Contents

Chapter 1

About this overview

The following sites present the logic used in the SudokuSolver algorithm.

Chapter 2

Candidate lists and the idea behind implemention

In an empty cell, any number is candidate that is not in the entire row, entire column or in the same 3x3-subblock.

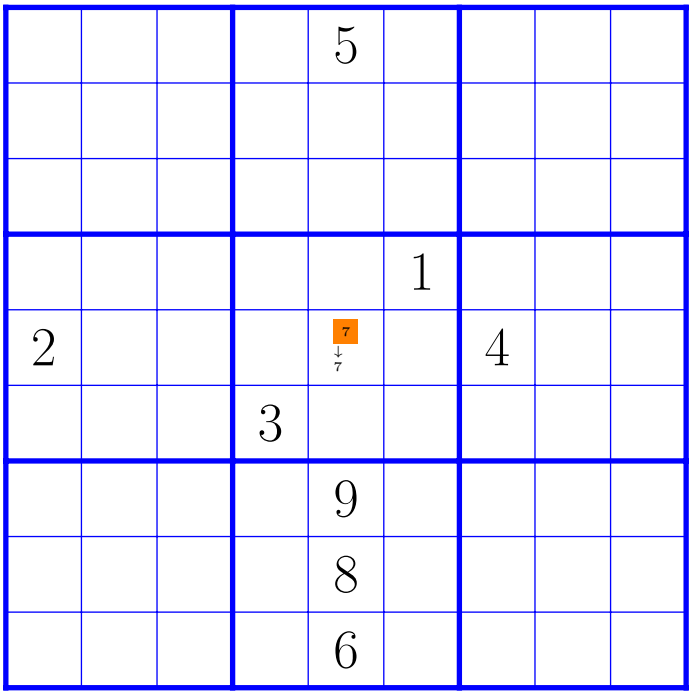
				5				
			2		8			
	6		4,5,8,9	2,4	1		7	
2			5,6,7,8,9	7	5,6,7,9	4		
	9		3	2,4,7	2,4,6,7	5		
				9				
				8				
				6				

Are all empty cells filled with their candidates, the algorithm takes the lists and tries to reduce candidate entries. In this sense, solving means nothing else than candidate elimination until only one last candidate is left. This must be the final digit of the cell. In order to discard invalid candidates from a specific list, a couple of implemented solving strategies use dependencies between cells and check constellations which make candidate entries logically invalid.

