

A basic Sudoku is...

a game of logic where the player must assign the numbers 1..9 to cells on a 9x9 grid. The placement of the numbers must be such that every number must appear exactly once in every row, column, and 3x3 block, where the board is split into nine 3x3 blocks.

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

#15 on Royle's web site

Our goals are to...

Promote the study of Constraint Processing (CP) among CS students, and explain CP techniques to the general public. Sudoku is perfect to this end because it allows us to visualize the operation of CP algorithms.



Understand and demystify humans' fascination with puzzles.

Discourage students from losing too much time playing by making a program to play games for them.



[Solver](#) & [Constructor](#) are available online

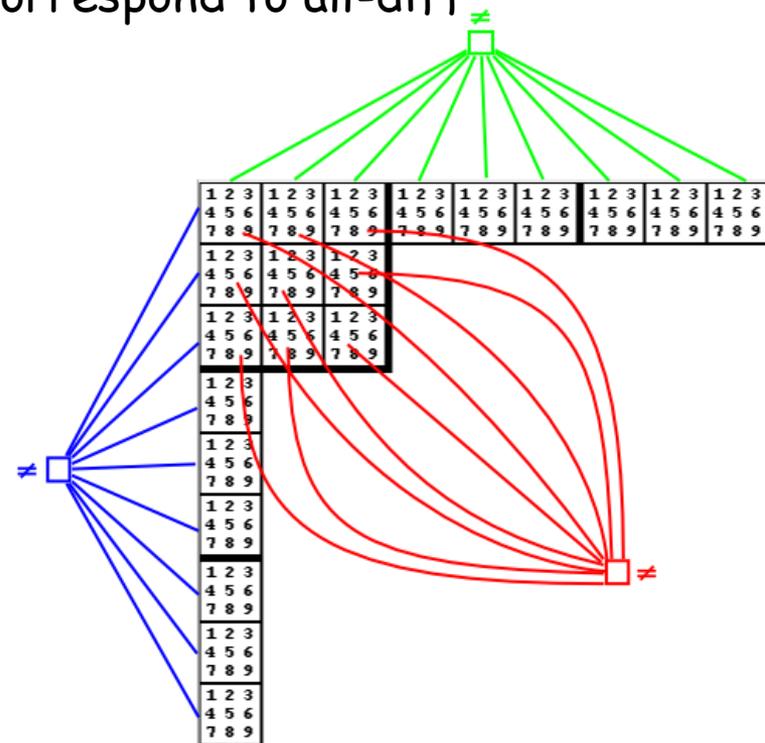
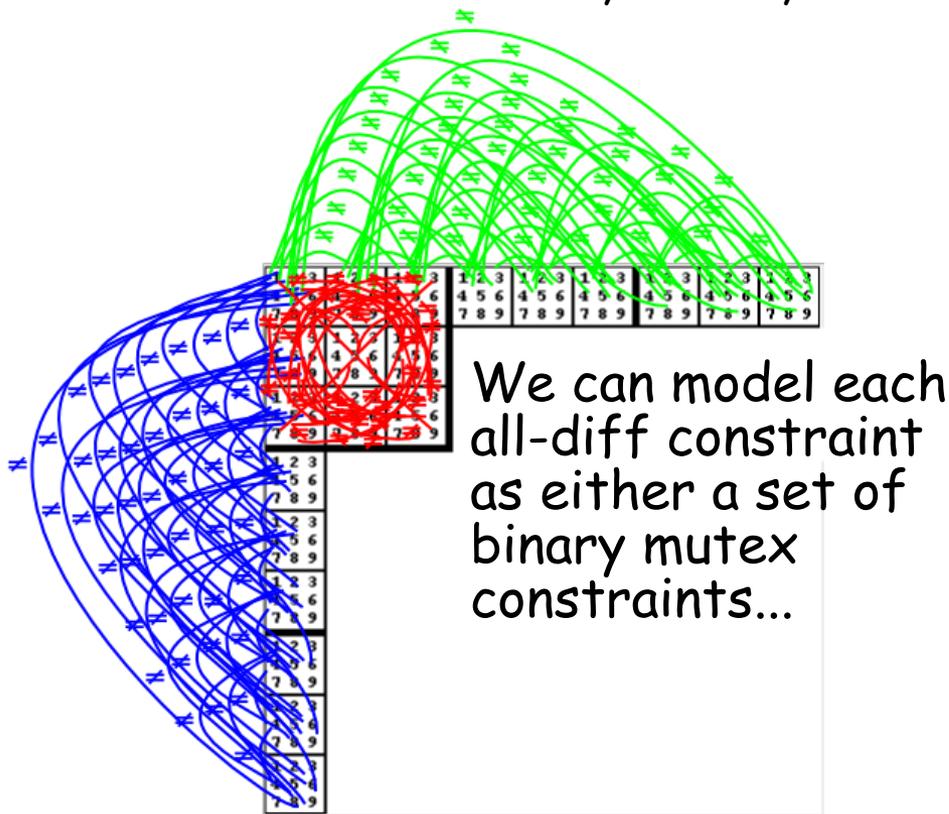


cse.unl.edu/~sudoku/Solver/ & cse.unl.edu/~sudoku/Constructor/

Our approach is to...

model Sudoku as a Constraint Satisfaction Problem (CSP) and use it to illustrate various CP techniques.

We model Sudoku as a CSP as follows. Each cell is a variable whose domain is made of the numbers 1...9. The constraints correspond to all-diff constraints on each row, column, and block.



We implement both models to allow the player to compare and understand the effectiveness of constraint propagation operating on each model.

