PMATH 330 Introduction to Mathematical Logic

Lecture Notes

by Stephen New

Chapter 1. Propositional Formulas

1.1 Definition: The **alphabet** (or **symbol set**) of propositional logic consists of the following symbols:

$$\neg,\ \wedge,\ \vee,\ \rightarrow,\ \leftrightarrow,\ (,\ ),\ P_1,\ P_2,\ P_3,\ \cdots$$

The symbols $\neg, \wedge, \vee, \rightarrow$ and $\leftrightarrow$ are called the **logic symbols**. Individually, the logic symbols are given the following names:

$\neg$	not
$\wedge$	and
$\vee$	or
$\rightarrow$	implies
$\leftrightarrow$	if and only if

The symbols $P_1, P_2, P_3 \cdots$ are called the **propositional variable symbols**. Usually, we shall write $P, Q, R, \cdots$ instead of $P_1, P_2, P_3, \cdots$.

1.2 Definition: An empty or finite list of symbols from the above symbol set is called a **string**, and the number of symbols occuring in a string is called its **length**; we consider P_i to be a single symbol. The **empty string**, denoted by $\emptyset$, has length 0. If X and Y are two strings then the string XY is called the **concatenation** of X and Y. If U, X and V are strings and if Y is the string $Y = UXV$ then we say that X is a **substring** of Y.

1.3 Example: Let $X = \neg \wedge \wedge \neg \wedge$. Determine how many substrings X has.

Solution: We list all the substrings of X ordered by length:

$$\emptyset,\ \neg,\ \wedge,\ \neg\wedge,\ \wedge\wedge,\ \wedge\neg,\ \neg\wedge\wedge,\ \wedge\wedge\neg,\ \wedge\neg\wedge,\ \neg\wedge\wedge\neg,\ \wedge\wedge\neg\wedge,\ X$$

Altogether X has 12 substrings.

1.4 Definition: In propositional logic, a **formula** (in standard notation) is a non-empty string which can be obtained using finitely many applications of the following rules:

1. Each propositional variable symbol is a formula.
2. If F is a formula then $\neg F$ is also a formula.
3. If F and G are formulas and $\times \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ then $(F \times G)$ is a formula.

Since the symbols $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are used to join together two formulas, they are called **binary connective symbols**. The symbol $\neg$ is called a **unary connective symbol**.

1.5 Example: $X = (\neg P \wedge (Q \rightarrow P))$ is a formula but $Y = \wedge\neg P)\neg$ is not.

1.6 Definition: A **derivation** for a formula F is a list of formulas $F_1, F_2, \cdots, F_n$ with $F = F_n$ (or sometimes $F = F_k$ for some other value of k) in which each formula F_k is either

1. a propositional variable,
2. of the form $\neg F_i$ for some $i < k$, or
3. of the form $(F_i \times F_j)$ for some $i, j < k$ and some $\times \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Often we shall arrange the list $F_1, F_2, \cdots, F_n$ in a column, with one formula on each line, and provide justification on each line.

1.7 Example: Make a derivation for the formula $F = (P \vee (\neg Q \to (R \wedge P)))$.

Solution: One possible derivation is P, R, $(R \wedge P)$, Q, $\neg Q$, $(\neg Q \to (R \wedge P))$, F.
We write this derivation in a column and provide justification:

1.	P	rule 1
2.	R	rule 1
3.	$(R \wedge P)$	rule 3, with $\times = \wedge$, on formulas 2 and 1
4.	Q	rule 1
5.	$\neg Q$	rule 2 on formula 4
6.	$(\neg Q \to (R \wedge P))$	rule 3, with $\times = \to$, on formulas 5 and 3
7.	F	rule 3, with $\times = \vee$, on formulas 1 and 6

1.8 Note: A derivation for F provides a proof that F is a formula. It is in general more difficult to prove that a given string X is *not* a formula. One way to do this is to find a property which all formulas must satisfy but which X does not.

1.9 Definition: Let $\Phi(F)$ be a statement about a string F. We can show that $\Phi(F)$ is true for all formulas F as follows:

1. Let $F = P$, where P is any propositional variable, and show that $\Phi(F)$ is true.
2. Let $F = \neg G$, where G is any formula with $\Phi(G)$ true, and show that $\Phi(F)$ is true.
3. Let $F = (G \times H)$, where G and H are any formulas with $\Phi(G)$ and $\Phi(H)$ both true, and $\times$ is any binary connective, and show that $\Phi(F)$ is also true.

This method of proof is called **induction on formulas**.

1.10 Example: Prove that the string $X = (P\neg \wedge Q)$ is not a formula.

Solution: Let $\Phi(F)$ be the statement "$P\neg$ is not a substring of F". We show that $\Phi(F)$ is true for all formulas F. Case 1: let $F = S$, where S is a propositional variable (possibly equal to P or to Q). Clearly, $P\neg$ is not a substring of $F = S$. Case 2: let $F = \neg G$, where G is a formula in which $P\neg$ is not a substring. Then $P\neg$ is not a substring of $F = \neg G$ since if it was, then it would have to be a substring of G. Case 3: let $F = (G \times H)$, where G and H are formulas in which $P\neg$ is not a substring, and $\times \in \{\wedge, \vee, \to, \leftrightarrow\}$. Then $P\neg$ is not a substring of $F = (G \times H)$ since if it was, then it would have to be a substring either of G or of H. By induction on formulas, $P\neg$ is not a substring of any formula F. Since $P\neg$ is a substring of X, X cannot be a formula.

1.11 Theorem: *(Unique Readability of Formulas) Let F be a formula. Then*

(1) F does not begin with any formula $G \neq F$. To be precise, if X is a string and G is a formula and $F = GX$ then $X = \emptyset$ (and so $F = G$), and

(2) The first symbol of F is either a propositional variable P, in which case $F = P$; or the symbol $\neg$, in which case $F = \neg G$ for some uniquely determined formula G; or the symbol (, in which case $F = (G \times H)$ for some uniquely determined formulas G and H and some uniquely determined binary connective $\times$.

Proof: Part (2) follows immediately from the definition of a formula, except for the uniqueness of G and H and $\times$. The uniqueness follows from part (1).

To prove part (1), let $\Phi(F)$ be the statement "for any formula $G \neq F$, F does not begin with G and G does not begin with F". To be precise, $\Phi(F)$ is the statement "for any string X and any formula G, (if $F = GX$ then $X = \emptyset$, and if $G = FX$ then $X = \emptyset$).

Case 1: Let $F = P$, where P is a propositional variable. Let X be a string and let G be a formula. If $F = GX$, then since the string $F = P$ has length 1, and the string G has length at least 1 (G cannot be the empty string), we must have $F = G$ and $X = \emptyset$. If $G = FX = PX$, then the formula G starts with P, so G was obtained by an application of rule 1 in the definition of a formula, and so $G = P$ and $X = \emptyset$.

Case 2: Let $F = \neg H$, where H is a formula with $\Phi(H)$ true. Let X be a string and let G be a formula. Suppose first that $F = GX$. Then $\neg H = GX$, so G begins with the symbol $\neg$ and so G was obtained using rule 3 in the definition of a formula; say $G = \neg K$. Now we have $\neg H = \neg KX$ so H begins with K. Since $\Phi(H)$ is true, we must have $H = K$ and hence $X = \emptyset$. This completes the proof that if $F = GX$ then $X = \emptyset$. Now suppose that $G = FX = \neg HX$. Then G begins with $\neg$; say $G = \neg K$. Now we have $\neg K = \neg HX$ and so K begins with H. Since $\Phi(H)$ is true, we must have $K = H$ and hence $X = \emptyset$.

Case 3: Let $F = (H \times K)$ where H and K are formulas with $\Phi(H)$ and $\Phi(K)$ true, let X be a string and let G be a formula. Suppose that $(H \times K) = GX$. Then the string G must begin with the symbol (. So G was obtained by an application of rule 3 in the definition of a formula. Say $G = (L * M)$, where L and M are formulas and $*$ is a binary connective. We have $(H \times K) = GX = (L * M)X$ and so either H begins with L or L begins with H. Since $\Phi(H)$ is true, we must have $H = L$. We now have $(L * M)X = (H \times K) = (L \times K)$ and so $* = \times$ and either K begins with M or M begins with K. Since $\Phi(K)$ is true, we must have $K = M$. We now have $(L * M)X = (H \times K) = (L * M)$ and so $X = \emptyset$. This completes the proof that if $(H \times K) = GX$ then $X = \emptyset$. Similarly we can show that if $G = (H \times K)X$ then $X = \emptyset$.

By induction on formulas, $\Phi(F)$ is true for all formulas F.

1.12 Remark: Unique readability of formulas ensures that we could, if we wished, make an algorithm to test a given string to determine whether it is a formula or a list of formulas. We shall not bother to do this.

1.13 Example: Use unique readability to prove that $X = (P\neg \wedge Q)$ is not a formula.

Solution: The first symbol in X is the open bracket. This is followed by the formula P which should then be followed by a binary connective. But P is followed by $\neg$, which is not a binary connective.

1.14 Remark: By part (1) of the unique readability theorem, in a list of formulas $F_1, F_2, \cdots, F_n$ it is not necessary to separate the formulas by commas. The same is not true, for example, of a list of integers.

1.15 Note: In addition to proving properties using induction on formulas, we can also make inductive definitions.

1.16 Example: Define a function λ on formulas inductively as follows. If $F = P$, where P is a propositional variable, then define $\lambda(F) = 1$. If $F = \neg G$, where G is a formula, then define $\lambda(F) = 1 + \lambda(G)$. If $F = (G \times H)$, where G and H are formulas and $\times$ is a binary connective, then define $\lambda(F) = 3 + \lambda(G) + \lambda(H)$. It is not hard to see (and it is not hard to prove by induction on formulas) that $\lambda(F)$ is just the length of F.

1.17 Definition: For a formula F, we define the set $SF(F)$ of **subformulas** of F inductively as follows. If $F = P$ then define $SF(F) = \{F\}$. If $F = \neg G$, where G is a formula, then define $SF(F) = \{F\} \cup SF(G)$. If $F = (G \times H)$, where G and H are formulas and $\times$ is a binary connective, then define $SF(F) = \{F\} \cup SF(G) \cup SF(H)$.

1.18 Note: It is not very hard to see (though it is a bit long and tedious to prove) that a subformula of F is the same thing as a formula which is a substring of F. We could define a subformula in this way, but it is often easier to prove things about subformulas using the above inductive definition.

1.19 Definition: The standard notation that we have been using for formulas is called **infix notation**. There is another notation for formulas known as **prefix notation** in which for example the formula $(F \wedge G)$ is written as $\wedge FG$. To be precise, a **formula** in **prefix notation** is a non-empty string which can be obtained by using finitely many applications of the following rules:

1. Every propositional variable is a formula in prefix notation.
2. If F is a formula in prefix notation, then so is $\neg F$.
3. If F and G are formulas in prefix notation and $\times$ is a binary connective, then $\times FG$ is a formula in prefix notation.

1.20 Note: We can make a **derivation** for a formula in prefix notation just as we did earlier for a formula (in infix notation). There is a one-to-one correspondence between formulas and formulas in prefix notation. A formula and its prefix notation counterpart can be given derivations with the same justifying rules used on each line.

1.21 Example: Convert the formula $F = (((P \to \neg(Q \wedge \neg R)) \vee \neg(\neg P \leftrightarrow R))$ to prefix notation.

Solution: With practice, this conversion can be done without the bother of writing out a derivation, but here is a derivation for F (on the left) and simultaneously for its prefix notation counterpart (on the right):

1.	R	rule 1	R
2.	P	rule 1	P
3.	$\neg P$	rule 2 on 2	$\neg P$
4.	$(\neg P \leftrightarrow R)$	rule 3 $\leftrightarrow$ on 3 and 1	$\leftrightarrow \neg PR$
5.	$\neg(\neg P \leftrightarrow R)$	rule 2 on 4	$\neg \leftrightarrow \neg PR$
6.	$\neg R$	rule 2 on 1	$\neg R$
7.	Q	rule 1	Q
8.	$(Q \wedge \neg R)$	rule 3$\wedge$ on 7 and 6	$\wedge Q \neg R$
9.	$\neg(Q \wedge \neg R)$	rule 2 on 8	$\neg \wedge Q \neg R$
11.	$(P \to \neg(Q \wedge \neg R))$	rule 3 $\to$ on 2 and 9	$\to P \neg \wedge Q \neg R$
12.	F	rule 3$\vee$ on 11 and 5	$\vee \to P \neg \wedge Q \neg R \neg \leftrightarrow \neg PR$

1.22 Note: A formula in standard notation is easier to read than a formula in prefix notation, but prefix notation has the advantage that there is a nice easy algorithm, called the **tub algorithm**, for testing a given string to determine whether it is a formula (or a list of formulas) in prefix notation. The algorithm works as follows.

