

EC3272 Artificial Intelligence Project

By:

Sneha Tamang
Akriti Gautam
Isha Khadka
Prince Thapa

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

Introduction

> *Sudoku*

Prolog

Objectives

Facts & Constraints

System Workflow

System commands

Demonstration

Conclusion

Sudoku

- A logical number placement puzzle played on a 9x9 grid divided into 3x3 sub grids.
- Fill all cells with numbers 1-9 without repeating in rows, columns, or subgrids.

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

Introduction

Sudoku

> *Prolog*

Objectives

Facts & Constraints

System Workflow

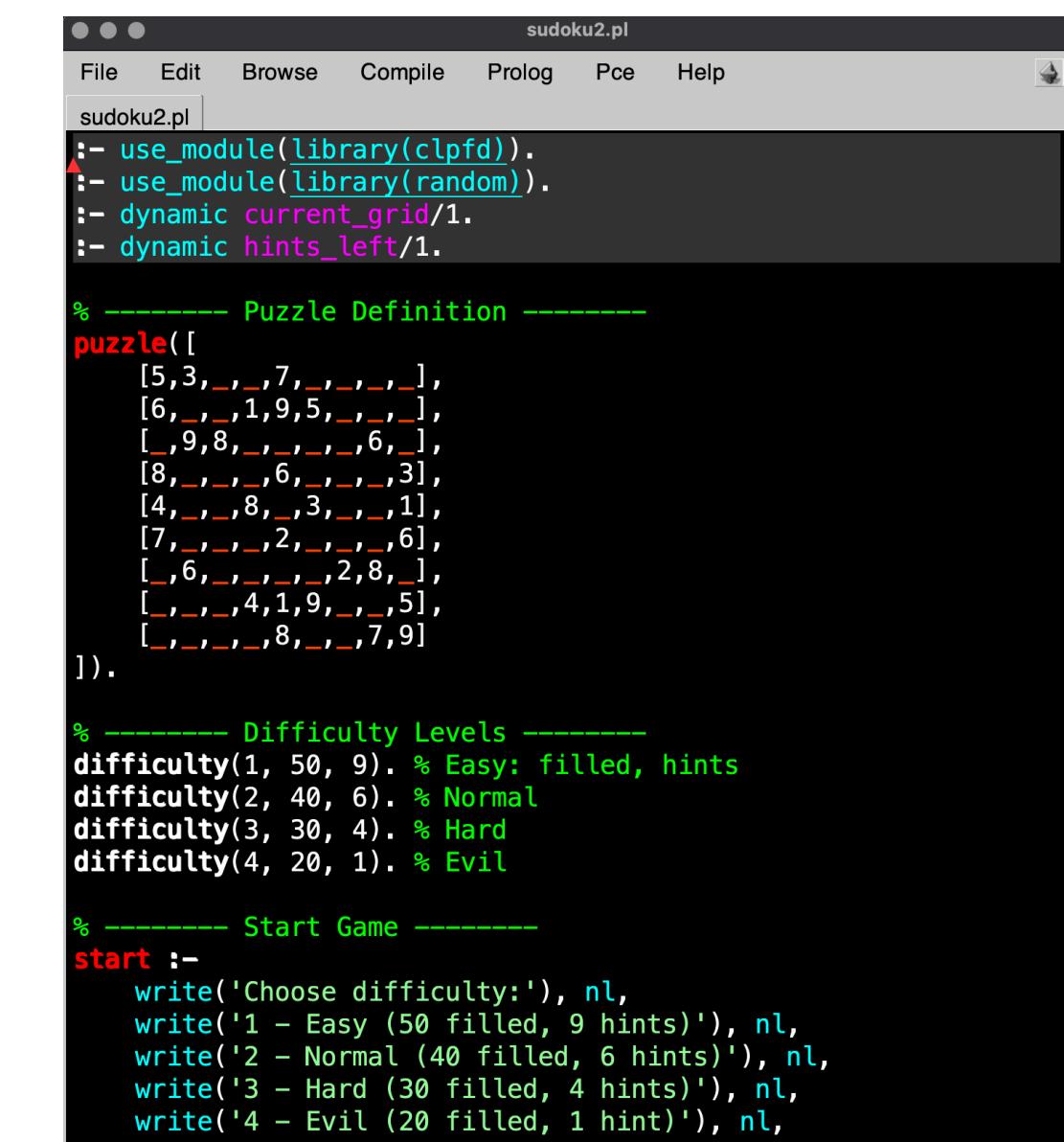
System commands

Demonstration

Conclusion

Prolog

- A declarative programming language focused on logic and reasoning.
- Ideal for solving problems defined by rules and constraints.



