Notes for Discrete Mathematics (2. Ed) by Rosen: Chapter 1: Logic and Proofs

Andreas Hansen

May 2025

Chapter 1.3: Propositional Equivalences (pg. 26-37)

Tautology and Contradiction: (pg. 26-27)

Definition: Tautology. A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a tautology.

Definition: Contradiction. A compound proposition that is always false is called a contradiction

Definition: Contingency. A compound proposition that is neither a tautology nor a contradiction is called a contingency

Example 1: Examples of tautologies and contradictions using just one propositional variable

Exercise:

Consider the truth tables of $p \vee \neg p$ and $p \wedge \neg p$

Strategy: Remember the conjunction ($\wedge$) and disjunction ($\vee$) operators and write first truth and false for the propositional variable p and then for its negation $\neg p$.

Truth Table:

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

Conclusion:
The compound proposition $p \vee \neg p$ is a **tautology** — always true.
The compound proposition $p \wedge \neg p$ is a **contradiction** — always false.

Logical Equivalences: (pg. 27-30)

Definition: Logically Equivalent

The compound propositions p and q are called **logically equivalent** if

$$p \leftrightarrow q \text{ is a tautology}$$

The notation $p \equiv q$ denotes that p and q are logically equivalent.

Remark 1

The symbol $\equiv$ is not a logical connective, and $p \equiv q$ is not a compound proposition but rather is the statement that $p \leftrightarrow q$ is a tautology. The symbol $\Leftrightarrow$ is sometimes used instead of $\equiv$ to denote logical equivalence.

Definition: De Morgan's Laws

A pair of transformation rules, that state:

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

They can also be expressed as:

Logic	Boolean	English (simple)	English (formal)
$\neg(p \wedge q) \equiv \neg p \vee \neg q$	$\overline{p \wedge q} = \overline{p} \vee \overline{q}$	The negation of "A and B" is the same as "not A or not B"	The complement of the union of two sets is the same as the intersection of their complements
$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\overline{p \vee q} = \overline{p} \wedge \overline{q}$	The negation of "A or B" is the same as "not A and not B"	The complement of the intersection of two sets is the same as the union of their complements

Example 1: Show 2. De Morgan´s Law with truth table

Exercise:

Show that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent (using truth table)

Truth table:

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

Conclusion: Because the truth values of the compound propositions $\neg(p \vee q)$ and $\neg p \wedge \neg q$ agree for all possible combinations of the truth values of p and q, it follows that $\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q$ is a tautology and that these compound propositions are logically equivalent.

Example 2: Show conditional-disjunction equivalence

Exercise:

Show that $p \to q$ and $\neg p \vee q$ are logically equivalent (using truth table)

Truth table:

p	q	$\neg p$	$\neg p \vee q$	$p \to q$
T	T	T	F	F
T	F	T	F	F
F	T	T	F	T
F	F	F	T	T

Conclusion: Because the truth values of the compound propositions $\neg(p \vee q)$ and $\neg p \wedge \neg q$ agree for all possible combinations of the truth values of p and q, it follows that $\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q$ is a tautology and that these compound propositions are logically equivalent.

Remark 2

If we establish a logical equivalence of two compound propositions involving three different propositional variables p, q, and r. To use a truth table to establish such a logical equivalence, we need eight rows, one for each possible combination of truth values of these three variables. We symbolically represent these combinations by listing the truth values of p, q, and r, respectively.

Remark 3

These eight combinations of truth values are **TTT**, **TTF**, **TFT**, **TFF**, **FTT**, **FTF**, **FFT**, and **FFF**; we use this order when we display the rows of the truth table. Note that we need to double the number of rows in the truth tableswe use to show that compound propositions are equivalent for each additional propositional variable, so that 16 rows are needed to establish the logical equivalence of two compound propositions involving four propositional variables, and so on.

