EXPLANATION OF SAT SOLVER

EXPLANATION OF BOOLEAN EXPRESSIONS

GRAPHS

PROBLEM STATEMENT AND HYPOTHESIS AND PROCEDURE

TITLE

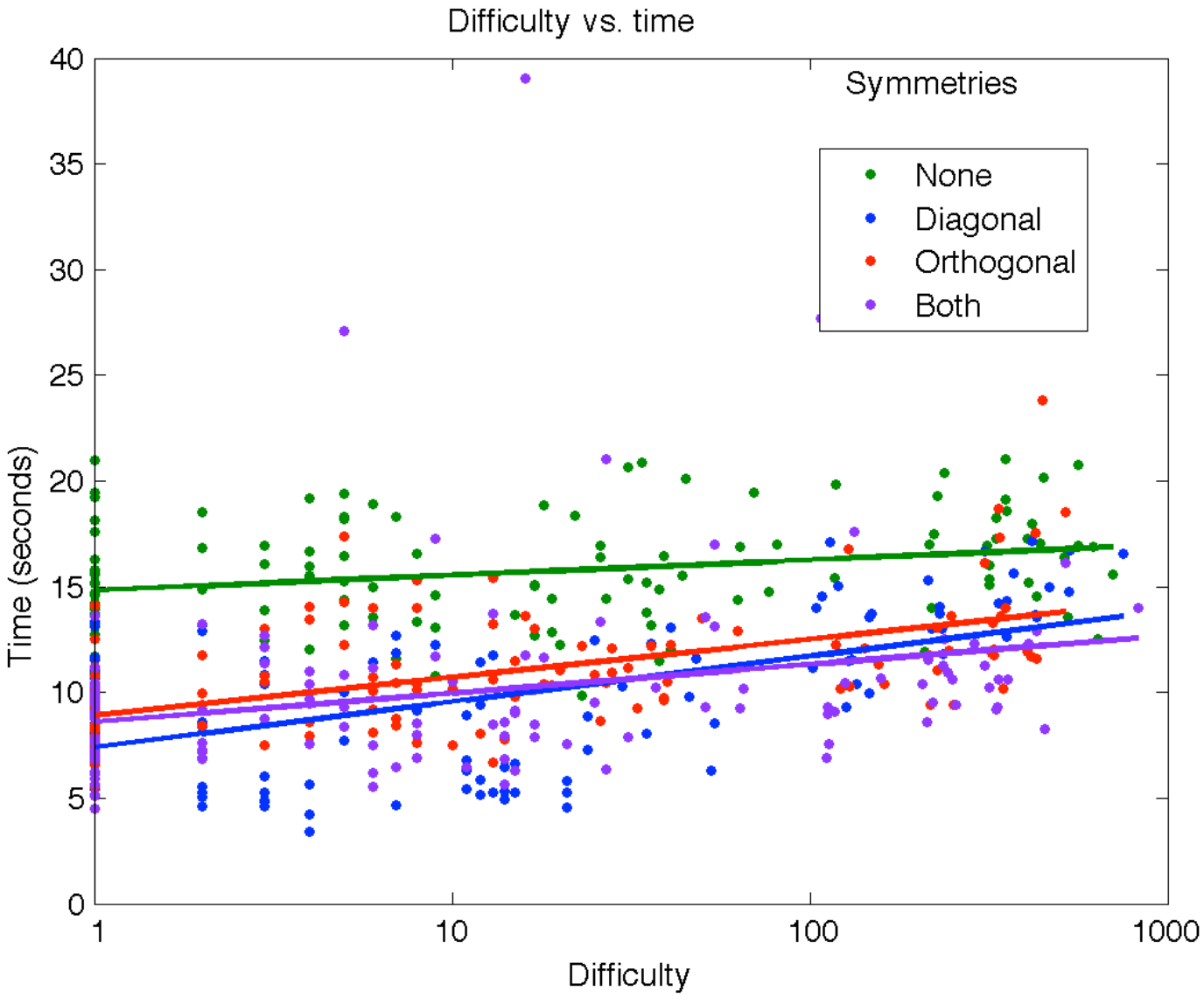
CONCLUSION AND OTHER MISCELLANEOUS STUFF

MORE GRAPHICS

EXPLANATION OF SUDOKU SOLVER

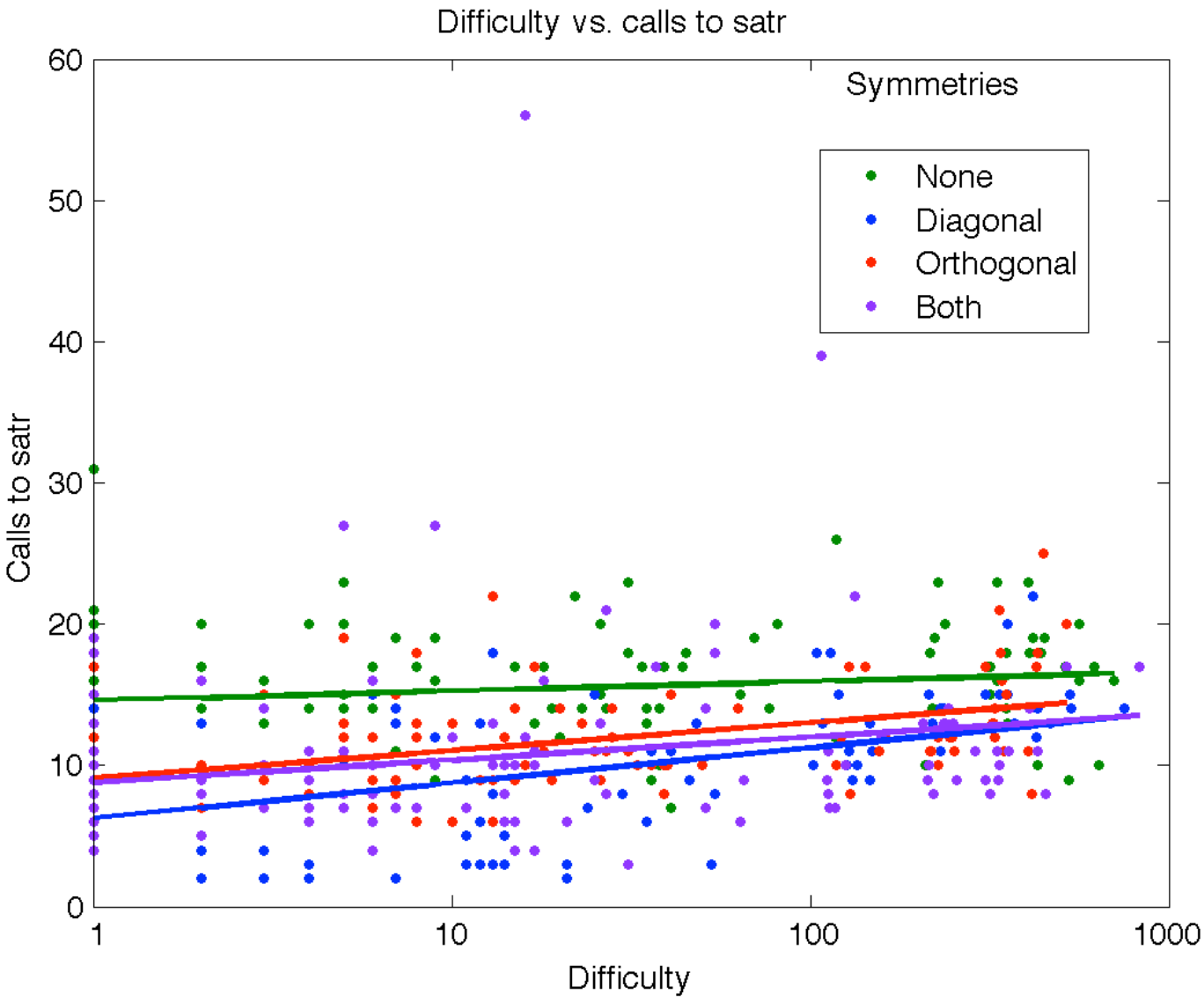
