Logic and Proofs

Chapter 1

Propositional Logic

Section 1.1

Video 1: Propositional Logic and Basic Logical Connectives

- Propositional Logic
- Propositions
- Basic Logical Connectives
  - Negation
  - Conjunction
  - Disjunction
- Truth tables

What is Logic?

- Language of mathematics
- Basis for automated reasoning
- Makes human language precise

- **Logic is about statements that are either true or false**

- **Propositional logic** is the most basic form of logic

Some facts on Propositional Logic

- Propositional logic allows to
  - Formulate search queries in Google
  - Describe computer circuits
  - Formally describe games, like Sudoku
  - Specify properties of software systems etc

- Propositional logic exhibits problems we have in natural language with interpreting expressions such as “or”, “if … then”

- Anything expressed in propositional logic can be **automatically decided** whether it is true or false

Propositions

- A **proposition** is a declarative sentence that is either **true** or **false**

- Examples of propositions
  a) The Moon is made of green cheese.
  b) The earth is round.
  c) $1 + 0 = 1$
  d) $0 + 0 = 2$

Propositions

- A **proposition** is a declarative sentence that is either **true** or **false**

- Examples that are not propositions
  a) Sit down!
  b) What time is it?
  c) $x + 1 = 2$
  d) $x + y = z$

Language of Propositional Logic: Atomic Propositions

Propositions that cannot be expressed in terms of simpler propositions are called **atomic propositions**

- Letters denote **Propositional Variables**: *p*, *q, r*, *s*, …
  - Example: p denotes "The Earth is round."

- The proposition that is always **true** is denoted by **T**
- The proposition that is always **false** is denoted by **F**

Language of Propositional Logic: Compound Propositions

- **Compound Propositions** are constructed from **logical connectives** and other propositions
  - Negation $\neg$
  - Conjunction $\wedge$
  - Disjunction $\vee$
  - Implication $\rightarrow$
  - Biconditional $\leftrightarrow$

Negation

Let p be a proposition. The *negation of* p, denoted by $\neg p$ (also denoted by $\overline{p}$), is the statement

"It is not the case that p."

The proposition $\neg p$ is read "not p." The truth value of the negation of p, $\neg p$, is the opposite of the truth value of p.

Example:

If the letter *p* denotes "The earth is round."
then $\neg p$ denotes "It is not the case that the earth is round,"
(or more simply "The earth is not round.")

Truth Tables

A **truth table** lists all possible truth values of the propositional variables occurring in a compound proposition, and the corresponding truth values of the compound proposition

TABLE 1 The Truth Table for the Negation of a Proposition.

p	$\neg p$
T	F
F	T

Conjunction

Let p and q be propositions. The *conjunction* of p and q, denoted by $p \wedge q$, is the proposition "p and q." The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

Example:

If the letter p denotes "I am at home." and q denotes "It is raining." then $p \wedge q$ denotes "I am at home and it is raining."

TABLE 2 **The Truth Table for the Conjunction of Two Propositions.**

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction

Let p and q be propositions. The *disjunction* of p and q, denoted by $p \vee q$, is the proposition "p or q." The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

Example:

If the letter p denotes "I am at home." and q denotes "It is raining." then $p \vee q$ denotes "I am at home or it is raining."

TABLE 3 The Truth Table for the Disjunction of Two Propositions.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

The Connective Or in Natural Language

- In natural language "or" has two distinct meanings
  - "Inclusive Or" - called also disjunction.
    For $p \vee q$ to be true, either one or both of p and q must be true.
  - "Exclusive Or" - called also Xor.
    In $p \oplus q$, one of p and q must be true, but not both.
- The truth table for $\oplus$ is:

TABLE 4 **The Truth Table for the Exclusive Or of Two Propositions.**

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Example

- Inclusive or

  In the sentence “Students who have taken CS202 or Math120 may take this class,” we assume that students need to have taken one of the prerequisites, but may have taken both.

- Exclusive or

  When reading the sentence “Soup or salad comes with this entrée,” we do not expect to be able to get both soup and salad.

Summary

- Propositional logic
- Proposition
- Atomic and Compound proposition
- Truth table
- Negation, Disjunction, Conjunction
- Exclusive or

- Next: Implication and more on compound propositions

Video 2: Implication and Compound Propositions

- Logical connectives
  - Implication
  - Biconditional
- Compound propositions
  - Precendence
  - Truth tables
  - Tautologies, Contradictions, and Contingencies

Implication

Let p and q be propositions. The *conditional statement* $p \to q$ is the proposition "if p, then q." The conditional statement $p \to q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \to q$, p is called the *hypothesis* (or *antecedent* or *premise*) and q is called the *conclusion* (or *consequence*).