We built a Java applet that allows the user to play Sudoku with the assistance of CP techniques. The player can...

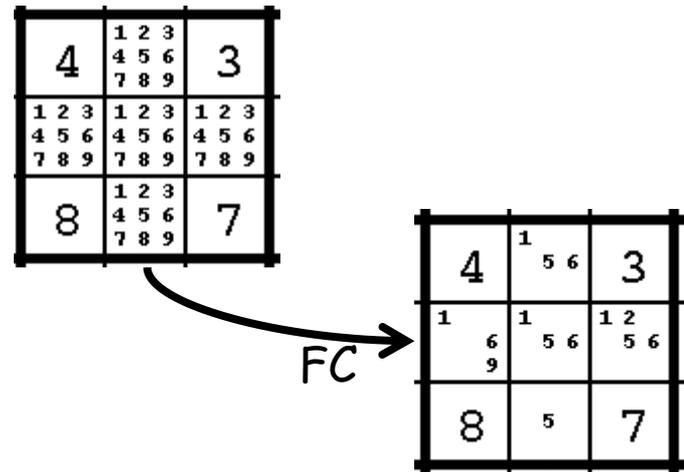


The screenshot shows a Sudoku puzzle grid with various annotations and controls:

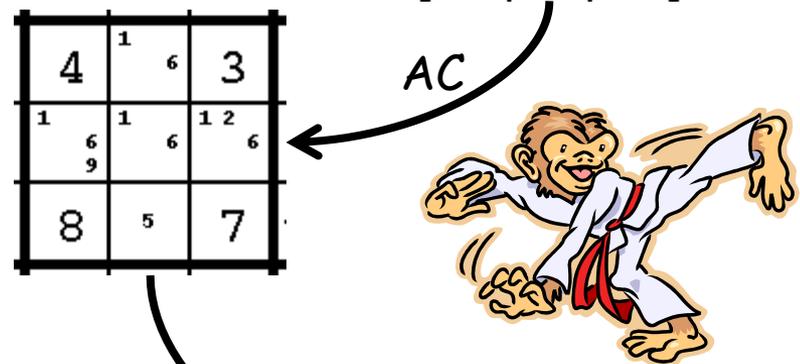
- Load Instance:** A button at the top left, circled in green, with an arrow pointing to the text "...load a problem instance from the online library".
- Interactive mode:** A panel at the bottom left with a "Show domains" checkbox checked, circled in green, and an arrow pointing to the text "...display the remaining values in the domain of each cell".
- Propagation:** A panel at the bottom center with options for "Binary" (AC, SAC) and "Non-Binary" (GAC, SGAC).
- Interactive-Search:** A panel at the bottom right with radio buttons for "None", "Binary" (BC, FC, MAC), and "Type" (Singleton, Vital).
- Undo/Redo:** Buttons in the top right, circled in green, with an arrow pointing to the text "...undo/redo any action".
- Assign Singletons:** A button in the top right, circled in green, with an arrow pointing to the text "...assign all cells whose domains have a single value".
- # Solutions:** A label in the top right, circled in green, with an arrow pointing to the text "...check the number of solutions left".
- Reset Problem:** A button in the top right.
- Grid:** A 9x9 grid with numbers and domain lists. A cell containing the number "5" is circled in green, with an arrow pointing to the text "...assign values to cells and play the game without aid".

Using the Constructor interface, the player can add new problems to the library, modify existing problems, and view all solutions of a puzzle in the grid

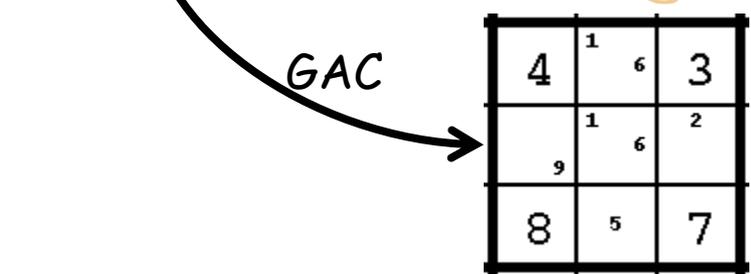
The first propagation technique we implement is Forward Checking (FC), which removes all values that are not consistent with an assigned value.



Binary Arc Consistency (AC) is stronger, iterating over every constraint until the entire problem is arc consistent.



Generalized Arc Consistency (GAC) is even stronger, considering every non-binary all-diff constraint and removing inconsistent values.



The player can run each algorithm on individual constraints, or on the entire problem.

Shaving techniques enforce a stronger level of consistency. Singleton AC (SAC) considers every value x for each cell y . The algorithm assigns x to y and runs AC. If the problem is no longer consistent (i.e., domain wipeout occurs) we remove x from the domain of y .

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



SGAC is similar to SAC, but executes GAC instead of AC.

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



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

Empirically, we have seen that SGAC solves every 9x9 Sudoku that has a unique solution.

We also implemented a Backtrack Search (BT) that enumerates and counts all possible solutions to a problem that has more than one solution (a not well posed Sudoku).





When the player makes a mistake by assigning a value that violates a constraint, Solver detects the error and highlights the variables in the broken constraint

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

If the player becomes completely stuck or wants sound advise, he or she can use the Hint feature.



The application displays the number of hints found and allows the user to iterate through them. The user can increment the level of consistency and choose what kind of hint to display.

Hint: 1 of 38 ▶

Level: SAC ▶

Type:

Singleton

Vital

A Singleton hint highlights a cell that will become a singleton after the selected consistency level is enforced.

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

A Vital hint highlights a cell that will contain a value that will not appear in any other cell in the same row, block, or column after the selected consistency level is enforced.