GRAPHICS

Figure 1

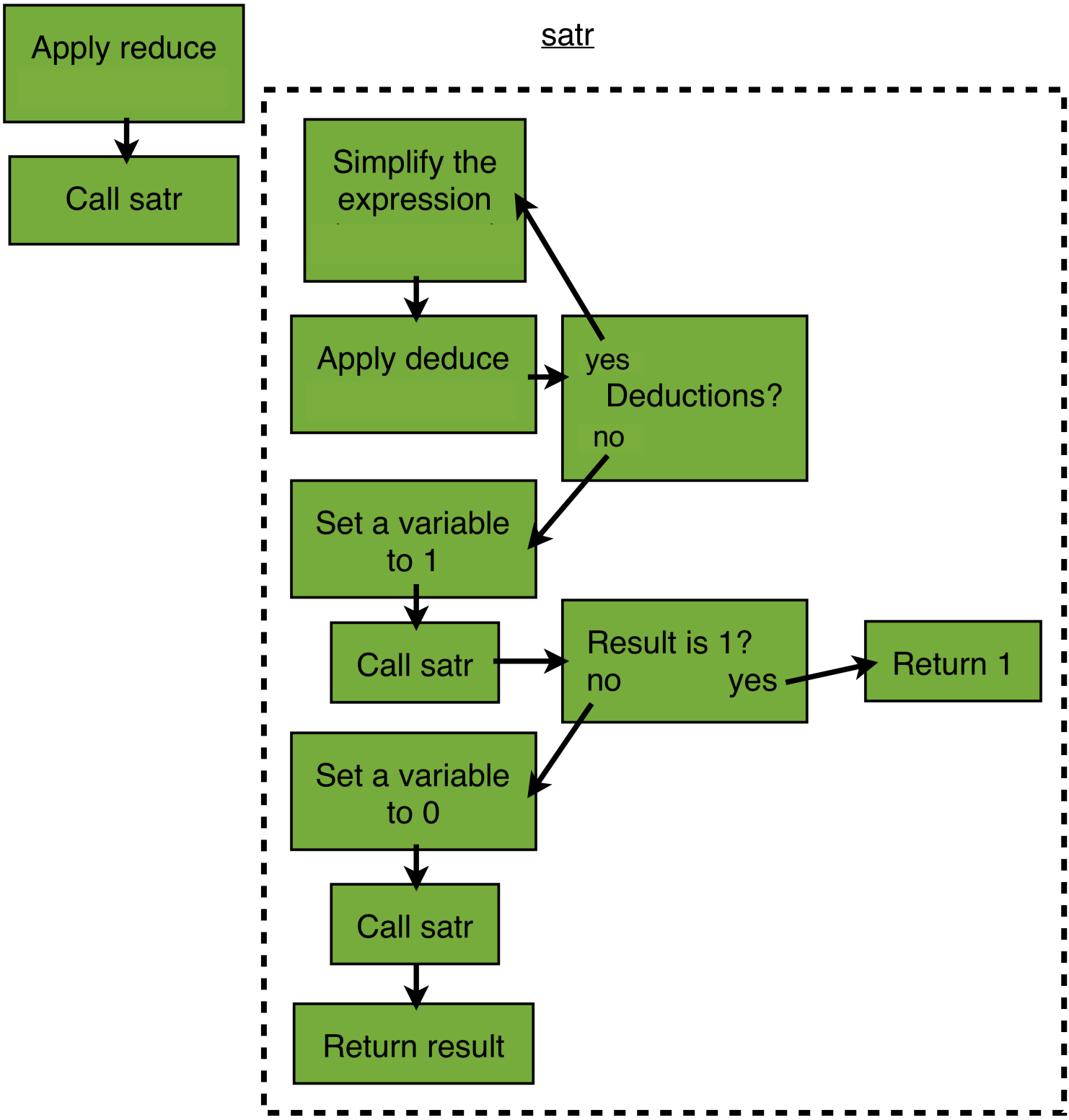


Solve time as a function of the perceived difficulty for variously symmetric Sudoku puzzles. The lines are the best fit to the data sets.

Figure 2



Number of recursions (calls to *satr*) as a function of the perceived difficulty for variously symmetric Sudoku puzzles. The lines are the best fit to the data sets.



The flow chart of my program.

Boolean Expression for Sudoku

$$\left(\bigwedge_{r=1}^{9} \bigwedge_{n=1}^{9} \bigvee_{c=1}^{9} S_{rcn} \right) \bigwedge$$ Rule for rows