Step 0: Let n be a counter (to count the number of formulas). Look at the last symbol in the given string.

Step 1: If the last symbol is not a propositional variable, then the given string cannot be a list of prefix notation formulas. If the the last symbol is a propositional variable, then set the counter to $n = 1$ and begin to work through the given string, from right to left, one symbol at a time.

Step 2: Look at the next symbol to the left: If it is a propositional variable then increase the value of n by 1; if it is the symbol $\neg$ then leave the value of n unchanged; if

it is a binary connective then reduce the value of n by 1.

Step 3: At this stage, if $n = 0$ then the given string cannot be a list of prefix notation formulas, and if $n > 0$ then the rightmost portion of the given string (the portion which has been examined so far) is a list of n formulas in prefix notation. If we have not yet examined every symbol in the given string, then go back to step 2.

1.23 Example: Let $X = \wedge\neg P \vee \neg \to QR\neg\neg P \leftrightarrow Q \wedge PR$. Use the tub algorithm to determine whether X is a list of formulas in prefix notation. If so, convert it to a list of formulas in standard notation.

Solution: The results of the tub algorithm are shown below. The successive values of the counter n are shown beneath each symbol, and should be read from right to left.

$$\begin{array}{c} \wedge\ \neg\ P\ \vee\ \neg\ \to\ Q\ R\ \neg\ \neg\ P\ \leftrightarrow\ Q\ \wedge\ P\ R \\ 2\ 3\ 3\ 2\ 3\ 3\ 4\ 3\ 2\ 2\ 2\ 1\ 2\ 1\ 2\ 1\ 0 \end{array}$$

The final counter value is 2, so the given string is a list of two formulas in prefix notation. When this algorithm is performed by hand, it is convenient to draw tubs underneath the prefix-notation formulas as they are counted (hence the name of the algorithm) instead of recording the successive values of the counter n: the tubs are shown below:

The tubs help to convert the two prefix-notation formulas into standard notation (without bothering to write out verification columns). The corresponding list of formulas in standard notation is $(\neg P \wedge (\neg(Q \to R) \vee \neg\neg P))$, $(Q \leftrightarrow (P \wedge R))$

1.24 Definition: For formulas F and G and for a propositional variable symbol P, we define a formula $[F]_{P\mapsto G}$ inductively as follows: for a propositional variable symbol Q, for formulas H and K, and for $\times \in \{\wedge, \vee, \to, \leftrightarrow\}$, we set

1. $[Q]_{P\mapsto G} = \begin{cases} Q \text{ , if } Q \neq P \\ G \text{ , if } Q = P \end{cases}$
2. $[\neg H]_{P\mapsto G} = \neg[H]_{P\mapsto G}$
3. $[(H \times K)]_{P\mapsto G} = \big([H]_{P\mapsto G} \times [K]_{P\mapsto G}\big)$.

It is not hard to check, using induction on formulas, that $[F]_{P\mapsto G}$ is indeed a formula; it is called the **formula obtained from F by replacing each occurrence of P by G.**

1.25 Example: Let $F = \big(\neg P \to (Q \wedge P)\big)$ and $G = (P \vee \neg Q)$. Find $[F]_{P\mapsto G}$.

Solution: Using the inductive definition, we have

$$\begin{aligned} [F]_{P\mapsto G} &= \big[\big(\neg P \to (Q \wedge P)\big)\big]_{P\mapsto G} \\ &= \big([\neg P]_{P\mapsto G} \to [(Q \wedge P)]_{P\mapsto G}\big) \\ &= \big(\neg[P]_{P\mapsto G} \to ([Q]_{P\mapsto G} \wedge [P]_{P\mapsto G})\big) \\ &= \big(\neg G \to (Q \wedge G)\big) \\ &= \big(\neg(P \vee \neg Q) \to (Q \wedge (P \vee \neg Q))\big) \end{aligned}$$

Notice that we could have obtained $[F]_{P\mapsto G}$ in one step simply by replacing each occurence of P in F by the formula G.

Chapter 2. Truth-Evaluations and Truth-Tables

2.1 Note: In this and future chapters, we shall often omit the outermost pair of brackets from formulas, for example we might write $(F \vee G)$ as $F \vee G$.

2.2 Definition: A **truth-evaluation** is a map $\alpha : \{P_1, P_2, P_3, \cdots\} \to \{0, 1\}$, where we are using 1 to represent true and 0 to represent false. Given a truth-evaluation α, we define the **truth-value** $\alpha(F)$, for any formula F, inductively as follows:

1. $\alpha(P)$ is already known for all propositional variables P.
2. If G is a formula then $\alpha(\neg G)$ is defined according to the table

$$\begin{array}{cc} G & \neg G \\ 1 & 0 \\ 0 & 1 \end{array}$$

so we have $\alpha(\neg G) = \begin{cases} 1 & \text{if } \alpha(G) = 0, \text{ and} \\ 0 & \text{if } \alpha(G) = 1. \end{cases}$

3. If G and H are formulas and $\times \in \{\wedge, \vee, \to, \leftrightarrow\}$, then $\alpha(G \times H)$ is defined according to the table

$$\begin{array}{cccccc} G & H & (G \wedge H) & (G \vee H) & (G \to H) & (G \leftrightarrow H) \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

so for example we have $\alpha(G \wedge H) = \begin{cases} 1 & \text{if } \alpha(G) = 1 \text{ and } \alpha(H) = 1, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$

When $\alpha(F) = 1$ we say that F is **true** under the truth-evaluation α, and when $\alpha(F) = 0$ we say that F is **false** under α.

2.3 Example: Let $F = ((P \wedge \neg(Q \to P)) \vee (R \leftrightarrow \neg Q)$, and let α be any truth-evaluation with $\alpha(P) = 1$, $\alpha(Q) = 0$ and $\alpha(R) = 1$. Determine whether F is true under α.

Solution: We make a derivation $F_1\ F_2\ \cdots\ F_n$ for F, and under each formula F_i we put the truth value $\alpha(F_i)$, which we find using the above definition.

P	Q	R	$Q \to P$	$\neg(Q \to P)$	$P \wedge \neg(Q \to P)$	$\neg Q$	$R \leftrightarrow \neg Q$	F
1	0	1	1	0	0	1	1	1

The table shows that $\alpha(F) = 1$.

2.4 Definition: A truth-evaluation on $P_1, \cdots, P_n$ is a map $\alpha : \{P_1, P_2, \cdots, P_n\} \to \{0, 1\}$ (there are 2^n such truth-evaluations). A **truth-table** (on $P_1, P_2, \cdots P_n$) is a table in which

1. the header row is a derivation $F_1\ F_2\ \cdots\ F_n\ \cdots\ F_l$, where the formulas F_i contain no propositional variables other than $P_1, \cdots, P_n$, with $F_i = P_i$ for $i \leq n$.
2. There are 2^n rows (not counting the header row): for each of the 2^n truth-evaluations α on $P_1, \cdots, P_n$, there is a row of the form $\alpha(F_1)\ \alpha(F_2)\ \cdots\ \alpha(F_l)$.
3. The rows are ordered so that first n columns (headed by $P_1, \cdots, P_n$) list the binary numbers in decreasing order from $11\cdots1$ at the top down to $00\cdots0$ at the bottom.

We say it is a truth-table for F when F is one of the formulas F_i. The column headed by F_i will be called the F_i-column.

2.5 Example: Make a truth-table for the formula $F = \neg((P \vee \neg Q) \to R)$.

Solution: We make a table, as in example 2.2, but with $2^3 = 8$ rows.

P	Q	R	$\neg Q$	$P \vee \neg Q$	$(P \vee \neg Q) \to R$	F
1	1	1	0	1	1	0
1	1	0	0	1	0	1
1	0	1	1	1	1	0
1	0	0	1	1	0	1
0	1	1	0	0	1	0
0	1	0	0	0	1	0
0	0	1	1	1	1	0
0	0	0	1	1	0	1

2.6 Definition: Let F and G be formulas, and let $\mathcal{S}$ be a set of formulas.

F is a **tautology**, written $\models F$, means that for all truth-evaluations α, $\alpha(F) = 1$.

F is **truth-equivalent** to G, written F treq G, means that for all truth-evaluations α, $\alpha(F) = \alpha(G)$.

$\mathcal{S}$ is **satisfiable** means that there exists a truth-evaluation α such that for all $F \in \mathcal{S}$, $\alpha(F) = 1$. Such a truth-evaluation α is said to **satisfy** $\mathcal{S}$. When $\mathcal{S} = \{F\}$ is satisfiable, we often omit the set-brackets and say that F is satisfiable.

The argument "$\mathcal{S}$ therefore F" is **valid**, written $\mathcal{S} \models F$, means that for all truth-evaluations α, if ($\alpha(G) = 1$ for every $G \in \mathcal{S}$) then $\alpha(F) = 1$. The formulas in $\mathcal{S}$ are called the **premises** of the argument, and the formula F is called the **conclusion**. When $\mathcal{S}$ is finite we often omit the set-brackets.

2.7 Example: Let $F = (P \leftrightarrow ((Q \wedge \neg R) \vee S)) \vee (P \to \neg S)$. Determine whether F is a tautology.

Solution: We make a truth-table for F.

P	Q	R	S	$\neg R$	$Q \wedge \neg R$	$(Q \wedge \neg R) \vee S$	$P \leftrightarrow ((Q \wedge \neg R) \vee S)$	$\neg S$	$P \to \neg S$	F
1	1	1	1	0	0	1	1	0	0	1
1	1	1	0	0	0	0	0	1	1	1
1	1	0	1	1	1	1	1	0	0	1
1	1	0	0	1	1	1	1	1	1	1
1	0	1	1	0	0	1	1	0	0	1
1	0	1	0	0	0	0	0	1	1	1
1	0	0	1	1	0	1	1	0	0	1
1	0	0	0	1	0	0	0	1	1	1
0	1	1	1	0	0	1	0	0	1	1
0	1	1	0	0	0	0	1	1	1	1
0	1	0	1	1	1	1	0	0	1	1
0	1	0	0	1	1	1	0	1	1	1
0	0	1	1	0	0	1	0	0	1	1
0	0	1	0	0	0	0	1	1	1	1
0	0	0	1	1	0	1	0	0	1	1
0	0	0	0	1	0	0	1	1	1	1

Since all the entries in the F-column are 1, F is a tautology.

2.8 Example: Let $F = (P \vee Q) \to R$ and $G = (P \to R) \vee (Q \to R)$. Determine whether F treq G.

Solution: In general, F treq G if and only if the F-column is identical to the G-column in a truth table for F and G. We make a table:

P	Q	R	$P \vee Q$	F	$P \to R$	$Q \to R$	G
1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	0
1	0	1	1	1	1	1	1
1	0	0	1	0	0	1	1
0	1	1	1	1	1	1	1
0	1	0	1	0	1	0	1
0	0	1	0	1	1	1	1
0	0	0	0	1	1	1	1

The F-column is not the same as the G-column, for example on the 4^{th} row, F is false and G is true. So F is not truth equivalent to G.

2.9 Example: Let $F = (P \vee \neg Q) \to R$, $G = P \leftrightarrow (Q \wedge R)$, $H = (Q \to R)$ and $K = \neg(\neg Q \wedge R)$. Determine whether $\{F, G, H\} \models K$.

Solution: In general, we have $\{F_1, \cdots, F_n\} \models K$ if and only if in a truth-table for the formulas $F_1, \cdots, F_n$ and K, for every row in which $F_1, \cdots, F_n$ are all true, we also have K true. We make a truth-table for F, G, H and K:

P	Q	R	$\neg Q$	$P \vee \neg Q$	F	$Q \wedge R$	G	H	$\neg Q \wedge R$	K
1	1	1	0	1	1	1	1	1	0	1
1	1	0	0	1	0	0	0	0	0	1
1	0	1	1	1	1	0	0	1	1	0
1	0	0	1	1	0	0	0	1	0	1
0	1	1	0	0	1	1	0	1	0	1
0	1	0	0	0	1	0	1	0	0	1
0	0	1	1	1	1	0	1	1	1	0
0	0	0	1	1	0	0	1	1	0	1

On row 7, F G and H are all true but K is false. This implies that $\{F, G, H\} \not\models K$.