Chapter 3

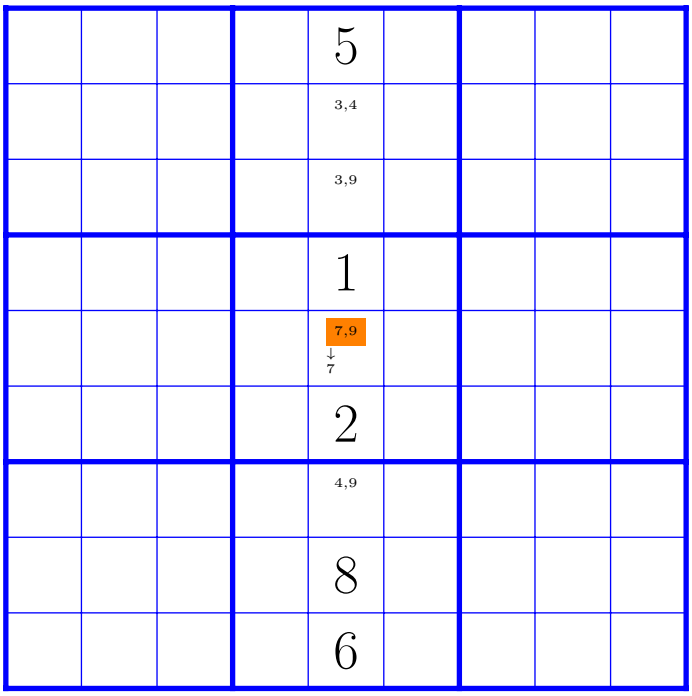
Solving Techniques

3.1 Naked Singles

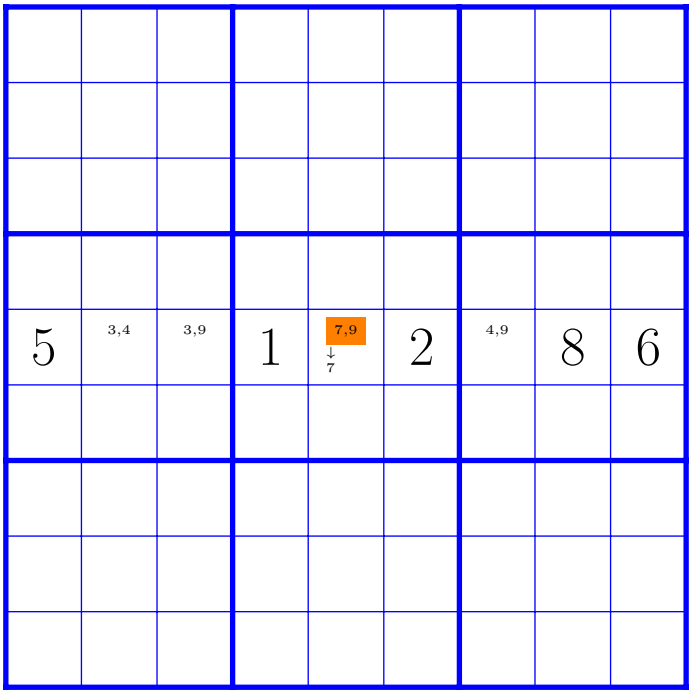


3.2 Hidden Singles

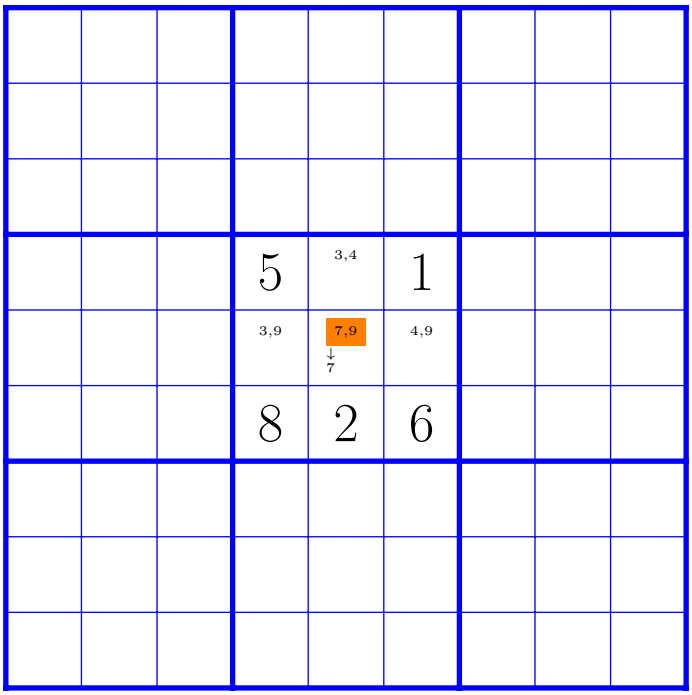
3.2.1 In Row



3.2.2 In Column

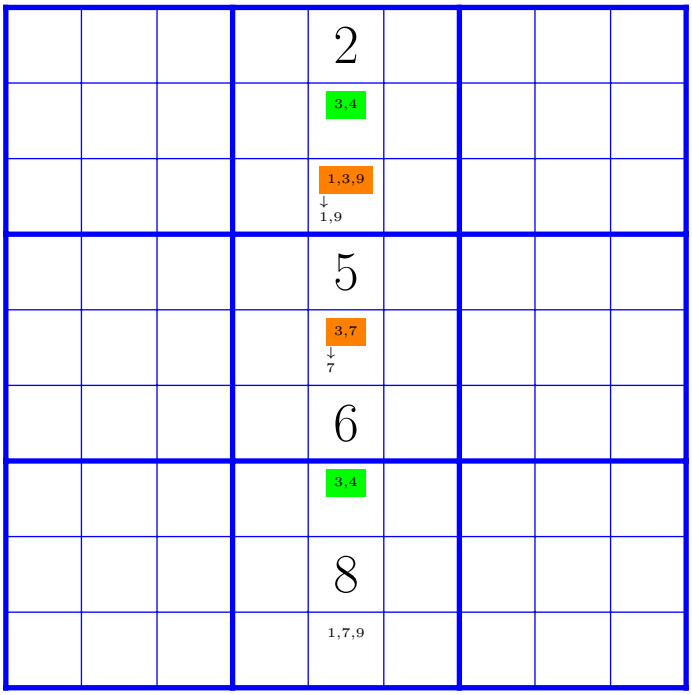


3.2.3 In Block