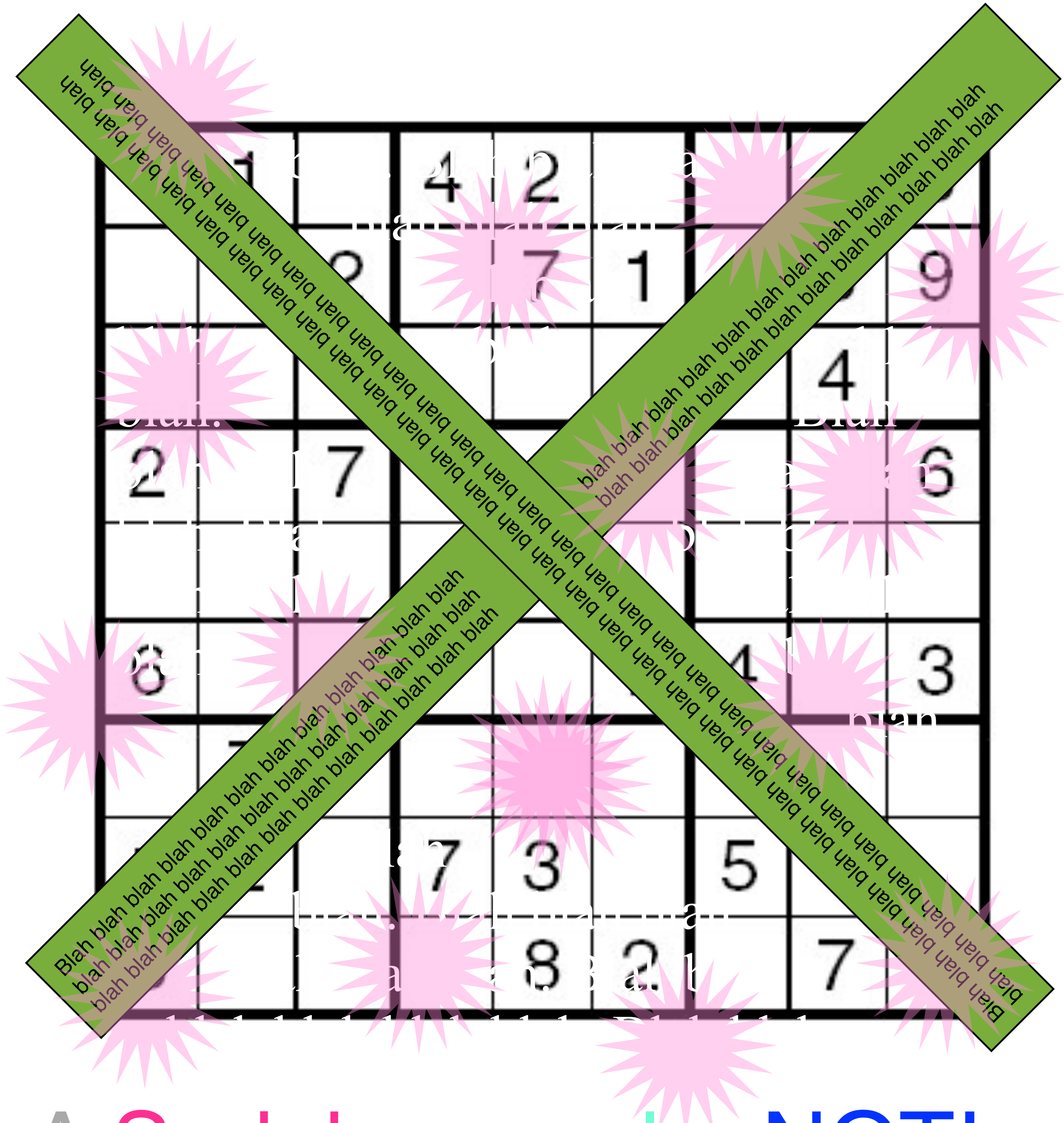
$$\left(\bigwedge_{c=1}^{9} \bigwedge_{n=1}^{9} \bigvee_{r=1}^{9} S_{rcn} \right) \bigwedge$$ Rule for columns

$$\left(\bigwedge_{r=1}^{9} \bigwedge_{c=1}^{9} \bigvee_{n=1}^{9} S_{rcn} \right) \bigwedge$$ Rule for having at least one number in each cell

$$\left(\bigwedge_{i=0}^{2} \bigwedge_{j=0}^{2} \bigwedge_{n=1}^{9} \bigvee_{r=1}^{3} \bigvee_{c=1}^{3} S_{r+3i,c+3j,n} \right)$$ Rule for 3x3 boxes

The Boolean variable S_{rcn} corresponds to whether the cell in row r and column c is equal to n.

Materials: None, other than computer.



A Sudoku puzzle. NOT!

Boolean Expressions

- A *Boolean* variable can equal either FALSE (0) or TRUE (1).