Remark 4

In general, 2^n rows are required if a compound proposition involves n propositional variables. Because of the rapid growth of 2^n, more efficient ways are needed to establish logical equivalences, such as by using ones we already know

Example 3: Distributive law of disjunction over conjunction

Exercise:

Show that $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ are logically equivalent (using truth table)

Truth table:

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

Logical Equivalences: Logic, Programming, and Law Names

Equivalence (Logic)	Equivalence (Programming)	Name
$p \vee \mathbf{F} \equiv p$	p || false == p	Identity law (OR)
$p \wedge \mathbf{T} \equiv p$	p && true == p	Identity law (AND)
$p \vee \mathbf{T} \equiv \mathbf{T}$	p || true == true	Domination law (OR)
$p \wedge \mathbf{F} \equiv \mathbf{F}$	p && false == false	Domination law (AND)
$p \vee p \equiv p$	p || p == p	Idempotent law (OR)
$p \wedge p \equiv p$	p && p == p	Idempotent law (AND)
$\neg(\neg p) \equiv p$	!!p == p	Double negation law
$p \vee q \equiv q \vee p$	p || q == q || p	Commutative law (OR)
$p \wedge q \equiv q \wedge p$	p && q == q && p	Commutative law (AND)
$(p \vee q) \vee r \equiv p \vee (q \vee r)$	(p || q) || r == p || (q || r)	Associative law (OR)
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	(p && q) && r == p && (q && r)	Associative law (AND)
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	p || (q && r) == (p || q) && (p || r)	Distributive law (OR over AND)
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	p && (q || r) == (p && q) || (p && r)	Distributive law (AND over OR)
$\neg(p \wedge q) \equiv \neg p \vee \neg q$	!(p && q) == !p || !q	De Morgan's law (NOT AND $\rightarrow$ OR)
$\neg(p \vee q) \equiv \neg p \wedge \neg q$	!(p || q) == !p && !q	De Morgan's law (NOT OR $\rightarrow$ AND)
$p \vee (p \wedge q) \equiv p$	p || (p && q) == p	Absorption law (OR)
$p \wedge (p \vee q) \equiv p$	p && (p || q) == p	Absorption law (AND)
$p \vee \neg p \equiv \mathbf{T}$	p || !p == true	Negation law (OR)
$p \wedge \neg p \equiv \mathbf{F}$	p && !p == false	Negation law (AND)

Logical Equivalences Involving Conditional Statements

Equivalence (Logic)	Equivalence (Programming)	Name
$p \to q \equiv \neg p \lor q$	!p || q	Conditional equivalence
$p \to q \equiv \neg q \to \neg p$	!p || q == !q || !p	Contrapositive equivalence
$p \lor q \equiv \neg p \to q$	!p || q	Implication from disjunction
$p \land q \equiv \neg(p \to \neg q)$	!(p => !q) or !(!p || !q)	Conjunction via implication
$\neg(p \to q) \equiv p \land \neg q$	!(!p || q) == p && !q	Negation of implication
$(p \to q) \land (p \to r) \equiv p \to (q \land r)$	(!p || q) && (!p || r) == !p || (q && r)	Distributive implication (AND)
$(p \to r) \land (q \to r) \equiv (p \lor q) \to r$	(!p || r) && (!q || r) == !(p || q) || r	Factoring implication (common result)
$(p \to q) \lor (p \to r) \equiv p \to (q \lor r)$	(!p || q) || (!p || r) == !p || (q || r)	Distributive implication (OR)
$(p \to r) \lor (q \to r) \equiv (p \land q) \to r$	(!p || r) || (!q || r) == !(p && q) || r	Factoring implication (common antecedent)

Logical Equivalences Involving Biconditional Statements

Equivalence (Logic)	Equivalence (Programming)	Name
$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$	(!p || q) && (!q || p)	Biconditional as conjunction of implications
$p \leftrightarrow q \equiv \neg q \leftrightarrow \neg p$	!q == !p	Invariance under negation (logical symmetry)
$p \leftrightarrow q \equiv (p \land q) \lor (\neg p \land \neg q)$	(p && q) || (!p && !q)	Biconditional as agreement in truth value
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$	!(p == q) == (p == !q)	Negation of biconditional / XOR form

Remark 5

The associative law for disjunction shows that the expression $p \lor q \lor r$ is well defined, in the sense that it does not matter whether we first take the disjunction of p with q and then the disjunction of $p \lor q$ with r, or if we first take the disjunction of q and r and then take the disjunction of p with $p \lor r$.

