

Modeling Sudoku as a CNF Formula

Spring 2019

CSCE 235H Introduction to Discrete Structures

URL: cse.unl.edu/~cse235h

All questions: [Piazza](#)

Compact CNF Notation

- Iterated conjunction operator:

$$a_1 \wedge a_2 \wedge \dots \wedge a_{n-1} \wedge a_n = \bigwedge_{i=1}^n a_i$$

- Iterated disjunction operator:

$$a_1 \vee a_2 \vee \dots \vee a_{n-1} \vee a_n = \bigvee_{i=1}^n a_i$$

Sudoku

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

Rules

- Each cell filled with a number 1...9
- All nine numbers appear in each
 - row,
 - column, and
 - 3x3 block(i.e., no duplicates)

Sudoku

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

Rules

- Each cell filled with a number 1...9
- All nine numbers appear in each
 - row,
 - column, and
 - 3x3 block(i.e., no duplicates)

Defining the Variables

- $p(i, j, n)$ asserts that
 - the cell in row i
 - column j
 - is assigned value n

$p(2, 5, 1)$
 $n=1$

$j = 5$

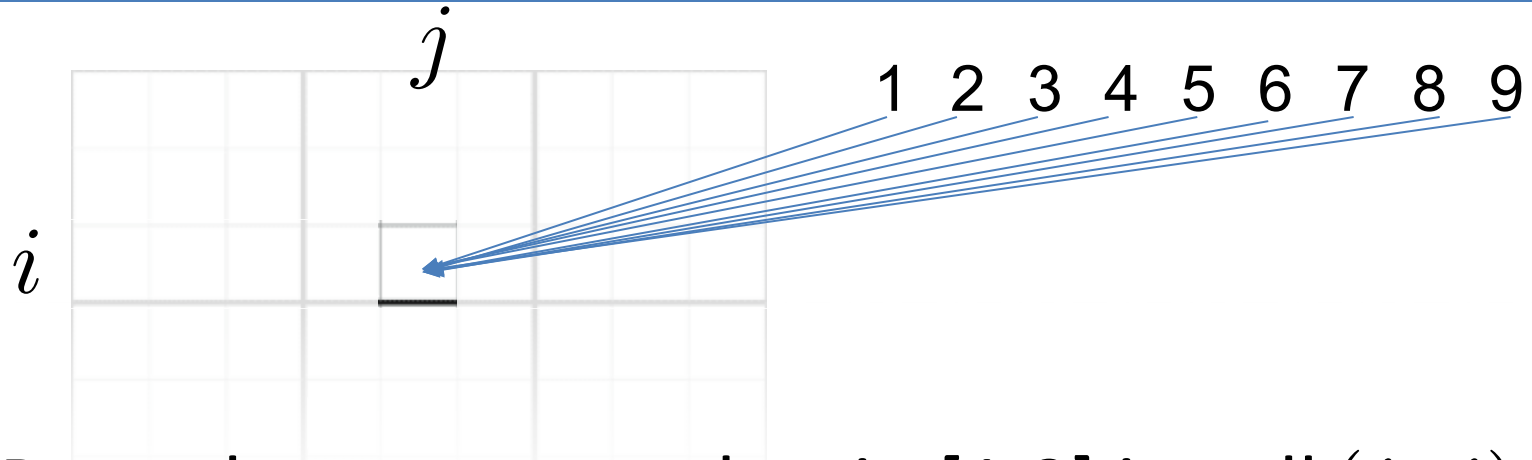
$i = 2$

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

Rules of the Game as Clauses

- Putting numbers in cells
 - Put a number in a given cell / Put a number in every cell
 - A given cell cannot take two numbers / No cell can take two numbers
- Constraining rows
 - Every number must appear in every row
- Constraining columns
 - Every number must appear in every column
- Constraining 3x3 blocks
 - Every number must appear in every 3x3 block

Putting Numbers in Cells (1)



- Put at least one number in $[1,9]$ in cell (i, j)

$$\begin{aligned} p(i, j, 1) \vee p(i, j, 2) \vee \dots \vee p(i, j, 9) \\ \equiv \bigvee_{n=1}^9 p(i, j, n) \end{aligned}$$

Putting Numbers in Cells (2)