2.10 Example: Determine whether $\{P \vee Q, \neg Q, P \to Q\} \models \neg P$.

Solution: We make a truth-table

P	Q	$P \vee Q$	$\neg Q$	$P \to Q$	$\neg P$
1	1	1	0	1	0
1	0	1	1	0	0
0	1	1	0	1	1
0	0	0	1	1	1

Notice that there are no rows in which the premises are all true (in other words, the set $\{P \vee Q, \neg Q, P \to Q\}$ is unsatisfiable). In this situation, the argument is valid (we don't even need to look at the last column of the table).

2.11 Example: It is not hard to check that $\{P \to Q, P\} \models Q$. This valid argument is called **modus ponens**. A verbal example of this argument is "If it is snowing then I will have to shovel my driveway. It is snowing. Therefore I will have to shovel my driveway". Another valid argument is the **chain rule** $\{P \to Q, Q \to R\} \models P \to R$. A verbal example of this is "Bats are mammals. Mammals can't fly. Therefore bats can't fly" (here we have $P =$"x is a bat", $Q =$"x is a mammal", and $R =$"x cannot fly"). Note that although this argument is valid, its conclusion is false.

2.12 Example: Let $F = (\neg P \to Q) \wedge (P \to \neg R)$, $G = \neg(P \wedge Q) \to R)$ and $H = \neg Q \vee (P \leftrightarrow R)$. Determine whether $\{F, G, H\}$ is satisfiable.

Solution: In general, $\{F_1, \cdots, F_n\}$ is satisfiable if and only if in a truth-table for the formulas $F_1, \cdots, F_n$, there is a row in which the formulas $F_1, \cdots, F_n$ are all true. We make a truth-table for the given formulas F, G and H.

P	Q	R	$\neg P$	$\neg P \to Q$	$\neg R$	$P \to \neg R$	F	$P \wedge Q$	$\neg(P \wedge Q)$	G	$\neg Q$	$P \leftrightarrow R$	H
1	1	1	0	1	0	0	0	1	0	1	0	1	1
1	1	0	0	1	1	1	1	1	0	1	0	0	0
1	0	1	0	1	0	0	0	0	1	1	1	1	1
1	0	0	0	1	1	1	1	0	1	0	1	0	1
0	1	1	1	1	0	1	1	0	1	1	0	0	0
0	1	0	1	1	1	1	1	0	1	0	0	1	1
0	0	1	1	0	0	1	0	0	1	1	1	0	0
0	0	0	1	0	1	1	0	0	1	0	1	1	1

A careful look at the table reveals that there is no row in which F, G and H are all true. So $\{F, G, H\}$ is not satisfiable.

2.13 Example: Let $\mathcal{S}_n = \{P_n\} \cup \{P_1 \to P_2, P_2 \to P_3, P_3 \to P_4, \cdots\}$. Find all the truth-evaluations which satisfy $\mathcal{S}_n$.

Solution: Let α be a truth-evaluation which satisfies $\mathcal{S}_n$. Let k be the smallest integer such that $\alpha(P_k) = 1$ (so we have $\alpha(P_i) = 0$ for all $i < k$). Since $P_n \in \mathcal{S}_n$, so $\alpha(P_n) = 1$, we must have $1 \leq k \leq n$. Since for all i we have $(P_i \to P_{i+1}) \in \mathcal{S}_n$ so $\alpha(P_i) = 1 \Longrightarrow \alpha(P_{i+1}) = 1$, we see that $1 = \alpha(P_k) = \alpha(P_{k+1}) = \alpha(P_{k+2}) = \cdots$. Conversely, if $1 \leq k \leq n$ and α_k is the truth-evaluation $\alpha_k(P_i) = \begin{cases} 0 \text{ if } i < k \\ 1 \text{ if } i \geq k \end{cases}$ then α_k satisfies $\mathcal{S}_n$.

2.14 Note: Let F and G be formulas. Consider the following table.

F	G	$F \leftrightarrow G$	$\neg G$	$F \to \neg G$
1	1	1	0	0
1	0	0	1	1
0	1	0	0	1
0	0	1	1	1

Notice that on the first row (when F and G are both true) we have $F \leftrightarrow G$ true and $F \to \neg G$ false. This might seem to imply that $(F \leftrightarrow G) \not\models (F \to \neg G)$, but it does not! For example, if $F = P$ and $G = \neg P \wedge Q$, then we never have F and G both true, so the combination of truth-values shown in the first row of the above table never actually occurs. The above table is not actually a truth-table as defined in 2.4. Rather, it is a table of possible combinations of truth-values which may or may not actually occur.

2.15 Theorem: *Let F, G and $F_1, \cdots, F_n$ be formulas and let $\mathcal{S}$ be a set of formulas. Write the formula $(\cdots((F_1 \wedge F_2) \wedge F_3) \wedge \cdots \wedge F_n)$ as $(F_1 \wedge \cdots \wedge F_n)$. Then*

(1) F is a tautology if and only if $\neg F$ is not satisfiable.
(2) F is satisfiable if and only if $\neg F$ is not a tautology.
(3) $\models F$ if and only if $\emptyset \models F$.
(4) $F \models G$ if and only if $\models (F \to G)$.
(5) F treq $G \iff \models (F \leftrightarrow G) \iff \big(F \models G$ and $G \models F\big)$.
(6) $\{F_1, \cdots, F_n\} \models G \iff (F_1 \wedge \cdots \wedge F_n) \models G \iff \models (F_1 \wedge \cdots \wedge F_n) \to G$.
(7) $\{F_1, \cdots, F_n\}$ is satisfiable if and only if $(F_1 \wedge \cdots \wedge F_n)$ is satisfiable.
(8) $\mathcal{S} \models F$ if and only if $\mathcal{S} \cup \{\neg F\}$ is not satisfiable.

Proof: We shall only prove a few parts of this theorem. First, let us prove part 3. Recall the mathematical convention that any statement of the form "for all $x \in \emptyset$, $A(x)$" is considered to be true (where $A(x)$ can be any statement about x), so that in particular, for every truth-evaluation α, the statement "$\alpha(G) = 1$ for all $G \in \emptyset$" is true. Also notice that for any statements A and B, when A is true, the truth of the statement B is the same as the truth of the statement "if A then B". So we have

$$\begin{aligned} \models F &\iff \text{for all truth-evaluations } \alpha,\ \alpha(F) = 1 \\ &\iff \text{for all truth-evaluations } \alpha, \text{ if } \alpha(G) = 1 \text{ for all } G \in \emptyset \text{ then } \alpha(F) = 1 \\ &\iff \emptyset \models F \end{aligned}$$

Next, let us prove part 4. Notice that from the definition of the truth of a formula of the form $(F \to G)$, we have $\alpha(F \to G) = 1 \iff \big($ if $\alpha(F) = 1$ then $\alpha(G) = 1\big)$. So

$$\begin{aligned} F \models G &\iff \text{for all truth evaluations } \alpha, \text{ if } \alpha(F) = 1 \text{ then } \alpha(G) = 1 \\ &\iff \text{for all truth-evaluations } \alpha,\ \alpha(F \to G) = 1 \\ &\iff \models (F \to G) \end{aligned}$$

Finally, let us prove part 8. We have

$$\begin{aligned} &\mathcal{S} \cup \{\neg F\} \text{ is not satisfiable} \\ &\iff \text{there is no truth-evaluation } \alpha \text{ such that } \alpha(G) = 1 \text{ for all } G \in \mathcal{S} \cup \{\neg F\} \\ &\iff \text{there is no truth-evaluation } \alpha \text{ such that } \big(\alpha(G) = 1 \text{ for all } G \in \mathcal{S} \text{ and } \alpha(\neg F) = 1\big) \\ &\iff \text{for all truth-evaluations } \alpha, \text{ we do not have } \big(\alpha(G) = 1 \text{ for all } G \in \mathcal{S} \text{ and } \alpha(F) = 0\big) \\ &\iff \text{for all truth-evaluations } \alpha, \text{ either we do not have } \alpha(G) = 1 \text{ for all } G \in \mathcal{S}, \text{ or } \alpha(F) = 1 \\ &\iff \text{for all truth-evaluations } \alpha, \text{ if } \alpha(G) = 1 \text{ for all } G \in \mathcal{S} \text{ then } \alpha(F) = 1 \\ &\iff \mathcal{S} \models F \end{aligned}$$

Chapter 3. Disjunctive and Conjunctive Normal Forms

3.1 Definition: A **literal** is a formula which is either of the form P or the form $\neg P$ for some propositional variable P. A **disjunctive-form constituent** (or a **DF constituent**) is a formula of the form

$$(\cdots((L_1 \wedge L_2) \wedge L_3) \wedge \cdots \wedge L_k)$$

for some literals L_i. Usually we shall omit all but the outer pair of brackets and write the above DF constituent as $(L_1 \wedge L_2 \wedge \cdots \wedge L_k)$. A **disjunctive-form formula** (or a **DF formula**) is a formula of the form

$$(\cdots((C_1 \vee C_2) \vee C_3) \vee \cdots \vee C_l)$$

where each C_i is a DF constituent. We shall usually omit the brackets. We also consider the empty string to be a DF formula.

3.2 Example: The string $(P \wedge Q \wedge \neg R) \vee (R \wedge \neg R \wedge Q) \vee (\neg P)$ is a DF formula.

3.3 Definition: A **disjunctive-normal-form constituent** (or a **DNF constituent**) on the set $\{P_1, P_2, \cdots, P_n\}$ is a formula of the form

$$(\cdots((L_1 \wedge L_2) \wedge L_3) \wedge \cdots \wedge L_n)$$

where for each i, either $L_i = P_i$ or $L_i = \neg P_i$. Usually we shall omit all but the outer pair of brackets.

Note that on $\{P_1, \cdots, P_n\}$, there are 2^n different DNF constituents, and they correspond naturally to the 2^n truth-evaluations, and hence to the 2^n rows in a truth-table, and to the n-digit binary numbers from $11\cdots1$ down to $00\cdots0$: the DNF constituent $C = (L_1 \wedge \cdots \wedge L_n)$ corresponds to the unique truth-evaluation α for which $\alpha(C) = 1$ which is given by $\alpha(L_i) = 1$ for all i, that is $\alpha(P_i) = \begin{cases} 1 \text{ , if } L_i = P_i \\ 0 \text{ , if } L_i = \neg P_i \text{ ,} \end{cases}$ and this corresponds in turn to the binary number $\alpha(P_1)\alpha(P_2)\cdots\alpha(P_n)$.

A **disjunctive-normal-form formula** (or **DNF formula**) on $\{P_1, P_2, \cdots, P_n\}$ is a formula of the form

$$(\cdots((C_1 \vee C_2) \vee C_3) \vee \cdots \vee C_l)$$

where the C_i are distinct DNF constituents on $\{P_1, \cdots, P_n\}$ which are ordered so that the corresponding binary numbers are in decreasing order. We shall usually omit some, or all, of the brackets. We also consider the empty string $\emptyset$ to be a DNF formula; and we shall call it **the empty DNF formula**.

3.4 Example: The string $(P \wedge \neg Q \wedge R) \vee (\neg P \wedge Q \wedge R) \vee (\neg P \wedge \neg Q \wedge R) \vee (\neg P \wedge \neg Q \wedge \neg R)$ is a DNF formula on $\{P, Q, R\}$.

3.5 Theorem: *(The DNF Theorem) Every formula which uses only propositional variables from $\{P_1, \cdots, P_n\}$ is truth-equivalent to a unique DNF formula on $\{P_1, \cdots, P_n\}$.*

Proof: On $\{P_1, \cdots, P_n\}$, each DNF constituent corresponds to a row of a truth-table. If C is a DNF constituent, then in the C-column of a truth-table, there is a 1 in the row which corresponds to C, and there are 0's in all the other rows. If F is a DNF formula then the F-column of a truth-table has one 1 for each DNF constituent in F. Thus there are 2^{2^n} DNF formulas on $\{P_1, \cdots, P_n\}$ and they correspond to all of the 2^{2^n} possible columns in a truth-table on $\{P_1, \cdots, P_n\}$. The empty DNF formula corresponds to the zero column.

