



인공지능 기말 프로젝트



스도쿠

12204132 강민정
12204133 강민지
12204152 박희윤
12204279 이소영

	6	2				8	9	
4					3			7
		1			7	2	6	
	2		7		6			
8	3	5			4			
	7			8		1		5
	9							6
2		8					3	
					9	7		



스도쿠 게임 설명과 규칙

스도쿠 게임이란?

9×9 칸에서 진행되는 숫자 퍼즐 게임

스도쿠를 구성하는 칸(cell)은 총 81칸, 3×3칸 9개로 세분화
각각의 가로줄(row)과 세로줄(column)에 1~9가 중복 없이 하나씩 들어간다.
3×3칸(box) 안에는 1~9가 중복 없이 하나씩 들어간다.



백트래킹(Backtracking)의 뜻

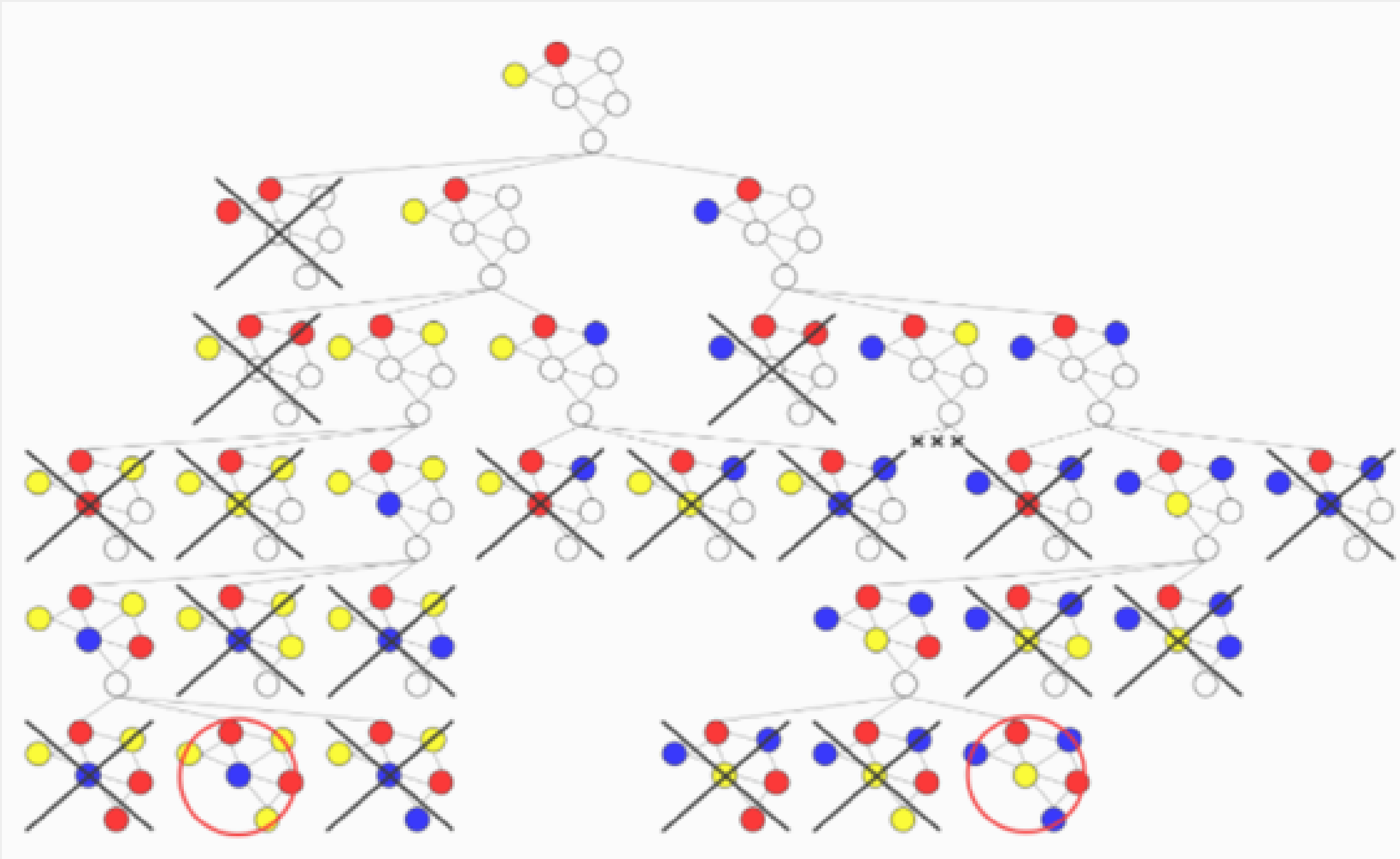
백트래킹이란?