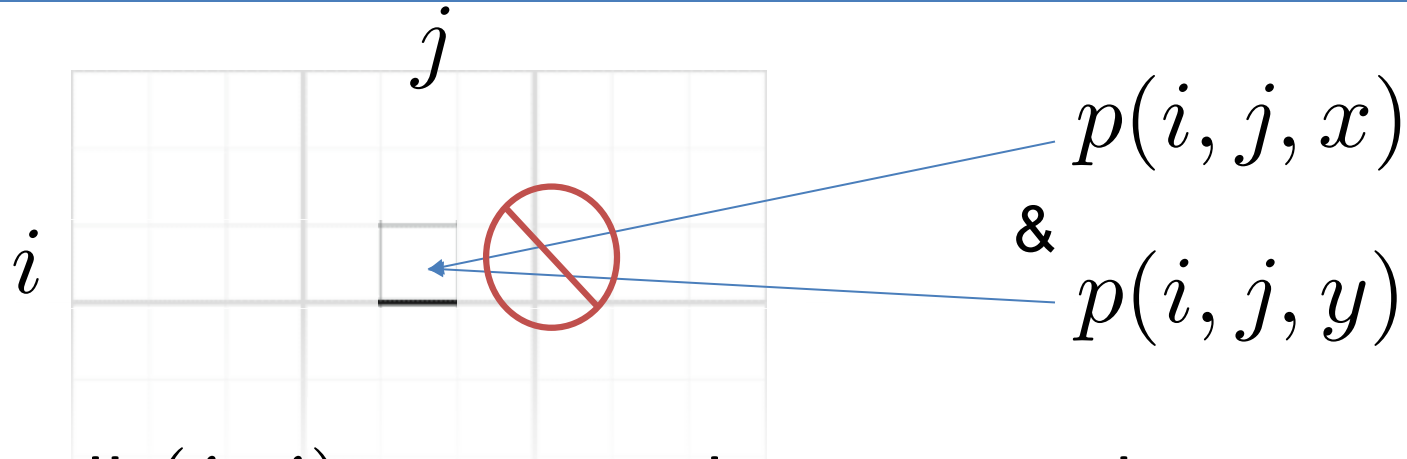
- Put a number in cell (i, j)

$$\bigvee_{n=1}^9 p(i, j, n)$$

- Put a number in every cell

$$\bigwedge_{i=1}^9 \bigwedge_{j=1}^9 \bigvee_{n=1}^9 p(i, j, n)$$

Putting Numbers in Cells (3)



- Cell (i, j) cannot take two numbers

$$\neg(p(i, j, 1) \wedge (p, i, j, 2)) \wedge \neg(p(i, j, 1) \wedge (p, i, j, 3)) \wedge \dots \wedge \neg(p(i, j, 1) \wedge (p, i, j, 9)) \wedge \\ \neg(p(i, j, 2) \wedge (p, i, j, 3)) \wedge \neg(p(i, j, 2) \wedge (p, i, j, 4)) \wedge \dots \wedge \neg(p(i, j, 2) \wedge (p, i, j, 9)) \wedge$$

$$\forall x, y (x \neq y \rightarrow \neg(p(i, j, x) \wedge p(i, j, y)))$$

$$\equiv \forall x, y (x \neq y \rightarrow (\neg p(i, j, x) \vee \neg p(i, j, y)))$$

Putting Numbers in Cells (4)

- Order of the literals does not matter so

$\forall x, y(x \neq y \rightarrow \dots)$ becomes $\forall x, y(x < y \rightarrow \dots)$