3.6 Example: Let $F = \neg((P \to \neg Q) \leftrightarrow \neg(\neg Q \wedge R))$. Put F into DNF, in other words find a DNF formula which is truth-equivalent to F.

Solution: We make a truth-table for F.

P	Q	R	$\neg Q$	$P \to \neg Q$	$Q \wedge R$	$\neg(Q \wedge R)$	$(P \to \neg Q) \leftrightarrow \neg(Q \wedge R)$	F
1	1	1	0	0	1	0	1	0
1	1	0	0	0	0	1	0	1
1	0	1	1	1	0	1	1	0
1	0	0	1	1	0	1	1	0
0	1	1	0	1	1	0	0	1
0	1	0	0	1	0	1	1	0
0	0	1	1	1	0	1	1	0
0	0	0	1	1	0	1	1	0

From the last column we see that F treq $(P \wedge Q \wedge \neg R) \vee (\neg P \wedge Q \wedge R)$.

3.7 Definition: A **conjunctive-form constituent**, or a **CF constituent**, is a formula of the form

$$(\cdots((L_1 \vee L_2) \vee L_3) \vee \cdots \vee L_k)$$

where each L_i is a literal. Usually we shall omit all but the outer pair of brackets. A **conjunctive-form formula**, or a **CF formula**, is a formula of the form

$$(\cdots((C_1 \wedge C_2) \wedge C_3) \wedge \cdots \wedge C_l)$$

where each C_i is a CF constituent. Again, we shall usually omit some brackets.

3.8 Definition: A **conjunctive-normal-form constituent**, or a **CNF constituent**, on $\{P_1, P_2, \cdots, P_n\}$ is a formula of the form

$$(\cdots((L_1 \vee L_2) \vee L_3) \vee \cdots \vee L_n)$$

where for each i, either $L_i = P_i$ or $L_i = \neg P_i$. We shall usually omit brackets.

Note that on $\{P_1, \cdots, P_n\}$, there are 2^n distinct CNF constituents which correpond naturally to the 2^n truth-evaluations; the CNF constituent $C = (L_1 \vee \cdots \vee L_n)$ corresponds to the unique truth-evaluation α such that $\alpha(C) = 0$ which is given by $\alpha(L_i) = 0$ for all i, that is $\alpha(P_i) = \begin{cases} 1 \text{ , if } L_i = \neg P_i \\ 0 \text{ , if } L_i = P_i \text{ .} \end{cases}$

A **conjunctive-normal-form formula**, or a **CNF formula**, on $\{P_1, \cdots, P_n\}$ is a formula of the form

$$(\cdots((C_1 \wedge C_2) \wedge C_3) \wedge \cdots \wedge C_l)$$

where the C_i are distinct CNF constituents on $\{P_1, \cdots, P_n\}$ which appear in the same order that their corresponding truth-evaluations occur in a truth-table. Again, we shall usually omit some of the brackets. The empty string is also considered to be a CNF formula; it is called **the empty CNF formula**.

3.9 Theorem: *(The CNF Theorem) Every formula which only uses propositional variables from $\{P_1, \cdots, P_n\}$ is truth-equivalent to a unique CNF formula on $\{P_1, \cdots, P_n\}$.*

Proof: The proof is similar to the proof of the DNF Form Theorem, except that in a truth-table column for a CNF formula, there is one 0 for each constituent, instead of one 1 for each constituent.

3.10 Example: Let $F = (\neg P \vee Q \vee \neg R) \wedge (P \vee \neg Q \vee R) \wedge (P \vee Q \vee \neg R)$. Note that F is a CNF formula on $\{P, Q, R\}$. Make a truth-table for F.

Solution: F has three CNF constituents, so in its truth-table column there are three 0's. They occur in the rows which correspond to the CNF constituents, that is the rows which begin with the binary numbers 101, 010 and 001, that is in the 3^{rd}, 6^{th} and 7^{th} rows:

P	Q	R	F
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	1

3.11 Note: Given a DNF formula F we can put F into CNF as follows. First list all the DNF constituents which do *not* occur in F to put the formula $\neg F$ into DNF. Then interchange all the $\wedge$ symbols with the $\vee$ symbols and reverse all the literals, that is for each propositional variable P, replace each occurrence of $\neg P$ with P and *vice versa*. We can use a similar method to put a given CNF formula into DNF.

3.12 Definition: A given set of connectives is said to be **adequate** if every possible column in any truth-table can be realized as the column of a formula which only uses connectives from the given set.

3.13 Example: Since every formula is truth-equivalent to a DNF formula, and a DNF formula only uses connectives from the set $\{\neg, \wedge, \vee\}$, we see that $\{\neg, \wedge, \vee\}$ is an adequate set of connectives.

3.14 Example: Since $F \vee G$ treq $\neg(\neg F \wedge \neg G)$, we have

$$(C_1 \vee C_2 \vee \cdots \vee C_l) \text{ treq } \neg(\neg C_1 \wedge \neg C_2 \wedge \cdots \wedge \neg C_l)$$

and so every DNF formula is truth-equivalent to a formula which only uses connectives from the set $\{\neg, \wedge\}$. Thus $\{\neg, \wedge\}$ is an adequate set of connectives.

Similarly, since $(F \wedge G)$ treq $\neg(\neg F \vee \neg G)$, we see that $\{\neg, \vee\}$ is an adequate set of connectives.

3.15 Example: Show that $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$ is not an adequate set of connectives.

Solution: Let α be the truth-evaluation given by $\alpha(P) = 1$ for all propositional variables P. We claim that for every formula F which only uses connectives from $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$ we have $\alpha(F) = 1$. We prove this claim by induction on formulas which only use these connectives. If P is a propositional variable then $\alpha(P) = 1$ by the definition of α. If G and H are formulas which use only these connectives with $\alpha(G) = \alpha(H) = 1$ then we have $\alpha(G \wedge H) = \alpha(G \vee H) = \alpha(G \rightarrow H) = \alpha(G \leftrightarrow H) = 1$. By induction on formulas involving only these connectives, the claim is true. In particular, any truth-table column with a 0 on the first row cannot be realized as the column of any formula which uses only connectives from $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Chapter 4. Derivation of Truth-Equivalence

4.1 Theorem: *Truth-equivalence is an equivalence relation. This means that for all formulas F, G and H we have*

1. F treq F
2. F treq $G \Longrightarrow G$ treq F
3. $\big(F$ treq G *and* G treq $H\big) \Longrightarrow F$ treq H

Proof: These properties follow immediately from the definition of truth-equivalence. Here is a proof of part (3). Suppose that F treq G and G treq H. Let α be a truth-evaluation. Since F treq G we have $\alpha(F) = \alpha(G)$ and since G treq H we have $\alpha(G) = \alpha(H)$. Thus $\alpha(F) = \alpha(H)$. Since α was arbitrary, this shows that F treq H.

4.2 Definition: For any formulas F, G and H, it is easy to verify that we have the following 24 truth-equivalences, which we shall call the **Basic Truth-Equivalences**.

(Identity)	1.	F treq F
(Double Negation)	2.	F treq $\neg\neg F$
(Commutativity)	3.	$F \wedge G$ treq $G \wedge F$
	4.	$F \vee G$ treq $G \vee F$
(Associativity)	5.	$F \wedge (G \wedge H)$ treq $(F \wedge G) \wedge H$
	6.	$F \vee (G \vee H)$ treq $(F \vee G) \vee H$
(DeMorgan's Law)	7.	$\neg(F \wedge G)$ treq $(\neg F \vee \neg G)$
	8.	$\neg(F \vee G)$ treq $(\neg F \wedge \neg G)$
(Distributivity)	9.	$F \wedge (G \vee H)$ treq $(F \wedge G) \vee (F \wedge H)$
	10.	$F \vee (G \wedge H)$ treq $(F \vee G) \wedge (F \vee H)$
(Idempotence)	11.	$F \wedge F$ treq F
	12.	$F \vee F$ treq F
(Absorption)	13.	$F \wedge (F \vee G)$ treq F
	14.	$F \vee (F \wedge G)$ treq F
(Tautology)	15.	$F \wedge (G \vee \neg G)$ treq F
	16.	$F \vee (G \vee \neg G)$ treq $G \vee \neg G$
(Contradiction)	17.	$F \wedge (G \wedge \neg G)$ treq $G \wedge \neg G$
	18.	$F \vee (G \wedge \neg G)$ treq F
(Contrapositive)	19.	$F \to G$ treq $\neg G \to \neg F$
(Implication)	20.	$F \to G$ treq $\neg F \vee G$
	21.	$\neg(F \to G)$ treq $F \wedge \neg G$
(If and Only If)	22.	$F \leftrightarrow G$ treq $(F \wedge G) \vee (\neg F \wedge \neg G)$
	23.	$F \leftrightarrow G$ treq $(\neg F \vee G) \wedge (F \vee \neg G)$
	24.	$F \leftrightarrow G$ treq $(F \to G) \wedge (G \to F)$

4.3 Note: From these basic truth-equivalences, we can derive other truth-equivalences. Soon, we shall give a precise definition as to what constitutes such a derivation. Furthermore, we shall see that in fact every truth-equivalence can be derived from these basic truth-equivalences. Before giving the precise definition of a derivation, we provide one example.

4.4 Example: Let F and G be formulas. Use the basic truth-equivalences to derive the truth-equivalence $F \wedge (F \to G)$ treq $F \wedge G$.

Solution: We have

1.	$F \wedge (F \to G)$	treq	$F \wedge (\neg F \vee G)$	by Implication
2.		treq	$(F \wedge \neg F) \vee (F \wedge G)$	by Distributivity
3.		treq	$(F \wedge G) \vee (F \wedge \neg F)$	by Commutativity
4.		treq	$F \wedge G$	by Contradiction

4.5 Note: Notice that in the first line of the above example, the Implication Rule was applied to the subformula $(F \to G)$ in the formula $F \wedge (F \to G)$. In a truth-equivalence derivation, the basic truth-equivalences can be applied to sub-formulas. The following theorem is needed to make this precise.

4.6 Theorem: *(Substitution) Let F and G be formulas with F treq G, let H be any formula and let P be a propositional variable. Then*

$$[H]_{P \mapsto F} \text{ treq } [H]_{P \mapsto G}\,.$$

We say the truth-equivalence $[H]_{P \mapsto F}$ treq $[H]_{P \mapsto G}$ is **obtained by substitution** *from the truth-equivalence F treq G (or from the truth-equivalence G treq F).*

Proof: We shall prove the theorem by induction on the formula H, but first we make two preliminary remarks. Let K_1, K_2, L_1 and L_2 be formulas. Note that if K_1 treq K_2 then we also have $\neg K_1$ treq $\neg K_2$: indeed for any truth-evaluation α we have $\alpha(\neg K_1) = 1 \iff \alpha(K_1) = 0 \iff \alpha(K_2) = 0 \iff \alpha(\neg K_2) = 1$. Secondly, note that if K_1 treq K_2 and L_1 treq L_2 and if $* \in \{\wedge, \vee, \to, \leftrightarrow\}$, then we also have $(K_1 * L_1)$ treq $(K_2 * L_2)$: for example, for any truth-evaluation α we have $\alpha(K_1 \wedge L_1) = 1 \iff \big(\alpha(K_1) = 1 \text{ and } \alpha(L_1) = 1\big) \iff \big(\alpha(K_2) = 1 \text{ and } \alpha(L_2) = 1\big) \iff \alpha(K_2 \wedge L_2) = 1$.

Now we begin the inductive proof. Case 1: suppose that $H = Q$ where Q is a propositional variable. If $Q \neq P$ then we have $[H]_{P \mapsto F} = Q = [H]_{P \mapsto G}$. If $Q = P$ then we have $[H]_{P \mapsto F} = F$ and $[H]_{P \mapsto F} = H$. In either case we have $[H]_{P \mapsto F}$ treq $[H]_{P \mapsto G}$.

Case 2: suppose that $H = \neg K$ where $[K]_{P \mapsto F}$ treq $[K]_{P \mapsto G}$. Write $K_1 = [K]_{P \mapsto F}$ and $K_2 = [K]_{P \mapsto G}$. Then $[H]_{P \mapsto F} = \neg[K]_{P \mapsto F} = \neg K_1$ and $[H]_{P \mapsto G} = \neg[K]_{P \mapsto G} = \neg K_2$, and these are truth-equivalent by our first preliminary remark.