- Boolean variables use binary operators, such as OR (∨), AND (∧), and NOT (¬).

	x	y	x ∨ y
	0	0	0
OR	0	1	1
	1	0	1
	1	1	1

	x	y	x ∧ y
	0	0	0
AND	0	1	0
	1	0	0
	1	1	1

	x	¬ x
NOT	0	1
	1	0

APPLYING SAT SOLVING TO SUDOKU PUZZLES

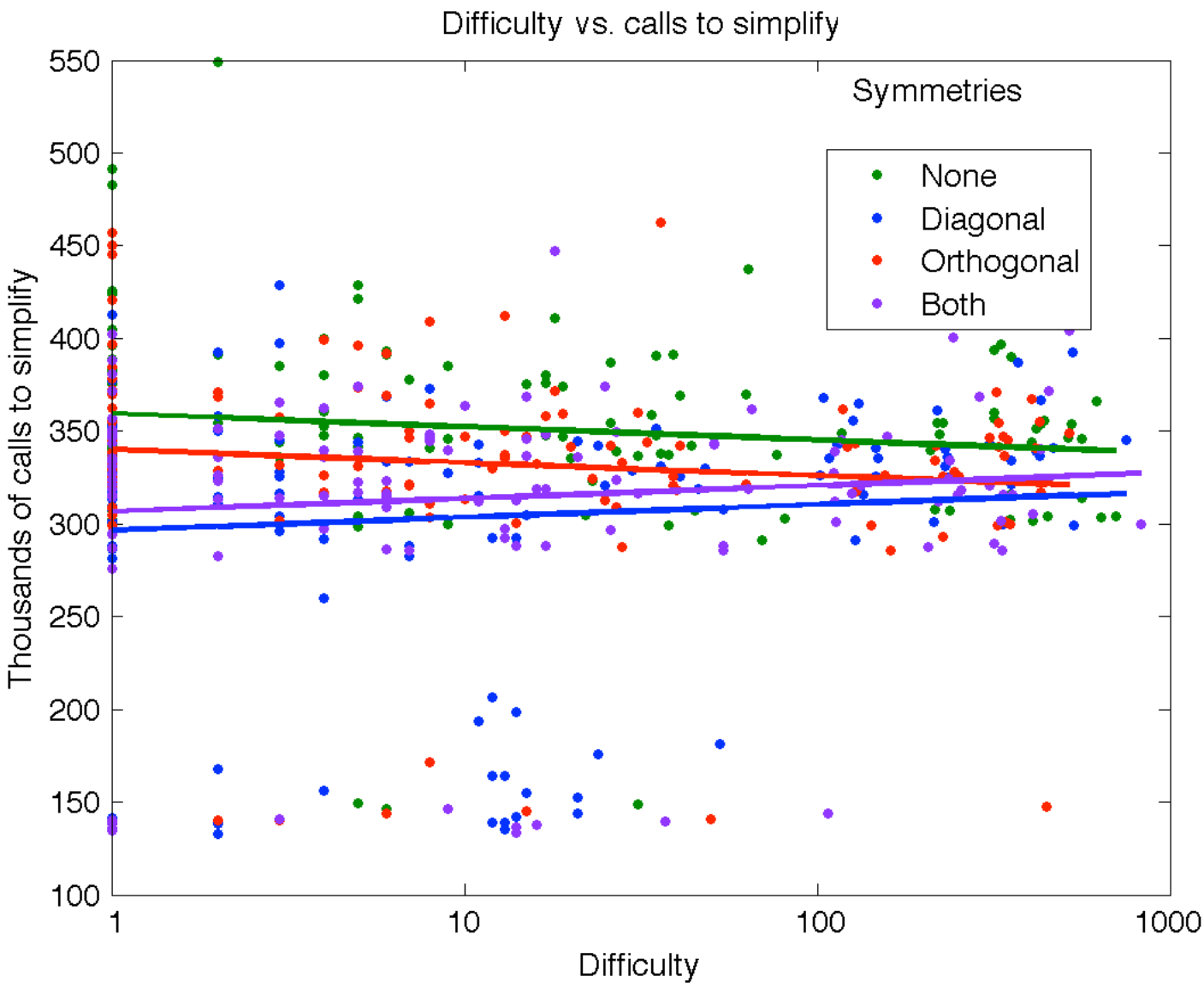
Radon Rosborough
Grade 8, Summit Charter Middle School
Boulder, Colorado

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

A diagonally symmetric Sudoku puzzle.