Remark 6

Similarly, the expression $p \land q \land r$ is well defined. By extending this reasoning, it follows that $p_1 \lor p_2 \lor \ldots \lor p_n$ and $p_1 \land p_2 \land \ldots \land p_n$ are well defined whenever $p_1, p_2, \ldots, p_n$ are propositions.

Extended Version of De Morgan's Laws

Explicit Form	Compact Form	Name
$\neg(p_1 \vee p_2 \vee \ldots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_n$	$\neg\left(\bigvee_{j=1}^{n} p_j\right) \equiv \bigwedge_{j=1}^{n} \neg p_j$	Generalized De Morgan's Law (OR to AND)
$\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \equiv \neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n$	$\neg\left(\bigwedge_{j=1}^{n} p_j\right) \equiv \bigvee_{j=1}^{n} \neg p_j$	Generalized De Morgan's Law (AND to OR)

Remark 7

A truth table with 2^n rows is needed to prove the equivalence of two compound propositions in n variables. Note that the number of rows doubles for each additional propositional variable added

Remark 8

Because 2^n grows extremely rapidly as n increases , the use of truth tables to establish equivalences becomes impractical as the number of variables grows. It is quicker to use other methods, such as employing logical equivalences that we already know.

Using De Morgan's Laws (pg. 30)

Remark 1

When using De Morgan's laws, remember to change the logical connective after you negate.

Remark 2

The two logical equivalences known as De Morgan's laws are particularly important. They tell us how to negate conjunctions and how to negate disjunctions.

Remark 3

In particular, the equivalence $\neg(p \vee q) \equiv \neg \wedge \neg q$ tells us that the negation of a disjunction is formed by taking the conjunction of the negations of the component propositions.

Remark 4

Similarly, the equivalence $\neg(p \wedge q) \equiv \neg p \vee \neg q$ tells us that the negation of a conjunction is formed by taking the disjunction of the negations of the component propositions.

Example 1: Using De Morgan's Laws

Exercise:
Use De Morgan's laws to express the negations of:

"Miguel has a cellphone and he has a laptop computer"
"Heather will go to the concert or Steve will go to the concert"

Let:

- **p:** Miguel has a cellphone
- **q:** Miguel has a laptop computer
- **r:** Heather will go to the concert
- **s:** Steve will go to the concert

Negation 1:

Original: $p \wedge q$ (Miguel has a cellphone and a laptop)
Negation: $\neg(p \wedge q)$
Apply De Morgan's Law: $\neg(p \wedge q) \equiv \neg p \vee \neg q$
English: *Miguel does not have a cellphone or he does not have a laptop.*

Negation 2:

Original: $r \vee s$ (Heather or Steve will go to the concert)
Negation: $\neg(r \vee s)$
Apply De Morgan's Law: $\neg(r \vee s) \equiv \neg r \wedge \neg s$
English: *Heather will not go to the concert and Steve will not go to the concert.*

Constructing New Logical Equivalences (pg. 31-32)

Remark 1

We use the fact that if p and q are logically equivalent and q and r are logically equivalent, then p and r are logically equivalent (See Exercise 60)

Example 1: Construct new logical equivalence

Exercise:
Show the logical equivalence:

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

Strategy:
We will establish this equivalence by developing a series of logical equivalences, using one of the equivalences from our previous tables, starting with $\neg(p \to q)$ and ending with $p \wedge \neg q$

Using logical equivalences:

$$\begin{aligned}
\neg(p \to q) &\equiv \neg(\neg p \vee q) \quad \textbf{(conditional-disjunction equivalence, pg. 3, example 2)} \\
&\equiv \neg(\neg p) \wedge \neg q \quad \textbf{(second De Morgan law, pg. 4)} \\
&\equiv p \wedge \neg q \quad \textbf{(double negation law, pg. 4)}
\end{aligned}$$