Case 3: suppose that $H = (K * L)$ where $* \in \{\wedge, \vee, \to, \leftrightarrow\}$ and where K and L are formulas with $[K]_{P \mapsto F}$ treq $[K]_{P \mapsto G}$ and $[L]_{P \mapsto F}$ treq $[L]_{P \mapsto G}$. Write $K_1 = [K]_{P \mapsto F}$, $K_2 = [K]_{P \mapsto G}$, $L_1 = [L]_{P \mapsto F}$ and $L_2 = [L]_{P \mapsto G}$. Then $[H]_{P \mapsto F} = \big([K]_{P \mapsto F} * [L]_{P \mapsto F}\big) = (K_1 * L_1)$ and $[H]_{P \mapsto G} = \big([K]_{P \mapsto G} * [L]_{P \mapsto G}\big) = (K_2 * L_2)$, and these are truth-equivalent by our second preliminary remark.

4.7 Example: The truth-equivalence $F \wedge (F \to G)$ treq $F \wedge (\neg F \vee G)$, which was found in the first line of the derivation of example 4.4, is obtained from the basic truth-equivalence $F \to G$ treq $\neg F \vee G$ by substitution. Indeed, if $H = F \wedge P$ where P is any propositional variable which does not occur in the formula F, then we have $[H]_{P \mapsto (F \to G)} = F \wedge (P \to G)$ and $[H]_{P \mapsto (\neg F \vee G)} = F \wedge (\neg F \vee G)$.

4.8 Definition: Let F and G be formulas with F treq G. A **derivation** of the truth-equivalence F treq G is a list of formulas $F_0, F_1, F_2, \cdots, F_l$ with $F_0 = F$ and $F_l = G$ such that for each $i \geq 1$ we have F_{i-1} treq F_i and this truth-equivalence is obtained from one of the basic truth-equivalences by substitution. Usually we shall display the formulas $F_0, F_1, \cdots, F_l$ in a column and provide justification on each line.

4.9 Example: Let F, G and H be formulas. Find a derivation for distributivity of $\vee$ over $\wedge$ from the right, that is for the truth-equivalence $(F \wedge G) \vee H$ treq $(F \vee H) \wedge (G \vee H)$.

Solution:

0.	$(F \wedge G) \vee H$	
1.	$H \vee (F \wedge G)$	Commutativity
2.	$(H \vee F) \wedge (H \vee G)$	Distributivity (from the left)
3.	$(F \vee H) \wedge (H \vee G)$	Commutativity
3.	$(F \vee H) \wedge (G \vee H)$	Commutativity

4.10 Example: Derive the truth-equivalence $F \to (G \to H)$ treq $(F \wedge G) \to H$.

Solution:

0.	$F \to (G \to H)$	
1.	$\neg F \vee (G \to H)$	Implication
2.	$\neg F \vee (\neg G \vee H)$	Implication
3.	$(\neg F \vee \neg G) \vee H$	Associativity
4.	$\neg(F \wedge G) \vee H$	DeMorgan's Law
5.	$(F \wedge G) \to H$	Implication

4.11 Example: Derive the truth-equivalence $(F \wedge G) \to H$ treq $(F \to H) \vee (G \to H)$.

Solution:

0.	$(F \wedge G) \to H$	
1.	$\neg(F \wedge G) \vee H$	Implication
2.	$(\neg F \vee \neg G) \vee H$	DeMorgan's Law
3.	$(\neg F \vee \neg G) \vee (H \vee H)$	Idempotence
4.	$((\neg F \vee \neg G) \vee H) \vee H$	Associativity
5.	$(\neg F \vee (\neg G \vee H)) \vee H$	Associativity
6.	$(\neg F \vee (H \vee \neg G)) \vee H$	Commutativity
7.	$((\neg F \vee H) \vee \neg G) \vee H$	Associativity
8.	$(\neg F \vee H) \vee (\neg G \vee H)$	Associativity
9.	$(F \to H) \vee (\neg G \vee H)$	Implication
10.	$(F \to H) \vee (G \to H)$	Implication

4.12 Note: We now describe an algorithm for putting a given formula, which involves only propositional variables from $\{P_1, \cdots, P_n\}$, into disjunctive form or into disjunctive normal form on $\{P_1, \cdots, P_n\}$ using the basic truth-equivalences.

Step 1. Use the If and Only If truth-equivalence to eliminate all occurrences of the $\leftrightarrow$ symbol, and then use the Implication Rules to eliminate all occurrences of the $\to$ symbol.

Step 2. Use deMorgan's Law to move all occurrences of the $\neg$ symbol inside the brackets, and use Double Negation to eliminate extra $\neg$ symbols until all $\neg$ symbols occur in literals.

Step 3. Use Distributivity of $\wedge$ over $\vee$ together with Commutativity and Associativity until we are left with a DF formula in which all the literals in each DF constituent appear in alphabetical order with propositional variables occurring before their negations. At this stage we shall allow ourselves to omit some of the brackets from the formula.

Step 4. Use Idempotence to eliminate repeated copies of literals in each DF constituent, and use Contradiction to eliminate any DF constituents which contain a propositional variable and its negation. (Another optional step here is to use Idempotence and Absorption to eliminate any DF constituent which contains one of the other DF constituents as a subformula).

If, at this stage, the formula has been reduced to a formula of the form $(P_i \wedge \neg P_i)$, then it is truth-equivalent to the empty DNF formula.

Step 5. For each value of i from 1 to n, and for each DF constituent C which does not involve P_i, use Tautology and Distributivity to replace C, first by $C \wedge (P_i \vee \neg P_i)$ and then by $(C \wedge P_i) \vee (C \wedge \neg P_i)$. Also, as in step 4, reorder the literals within each constituent.

Step 6. Use Idempotence to eliminate any repeated copies of constituents, and use Commutativity and Associativity to place the constituents in the proper order.

4.13 Example: Let $F = \neg\big((P \vee Q) \to (Q \leftrightarrow R)\big)$. Put F into disjunctive normal form.

Solution: We provide two solutions. For the first solution, we make a truth-table for F.

P	Q	R	$P \vee Q$	$Q \leftrightarrow R$	$(P \vee Q) \to (Q \leftrightarrow R)$	F
1	1	1	1	1	1	0
1	1	0	1	0	0	1
1	0	1	1	0	0	1
1	0	0	1	1	1	0
0	1	1	1	1	1	0
0	1	0	1	0	0	1
0	0	1	0	0	1	0
0	0	0	0	1	1	0

From the truth-table, we see that F treq $(P \wedge Q \wedge \neg R) \vee (P \wedge \neg Q \wedge R) \vee (\neg P \wedge Q \wedge \neg R)$.

For the second solution, we use truth-equivalences, following the above algorithm.

Step 1: F treq $\neg\big((P \vee Q) \to ((Q \to R) \wedge (R \to Q))\big)$
treq $\neg(\neg(P \vee Q) \vee ((\neg Q \vee R) \wedge (\neg R \vee Q)))\big)$
Step 2: treq $(P \vee Q) \wedge \neg((\neg Q \vee R) \wedge (\neg R \vee Q))$
treq $(P \vee Q) \wedge \big(\neg(\neg Q \vee R) \vee \neg(\neg R \vee Q)\big)$
treq $(P \vee Q) \wedge \big((Q \wedge \neg R) \vee (R \wedge \neg Q)\big)$
Step 3: treq $\big((P \vee Q) \wedge (Q \wedge \neg R)\big) \vee \big((P \vee Q) \wedge (R \wedge \neg Q)\big)$
treq $\big((P \wedge (Q \wedge \neg R)) \vee (Q \wedge (Q \wedge \neg R))\big) \vee \big((P \wedge (R \wedge \neg Q)) \vee (Q \wedge (R \wedge \neg Q))\big)$
treq $(P \wedge Q \wedge \neg R) \vee (Q \wedge Q \wedge \neg R) \vee (P \wedge \neg Q \wedge R) \vee (Q \wedge \neg Q \wedge R)$
Step 4: treq $(P \wedge Q \wedge \neg R) \vee (Q \wedge \neg R) \vee (P \wedge \neg Q \wedge R)$
Step 5: treq $(P \wedge Q \wedge \neg R) \vee (P \wedge Q \wedge \neg R) \vee (\neg P \wedge Q \wedge \neg R) \vee (P \wedge \neg Q \wedge R)$
Step 6: treq $(P \wedge Q \wedge \neg R) \vee (P \wedge \neg Q \wedge R) \vee (\neg P \wedge Q \wedge \neg R)$

4.14 Remark: The list of truth-equivalent formulas in the above solution is *not* a derivation because we often used more than one basic truth-equivalence to get from one formula to the next. For example, to get from the first line to the second line we used the basic truth-equivalence Implication three times. We could extend the list to obtain a derivation.

4.15 Note: We can also use the above algorithm (with minor modifications) to put a given formula into conjunctive form or conjunctive normal form.

4.16 Example: Let $F = \neg\big(((P \leftrightarrow (\neg Q \vee R)) \wedge (S \to T)) \to \neg(Q \wedge T)\big)$. Use truth-equivalences to put F into conjunctive form.

Solution:

$F = \neg\big(((P \leftrightarrow (\neg Q \vee R)) \wedge (S \to T)) \to \neg(Q \wedge T)\big)$

Step 1: treq $\neg\big((((P \to (\neg Q \vee R)) \wedge ((\neg Q \vee R) \to P)) \wedge (S \to T)) \to \neg(Q \wedge T)\big)$

treq $\neg\big(\neg(((\neg P \vee (\neg Q \vee R)) \wedge (\neg(\neg Q \vee R) \vee P)) \wedge (\neg S \vee T)) \vee \neg(Q \wedge T)\big)$

Step 2: treq $(((\neg P \vee (\neg Q \vee R)) \wedge (\neg(\neg Q \vee R) \vee P)) \wedge (\neg S \vee T)) \wedge (Q \wedge T)$

treq $(((\neg P \vee (\neg Q \vee R)) \wedge ((Q \wedge \neg R) \vee P)) \wedge (\neg S \vee T)) \wedge (Q \wedge T)$

Step 3: treq $(\neg P \vee \neg Q \vee R) \wedge (P \vee Q) \wedge (P \vee \neg R) \wedge (\neg S \vee T) \wedge (Q) \wedge (T)$

Step 4: treq $(\neg P \vee \neg Q \vee R) \wedge (P \vee \neg R) \wedge (Q) \wedge (T)$

It was not actually necessary to perform the simplification in step 4.

4.17 Note: We can use the above algorithm to determine whether a given formula is a tautology, to determine whether two given formulas are truth-equivalent, to determine whether a given finite set of formulas is satisfiable, or to determine whether a given argument with a finite premise set is valid.

4.18 Example: Let $F = ((P \leftrightarrow R) \wedge \neg(Q \vee \neg S)) \to ((Q \vee S) \to R)$. Use truth-equivalences to determine whether $\models F$.

Solution: We could put F into disjunctive normal form on $\{P, Q, R, S\}$ and see if we obtain the DNF formula with all 16 DNF constituents. Instead we choose to put F into conjunctive normal form and see if we obtain the empty CNF formula. We have

$F = ((P \leftrightarrow R) \wedge \neg(Q \vee \neg S)) \to ((Q \vee S) \to R)$

Step 1. treq $(((P \to R) \wedge (R \to P)) \wedge \neg(Q \vee \neg S)) \to ((Q \vee S) \to R)$

treq $\neg(((\neg P \vee R) \wedge (\neg R \vee P)) \wedge \neg(Q \vee \neg S)) \vee (\neg(Q \vee S) \vee R)$

Step 2. treq $(\neg((\neg P \vee R) \wedge (\neg R \vee P)) \vee (Q \vee \neg S)) \vee ((\neg Q \wedge \neg S) \vee R)$

treq $((\neg(\neg P \vee R) \vee \neg(\neg R \vee P)) \vee (Q \vee \neg S)) \vee ((\neg Q \wedge \neg S) \vee R)$

treq $(((P \wedge \neg R) \vee (R \wedge \neg P)) \vee (Q \vee \neg S)) \vee ((\neg Q \wedge \neg S) \vee R)$

Step 3. treq $(P \wedge \neg R) \vee (R \wedge \neg P) \vee Q \vee \neg S \vee (\neg Q \wedge \neg S) \vee R$

treq $(P \wedge \neg R) \vee (\neg Q \wedge \neg S) \vee Q \vee R \vee \neg S$ by Absorption

treq $((P \vee \neg Q) \wedge (P \vee \neg S) \wedge (\neg R \vee \neg Q) \wedge (\neg R \vee \neg S)) \vee (Q \vee R \vee \neg S)$

treq $(P \vee Q \vee \neg Q \vee R \vee \neg S) \wedge (P \vee Q \vee R \vee \neg S \vee \neg S)$
$\wedge (Q \vee \neg Q \vee R \vee \neg R \vee \neg S) \wedge (Q \vee R \vee \neg R \vee \neg S \vee \neg S)$

Step 4. treq $(P \vee Q \vee R \vee \neg S)$

Since this is not the empty CNF formula, F is not a tautology.