The screenshot shows a window titled "sudoku2.pl" containing Prolog code. The code includes imports for clpfd and random libraries, and defines predicates for current_grid and hints_left. It then defines a puzzle grid with 9x9 cells, some filled with numbers and others empty (represented by underscores). Following this is a section for difficulty levels with four entries: Easy (50 filled, 9 hints), Normal (40 filled, 6 hints), Hard (30 filled, 4 hints), and Evil (20 filled, 1 hint). Finally, a start predicate is defined to write instructions to the user.

```
sudoku2.pl
:- use_module(library(clpfd)).
:- use_module(library(random)).
:- dynamic current_grid/1.
:- dynamic hints_left/1.

% ----- Puzzle Definition -----
puzzle([
    [5,3,_,_,7,_,_,_,_],
    [6,_,_,1,9,5,_,_,_],
    [_,9,8,_,_,_,6,_],
    [8,_,_,_,6,_,_,_,3],
    [_,_,_,8,3,_,_,1],
    [4,_,_,8,3,_,_,1],
    [7,_,_,_,2,_,_,6],
    [_,6,_,_,_,2,8,_],
    [_,_,_,4,1,9,_,_,5],
    [_,_,_,8,_,_,7,9]
]).

% ----- Difficulty Levels -----
difficulty(1, 50, 9). % Easy: filled, hints
difficulty(2, 40, 6). % Normal
difficulty(3, 30, 4). % Hard
difficulty(4, 20, 1). % Evil

% ----- Start Game -----
start :-
    write('Choose difficulty:'), nl,
    write('1 - Easy (50 filled, 9 hints)'), nl,
    write('2 - Normal (40 filled, 6 hints)'), nl,
    write('3 - Hard (30 filled, 4 hints)'), nl,
    write('4 - Evil (20 filled, 1 hint)'), nl,
```

Introduction

> Objectives

Facts & Constraints

System Workflow

System commands

Demonstration

Conclusion

Objectives

- Design and implement a complete 9x9 Sudoku grid.
- Provide interactive commands: look, fill, hint, solve and check.
- Enforce Sudoku rules and validate moves.
- Develop multiple difficulty levels.
- Include hints, warnings, and scoring features.
- Demonstrate logical problem solving using prolog.

Introduction

Objectives

> **Facts & Constraints**

System Workflow

System commands

Demonstration

Conclusion

Facts

```
% ----- Puzzle Definition -----
puzzle([
    [5,3,_,_,7,_,_,_,_],
    [6,_,_,1,9,5,_,_,_],
    [_,9,8,_,_,_,6,_],▲
    [8,_,_,6,_,_,_,3],
    [4,_,_,8,_,3,_,_,1],
    [7,_,_,2,_,_,_,6],
    [_,6,_,_,_,2,8,_],
    [_,_,_,4,1,9,_,_,5],
    [_,_,_,8,_,_,7,9]
]).
```

Constraints

- The rules of sudoku.

Introduction

Objectives

Facts & Constraints

> **System Workflow**

System commands

Demonstration

Conclusion

System Workflow

- How the program generated Sudoku puzzles?
- How the rules (constraints) are applied?
- How the user interacts (fills the squares)?
- How Prolog processes and validates the puzzle?

Introduction

Objectives

Facts & Constraints

System Workflow

> **System commands**

Demonstration

Conclusion

System Commands

- start.
- look.
- fill(row, col, num).
- hint.
- check.
- solve.
- show_initial.
- win.

One
Demo
Is
Worth
A
Thousand
Explanation

Introduction
Objectives
Facts & Constraints
System Workflow
System commands
> Demonstration
Conclusion

Introduction

Objectives

Facts & Constraints

System Workflow

System commands

Demonstration

> **Conclusion**

“Prolog—where the system doesn’t just execute, it reasons.”

- Logic over instructions.
- Reasoning & Inference
- Backtracking Mechanism
- Constraint Satisfaction
- Human-like Thinking
- Beyond Sudoku: Expert Systems, AI, Problem Solving

Any Questions ?