$$x = 1, y = 2, 3, 4, 5, 6, 7, 8, 9$$

$$x = 2, y = 3, 4, 5, 6, 7, 8, 9$$

$$x = 3, y = 4, 5, 6, 7, 8, 9$$

$$x = 4, y = 5, 6, 7, 8, 9$$

$$x = 5, y = 6, 7, 8, 9$$

$$x = 6, y = 7, 8, 9$$

$$x = 7, y = 8, 9$$

$$x = 8, y = 9$$

$$\bigwedge_{x=1}^8 \bigwedge_{y=x+1}^9 (\neg p(i, j, x) \vee \neg p(i, j, y))$$

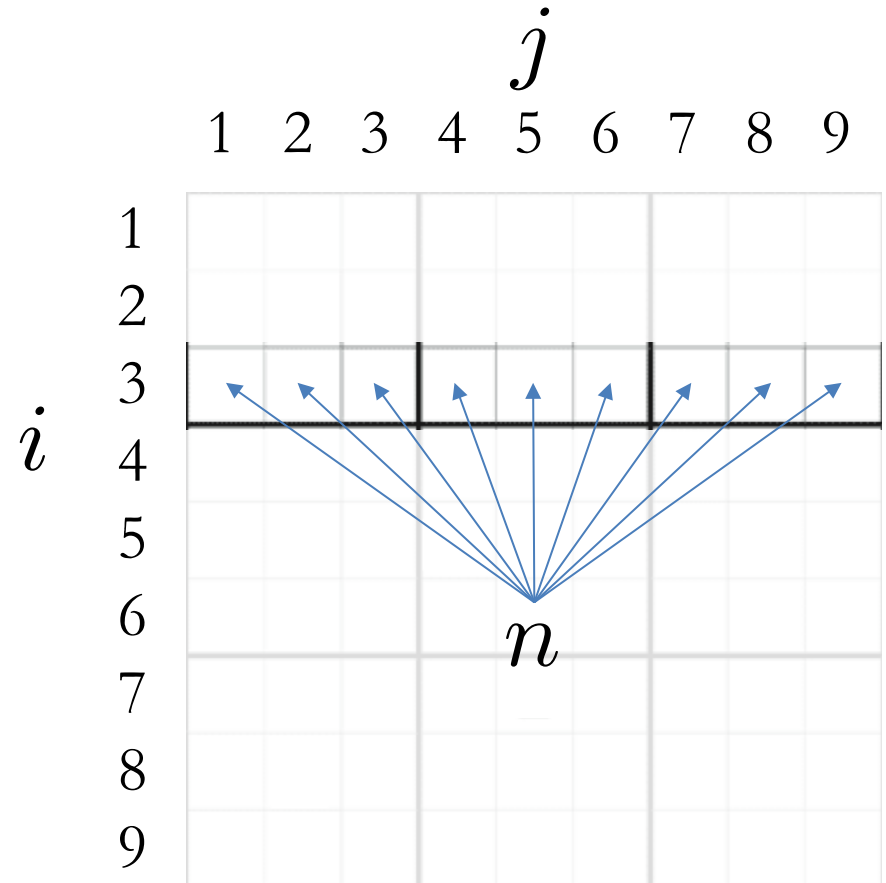
Putting Numbers in cells (5)

- Every cell takes at most one number

$$\bigwedge_{i=1}^9 \bigwedge_{j=1}^9 \bigwedge_{x=1}^8 \bigwedge_{y=x+1}^9 (\neg p(i, j, x) \vee \neg p(i, j, y))$$

Constraining Rows (1)

- For every
 - Row
 $i \in \{1, \dots, 9\}$
 - Number
 $n \in \{1, \dots, 9\}$
 - Select some column
 $j \in \{1, \dots, 9\}$



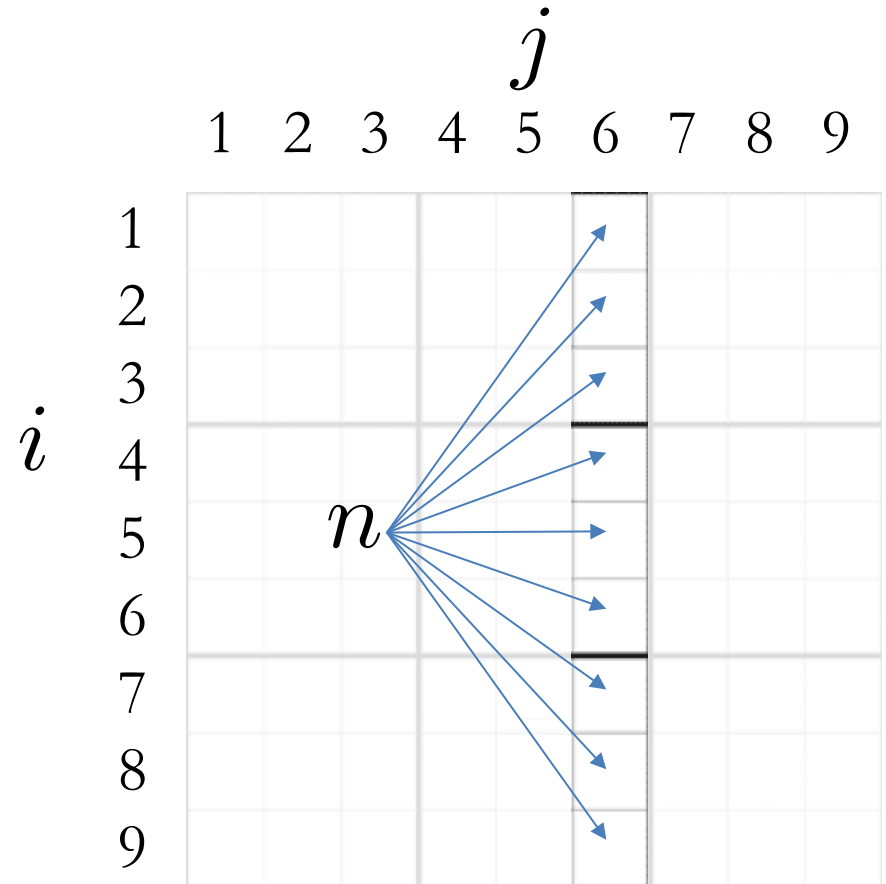
Constraining Rows (2)