Example 2: Construct new logical equivalence

Exercise:
Show the logical equivalence:

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

Strategy:
We will establish this equivalence by developing a series of logical equivalences, using one of the equivalences from our previous tables, starting with $\neg(p \vee (\neg p \wedge q))$ and ending with $\neg p \wedge \neg q$

Using logical equivalences:

$$\begin{aligned}
\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) \quad \textbf{(second De Morgan law, pg. 4)} \\
&\equiv \neg p \wedge (\neg(\neg p) \vee \neg q)) \quad \textbf{(first De Morgan law, pg. 4)} \\
&\equiv \neg p \wedge (p \vee \neg q) \quad \textbf{(double negation law, pg. 4)} \\
&\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) \quad \textbf{(distributive law (AND over OR), pg. 4)} \\
&\equiv \mathbf{F} \vee (\neg p \wedge \neg q) \quad \textbf{(negation law (AND), pg. 4)} \\
&\equiv (\neg p \wedge \neg q) \vee \mathbf{F} \quad \textbf{(commutative law (OR), pg. 4)} \\
&\equiv \neg p \wedge \neg q \quad \textbf{(identity law (OR), pg. 4)}
\end{aligned}$$

Example 3: Construct tautology

Exercise:
Show the tautology:

$$(p \wedge q) \rightarrow (p \vee q)$$

Strategy:
To show that this statement is a tautology, we will use logical equivalences to demonstrate that it is logically equivalent to **T**.

Using logical equivalences:

$$\begin{aligned}(p \wedge q) \rightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) \quad \textbf{(conditional-disjunction equivalence, pg. 3, ex. 2)}\\ &\equiv (\neg p \vee \neg q) \vee (p \vee q) \quad \textbf{(first De Morgan law, pg. 4)}\\ &\equiv (\neg p \vee p) \vee (\neg q \vee q) \quad \textbf{(associative + commutative (OR) laws, pg. 4)}\\ &\equiv \mathbf{T} \vee \mathbf{T} \quad \textbf{(tautology, pg. 1, ex. 1 + commutative (OR) law, pg. 4)}\\ &\equiv \mathbf{T} \quad \textbf{(domination law, pg. 4,)}\end{aligned}$$

Satisfiability (pg. 33)

Definition: Satisfiable. A compound proposition is satisfiable if there is an assignment of truth values to its variables that makes it true (that is, when it is a tautology or a contingency).

Definition: Unsatisfiable. When no such assignments exists, that is, when the compound proposition is false for all assignments of truth values to its variables, the compound proposition is unsatisfiable.

Remark 1

Note that a compound proposition is unsatisfiable if and only if its negation is true for all assignments of truth values to the variables, that is, if and only if its negation is a tautology.

Remark 2

When we find a particular assignment of truth values that makes a compound proposition true,we have shown that it is satisfiable; such an assignment is called a **solution** of this particular satisfiability problem.

Remark 3

However, to show that a compound proposition is unsatisfiable, we need to show that every assignment of truth values to its variables makes it false. Although we can always use a truth table to determine whether a compound proposition is satisfiable, it is often more efficient not to.

Example 1: Satisfiable compound propositions

Exercise:
Determine whether each of the compound propositions is satisfiable

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

Strategy:
Instead of using a truth table to solve this problem, we will reason about truth values.

First compound proposition:
We have $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ is true when the three variables p, q, and r have the same truth value. Hence, it is satisfiable as there is at least one assignment of truth values for p, q, and r that makes it true.

Second compound proposition:
Similarly, note that $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is true when at least one of p, q, and r is true and at least one is false. Hence, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is satisfiable, as there is at least one assignment of truth values for p, q, and r that makes it true.

Third compound proposition:
Finally, note that for $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ to be true:

Both:

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

and

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

must both be true.
For the first to be true, $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$, the three variables must have the same truth values.
For the second to be true, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ at least one of the three variables must be true and at least one must be false.

