**Constraint Satisfaction Problem (CSP) Formulation for Ultimate Tic-Tac-Toe (Variant)**

**Variables**

* Each cell in the game is a variable, identified by (big\_row, big\_col, small\_row, small\_col), where:
  + big\_row, big\_col ∈ {0, 1, 2} denote the 3x3 large board.
  + small\_row, small\_col ∈ {0, 1, 2} denote the 3x3 small board within a large board cell.
* Total variables: 81 (3 × 3 × 3 × 3).

**Domains**

* Each variable has a domain: {X, O, None}.
* Initially, all cells are None (empty).
* A move assigns X or O to a cell, reducing its domain to a singleton.

**Constraints**

1. **Active Board Constraint (Variant)**:
   * Once a small board at (big\_row, big\_col) is selected, all subsequent moves must occur in that board until it is won (three X or O in a row, column, or diagonal) or full (draw).
   * After a small board is won or full, the next move can be in any non-won, non-full small board.
   * At the start of the game, any small board can be chosen.
2. **Cell Occupancy Constraint**:
   * A cell (big\_row, big\_col, small\_row, small\_col) cannot be assigned X or O if it is already occupied (i.e., not None).
3. **Small Board Win Constraint**:
   * If a small board at (big\_row, big\_col) has a winning configuration, no further moves are allowed in that board.
   * A small board is won if:
     + Any row: board[big\_row][big\_col][i][0] = board[big\_row][big\_col][i][1] = board[big\_row][big\_col][i][2] ≠ None
     + Any column: board[big\_row][big\_col][0][i] = board[big\_row][big\_col][1][i] = board[big\_row][big\_col][2][i] ≠ None
     + Main diagonal: board[big\_row][big\_col][0][0] = board[big\_row][big\_col][1][1] = board[big\_row][big\_col][2][2] ≠ None
     + Anti-diagonal: board[big\_row][big\_col][0][2] = board[big\_row][big\_col][1][1] = board[big\_row][big\_col][2][0] ≠ None
   * If all cells are filled without a win, the board is a draw (D).
4. **Game Win Constraint**:
   * The game is won if three small boards form a line (row, column, or diagonal) with the same winner (X or O).
   * If all small boards are won or drawn without a game winner, the game is a draw.

**CSP Solver Implementation**

* **Backtracking Search**: Iterates through legal moves in the active small board, assigning X or O while respecting constraints.
* **Forward Checking**: Prunes moves that allow the opponent to win immediately in the current small board or the game.
* **Arc Consistency (AC-3)**: Ensures moves maintain consistency by evaluating opponent responses within the active board, pruning moves leading to poor outcomes.
* **Heuristics**:
  + **Minimum Remaining Values (MRV)**: Prioritizes moves that reduce opponent options in the current small board.
  + **Strategic Heuristics**: Favors moves that win the current small board or occupy central cells for tactical advantage.

This formulation reflects the variant rule where players must complete a small board before moving to another, ensuring focused gameplay within each small board.