Sudoku is NP-Complete

Computability (CSCI 538)

Fall 2014

Eric DiGiovine & Nathan Woods

Montana State University

MONTANA STATE UNIVERSITY

Based on...

Lynce, Inês; Ouaknine, Jöel

"Sudoku as a SAT Problem"

http://anytime.cs.umass.edu/aimath06/proceedings/P34.pdf

Outline

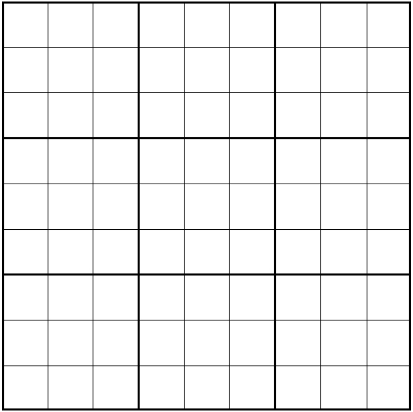
- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

Outline

- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

Sudoku Grid (9 $\times$ 9)

Represented by a 9x9 grid comprising of nine 3x3 sub-grids (boxes)



Sudoku Puzzle Rules

Fill cells with digits s.t. each:

- row,
- column,
- 3×3 "box"

contain the numbers one to nine

Note: initial cells are provided (minimum for experts is 17)

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

Solved Puzzle Instance

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

How many Sudoku puzzles are there?

Enumerating the Sudoku grid solutions directly yields:

$$6,670,903,752,021,072,936,960 \qquad (6.7 \times 10^{21})$$

Enough for every person (7 billion) to have:

$$952,986,250,288.7$$

953 billion separate puzzles per person

Not just a numbers game

Any unique set of 9 symbols works

In this example letters are used {M, R, P, E, A, B, O, D, Y}

				D			M	
		M			Y			E
		D	R			O		
Y				O				M
	E						A	
P				E				O
		E			R	M		
A			O			B		
	B			A				P

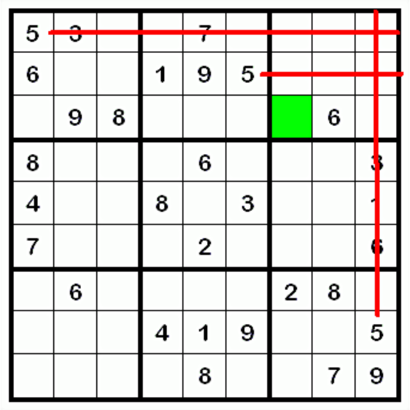
Solving Sudoku

A recent publication presents Sudoku as a Constraint Satisfaction Problem (CSP), using different propagation schemes for solving Sudoku.

But, Sudoku can also be encoded as a SAT problem

Reasoning not searching...

- Each puzzle has a unique solution and does not require the use of trial and error or guessing
- Each puzzle is solved merely with "reasoning"



NP-Complete vs. Poly-time

- The generalized Sudoku problem ($n \times n$) is NP-Complete (via reduction from Latin Squares)

- The generalized Sudoku puzzles (9×9) are poly-time solvable.

Outline

- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

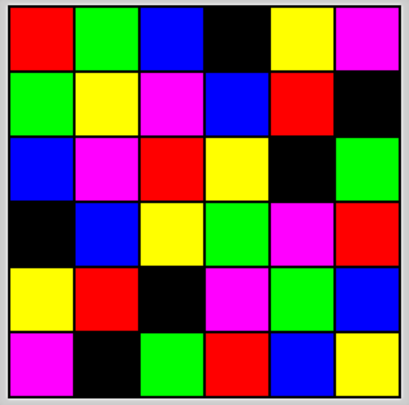
Sudoku extends LatinSquare

- Sudoku puzzles are a special case of Latin squares

- Any solution to a Sudoku puzzle is a Latin square

Latin Square

$n \times n$ array filled with n different symbols
each occurring exactly once in each row and column



More examples

$$\begin{bmatrix} 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 5 & 1 & 4 \\ 3 & 5 & 4 & 2 & 1 \\ 4 & 1 & 2 & 5 & 3 \\ 5 & 4 & 1 & 3 & 2 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 5 & 3 \\ 3 & 5 & 4 & 2 & 1 \\ 4 & 1 & 5 & 3 & 2 \\ 5 & 3 & 2 & 1 & 4 \end{bmatrix}$$

Latin Squares vs. Sudoku (9×9)

Latin Squares $5,524,751,496,156,892,842,531,225,600$
$5.5 \times 10^{27} = 5.5$ octillion

Sudoku $6,670,903,752,021,072,936,960$
$6.7 \times 10^{21} = 6.7$ sextillion

Recall

Sudoku imposes the additional restriction on Latin Squares that nine 3×3 adjacent sub squares must also contain the digits 1-9

