```
def solve_sudoku(grid):
    """
    Solves a 9x9 Sudoku puzzle using backtracking algorithm.

    Args:
        grid: 9x9 list of lists where 0 represents empty cells

    Returns:
        True if puzzle is solved, False if unsolvable
    """

    def is_valid(grid, row, col, num):
        """Check if placing num at (row, col) is valid"""

        # Check row
        for x in range(9):
            if grid[row][x] == num:
                return False

        # Check column
        for x in range(9):
            if grid[x][col] == num:
                return False

        # Check 3x3 box
        start_row = row - row % 3
        start_col = col - col % 3
        for i in range(3):
            for j in range(3):
                if grid[i + start_row][j + start_col] == num:
                    return False

        return True

    def find_empty(grid):
        """Find next empty cell (returns row, col or None)"""
        for i in range(9):
            for j in range(9):
                if grid[i][j] == 0:
                    return i, j
        return None

    # Find empty cell
    empty = find_empty(grid)
    if not empty:
        return True  # Puzzle solved

    row, col = empty

    # Try numbers 1-9
    for num in range(1, 10):
```

```
            if is_valid(grid, row, col, num):
                grid[row][col] = num

                # Recursively solve
                if solve_sudoku(grid):
                    return True

                # Backtrack
                grid[row][col] = 0

        return False

    def print_grid(grid):
        """Print the Sudoku grid in a readable format"""
        for i in range(9):
            if i % 3 == 0 and i != 0:
                print("------+-------+------")

            for j in range(9):
                if j % 3 == 0 and j != 0:
                    print("| ", end="")

                print(str(grid[i][j]) + " ", end="")

            print()

    # Example usage
    if __name__ == "__main__":
        # Example Sudoku puzzle (0 represents empty cells)
        puzzle = [
            [5, 3, 0, 0, 7, 0, 0, 0, 0],
            [6, 0, 0, 1, 9, 5, 0, 0, 0],
            [0, 9, 8, 0, 0, 0, 0, 6, 0],
            [8, 0, 0, 0, 6, 0, 0, 0, 3],
            [4, 0, 0, 8, 0, 3, 0, 0, 1],
            [7, 0, 0, 0, 2, 0, 0, 0, 6],
            [0, 6, 0, 0, 0, 0, 2, 8, 0],
            [0, 0, 0, 4, 1, 9, 0, 0, 5],
            [0, 0, 0, 0, 8, 0, 0, 7, 9]
        ]

        print("Original Sudoku puzzle:")
        print_grid(puzzle)
        print("\nSolving...\n")

        if solve_sudoku(puzzle):
            print("Solved Sudoku puzzle:")
            print_grid(puzzle)
        else:
            print("No solution exists for this puzzle.")
```