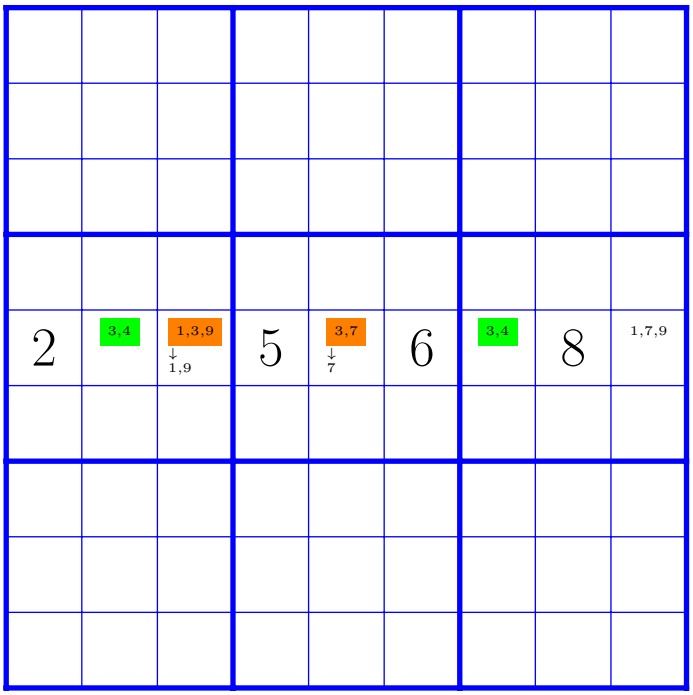


3.3 Naked Pairs

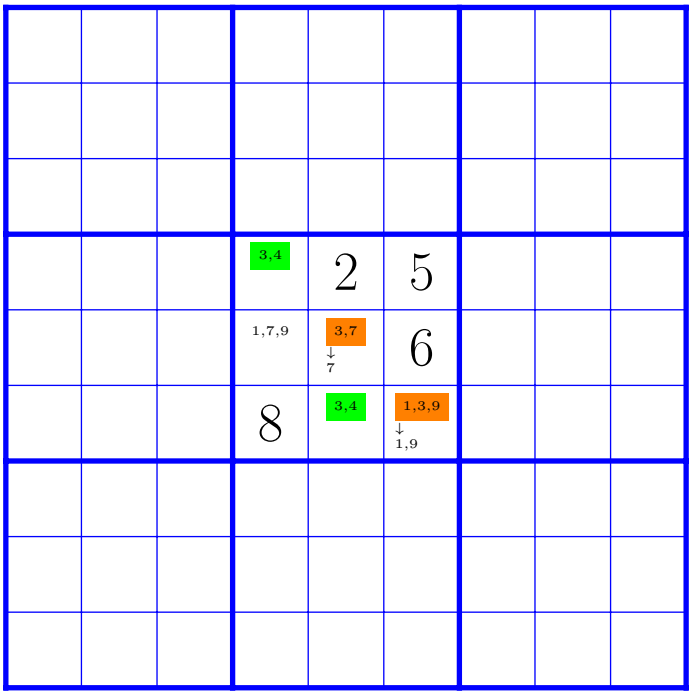
3.3.1 In Row



3.3.2 In Column



3.3.3 In Block



3.4 Hidden Pairs

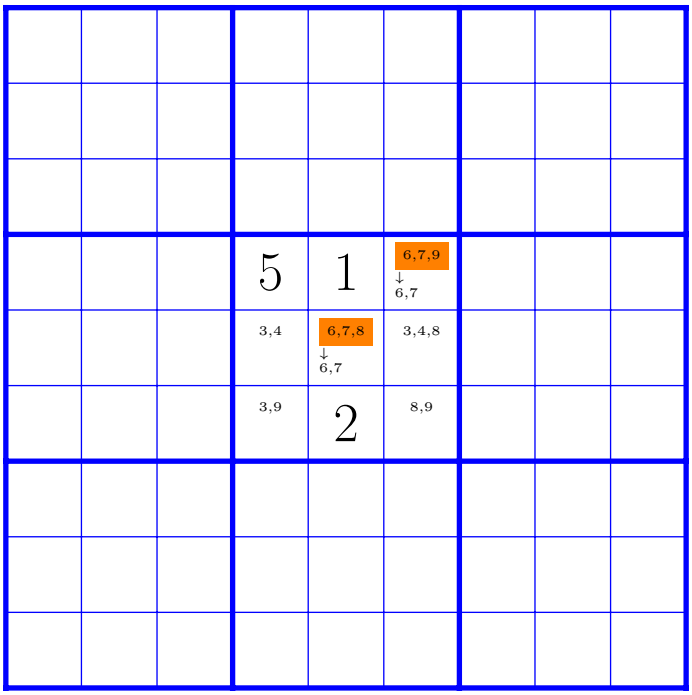
3.4.1 In Row

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

3.4.2 In Column