back(역) + tracking(추적) -> 역추적

가지치기(pruning)를 이용해 불필요한 탐색을 줄이는 알고리즘이다.
가지치기란 답이 될 수 없는 경우는 탐색 대상에서 제외하여,
효율적으로 답을 구하는 알고리즘이다.
반복문의 횟수를 줄여 필요한 연산량을 줄일 수 있으므로 효율적이다.



가지치기





코드 설명

구현 코드

```
from copy import deepcopy
from sys import exit
import pygame
import time
import random
```

```
def print_sudoku(board): # 스도쿠를 생성
    boardString = "" # 숫자 띄어쓰기 위함
    for i in range(9): # i 행 9개 생성
        for j in range(9): # j 열 9개 생성
            boardString += str(board[i][j]) + " "
            if j==2 or j==5:
                boardString += "|"
            if j == 8:
                boardString += "\n"
            if j == 8 and (i+1)%3 == 0 and (i+1) != 9:
                boardString += "====+====+=====\n"
    print(boardString)
```

```
def blank(board): # 빈칸을 찾아서 위치를 반환
    for i in range(9):
        for j in range(9):
            if board[i][j] == 0:
                return i,j
```

```
def check(board, position, num): #가로 열의 숫자 확인
    for i in range(9):
        if board[i][position[1]] == num and (i, position[1]) != position:
            return False
```

```
#세로 열의 숫자확인
for j in range(9):
    if board[position[0]][j] == num and (position[0], j) != position:
        return False
start_i = position[0] - position[0] % 3 # 시작 지점 위치
start_j = position[1] - position[1] % 3
```

```
# 사각형 내 숫자 확인
for i in range(3):
    for j in range(3):
        if board[start_i + i][start_j + j] == num and (start_i + i, start_j + j) != position:
            return False
    return True
```



코드 설명

구현 코드

```
def go_sudoku(board): # 백트래킹 알고리즘 사용
    empty = blank(board)
    if not empty: #빈칸이 없으면 스도쿠 완성 실행 종료
        return True
```

```
for nums in range(9):
```

```
    if check(board, empty,nums+1):
```

```
        board[empty[0]][empty[1]] = nums+1
```

```
        if go_sudoku(board): #재귀
```

```
            return True
```

```
        board[empty[0]][empty[1]] = 0 #숫자가 틀리면 다시 빈칸으로 변경
```

```
return False
```

```
#스도쿠 입력값
```

```
# if __name__ == '__main__':
```

```
# board = [
```

```
# [0, 0, 0, 0, 0, 0, 2, 0, 0],
```

```
# [0, 8, 0, 0, 0, 7, 0, 9, 0],
```

```
# [6, 0, 2, 0, 0, 0, 5, 0, 0],
```

```
# [0, 7, 0, 0, 6, 0, 0, 0, 0],
```

```
# [0, 0, 0, 9, 0, 1, 0, 0, 0],
```

```
# [0, 0, 0, 0, 2, 0, 0, 4, 0],
```

```
# [0, 0, 5, 0, 0, 0, 6, 0, 3],
```

```
# [0, 9, 0, 4, 0, 0, 0, 7, 0],
```

```
# [0, 0, 6, 0, 0, 0, 0, 0, 0]
```

```
# ]
```

```
# go_sudoku(board)
```

```
# print_sudoku(board)
```



구현 영상

랜덤으로 생성된 스도쿠를
인공지능이 백트래킹을 이용하여
값을 도출해 내고 있다.

Sudoku

7	6	8	3	4	9	2	5	1
3	2	1	8	7	5	9	4	6
4	9	5	6	1	2	3	8	7
8	3	6	9	2	4	1	7	5
1	4	9	5	3	7	6	2	8
2	5	7	1	6				3
	8					4		9
		2		8				
5				9				

00:00:00



인공지능 프로젝트

경청해주셔서 감사합니다.