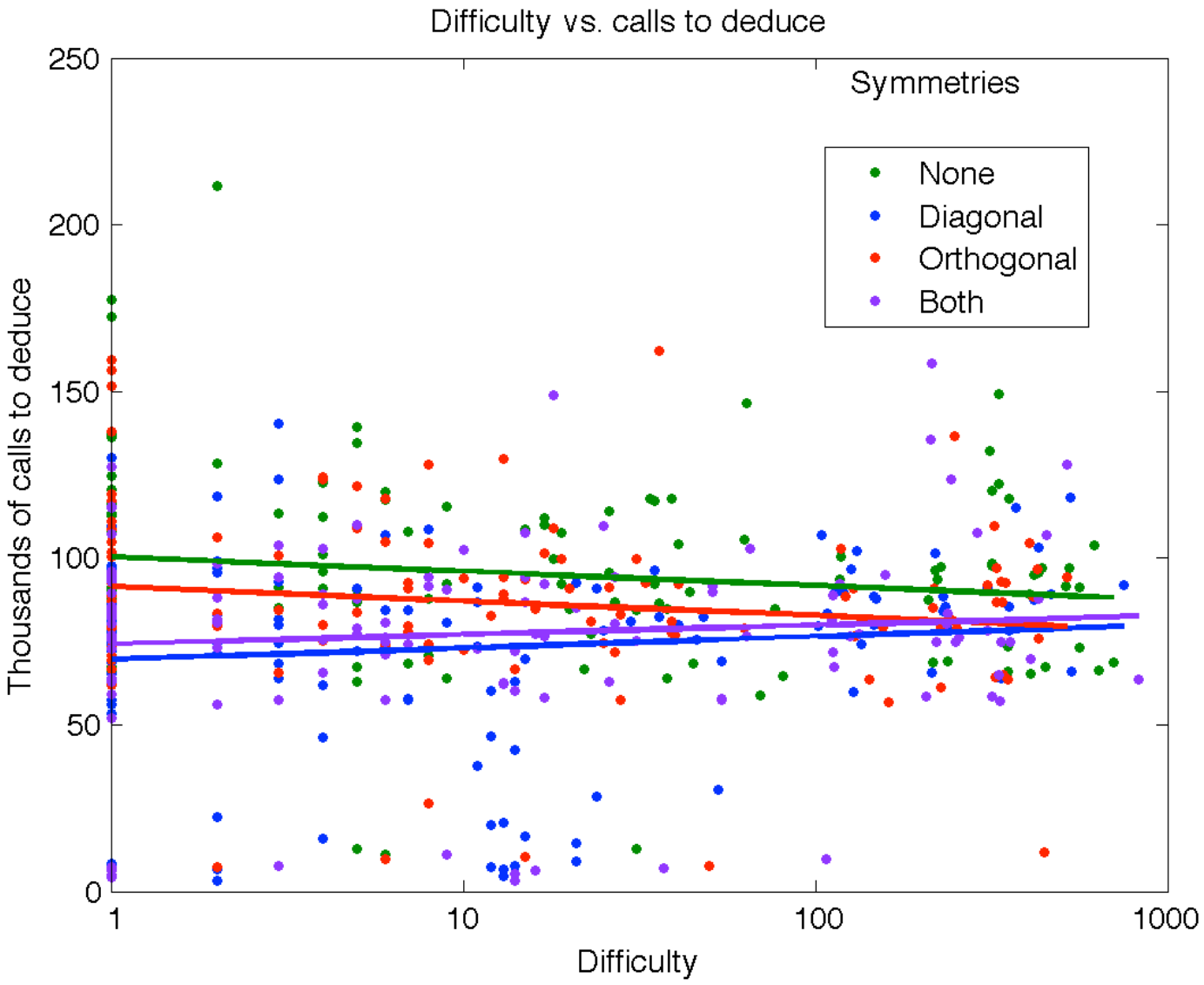
- Materials & Data
- Include program notes
- New figures
- Explain expression
- Maybe table explaining test cases?

Figure 3



Number of calls to *simplify* as a function of the perceived difficulty for variously symmetric Sudoku puzzles. The lines are the best fit to the data sets.

Figure 4



Number of calls to *deduce* as a function of the perceived difficulty for variously symmetric Sudoku puzzles. The lines are the best fit to the data sets.

reduce (deMorgan's Law):

NOT (x OR y) = NOT x AND NOT y
NOT (x AND y) = NOT x OR NOT y

NOT (x AND NOT (x OR y)) =
(NOT x) OR x OR y

simplify:

If y is TRUE, then

x AND (y OR z) =
x AND (TRUE OR z) =
x AND TRUE =
x

deduce

x AND (NOT x) AND (NOT y)

y must be FALSE
x must be both FALSE and TRUE
--> contradiction

x AND (y OR z)

x must be TRUE
y and z can be either, as long as at least one is TRUE

x AND (y OR z) =
TRUE AND (y OR z) =
y OR z