Example: If p denotes "I am at home." and q denotes "It is raining." then $p \to q$ denotes "If I am at home, then it is raining."

TABLE 5 The Truth Table for the Conditional Statement $p \to q$.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Understanding Implication

Implication does not require any connection between the antecedent and the consequent

Example: These implications are perfectly fine, but would not be used in ordinary English

- "If the moon is made of green cheese, then I have more money than Bill Gates. "
- "If the moon is made of green cheese, then I'm on welfare."

Understanding Implication

- One way to view the logical conditional is to think of an obligation or contract
  - Politician: “If I am elected, then I will lower taxes.”

- If the politician is elected and does not lower taxes, then the voters can say that he or she has broken the campaign pledge.
- If the politician is not elected, no one cares …

Understanding Implication

- If p is false, a conditional statement p → q is always true
  - "If the moon is made of green cheese, then …."

- If q is true, a conditional statement p → q is always true
  - "If …., then 1+1 = 2"

Properties of Implication

$p \rightarrow q$ is different from $q \rightarrow p$: this is a common logical fallacy

- “If the moon is made of green cheese, then the earth is round”
- “If the earth is round, then the moon is made of green cheese”

Implication in Natural Language

- Conditional statements are at the heart of any logical reasoning
- Therefore you find many way how they are expressed

"if p, then q"
"if p, q"
"p is sufficient for q"
"q if p"
"q when p"
"a necessary condition for p is q"
"q unless $\neg p$"

"p implies q"
"p only if q"
"a sufficient condition for q is p"
"q whenever p"
"q is necessary for p"
"q follows from p"
"q provided that p"

Implication in Mathematical Language

- In mathematics $p \rightarrow q$ is often formulated as

- A **necessary condition** for p is q
- A **sufficient condition** for q is p

- What if p is a necessary and sufficient condition for q?

Biconditional

Let p and q be propositions. The *biconditional statement* $p \leftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called *bi-implications*.

Example: If p denotes "I am at home." and q denotes "It is raining." then $p \leftrightarrow q$ denotes "I am at home if and only if it is raining."

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Expressing the Biconditional

- Some alternative ways "*p* if and only if *q*" is expressed in English:

  - *p* **is necessary and sufficient for** *q*
  - **if** *p* **then** *q* **, and conversely**
  - *p* **iff** *q*

Biconditional in Natural Language

- In natural language the biconditional is often implicit

“If you finish your meal, then you can have a dessert”

Precendence in Compound Propositions

- We can compose complex logical expressions from simpler ones

$$\neg(p \vee q)$$
$$(\neg p) \vee q$$

- To simplify notation **precedence** is used

$\neg p \vee q$ means $(\neg p) \vee q$

$p \to q \vee \neg r$ means $p \to (q \vee (\neg r))$

TABLE 8
Precedence of Logical Operators.

Operator	*Precedence*
$\neg$	1
$\wedge$ $\vee$	2 3
$\to$ $\leftrightarrow$	4 5

Tautologies, Contradictions, and Contingencies

- A **tautology** is a proposition which is always true
  - Example: $p \vee \neg p$
- A **contradiction** is a proposition which is always false
  - Example: $p \wedge \neg p$
- A **contingency** is a proposition which is neither a tautology nor a contradiction
  - Example: p

TABLE 1 Examples of a Tautology and a Contradiction.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Truth Tables For Compound Propositions

- Construction of a truth table
  - A row for every possible combination of values for the atomic propositions
  - A column for the compound proposition (usually at far right)
  - A column for the truth value of each sub-expression that occurs in the

TABLE 7 The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Summary

- Implication
- Biconditional
- Precendence
- Computing Truth tables
- Tautologies, Contradictions, and Contingencies

- Next: Propositional Equivalences

Propositional Equivalences

Section 1.3

Video 3: Logical Equivalences

- Important Logical Equivalences
- Showing Logical Equivalence
- Contrapositive, Converse and Inverse
- Equivalence Proofs

Logical Equivalence

Two compound propositions p and q are **logically equivalent** if $p \leftrightarrow q$ is a tautology

We write this as $p \equiv q$ where p and q are compound propositions

Two compound propositions p and q are equivalent if and only if the columns in a truth table giving their truth values agree

Example

Using a truth table we show that $\neg p \lor q \equiv p \rightarrow q$

TABLE 4 **Truth Tables for $\neg p \lor q$ and $p \rightarrow q$.**

p	q	$\neg p$	$\neg p \lor q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

De Morgan's Laws

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$

Using a truth table show that De Morgan's Second Law holds

TABLE 3 Truth Tables for $\neg(p \vee q)$ and $\neg p \wedge \neg q$.