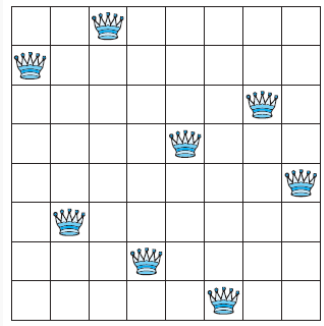
However, these conditions are contradictory. From these observations we conclude that no assignment of truth values to p, q, and r makes $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ true. Hence, it is unsatisfiable.

¨.

Applications of Satisfiability (pg. 33-37)

Example 1: The n-Queens Problem (1/2)

Exercise:
The n-queens problem asks for a placement of n queens on an n $\times$ n chessboard so that no queen can attack another queen. This means that no two queens can be placed in the same row, in the same column, or on the same diagonal. We display a solution to the eight-queens problem below.



Strategy:
To model the n-queens problem as a satisfiability problem, we introduce n^2 variables, $p(i, j)$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$.
For a given placement of a queens on the chessboard:

$p(i, j)$ is **true** when there is a queen on the square in the i'th row and j'th column.
and **false** otherwise.

Note that squares (i, j) and (i', j') are on the same diagonal if either:

$$i + i' = j + j' \quad \text{or} \quad i - i' = j - j'$$

Implementing solution:
Step 1: For no two of the n queens to be in the same row, there must be one queen in each row. We can show that there is one queen in each row by verifying that every row contains at least one queen and that every row contains at most one queen.

$\bigvee_{j=1}^{n} p(i, j)$ asserts that row i contains at least one queen
$Q_1 = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} p(i, j)$ **asserts that every row contains at least one queen**

Step 2: For every row to include at most one queen, it must be the case that $p(i, j)$ and $p(k, j)$ are not both true for integers j and k with $1 \leq j < k \leq n$.
Observe that $\neg p(i, j) \vee \neg p(i, k)$ asserts that at least one of $\neg p(i, j)$ and $\neg p(i, k)$ is true, which means that at least one of $p(i, j)$ and $p(i, k)$ is false. So, to check that there is:

$Q_2 = \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n-1} \bigwedge_{k=j+1}^{n} (\neg p(i, j) \vee \neg p(k, j))$ **asserts at most one queen in each row**

Step 3: We have do the same, but for coloumns:

$Q_3 = \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n-1} \bigwedge_{k=i+1}^{n} (\neg p(i, j) \vee \neg p(k, j))$ **no column contains more than one queen**

Example 1: The n-Queens Problem (2/2)

Step 4: Now we consider the diagonals, and we have to make sure, that no diagonal contains two queens. We will first look at the main diagonal, which are characterized by having constant values of $i - j$. So we have to enforce that for all cells (i, j) there is no other queen on $(i + k, j + k)$, where k increases along the diagonal:

$$Q_4 = \bigwedge_{i=1}^{n-1} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{min(n-i,n-j)} (\neg p(i, j) \vee \neg p(i + k, j + k)) \quad \textbf{Main diagonal}$$

Step 5: These anti-diagonals are characterized by constant values of $i + j$. So we have to enforce that for all cells (i, j) there is no other queen on $(i + k, j - k)$, where k increases along the diagonal:

$$Q_5 = \bigwedge_{i=1}^{n-1} \bigwedge_{j=2}^{n} \bigwedge_{k=1}^{min(n-i,j-1)} (\neg p(i, j) \vee \neg p(i + k, j - k)) \quad \textbf{Anti-diagonal}$$

Step 6: Putting all this together, we find that the solutions of the n-queens problem are given by the assignments of truth values to the variables $p(i, j)$

$$i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, n \text{ that make the following true}$$

$$Q = Q_1 \wedge Q_2 \wedge Q_3 \wedge Q_4 \wedge Q_5$$

Example 2: Sudoku puzzles (1/2)

Exercise:
Sudoku puzzles are constructed using a 9 × 9 grid made up of nine 3 × 3 subgrids, known as blocks
For each puzzle, some of the 81 cells, called givens, are assigned one of the numbers 1, 2,. . . , 9, and the other cells are blank. The puzzle is solved by assigning a number to each blank cell so that every row, every column, and every one of the nine 3 × 3 blocks contains each of the nine possible numbers.
Note that instead of using a 9 × 9 grid, Sudoku puzzles can be based on $n^2 \times n^2$ grids, for any positive integer n, with the $n^2 \times n^2$ grid made up of n^2 $n \times n$ subgrids.