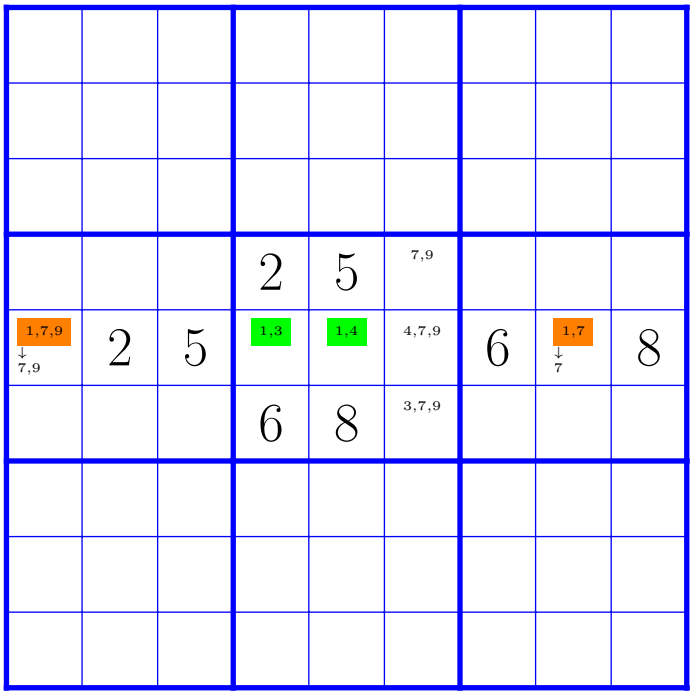
5	3,4	3,9	1	6,7,8 ↓ 6,7	2	6,7,9 ↓ 6,7	3,4,8	8,9

3.4.3 In Block

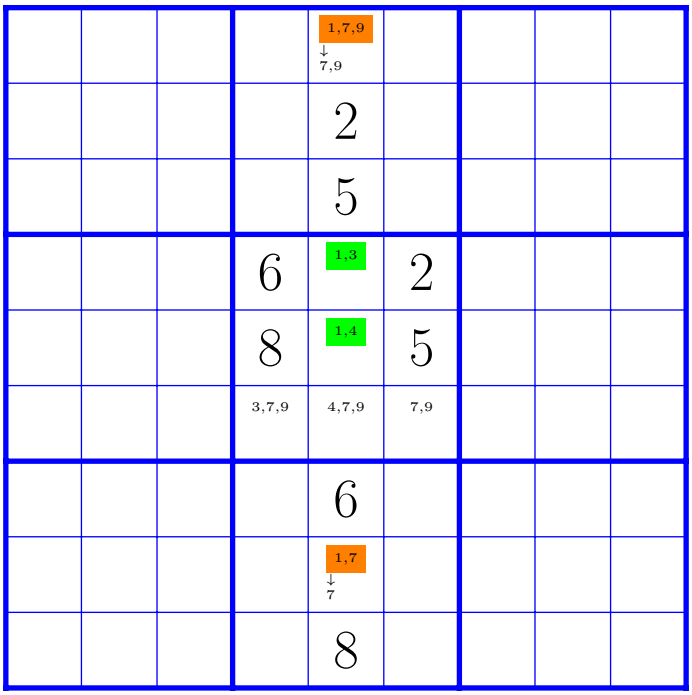


3.5 Locked candidates

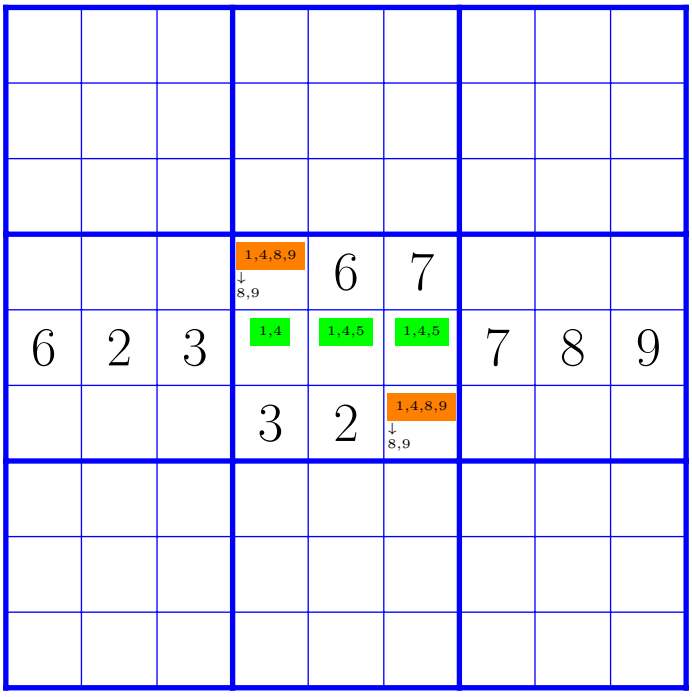
3.5.1 Row within block



3.5.2 Column within block



3.5.3 Block within row



3.5.4 Block within row