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

Equivalences with Basic Connectives

Equivalence	***Name***
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law

$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws

Equivalences with Implications

TABLE 7 Logical Equivalences Involving Conditional Statements.

$p \to q \equiv \neg p \lor q$

$p \to q \equiv \neg q \to \neg p$

$p \lor q \equiv \neg p \to q$

$p \land q \equiv \neg(p \to \neg q)$

$\neg(p \to q) \equiv p \land \neg q$

$(p \to q) \land (p \to r) \equiv p \to (q \land r)$

$(p \to r) \land (q \to r) \equiv (p \lor q) \to r$

$(p \to q) \lor (p \to r) \equiv p \to (q \lor r)$

$(p \to r) \lor (q \to r) \equiv (p \land q) \to r$

TABLE 8 Logical Equivalences Involving Biconditional Statements.

$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$

$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$

$p \leftrightarrow q \equiv (p \land q) \lor (\neg p \land \neg q)$

$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Contrapositive, Converse and Inverse

$\neg q \rightarrow \neg\, p$ is the **contrapositive** of $p \rightarrow q$, $\quad p \rightarrow q \equiv \neg q \rightarrow \neg p$

From $p \rightarrow q$ we can form the following conditional statements

- $q \rightarrow p$ is the **converse** of $p \rightarrow q$
- $\neg\, p \rightarrow \neg\, q$ is the **inverse** of $p \rightarrow q$ (and the contrapositive of $q \rightarrow p$)

They are equivalent to each other, but not equivalent to $p \rightarrow q$

Applying Logical Equivalences

The propositions in a known equivalence can be replaced by any compound proposition

Example: since we know that

$$p \to q \equiv \neg q \to \neg p$$

we also know that, for example

$$(p_1 \vee p_2) \to (q_1 \wedge q_2) \equiv \neg\,(q_1 \wedge q_2) \to \neg\,(p_1 \vee p_2)$$

Constructing New Logical Equivalences

- We can show that two expressions are logically equivalent by developing a series of logically equivalent statements
- To prove that $A \equiv B$ we produce a series of equivalences beginning with A and ending with B
- This is called an **equivalence proof**

$$\begin{gathered} A \equiv A_1 \\ \vdots \\ A_n \equiv B \end{gathered}$$

Example: Equivalence Proofs

Show that $\neg(p \vee (\neg p \wedge q))$

is logically equivalent to $\neg p \wedge \neg q$

$$
\begin{array}{rcll}
\neg(p \vee (\neg p \wedge q)) & \equiv & \neg p \wedge \neg(\neg p \wedge q) & \text{by the second De Morgan law} \\
& \equiv & \neg p \wedge [\neg(\neg p) \vee \neg q] & \text{by the first De Morgan law} \\
& \equiv & \neg p \wedge (p \vee \neg q) & \text{by the double negation law} \\
& \equiv & (\neg p \wedge p) \vee (\neg p \wedge \neg q) & \text{by the second distributive law} \\
& \equiv & F \vee (\neg p \wedge \neg q) & \text{because } \neg p \wedge p \equiv F \\
& \equiv & (\neg p \wedge \neg q) \vee F & \text{by the commutative law} \\
& & & \text{for disjunction} \\
& \equiv & (\neg p \wedge \neg q) & \text{by the identity law for } \mathbf{F}
\end{array}
$$

Example: Equivalence Proofs

Show that $(p \land q) \to (p \lor q)$ is a tautology.

$$
\begin{array}{rcll}
(p \land q) \to (p \lor q) & \equiv & \neg(p \land q) \lor (p \lor q) & \text{by truth table for } \to \\
& \equiv & (\neg p \lor \neg q) \lor (p \lor q) & \text{by the first De Morgan law} \\
& \equiv & (\neg p \lor p) \lor (\neg q \lor q) & \text{by associative and} \\
& & & \text{commutative laws} \\
& & & \text{laws for disjunction} \\
& \equiv & T \lor T & \text{by truth tables} \\
& \equiv & T & \text{by the domination law}
\end{array}
$$

Summary

- Showing Logical Equivalence
- De Morgan’s Laws
- Many Logical Equivalences
- Contrapositive, Converse and Inverse
- Equivalence Proofs

- Next: Normal Forms

Video 4: Normal Forms, Satisfiability

- Normal Forms
  - Disjunctive Normal Form
  - Conjunctive Normal Form
- Propositional Satisfiability