- Every number must appear in every row

$$\bigwedge_{i=1}^9 \bigwedge_{n=1}^9 \bigvee_{j=1}^9 p(i, j, n)$$

Constraining Columns (1)

- For every
 - Column
 $j \in \{1, \dots, 9\}$
 - Number
 $n \in \{1, \dots, 9\}$
 - Select some row
 $i \in \{1, \dots, 9\}$



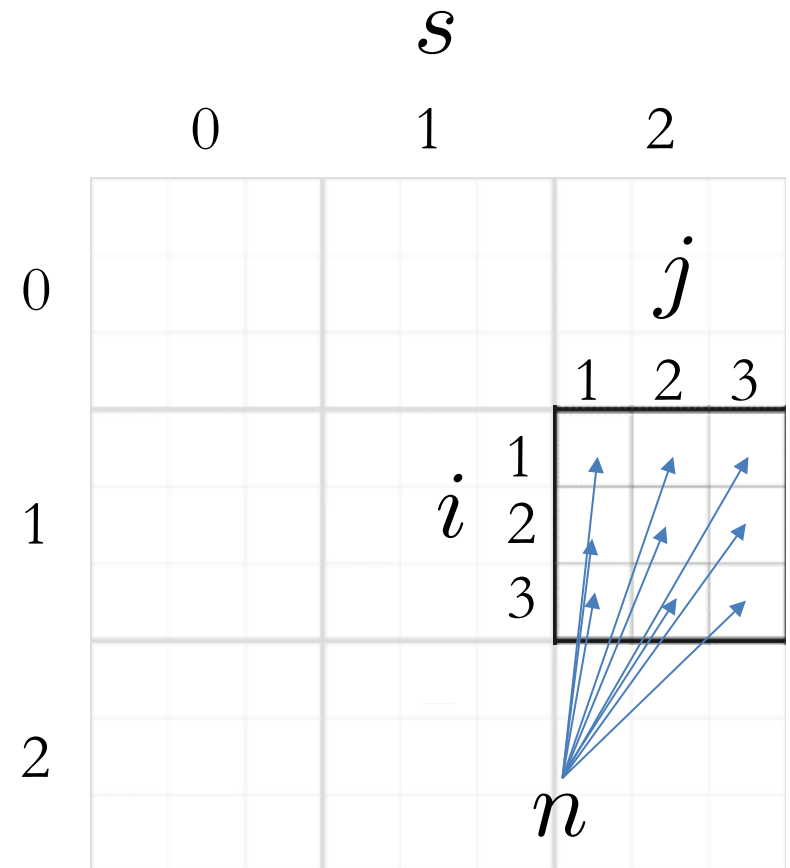
Constraining Columns (2)

- Every number must appear in every column

$$\bigwedge_{j=1}^9 \bigwedge_{n=1}^9 \bigvee_{i=1}^9 p(i, j, n)$$

Constraining 3x3 Blocks (1)

- For every
 - Block
 - $(r, s) \in \{0, 1, 2\} \times \{0, 1, 2\}$
 - Number
 - $n \in \{1, \dots, 9\}$
 - Select some cell in block
 - $(i, j) \in \{1, 2, 3\} \times \{1, 2, 3\}$



Constraining 3x3 Blocks (2)

- Every number must appear in every 3x3 block

$$\bigwedge_{r=0}^2 \bigwedge_{s=0}^2 \bigwedge_{n=1}^9 \bigvee_{i=1}^3 \bigvee_{j=1}^3 p(3r + i, 3s + j, n)$$

Sudoku CNF Formula

- 729 variables
- 324 clauses with 9 literals
- 2916 clauses with 2 literals
- 3240 total clauses (+ clauses for initial setup)

Redundant Clauses

- We can model a Sudoku as a SAT sentence in many ways
- Some models may have redundant clauses
 - Removing redundant clauses does not change the problem and yields an equivalent formula
 - Example: we can generate redundant clauses by applying inference rules on some clauses in the model
- Redundant clauses
 - Do not change the model
 - May speed the solving process.. Useful!

Declaring the Hints/Clues

- We add propositions as unary clauses corresponding to hints/clues (filled cells)

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

⋮
 $\wedge p(2, 2, 8)$
 $\wedge p(2, 3, 9)$
 $\wedge p(2, 4, 4)$
 $\wedge p(2, 5, 1)$
 $\wedge p(3, 3, 6)$
 $\wedge \dots$

Solving Sudoku

- Total number of possible assignments

$$2^{729} = 2.824014 \times 10^{219}$$

- Testing one billion assignments a second

$$8.94876 \times 10^{202} \text{ years}$$

- SAT solvers solve a Sudoku in milliseconds by aggressively pruning the search tree