Universidade Federal de Ouro Preto
Lecture Notes
Backtracking

Prof. Rodrigo Silva

May 30, 2023

1 Backtracking

Algorithm 1 Backtracking Algorithm

```
1: function BACKTRACKING(problem)
2:     if ISSOLUTION(problem) then
3:         return problem                                   ▷ Found a solution
4:     for all option in GENERATEOPTIONS(problem) do
5:         if ISVALID(option) then
6:             APPLYOPTION(option)
7:             result ← BACKTRACKING(problem)
8:             if result ≠ None then
9:                 return result                            ▷ Found a solution
10:            UNDOOPTION(option)                           ▷ Backtrack
11:    return None                                          ▷ No solution
```

2 N-Queen Problem

Algorithm 2 N-Queens Backtracking Algorithm

```
1: function SOLVENQUEENS(n)
2:     board ← empty n × n chessboard
3:     queens ← empty list
4:     PLACEQUEENS(board, queens, 0, n)
5:     return queens                                        ▷ Found a solution or None if no solution
6:
7: function PLACEQUEENS(board, queens, row, n)
8:     if row ≥ n then
9:         return true                                      ▷ All queens are placed
10:    for col in 0 to n − 1 do
11:        if ISSAFE(board, row, col) then
12:            MARKQUEEN(board, row, col)
13:            queens.append((row, col))
14:            if PLACEQUEENS(board, queens, row+1, n) then
15:                return true                              ▷ Queens placed successfully
16:            UNMARKQUEEN(board, row, col)
17:            queens.pop()
18:    return false                                         ▷ Cannot place queens in this configuration
```

3 Sudoku

Algorithm 3 Sudoku Backtracking Algorithm

```
1: function SolveSudoku(board)
2:     if IsBoardComplete(board) then
3:         return board                                   ▷ Found a solution
4:     row, col ← FindEmptyCell(board)
5:     for num in [1, 2, 3, 4, 5, 6, 7, 8, 9] do
6:         if IsValidMove(board, row, col, num) then
7:             board[row][col] ← num
8:             result ← SolveSudoku(board)
9:             if result ≠ None then
10:                return result                          ▷ Found a solution
11:            board[row][col] ← 0                        ▷ Backtrack
12:    return None                                        ▷ No solution found
```