Disjunctive Normal Form

- A propositional formula is in **disjunctive normal form** if
  - it consists of a disjunction of compound expressions
  - where each compound expressions consists of a conjunction of a set of propositional variables or their negation

- Disjunctive Normal Form is important for circuit design methods

Examples

$(p \land \neg q) \lor (\neg p \land q)$	yes
$p \lor (\neg p \land q)$	yes
$p \land (\neg p \lor q)$	no
$\neg(\neg p \lor q)$	no

Example

Find the Disjunctive Normal Form (DNF) of

$$(p \lor q) \rightarrow \neg r$$

This proposition is true when r is false or when both p and q are false.

$$(\neg p \land \neg q) \lor \neg r$$

Construction of Disjunctive Normal Form

Every compound proposition can be put in disjunctive normal form

1. Construct the truth table for the proposition
2. Select rows that evaluate to **T**
3. For the propositional variables in the selected rows add a conjunct which includes the positive form of the propositional variable if the variable is assigned **T** in that row and the negated form if the variable is assigned **F** in that row

The resulting proposition can be further simplified using the equivalence

$(p \land q) \lor (p \land \neg q) \equiv p$

Conjunctive Normal Form

- A compound proposition is in *Conjunctive Normal Form* (CNF) if it is a conjunction of disjunctions.
- Every proposition can be put in an equivalent CNF.
- Conjunctive Normal Form (CNF) can be obtained by eliminating implications, moving negation inwards and using the distributive and associative laws.
- Important in theorem proving used in artificial Intelligence (AI).

Example

Put the following into CNF: $\neg(p \rightarrow q) \vee (r \rightarrow p)$

1. Eliminate implication signs

$$\neg(\neg p \vee q) \vee (\neg r \vee p)$$

2. Move negation inwards; eliminate double negation

$$(p \wedge \neg q) \vee (\neg r \vee p)$$

3. Convert to CNF using associative/distributive laws

$$(p \vee \neg r \vee p) \wedge (\neg q \vee \neg r \vee p)$$

Complexity of DNF and CNF

- Both DNF and CNF can be much larger than the original proposition

- More precisely
  - There exists examples of propositions with n clauses for which the CNF (DNF) has 2^n clauses

Propositional Satisfiability

- A compound proposition is **satisfiable** if there is an assignment of truth values to its variables that make it true.
- When no such assignments exist, the compound proposition is **unsatisfiable**.
- A compound proposition is unsatisfiable if and only if its negation is a tautology.

Examples

Determine the satisfiability of the following compound propositions:

$$(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$$

Satisfiable. Assign **T** to *p, q,* and *r*.

$$(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$$

Satisfiable. Assign **T** to *p* and ***F*** to *q*.

$$(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$$

Not satisfiable. Check each possible assignment of truth values to the propositional variables and none will make the proposition true.

Summary

- Normal Forms
  - Disjunctive Normal Form
  - Conjunctive Normal Form
  - Every proposition can be converted to DNF and CNF
  - The resulting proposition can explode in size
- Propositional Satisfiability
  - Does there exist an assignment of truth values such that the proposition is true?

Applications of Propositional Logic

Section 1.2

Video 5: Applications of Propositional Logic

- Translating Natural Language to Propositional Logic
- Boolean Search
- Logic Puzzles
- Logic Circuits
- Sudoku

Translating Natural Language Sentences

- Steps to convert a natural language sentence to a statement in propositional logic
  - Identify atomic propositions and represent using propositional variables
  - Determine appropriate logical connectives
- "If I go skiing or to the party, I will not go shopping."
  - $p :=$ I go skiing
  - $q :=$ I go to the party
  - $r :=$ I will go shopping

If *p* or *q* then not *r*

$$(p \vee q) \rightarrow \neg r$$

Boolean Queries for Document Search

If you ask on Google the query “lausanne”, it denotes the proposition “The word ‘Lausanne’ appears in the document”

You can then form complex queries using logical operators (Boolean Queries)

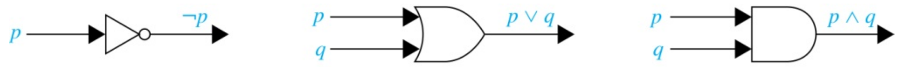
- lausanne $\land$ university
- lausanne $\land$ (university $\lor$ studying)
- lausanne $\land$ (university $\lor$ studying) $\land$ economics
- lausanne $\land$ (university $\lor$ studying) $\land$ economics $\land \neg$ unil

Attention: often different syntax is used

- Lausanne AND (university OR studying) AND NOT unil
- Google: Lausanne university|studying economics -unil

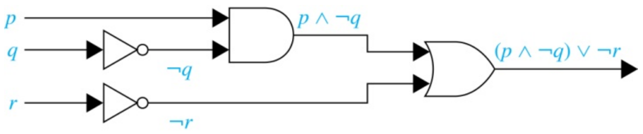
Logic Circuits

- Electronic circuits; each input/output signal can be viewed as a 0 or 1.
  - 0   represents **False**
  - 1   represents **True**
- Complicated circuits are constructed from three basic circuits called gates.