Strategy:
To encode a Sudoku puzzle, 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 j'th column. There are $9 \times 9 \times 9 = 729$ such propositions, as i, j, and n all range from 1 to 9. For example, for the puzzle in Figure 2, the number 6 is given as the value in the fifth row and first column. Hence, we see that $p(5, 1, 6)$ is true, but $p(5, j, 6)$ is false for $j = 2, 3, \ldots, 9$.

Given a particular Sudoku puzzle, we begin by encoding each of the given values. Then, we construct compound propositions that assert that every row contains every number, every column contains every number, every 3 × 3 block contains every number, and each cell contains no more than one number.
It follows, as the reader should verify, that the Sudoku puzzle is solved by finding an assignment of truth values to the 729 propositions $p(i, j, n)$ with i, j, and n each ranging from 1 to 9 that makes the conjunction of all these compound propositions true.
After listing these assertions, we will explain how to construct the assertion that every row contains every integer from 1 to 9.
We will leave the construction of the other assertions that every column contains every number and each of the nine 3 × 3 blocks contains every number to the exercises

Example 2: Sudoku puzzles (2/2)

Solution:
Step 1:
For each cell with a given value, we assert $p(i, j, n)$ when the cell in row i and column j has the given value n.
Step 2:
First, to **assert that row i contains the number n**, we form

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

Step 3:
To assert that **row i contains all n numbers**, we form the conjunction of these disjunctions over all nine possible values of n, giving us

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

Step 4:
Finally, to **assert that every row contains every number**, we take the conjunction

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

Step 5:
(Exercises 71 and 72 ask for explanations of the assertions that every column contains every number and that each of the nine 3×3 blocks contains every number.)

$\bigwedge_{j=1}^{9} \bigwedge_{n=1}^{9} \bigvee_{i=1}^{9} p(i, j, n)$ **assert that every column contains every number**

And that **assert that each of the nine 3×3 blocks contains every number;**
$$\bigwedge_{r=0}^{2} \bigwedge_{s=0}^{2} \bigwedge_{n=1}^{9} \bigvee_{i=1}^{3} \bigvee_{j=1}^{3} p(3r + i, 3s + j, n)$$

Solving Satisfiability Problems (pg. 37)

Remark 1

A **truth table** can be used to determine whether a compound proposition is **satisfiable**, or equivalently, whether its negation is a tautology (see **Exercise 64**).
This can be done by hand for a compound proposition with a small number of variables, but when the number of variables grows, this becomes impractical.

Remark 2

Because the problem scales as 2^n, where n is the number of variables.
For instance, there are $2^{20} = 1,048,576$ rows in the truth table for a compound proposition with 20 variables. Thus, you need a computer to help you determine, in this way, whether a compound proposition in 20 variables is satisfiable.

Remark 3

When many applications are modeled, questions concerning the satisfiability of compound propositions with hundreds, thousands, or millions of variables arise. Note, for example, that when there are 1000 variables, checking every one of the 2^{1000} (a number with more than 300 decimal digits) possible combinations of truth values of the variables in a compound proposition cannot be done by a computer in even trillions of years

Remark 4

No procedure is known that a computer can follow to determine in a reasonable amount of time whether an arbitrary compound proposition in such a large number of variables is satisfiable. However, progress has been made developing methods for solving the satisfiability problem for the particular types of compound propositions that arise in practical applications, such as for the solution of Sudoku puzzles.

Remark 5

Many computer programs have been developed for solving satisfiability problems which have practical use.
In our discussion of the subject of algorithms in Chapter 3, we will discuss this question further. In particular, we will explain the important role **the propositional satisfiability problem plays in the study of the complexity of algorithms**