Outline

- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

SAT Encoding for Sudoku

Relatively simple to encode, requires a lot of variables and clauses

Encoding Sudoku puzzles into CNF requires:

$$9 \times 9 \times 9 = 729 \text{ propositional variables}$$

Propositional Variables

For each entry in the 9×9 grid, we associate 9 variables

$$S_{xyz} = \text{true}$$

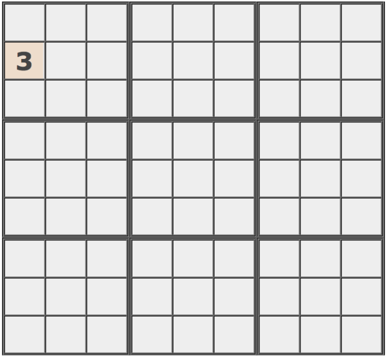
iff the entry in column x and row y is assigned number z

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

Propositional Variables

$S_{123} = \text{true}$

means $S[1,2] = 3$



Unit Clauses

Pre-assigned entries of a puzzle will be represented as unit clauses.

Unit Clause

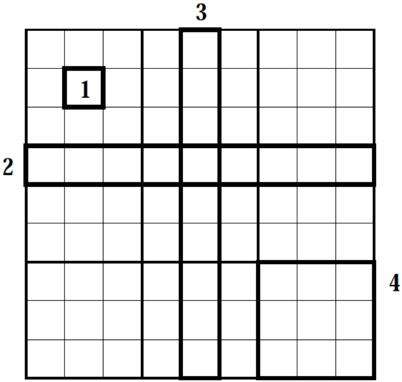
a clause that are composed of a single literal

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

S_{458} $\quad S_{546}$ $\quad S_{562}$ $\quad S_{653}$ $\quad \ldots$

Constraints

1. each entry / cell
2. each row
3. each column
4. each 3×3 sub-grid



Two types of Encodigs

Similar to the SAT encoding for partial latin squares, you can either use minimal or extended encoding.

Extended encoding adds redundant constraints (which turn out to be more intuitive)

Outline

Minimal Encoding Constraints — Each Entry

There is **at least** one number in each entry

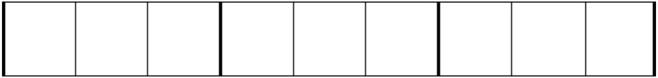
$$\bigwedge_{x=1}^{9} \bigwedge_{y=1}^{9} \bigvee_{z=1}^{9} S_{xyz}$$

$$(S_{111} \vee S_{112} \vee \cdots \vee S_{119}) \wedge \cdots \wedge (S_{991} \vee S_{992} \vee \cdots \vee S_{999})$$

Minimal Encoding Constraints — Each Row

Each number appears **at most** once in each row

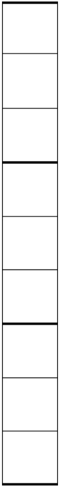
$$\bigwedge_{y=1}^{9}\bigwedge_{z=1}^{9}\bigwedge_{x=1}^{8}\bigwedge_{i=x+1}^{9}(\neg S_{xyz} \vee \neg S_{iyz})$$



Minimal Encoding Constraints — Each Column

Each number appears **at most** once in each column

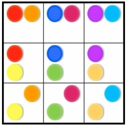
$$\bigwedge_{x=1}^{9}\bigwedge_{z=1}^{9}\bigwedge_{y=1}^{8}\bigwedge_{i=y+1}^{9}(\neg S_{xyz} \vee \neg S_{xiz})$$

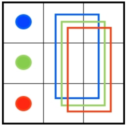


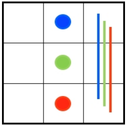
Minimal Encoding Constraints — Each 3×3 sub-grid

Each number appears **at most** once in each 3×3 sub-grid

$$\bigwedge_{z=1}^{9} \bigwedge_{i=0}^{2} \bigwedge_{j=0}^{2} \bigwedge_{x=1}^{3} \bigwedge_{y=1}^{3} \bigwedge_{k=y+1}^{3} \left[\neg S_{(3i+x)(3j+y)z} \vee \neg S_{(3i+x)(3j+k)z} \right]$$







$$\bigwedge_{z=1}^{9} \bigwedge_{i=0}^{2} \bigwedge_{j=0}^{2} \bigwedge_{x=1}^{3} \bigwedge_{y=1}^{3} \bigwedge_{k=x+1}^{3} \bigwedge_{\ell=1}^{3} \left[\neg S_{(3i+x)(3j+y)z} \vee \neg S_{(3i+k)(3j+\ell)z} \right]$$

Outline

- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

Extended Encoding — Each Entry

At most one number in each entry

$$\bigwedge_{x=1}^{9} \bigwedge_{y=1}^{9} \bigwedge_{z=1}^{8} \bigwedge_{i=z+1}^{9} \left[\neg S_{xyz} \vee \neg S_{xyi}\right]$$

Extended Encoding — Each Row + Column

Each number appears **at least** once in each:

Row

$$\bigwedge_{y=1}^{9} \bigwedge_{z=1}^{9} \bigvee_{x=1}^{9} S_{xyz}$$

Column

$$\bigwedge_{x=1}^{9} \bigwedge_{z=1}^{9} \bigvee_{y=1}^{9} S_{xyz}$$

Extended Encoding — Each 3×3 sub-grid

Each number appears **at least** once in each 3×3 sub-grid:

$$\bigvee_{z=1}^{9} \bigwedge_{i=0}^{2} \bigwedge_{j=0}^{2} \bigwedge_{x=1}^{3} \bigwedge_{y=1}^{3} S_{(3i+x)(3j+y)z}$$

1	1	1
1	1	1
1	1	1

Number of Clauses

Minimal	$8,829$ clauses
Extended	$11,988$ clauses

Size of encoding is $O(n^4)$,
where n is the order of the Sudoku puzzle ($n = 9$)

Outline

Outline

- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

Resolution

Resolution is a well known inference technique for satisfiability

Definition

Each variable x is eliminated by applying resolution between each pair of clauses containing x and $\neg x$

Resolution Details

Either the empty clause is derived and the formula is un-satisfiable, or a set of clauses containing only pure literals is obtained and is satisfiable

Pure Literal: A literal is said to be pure whenever its variable only occurs in either positive or negative form throughout the formula

Restricted Resolution

In general resolution is not feasible

Different algorithms for simplifying propositional formulas restrict the application of resolution to only simplified forms

Resolution Example

Knowledge Base

$$\omega_r = (x_i \vee \alpha)$$
$$\omega_s = (\neg x_i \vee \alpha)$$

Example

$$\begin{aligned}\omega_t &= \rho(\omega_r, \omega_s, x_i) \\ &= \rho((x_i \vee \alpha), (\neg x_i \vee \alpha), x_i) \\ &= (\alpha)\end{aligned}$$

Outline

- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

Unit Propagation

Unit propagation is a simplification rule in backtrack-search SAT algorithms

The procedure is based on **unit clauses**, i.e. clauses that are composed of a single literal. If a set of clauses contains the unit clause ℓ, the other clauses are simplified by the application of the following rules:

1. Every clause containing ℓ is removed
2. Remove $\neg\ell$ from all clauses

Unit Propagation Example

$$\{a \vee b, \qquad \neg a \vee c, \qquad \neg c \vee d, \qquad a\}$$

Original Clause:	$a \vee b$	$\neg a \vee c$	$\neg c \vee d$	a
	(removed)	($\neg a$ deleted)	(n/a)	(n/a)
Modified Clause:		c	$\neg c \vee d$	a

Outline

- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

Other Inference Rules

Failed Literal Rule

If forcing an assignment to a variable $x = v(x)$ (where $v(x) \in \{0, 1\}$) and then performing unit propagation yields a conflict (when a unit clause is derived), then $x = 1 - v(x)$ is a necessary assignment.

Binary Failed Literal Rule

If the assignments $x_i = v(x_i)$ and $x_j = v(x_j)$ yield a conflict, then add the clause $(x_i = 1 - v(x_i) \vee x_j = 1 - v(x_j))$ to the formula.

Hyper-binary Resolution Rule

Given

$(\neg \ell_1 \vee x_i) \wedge (\neg \ell_2 \vee x_i) \wedge \cdots \wedge (\neg \ell_k \vee x_i) \wedge (\ell_1 \vee \ell_2 \vee \cdots \vee \ell_k \vee x_j)$

infer $(x_i \vee x_j)$.

Outline

- Sudoku Background
- Latin Squares vs. Sudoku
- Sudoku as a SAT Problem
  - Minimal Encoding
  - Extended Encoding
- SAT Inference Techniques
  - Resolution
  - Unit Propagation
  - Other Inference Rules
- SAT Experiments

Experiments

24,260 very hard Sudoku puzzles

Each puzzle contained only 17 pre-assigned entries

Configurations

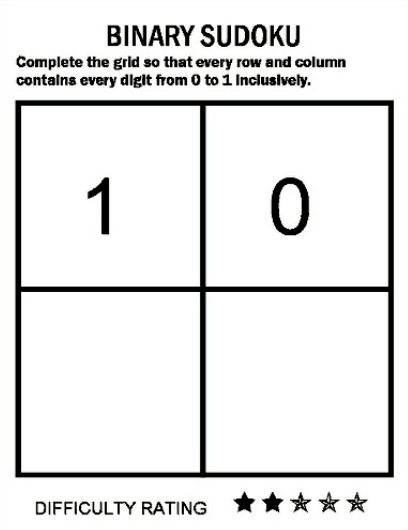
1. Unit propagation
2. Unit propagation + Failed literal rule
3. Unit propagation + Binary failed literal rule
4. Unit propagation + Hyper-binary resolution

Conclusions

Puzzles are meant to be solved without search
(merely with reasoning)

Results show Unit Propagation is not powerful enough to solve all the puzzles, but more sophisticated (yet poly-time) inference techniques are very successful in solving them

Questions


BINARY SUDOKU
Complete the grid so that every row and column contains every digit from 0 to 1 inclusively.
1
0
DIFFICULTY RATING

References

Lynce, Inês; Ouaknine, Jöel

"Sudoku as a SAT Problem"

http://anytime.cs.umass.edu/aimath06/proceedings/P34.pdf