4.19 Theorem: *(Completeness) Let F and G be formulas with F* treq *G. Then there exists a derivation for the truth-equivalence F* treq *G.*

Proof: We only provide an informal proof. Since F treq G, F and G are truth-equivalent to the same DNF formula, say H. Using the above algorithm, we can make a derivation for the truth-equivalence F treq H and a derivation for the truth-equivalence G treq H. The derivation of G treq H can be reversed to obtain a derivation for H treq G. Then the derivations for F treq H and for H treq G can be combined to obtain a derivation for F treq G.

The reason that this proof is not a rigorous one is that our presentation of the above algorithm was somewhat informal, and we did not provide a proof that the above algorithm always works.

Chapter 5. Satisfiability and the Davis-Putnam Procedure

5.1 Definition: A **clause** is a set of literals. Let $\mathcal{S}$ be a set of clauses. We say that a truth-evaluation α **satisfies** $\mathcal{S}$ when for every clause $C \in \mathcal{S}$ there exists a literal $L \in C$ such that $\alpha(L) = 1$. We say that $\mathcal{S}$ is **satisfiable** if there exists a truth-evaluation which satisfies $\mathcal{S}$. Thus $\mathcal{S}$ is satisfiable means that there exists a truth-evaluation α such that for every clause $C \in \mathcal{S}$ there exists a literal $L \in C$ such that $\alpha(L) = 1$.

5.2 Example: Let $\mathcal{S} = \big\{\{P, Q\}, \{P, \neg Q, R\}, \{\neg P, R\}, \{\neg P, \neg R\}\big\}$. Determine whether $\mathcal{S}$ is satisfiable, and if so describe all the truth-evaluations α which satisfy $\mathcal{S}$.

Solution: Suppose that α is a truth-evaluation which satisfies $\mathcal{S}$. Either $\alpha(P) = 1$ or $\alpha(P) = 0$. Suppose first that $\alpha(P) = 1$. Since α satisfies $\mathcal{S}$, and since $\{\neg P, R\} \in \mathcal{S}$, we must have either $\alpha(\neg P) = 1$ or $\alpha(R) = 1$. Since $\alpha(P) = 1$, we must have $\alpha(R) = 1$. But similarly, since $\{\neg P, \neg R\} \in \mathcal{S}$, we must have $\alpha(\neg R) = 1$, which gives a contradiction. Thus we must have $\alpha(P) = 0$. Now, since α satisfies $\mathcal{S}$ and $\{P, Q\} \in \mathcal{S}$ and $\alpha(P) = 0$, we must have $\alpha(Q) = 1$. Also, since $\{P, \neg Q, R\} \in \mathcal{S}$ and $\alpha(P) = 0$ and $\alpha(Q) = 1$, we must have $\alpha(R) = 1$. Thus we have shown in order for α to satisfy $\mathcal{S}$ we must have $\alpha(P) = 0$ and $\alpha(Q) = \alpha(R) = 1$. Conversely, any such truth-evaluation α satisfies $\mathcal{S}$.

5.3 Note: If $\mathcal{S} = \emptyset$ then according to the above definition, $\mathcal{S}$ is satisfiable, and indeed every truth-evaluation satisfies $\mathcal{S}$. By contrast, if $\mathcal{S} = \big\{\emptyset\big\}$ then $\mathcal{S}$ is not satisfiable.

5.4 Definition: Given a CF constituent $C = (L_1 \vee L_2 \vee \cdots \vee L_k)$, the **clause associated to** C is the clause $D = \{L_1, L_2, \cdots, L_k\}$. Given a CF formula $F = C_1 \wedge C_2 \wedge \cdots \wedge C_l$, where each C_i is a CF constituent, the **clause set associated to** F is the clause set $\mathcal{S} = \big\{D_1, D_2, \cdots, D_l\big\}$ where each D_i is the clause associated to the CF constituent C_i.

5.5 Note: Let F be a CF formula with associated clause set $\mathcal{S}$. Note that for a truth-evaluation α, α satisfies $\mathcal{S}$ if and only if $\alpha(F) = 1$. So to determine whether F is satisfiable, it suffices to determine whether $\mathcal{S}$ is satisfiable.

5.6 Example: Let $F = (P \vee Q) \wedge (P \vee \neg Q \vee R) \wedge (\neg P \vee R) \wedge (\neg P \vee \neg R)$. Then F is a CF formula, and the associated clause set of F is the clause set $\mathcal{S}$ in the above example. If you make a truth-table for F, you will find that the F-column has a single 1 on row 5, which corresponds to the truth-evaluation α on $\{P, Q.R\}$ with $\alpha(P) = 0$ and $\alpha(Q) = \alpha(R) = 1$.

5.7 Example: Let $\mathcal{S}$ be the set of all 8 clauses on $\{P, Q, R\}$, that is

$$\mathcal{S} = \big\{\{P, Q, R\}, \{P, Q, \neg R\}, \{P, \neg Q, R\}, \{P, \neg Q, \neg R\},$$
$$\{\neg P, Q, R\}, \{\neg P, Q, \neg R\}, \{\neg P, \neg Q, R\}, \{\neg P, \neg Q, \neg R\}\big\}.$$

Then $\mathcal{S}$ is the clause set associated to the CNF formula F on $\{P, Q, R\}$ with all 8 constituents. This formula is not satisfiable, so $\mathcal{S}$ is not satisfiable.

5.8 Remark: The reason that we work with clause sets instead of CF formulas is that we no longer have to deal explicitly with the truth-equivalences Commutativity, Associativity and Idempotence: for CF formulas, Commutativity and Associativity allow us to omit some brackets and to rearrange the literals within each CF constituent and to rearrange the constituents, and Idempotence allows us to omit repeated copies of literals in each constituent and to omit repeated constituents; for clause sets, all this is taken care of implicitly in the use of set notation.

5.9 Theorem: *Let $\mathcal{S}$ be a set of clauses, and let P be a propositional variable. Write*

$$\mathcal{T} = \big\{C \in \mathcal{S} \big| P \notin C, \neg P \notin C\big\},$$
$$\mathcal{U} = \big\{D \setminus \{P\} \big| D \in \mathcal{S}, P \in D, \neg P \notin D\big\},$$
$$\mathcal{V} = \big\{E \setminus \{\neg P\} \big| E \in \mathcal{S}, P \notin E, \neg P \in E\big\} \text{ and}$$
$$\mathcal{W} = \big\{(D \setminus \{P\}) \cup (E \setminus \{\neg P\}) \big| D, E \in \mathcal{S}, P \in D, \neg P \notin D, P \notin E, \neg P \in E\big\}.$$

(1) $\mathcal{S}$ is satisfiable $\iff$ $\big(\mathcal{T} \cup \mathcal{U}$ is satisfiable or $\mathcal{T} \cup \mathcal{V}$ is satisfiable$\big)$.

(2) $\mathcal{S}$ is satisfiable $\iff$ $\mathcal{T} \cup \mathcal{W}$ is satisfiable.

Replacing the clause set $\mathcal{S}$ by the two clause sets $\mathcal{T} \cup \mathcal{U}$ and $\mathcal{T} \cup \mathcal{V}$ is called **splitting $\mathcal{S}$ into cases on** *P, and replacing $\mathcal{S}$ by the clause set $\mathcal{T} \cup \mathcal{W}$ is called* **resolving $\mathcal{S}$ on** *P.*

Proof: First let us prove part (1). Suppose that $\mathcal{S}$ is satisfiable. Let α be a truth evaluation which satisfies $\mathcal{S}$ (that is, for every clause $C \in \mathcal{S}$, $\alpha(L) = 1$ for some literal $L \in C$). We claim that if $\alpha(P) = 1$ then α satisfies $\mathcal{T} \cup \mathcal{V}$, and if $\alpha(P) = 0$ then α satisfies $\mathcal{T} \cup \mathcal{U}$. Suppose that $\alpha(P) = 1$. Let C be a clause in $\mathcal{T} \cup \mathcal{V}$. If $C \in \mathcal{T}$ then $C \in \mathcal{S}$ and so $\alpha(L) = 1$ for some literal $L \in C$. If $C \in \mathcal{V}$ then $C = E \setminus \{\neg P\}$ for some $E \in \mathcal{S}$ with $P \notin E$ and $\neg P \in E$, and we have $\alpha(L) = 1$ for some literal $L \in E$, but $L \neq \neg P$ since $\alpha(\neg P) = 0$, and so $L \in C$. Thus α satisfies $\mathcal{T} \cup \mathcal{V}$. Similarly, if $\alpha(P) = 0$ then α satisfies $\mathcal{T} \cup \mathcal{U}$.

Conversely, suppose that either $\mathcal{T} \cup \mathcal{U}$ is satisfiable or $\mathcal{T} \cup \mathcal{V}$ is satisfiable. Let us suppose that $\mathcal{T} \cup \mathcal{U}$ is satisfiable. Let α be a truth-evaluation which satisfies $\mathcal{T} \cup \mathcal{U}$ (that is, for every clause $C \in \mathcal{T} \cup \mathcal{U}$ we have $\alpha(L) = 1$ for some literal $L \in C$). Let β be the truth-evaluation given by $\beta(X) = \alpha(X)$ for all propositional variables X other than P, and $\beta(P) = 0$. We claim that β satisfies $\mathcal{S}$. Let C be a clause in $\mathcal{S}$. We need to show that $\beta(L) = 1$ for some literal $L \in C$. We consider several cases. If $\neg P \in C$, then since $\beta(\neg P) = 1$ we know that $\beta(L) = 1$ for the literal $L = \neg P \in C$. Suppose $\neg P \notin C$. If $P \in C$ then $C \setminus \{P\} \in \mathcal{U}$ so $\alpha(L) = 1$ for some literal $L \in C \setminus \{P\}$, and since $L \neq P$ we have $\beta(L) = \alpha(L) = 1$. If $P \notin C$, then $C \in T$ so $\alpha(L) = 1$ for some $L \in T$, and since $L \neq P$, we have $\beta(L) = \alpha(L) = 1$. This covers all cases, and we see that indeed β satisfies $\mathcal{S}$. Similarly, if α is a truth-evaluation which satisfies $\mathcal{T} \cup \mathcal{V}$, then the truth-evaluation γ given by $\gamma(X) = \alpha(X)$ for $X \neq P$ and $\gamma(P) = 1$ can be shown to satisfy $\mathcal{S}$.

Now let us prove part (2). Suppose that $\mathcal{S}$ is satisfiable, and let α be a truth-evaluation which satisfies $\mathcal{S}$. By our proof of part (1), we know that either α satisfies $\mathcal{T} \cup \mathcal{U}$ or α satisfies $\mathcal{T} \cup \mathcal{V}$. We claim that α also satisfies $\mathcal{T} \cup \mathcal{W}$. Let C be any clause in $\mathcal{T} \cup \mathcal{W}$. If $C \in \mathcal{T}$ then we also have $C \in \mathcal{S}$ and so $\alpha(L) = 1$ for some literal in C. Suppose instead that $C \in \mathcal{W}$, and say $C = (D \setminus \{P\}) \cup (E \setminus \{\neg P\})$, where D and E are clauses in $\mathcal{S}$ and $P \in D$, $\neg P \notin D$, $P \notin E$ and $\neg P \in E$. Note that $(D \setminus \{P\}) \in \mathcal{U}$ and $(E \setminus \{\neg P\}) \in \mathcal{V}$. If α satisfies $\mathcal{T} \cup \mathcal{U}$ then $\alpha(L) = 1$ for some $L \in (D \setminus \{P\})$, and if α satisfies $\mathcal{T} \cup \mathcal{V}$ then $\alpha(L) = 1$ for some $L \in (E \setminus \{\neg P\})$. In either case, we have $\alpha(L) = 1$ for some $L \in C$.

