\ v)\}$
IN $Cardinality(PossibleRs) = 1 \land r \in PossibleRs$
$\lor$ LET $BoxRowStart\ \triangleq\ ((r-1) \div 2) * 2 + 1$
$BoxColStart\ \triangleq\ ((c-1) \div 3) * 3 + 1$
$PossibleCellsInBox\ \triangleq\ \{\langle br,\ bc \rangle \in \text{DOMAIN } grid :$
$br \in BoxRowStart\,..\,(BoxRowStart + 1) \land$
$bc \in BoxColStart\,..\,(BoxColStart\ \ + 2) \land$
$IsPossible(grid,\ br,\ bc,\ v)\}$
IN $Cardinality(PossibleCellsInBox) = 1 \land \langle r,\ c \rangle \in PossibleCellsInBox$
$\} : \text{TRUE}$
IN $hint' = [type \mapsto \text{“LastRemaining”},\ dieValue \mapsto CellToHint[3],\ row \mapsto CellToHint[1],\ col \mapsto CellToH$
$\land\ pc' = \text{“done”}$
$\land$ UNCHANGED $\langle grid,\ solution \rangle$

Tier 3 Action: Brute-force fallback to find any valid move.

$FindBruteForce \triangleq$
$\land\ pc = \text{“find\_brute\_force”}$
$\land\ \exists\, r,\, c \in 1\,..\,GRID\_SIZE : grid[r,\, c] = 0$
$\land$ LET $CellToHint \triangleq$ CHOOSE $cell \in \{\langle r,\, c\rangle \in$ DOMAIN $grid : grid[r,\, c] = 0\} :$ TRUE
IN $hint' = [type \mapsto \text{“BruteForce”},\ dieValue \mapsto solution[CellToHint[1],\ CellToHint[2]],\ row \mapsto CellToHi$
$\land\ pc' = \text{“done”}$
$\land$ UNCHANGED $\langle grid,\, solution\rangle$

Control flow actions.

$StartHeuristicCheck \triangleq$
$\land\ pc = \text{“start”}$
$\land\ pc' = \text{“find\_only\_choice”}$
$\land$ UNCHANGED $\langle grid,\, solution,\, hint\rangle$

$ProceedToTier2 \triangleq$
$\land\ pc = \text{“find\_only\_choice”}$
$\land\ \neg\exists\, r,\, c \in 1\,..\,GRID\_SIZE :$
$\land\ grid[r,\, c] = 0$
$\land$ LET
$val \triangleq solution[r,\, c]$
$PossibleInRow \triangleq \{col \in 1\,..\,GRID\_SIZE : IsPossible(grid,\, r,\, col,\, val)\}$
$PossibleInCol \triangleq \{row \in 1\,..\,GRID\_SIZE : IsPossible(grid,\, row,\, c,\, val)\}$
$PossibleInBox \triangleq \{\langle br,\, bc\rangle \in$ DOMAIN $grid :$
LET $BoxRowStart \triangleq ((r-1) \div 2) * 2 + 1$
$BoxColStart \triangleq ((c-1) \div 3) * 3 + 1$
IN $br \in BoxRowStart\,..\,(BoxRowStart + 1) \land$
$bc \in BoxColStart\,..\,(BoxColStart + 2) \land$
$IsPossible(grid,\, br,\, bc,\, val)\}$
IN $\lor\ Cardinality(PossibleInRow) = 1$
$\lor\ Cardinality(PossibleInCol) = 1$
$\lor\ Cardinality(PossibleInBox) = 1$
$\land\ pc' = \text{“find\_last\_remaining”}$
$\land$ UNCHANGED $\langle grid,\, solution,\, hint\rangle$

$ProceedToTier3 \triangleq$
$\land\ pc = \text{“find\_last\_remaining”}$
$\land\ \neg($ $\lor$ Search in a row
$(\exists\, r \in 1\,..\,GRID\_SIZE,\, v \in ValidValue :$
LET $PossibleCells \triangleq \{c \in 1\,..\,GRID\_SIZE : IsPossible(grid,\, r,\, c,\, v)\}$
IN $Cardinality(PossibleCells) = 1)$
$\lor$ Search in a column
$(\exists\, c \in 1\,..\,GRID\_SIZE,\, v \in ValidValue :$

```
            LET PossibleCells ≜ {r ∈ 1 .. GRID_SIZE : IsPossible(grid, r, c, v)}
            IN  Cardinality(PossibleCells) = 1)
      ∨ Search in a box
        (∃ b_row ∈ 0 .. 2, b_col ∈ 0 .. 1, v ∈ ValidValue :
            LET BoxRowStart ≜ b_row * 2 + 1
                BoxColStart ≜ b_col * 3 + 1
                PossibleCells ≜ {⟨br, bc⟩ ∈ DOMAIN grid :
                                    br ∈ BoxRowStart .. (BoxRowStart + 1) ∧
                                    bc ∈ BoxColStart .. (BoxColStart + 2) ∧
                                    IsPossible(grid, br, bc, v)}
            IN  Cardinality(PossibleCells) = 1))
   ∧ pc' = "find_brute_force"
   ∧ UNCHANGED ⟨grid, solution, hint⟩

NoHintFound ≜
   ∧ pc = "find_brute_force"
   ∧ ∀ r, c ∈ 1 .. GRID_SIZE : grid[r, c] ≠ 0
   ∧ pc' = "done"
   ∧ UNCHANGED ⟨grid, solution, hint⟩

(* ---------------------------------------------------------------------- *)
(* The next-state relation for the entire specification.                  *)
(* ---------------------------------------------------------------------- *)
Next ≜
   ∨ StartHeuristicCheck
   ∨ FindOnlyChoice
   ∨ ProceedToTier2
   ∨ FindLastRemaining
   ∨ ProceedToTier3
   ∨ FindBruteForce
   ∨ NoHintFound

Spec ≜ Init ∧ □[Next]_vars

(* ---------------------------------------------------------------------- *)
(* Liveness property to check that the algorithm eventually terminates.   *)
(* ---------------------------------------------------------------------- *)
Termination ≜ ◇(pc = "done")

THEOREM Spec ⇒ □TypeOK
```