Inverter   OR gate   AND gate

  - The inverter (**NOT gate**)takes an input bit and produces the negation of that bit.
  - The **OR gate** takes two input bits and produces the value equivalent to the disjunction of the two bits.
  - The **AND gate** takes two input bits and produces the value equivalent to the conjunction of the two bits.
- More complicated digital circuits can be constructed by combining these basic circuits to produce the desired output given the input signals by building a circuit for each piece of the output expression and then combining them. For example:



Logic Puzzles: Knights and Knaves

An island has two types of inhabitants, *knights*, who always tell the truth, and *knaves*, who always lie

You go to the island and meet A and B

- A says “B is a knight.”
- B says “The two of us are of opposite types.”

Question: What are the types of A and B?

Knights and Knaves - Solution

- A says "B is a knight."
- B says "The two of us are of opposite types."

Let p :≔ "A is a knight" and q :≔ "B is a knight"

Then $\neg p$ represents the proposition that "A is a knave" and $\neg q$ that "B is a knave"

If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \wedge \neg q) \vee (\neg p \wedge q)$ would have to be true, but it is not. So, A is not a knight and therefore $\neg p$ must be true.

If A is a knave, then B must not be a knight since knaves always lie. So, then both $\neg p$ and $\neg q$ hold since both are knaves.

Sudoku

- A **Sudoku puzzle** is represented by a 9×9 grid made up of nine 3×3 subgrids, known as **blocks**. Some of the 81 cells of the puzzle are assigned one of the numbers 1,2, …, 9.
- The puzzle is solved by assigning numbers to each blank cell so that every row, column and block contains each of the nine possible numbers.

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

Encoding as a Satisfiability Problem

- Let $p(i, j, n)$ denote the proposition that is true when the number n is in the cell in the i^{th} row and the j^{th} column.
- There are $9 \times 9 \times 9 = 729$ such propositions.
- In the sample puzzle $p(5,1,6)$ is true, but $p(5,j,6)$ is false for $j = 2,3,\ldots 9$

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

Encoding

- For each cell with a given value, assert *p*(*i*,*j*,*n*), when the cell in row *i* and column *j* has the given value.
- Assert that every row contains every number.

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

$\bigvee_{j=1}^{n} p_j$ is used for $p_1 \vee p_2 \vee \ldots \vee p_n$

$\bigwedge_{j=1}^{n} p_j$ is used for $p_1 \wedge p_2 \wedge \ldots \wedge p_n$

- Assert that every column contains every number.

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

Encoding (cont)

- Assert that each of the 3 x 3 blocks contain every number (tricky)

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

- Assert that no cell contains more than one number. Take the conjunction over all values of *n*, *n'*, *i*, and j, where each variable ranges from 1 to 9 and $n \neq n'$, of the following expressions

$$p(i, j, n) \to \neg p(i, j, n')$$

Solving Satisfiability Problems

- To solve a Sudoku puzzle, we need to find an assignment of truth values to the 729 variables of the form $p(i, j, n)$ that makes the conjunction of the assertions true. Those variables that are assigned T yield a solution to the puzzle.

Solving Satisfiability Problems Computationally

- A truth table can always be used to determine the satisfiability of a compound proposition. But this is too complex even for modern computers for large problems.
- There has been much work on developing efficient methods for solving satisfiability problems as many practical problems can be translated into satisfiability problems.

Problem

How many rows are there in a truth table with *n* propositional variables?

Answer: 2^n

Sudoku: 2^{729} =
2824013958708217496949108842204627863351353911851577524683401930862693830361198499905873920995229996970897865498283996578123296865878390947626553088486946106430796091482716120572632072492703527723757359478834530365734912
(this is pretty large)

Summary

- Translating Natural Language to Propositional Logic
  - Making language precise
- Boolean Search
  - Expressing search
- Logic Circuits
  - Designing systems
- Logic Puzzles
  - Solving problems by logical reasoning
- Sudoku
  - Solving problems can be difficult/expensive