**1. State: (cr, cc, row[0..9], col[0..9], reg[a1..b1][a2..b2], x, y)**

cr : Hàng hiện tại đang xét (current row)

cc : Cột hiện tại đang xét (current column)

row[0..9] : mảng lưu tất cả số trong hàng đang xét (nếu không có số, gán -1)

col[0..9] : mảng lưu tất cả số trong cột đang xét (nếu không có số, gán -1)

reg[a1..b1][a2..b2] : mảng lưu các vị trí nằm trong hình vuông 3x3 chứa vị trí đang xét của bảng sudoku. Với một số điều kiện ràng buộc sau :

b1 – a1 = 3, a1<=cr<=b1, a1<-{0;3;6}, b1<-{2;5;8}

b2 – a2 = 3, a2<=cc<=b2, a2<-{0;3;6}, b2<-{2;5;8}

Hai điều kiện trên nhằm xác định các ô vuông 3x3 chứa vị trí hiện tại đang xét

**2. Initial state: (0, 0, row <- gán các số trong dòng 0, col <- gán các số trong cột 0, reg <- gán các số ở ô vuông 3x3 đầu tiên, 0, 0)**

**3. Goal state: (a, b, row contains distinct values, col contains distinct values, reg contains distinct values, c, d)**

Khi xét hết bảng sudoku đều thu được các giá trị phân biệt trong mảng row, col và reg

**4. Legal moves: Current state -> Next state**

(a, b, row, col, reg, x, y) -> (a, b, row[x] = c, col, reg, x, y), nếu row[x] = -1 và c không thuộc row, col và reg

(a, b, row, col, reg, x, y) -> (a, b, row, col[y] = d, reg, x, y), nếu col[y] = -1 và d không thuộc row, col và reg

(a, b, row, col, reg, x, y) -> (a, b+1, row, col, reg, x+1, y), nếu b, x < 9

(a, b, row, col, reg, x, y) -> (a, b-1, row, col, reg, x-1, y), nếu b, x > 0

(a, b, row, col, reg, x, y) -> (a+1, b, row, col, reg, x, y+1), nếu a, y < 9

(a, b, row, col, reg, x, y) -> (a-1, b, row, col, reg, x, y-1), nếu a, y > 0