Conversely, suppose that $\mathcal{T} \cup \mathcal{W}$ is satisfiable. Choose a truth-evaluation α which satisfies $\mathcal{T} \cup \mathcal{W}$. This means that α satisfies $\mathcal{T}$ and α satisfies $\mathcal{W}$. We claim that either α satisfies $\mathcal{U}$ and hence also $\mathcal{T} \cup \mathcal{U}$, or α satisfies $\mathcal{V}$ and hence also $\mathcal{T} \cup \mathcal{V}$. Note that in either case, $\mathcal{S}$ is satisfiable by part (1). Suppose that α does not satisfy $\mathcal{U}$. Choose a clause $C \in \mathcal{U}$ such that $\alpha(L) = 0$ for every literal $L \in C$, and say $C = (D \setminus \{P\})$ where $D \in \mathcal{S}$, $P \in D$ and $\neg P \notin D$. Then for every clause $E \in \mathcal{S}$ with $P \notin E$ and $\neg P \in E$ we have $(D \setminus \{P\}) \cup (E \setminus \{\neg P\}) \in \mathcal{W}$, and α satisfies $\mathcal{W}$ so we have $\alpha(L) = 1$ for some $L \in (D \setminus \{P\}) \cup (E \setminus \{\neg P\})$, but $\alpha(L) = 0$ for every $L \in (D \setminus \{P\})$ so we must have $\alpha(L) = 1$ for some $L \in (E \setminus \{\neg P\})$. This shows that α satisfies $\mathcal{V}$.

5.10 Example: Let $\mathcal{S} = \big\{\{P, Q, \neg R\}, \{P, \neg Q\}, \{P, R\}, \{\neg P, Q\}, \{\neg P, \neg Q, \neg R\}, \{\neg Q, R\}\big\}$. Then the clause sets $\mathcal{T}$, $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{W}$ of the above theorem are

$$\mathcal{T} = \big\{\{\neg Q, R\}\big\}$$
$$\mathcal{U} = \big\{\{Q, \neg R\}, \{\neg Q\}, \{R\}\big\}$$
$$\mathcal{V} = \big\{\{Q\}, \{\neg Q, \neg R\}\big\}$$
$$\mathcal{W} = \big\{\{Q, \neg R\}, \{Q, \neg Q, \neg R\}, \{Q, \neg Q\}, \{\neg Q, \neg R\}, \{Q, R\}, \{\neg Q, R, \neg R\}\big\}$$

5.11 Definition: We describe an algorithm, called the **Davis-Putnam Procedure**, for determining whether or not a given clause set $\mathcal{S}$ is satisfiable.

Step 1 (Clean-Up). In each clause, remove any repeated copies of a propositional variable.
Remove any clauses which contain a propositional variable and its negation.
Put the literals in alphabetical order within each clause.
Remove any clause which contains one of the other clauses as a subset.

Step 2 (Test). If we are left with the empty clause set $\emptyset$ then $\mathcal{S}$ is satisfiable.
If the remaining clause set contains $\big\{\emptyset\big\}$ then $\mathcal{S}$ is not satisfiable.
If the remaining clause set contains all 2^n clauses on some set of n propositional variables (for example, if the clause set contains all 4 clauses $\{P, Q\}$, $\{P, \neg Q\}$, $\{\neg P, Q\}$, $\{\neg P, \neg Q\}$) then $\mathcal{S}$ is not satisfiable.
If every clause contains a propositional variable or if every clause contains a negated propositional variable, then $\mathcal{S}$ is satisfiable.

Step 3 (Resolution). If there is any clause with only one element, then let P be the first propositional variable symbol such that there is a clause of the form $\{P\}$ or the form $\{\neg P\}$.
Otherwise, if there is any propositional variable Q such that exactly one of the two literals Q and $\neg Q$ occurs in the union of all the clauses, then let P be the first such variable Q.
Otherwise let P be the first propositional variable symbol which occurs in any clause.
Resolve on P and then return to step 1.

5.12 Example: Let $\mathcal{S}$ be the clause set in the above example. Use that Davis-Putnam Procedure to determine whether $\mathcal{S}$ is satisfiable.

Solution: We start with

$$\mathcal{S} = \big\{\{P, Q, \neg R\}, \{P, \neg Q\}, \{P, R\}, \{\neg P, Q\}, \{\neg P, \neg Q, \neg R\}, \{\neg Q, R\}\big\}.$$

In step 1, none of the clean-up operations have any effect, and in step 2, the test for satisfiability is inconclusive. Moving to step 3, we resolve on P to obtain the clause set

$$\big\{\{Q, \neg R\}, \{Q, \neg Q, \neg R\}, \{Q, \neg Q\}, \{\neg Q, \neg R\}, \{Q, R\}, \{\neg Q, R, \neg R\}, \{\neg Q, R\}\big\}$$

The clean-up step leaves us with the clause set

$$\big\{\{Q, \neg R\}, \{\neg Q, \neg R\}, \{Q, R\}, \{\neg Q, R\}\big\}$$

Since this clause set contains all 4 clauses on $\{Q, R\}$, it is not satisfiable, and hence neither was the original clause set $\mathcal{S}$.

5.13 Example: Let $\mathcal{S}$ be the clause set which consists of the following clauses

$$\{P, Q, \neg R\}, \{P, \neg Q, S\}, \{P, \neg S, T\}, \{P, R, U\}, \{\neg P, Q, S, \neg U\}, \{\neg P, \neg Q\}, \{\neg P, R, T\},$$
$$\{Q, S\}, \{\neg Q, T, U\}, \{R, \neg S, T, U\}, \{R, \neg S, \neg U\}, \{\neg R, \neg S, U\}, \{\neg R, T, \neg U\}, \{\neg R, \neg U\}$$

Use the Davis-Putnam Procedure to determine whether $\mathcal{S}$ is satisfiable.

Solution: First we perform the clean-up step, and two of the clauses can be eliminated because they each contain one of the other clauses as a subset.

$$\{P,Q,\neg R\},\{P,\neg Q,S\},\{P,\neg S,T\},\{P,R,U\},\{\cancel{\neg P},\cancel{Q},\cancel{S},\cancel{\neg U}\},\{\neg P,\neg Q\},\{\neg P,R,T\},$$
$$\{Q,S\},\{\neg Q,T,U\},\{R,\neg S,T,U\},\{R,\neg S,\neg U\},\{\neg R,\neg S,U\},\{\cancel{\neg R},\cancel{T},\cancel{\neg U}\},\{\neg R,\neg U\}$$

The test for satisfiability is inconclusive at this stage, so we move on to the resolution step. Notice that T occurs but $\neg T$ does not occur in any clause, so we resolve on T. When we resolve on T, all the clauses containing T disappear.

$$\{P,Q,\neg R\},\{P,\neg Q,S\},\{P,R,U\},\{\neg P,\neg Q\},\{Q,S\},\{R,\neg S,\neg U\},\{\neg R,\neg S,U\},\{\neg R,\neg U\}$$

None of the clean-up operations have any effect, and the test for satisfiability is still inconclusive, so we resolve on the first propositional variable, namely P, then clean up.

$$\{\cancel{Q},\cancel{\neg Q},\cancel{\neg R}\},\{\neg Q,S\},\{\neg Q,R,U\},\{Q,S\},\{R,\neg S,\neg U\},\{\neg R,\neg S,U\},\{\neg R,\neg U\}$$

The test step is inconclusive so we resolve on the next propositional variable symbol, namely Q, and we clean up.

$$\{S\},\{\cancel{R},\cancel{S},\cancel{U}\},\{R,\neg S,\neg U\},\{\neg R,\neg S,U\},\{\neg R,\neg U\}$$

Since there is a singleton clause $\{S\}$, we resolve on the propositional variable S. When we do so, the singleton clause $\{S\}$ disappears, and each copy of the literal $\neg S$ disappears from all the other clauses.

$$\{R,\neg U\},\{\neg R,U\},\{\neg R,\neg U\}$$

Finally, in the testing step, since each of the remaining clauses contains a negated propositional variable, this set of clauses is satisfiable (by the truth-evaluation α on $\{R,U\}$ given by $\alpha(R)=\alpha(U)=0$). Thus the original clause set $\mathcal{S}$ is satisfiable.

5.14 Note: If a clause set $\mathcal{S}$ is satisfiable, then not only can we use the Davis-Putnam Procedure to verify that $\mathcal{S}$ is satisfiable, but we can also find a truth-evaluation which satisfies $\mathcal{S}$ by retracing the steps of the procedure.

5.15 Example: Let $\mathcal{S}$ be the clause set of the previous example, in which we used the Davis-Putnam procedure to show that $\mathcal{S}$ is satisfiable. Retrace the steps to find a truth-evaluation α on $\{P,Q,R,S,T,U\}$ which satisfies $\mathcal{S}$.

Solution: Let α be a truth-evaluation which satisfies $\mathcal{S}$. Then α satisfies the clause sets which were obtained at each step in the Davis-Putnam Procedure. To satisfy the 5^{th} and final clause set, we must have $\alpha(R)=\alpha(U)=0$. To make a literal true in the clause $\{S\}$, which is in the 4^{th} clause set, we must also have $\alpha(S)=1$. To make a literal true in the clause $\{\neg Q,R,U\}$, which is in the 3^{rd} clause set, we also need $\alpha(Q)=0$. To make one of the literals true in the clause $\{P,R,U\}$, which is in the 2^{nd} clause set, we need $\alpha(P)=1$. Finally, to make a literal true in $\{\neg P,R,T\}$ we must have $\alpha(T)=1$. Thus the only truth-evaluation α on $\{P,Q,R,S,T,U\}$ which satisfies $\mathcal{S}$ is given by $\alpha(P)=\alpha(S)=\alpha(T)=1$ and $\alpha(Q)=\alpha(R)=\alpha(U)=0$.

5.16 Example: Determine whether

$$\big\{(P\wedge\neg R)\to T,(P\wedge\neg T)\to S,(P\wedge R)\to Q,(T\wedge\neg R)\to\neg P,(\neg P\wedge\neg R)\to Q,$$
$$(R\wedge\neg T)\to P,(Q\wedge T)\to P,\neg P\to S,(Q\wedge S)\to T,(S\wedge T)\to Q\big\}\models(P\wedge Q\wedge R\wedge T)\,.$$

Solution: Write the 10 premises as $F_1, F_2, \cdots, F_{10}$ and let $G = (P \wedge Q \wedge R \wedge T)$ be the conclusion. We have $\{F_1, \cdots, F_{10}\} \models G$ if and only if $(F_1 \wedge \cdots \wedge F_{10} \wedge \neg G)$ is not satisfiable. If we put $(F_1 \wedge \cdots F_{10} \wedge \neg G)$ into conjunctive form then the associated clause set is

$$\mathcal{S} = \Big\{ \{\neg P, R, T\}, \{\neg P, S, T\}, \{\neg P, Q, \neg R\}, \{\neg P, R, \neg T\}, \{P, Q, R\}, \{P, \neg R, T\},$$
$$\{P, \neg Q, \neg T\}, \{P, S\}, \{\neg Q, \neg S, T\}, \{Q, \neg S, \neg T\}, \{\neg P, \neg Q, \neg R, \neg T\} \Big\}.$$

We split into cases on the variable P, then apply the Davis-Putnam Procedure to each case. Write $\mathcal{T}$, $\mathcal{U}$ and $\mathcal{V}$ as in theorem 5.9.
Case 1:

$$\mathcal{T} \cup \mathcal{U} = \Big\{ \{\neg Q, \neg S, T\}, \{Q, \neg S, \neg T\}, \{Q, R\}, \{\neg R, T\}, \{\neg Q, \neg T\}, \{S\} \Big\}$$
$$\text{on } S: \ \Big\{ \{\neg Q, T\}, \{Q, \neg T\}, \{Q, R\}, \{\neg R, T\}, \{\neg Q, \neg T\} \Big\}$$
$$\text{on } R: \ \Big\{ \{\neg Q, T\}, \{Q, \neg T\}, \{Q, T\}, \{\neg Q, \neg T\} \Big\}$$

All 4 possible clauses on Q, T occur, so $\mathcal{T} \cup \mathcal{U}$ is not satisfiable.
Case 2:

$$\mathcal{T} \cup \mathcal{V} = \Big\{ \{\neg Q, \neg S, T\}, \{Q, \neg S, \neg T\}, \{R, T\}, \{S, T\}, \{Q, \neg R\}, \{R, \neg T\}, \{\neg Q, \neg R, \neg T\} \Big\}$$
$$\text{on } Q: \ \Big\{ \{\not\neg S, \not T, \not\neg T\}, \{\neg R, \neg S, T\}, \{\not\neg R, \not\neg S, \not\neg T\}, \{R, T\}, \{S, T\}, \{\neg R, \neg T\}, \{R, \neg T\} \Big\}$$
$$\text{on } R: \ \Big\{ \{\neg S, T\}, \{\not\neg S, \not T, \not\neg T\}, \{\not T, \not\neg T\}, \{S, T\}, \{\neg T\} \Big\}$$
$$\text{on } S: \ \Big\{ \{T\}, \{\neg T\} \Big\}$$

Thus $\mathcal{T} \cup \mathcal{V}$ is not satisfiable, so $\mathcal{S}$ is not satisfiable and hence the argument is valid.

Chapter 6. Derivation of Valid Arguments

6.1 Theorem: *(Basic Validity Rules) Let F, G and H be formulas, and let $\mathcal{S}$ and $\mathcal{T}$ be sets of formulas. Then we have the following 36 rules, called the* **Basic Validity Rules**.

(Premise)	1. *If* $F \in \mathcal{S}$ *then* $\mathcal{S} \models F$
(Adding Premises)	2. *If* $\mathcal{S} \models F$ *and* $\mathcal{S} \subseteq \mathcal{T}$ *then* $\mathcal{T} \models F$
(The Chain Rule)	3. *If* $\mathcal{S} \models F$ *and* $\mathcal{S} \cup \{F\} \models G$ *then* $\mathcal{S} \models G$
(Proof by Cases)	4. *If* $\mathcal{S} \cup \{F\} \models G$ *and* $\mathcal{S} \cup \{\neg F\} \models G$ *then* $\mathcal{S} \models G$
(Contradiction)	5. *If* $\mathcal{S} \models F$ *and* $\mathcal{S} \models \neg F$ *then* $\mathcal{S} \models G$
	6. *If* $\mathcal{S} \cup \{\neg F\} \models G$ *and* $\mathcal{S} \cup \{\neg F\} \models \neg G$ *then* $\mathcal{S} \models F$
	7. *If* $\mathcal{S} \cup \{F\} \models G$ *and* $\mathcal{S} \cup \{F\} \models \neg G$ *then* $\mathcal{S} \models \neg F$
(Conjunction)	8. *If* $\mathcal{S} \models F$ *and* $\mathcal{S} \models G$ *then* $\mathcal{S} \models F \wedge G$
	9. *If* $\mathcal{S} \models F \wedge G$ *then* $\mathcal{S} \models F$
	10. *If* $\mathcal{S} \models F \wedge G$ *then* $\mathcal{S} \models G$
	11. $\mathcal{S} \cup \{F \wedge G\} \models H \iff \mathcal{S} \cup \{F, G\} \models H$
(Disjunction)	12. *If* $\mathcal{S} \models F$ *then* $\mathcal{S} \models F \vee G$
	13. *If* $\mathcal{S} \models G$ *then* $\mathcal{S} \models F \vee G$
	14. *If* $\mathcal{S} \cup \{\neg F\} \models G$ *then* $\mathcal{S} \models F \vee G$
	15. *If* $\mathcal{S} \cup \{\neg G\} \models F$ *then* $\mathcal{S} \models F \vee G$
	16. *If* $\mathcal{S} \models F \vee G$ *and* $\mathcal{S} \models \neg F$ *then* $\mathcal{S} \models G$
	17. *If* $\mathcal{S} \models F \vee G$ *and* $\mathcal{S} \models \neg G$ *then* $\mathcal{S} \models F$
	18. $\mathcal{S} \cup \{F \vee G\} \models H \iff \big(\mathcal{S} \cup \{F\} \models H$ *and* $\mathcal{S} \cup \{G\} \models H\big)$
(Implication)	19. *If* $\mathcal{S} \models \neg F$ *then* $\mathcal{S} \models F \to G$
	20. *If* $\mathcal{S} \models G$ *then* $\mathcal{S} \models F \to G$
	21. *If* $\mathcal{S} \cup \{F\} \models G$ *then* $\mathcal{S} \models F \to G$ *(also called Deduction)*
	22. *If* $\mathcal{S} \cup \{\neg G\} \models \neg F$ *then* $\mathcal{S} \models F \to G$
	23. *If* $\mathcal{S} \models F \to G$ *and* $\mathcal{S} \models F$ *then* $\mathcal{S} \models G$ *(also called Modus Ponens)*
	24. *If* $\mathcal{S} \models F \to G$ *and* $\mathcal{S} \models \neg G$ *then* $\mathcal{S} \models \neg F$
	25. $\mathcal{S} \cup \{F \to G\} \models H \iff \big(\mathcal{S} \cup \{\neg F\} \models H$ *and* $\mathcal{S} \cup \{G\} \models H\big)$
(If and Only If)	26. *If* $\mathcal{S} \models F$ *and* $\mathcal{S} \models G$ *then* $\mathcal{S} \models F \leftrightarrow G$
	27. *If* $\mathcal{S} \models \neg F$ *and* $\mathcal{S} \models \neg G$ *then* $\mathcal{S} \models F \leftrightarrow G$
	28. *If* $\mathcal{S} \models F \to G$ *and* $\mathcal{S} \models G \to F$ *then* $\mathcal{S} \models F \leftrightarrow G$
	29. *If* $\mathcal{S} \models F \leftrightarrow G$ *and* $\mathcal{S} \models F$ *then* $\mathcal{S} \models G$
	30. *If* $\mathcal{S} \models F \leftrightarrow G$ *and* $\mathcal{S} \models G$ *then* $\mathcal{S} \models F$
	31. *If* $\mathcal{S} \models F \leftrightarrow G$ *and* $\mathcal{S} \models \neg F$ *then* $\mathcal{S} \models \neg G$
	32. *If* $\mathcal{S} \models F \leftrightarrow G$ *and* $\mathcal{S} \models \neg G$ *then* $\mathcal{S} \models \neg F$
	33. $\mathcal{S} \cup \{F \leftrightarrow G\} \models H \iff \big(\mathcal{S} \cup \{F, G\} \models H$ *and* $\mathcal{S} \cup \{\neg F, \neg G\} \models H\big)$
(Tautology)	34. *If* F *is a tautology then* $\mathcal{S} \models F$
(Truth-Equivalence)	35. *If* F treq G *then* $\big(\mathcal{S} \models F \iff \mathcal{S} \models G\big)$
	36. *If* F treq G *then* $\big(\mathcal{S} \cup \{F\} \models H \iff \mathcal{S} \cup \{G\} \models H\big)$

Proof: We provide two sample proofs. First we prove the second of the contradiction rules. Suppose that $\mathcal{S} \cup \{\neg F\} \models G$ and that $\mathcal{S} \cup \{\neg F\} \models \neg G$. $\big[$We must show that $\mathcal{S} \models F\big]$. Let α be a truth-evaluation. Suppose that $\alpha(H) = 1$ for every $H \in \mathcal{S}$. $\big[$We must show that

$\alpha(F) = 1\big]$. Suppose, for a contradiction, that $\alpha(F) = 0$ so that $\alpha(\neg F) = 1$. Then we have $\alpha(H) = 1$ for every $H \in \mathcal{S} \cup \{\neg F\}$. Since $\mathcal{S} \cup \{\neg F\} \models G$ we have $\alpha(G) = 1$, and since $\mathcal{S} \cup \{\neg F\} \models \neg G$ we also have $\alpha(\neg G) = 1$. But this is impossible, so we have a contradiction and so $\alpha(F) = 1$, as required.

Next we prove the first Truth-Equivalence Rule. Suppose that F treq G. Suppose that $\mathcal{S} \models F$. $\big[$We must show that $\mathcal{S} \models G\big]$. Let α be a truth-evaluation such that $\alpha(H) = 1$ for every $H \in \mathcal{S}$. $\big[$We must show that $\alpha(G) = 1\big]$. Since $\mathcal{S} \models F$ we have $\alpha(F) = 1$. Since F treq G we have $\alpha(G) = \alpha(F) = 1$, as required.

6.2 Definition: Let $\mathcal{S}$ be a set of formulas and let F be a formula. Suppose that $\mathcal{S} \models F$. A **derivation** of the valid argument $\mathcal{S} \models F$ is a finite list of valid arguments $\mathcal{S}_1 \models F_1, \mathcal{S}_2 \models F_2, \cdots, \mathcal{S}_l \models F_l$ with each $\mathcal{S}_k$ finite, $\mathcal{S}_l \subseteq \mathcal{S}$ and $F_l = F$, such that each valid argument $\mathcal{S}_k \models F_k$ is obtained from zero or more previous valid arguments $\mathcal{S}_i \models F_i$ with $i < k$ using one of the Basic Validity Rules; we shall only use the Truth-Equivalence Rule in the case that the truth-equivalence F treq G is obtained by substitution from one of the 24 basic truth-equivalences, and we shall only use the Tautology Rule in the case that the tautology is of the form $F \to F$.

6.3 Example: Let F, G and H be formulas. Make a derivation for the valid argument $\big\{F \to (G \wedge H), (F \wedge G) \vee H\big\} \models H$.

Solution:

1.	$F \to (G \wedge H), (F \wedge G) \vee H, \neg H \models \neg H$	Premise
2.	$F \to (G \wedge H), (F \wedge G) \vee H, \neg H \models (F \wedge G) \vee H$	Premise
3.	$F \to (G \wedge H), (F \wedge G) \vee H, \neg H \models F \wedge G$	Disjunction on 1, 2
4.	$F \to (G \wedge H), (F \wedge G) \vee H, \neg H \models F$	Conjunction on 3
5.	$F \to (G \wedge H), (F \wedge G) \vee H, \neg H \models F \to (G \wedge H)$	Premise
6.	$F \to (G \wedge H), (F \wedge G) \vee H, \neg H \models G \wedge H$	MP on 4, 5
7.	$F \to (G \wedge H), (F \wedge G) \vee H, \neg H \models H$	Conjunction on 6
8.	$F \to (G \wedge H), (F \wedge G) \vee H \models H$	Contradiction on 1,7

6.4 Example: Let $F = (P \vee \neg Q) \to R$, $G = P \leftrightarrow (Q \wedge \neg R)$ and $H = \neg(R \to P)$. Derive the valid argument $\{F, G\} \models H$.

Solution:

1.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models (P \vee \neg Q) \to R$	Premise
2.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models \neg R$	Premise
3.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models \neg(P \vee \neg Q)$	Implication on 1,2
4.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models \neg P \wedge \neg\neg Q$	Truth-Equiv on 3
5.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models \neg P \wedge Q$	Truth-Equiv on 4
6.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models \neg P$	Conjunction on 5
7.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models Q$	Conjunction on 4
8.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models \neg R$	Premise
9.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models Q \wedge \neg R$	Conjunction on 7,8
10.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models P \leftrightarrow (Q \wedge \neg R)$	Premise
11.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), \neg R \models P$	If and Only If on 10,9
12.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R) \models R$	Contradiction on 11,6
13.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), P \models P$	Premise
14.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), P \models P \leftrightarrow (Q \wedge \neg R)$	Premise
15.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), P \models Q \wedge \neg R$	If and Only If on 14,13
16.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), P \models \neg R$	Conjunction on 15
17.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R), P \models R$	Add Premises on 12

18.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R) \models \neg P$	Contradiction on 17,16
19.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R) \models R \wedge \neg P$	Conjunction on 12,18
20.	$(P \vee \neg Q) \to R, P \leftrightarrow (Q \wedge \neg R) \models \neg(R \to P)$	Truth-Equiv on 19

6.5 Definition: We now define an apparently much weaker formal proof system for propositional logic. First we single out three tautologies and three truth-equivalences, and we replace each truth-equivalence by two tautologies to obtain nine tautologies listed below, which we call the nine **axioms** of the proof system. Also we single out one valid argument, namely **modus ponens**, which we call the **rule of inference** of the proof system:

(Weak Implication)	A1	$F \to (G \to F)$
(Contrapositive)	A2	$(\neg F \to \neg G) \to (G \to F)$
(Deduction)	A3	$\big(F \to (G \to H)\big) \to \big((F \to G) \to (F \to H)\big)$
(And/Implies)	A4	$(F \wedge G) \to \neg(F \to \neg G)$
	A5	$\neg(F \to \neg G) \to (F \wedge G)$
(Or/And)	A6	$\neg(F \vee G) \to (\neg F \wedge \neg G)$
	A7	$(\neg F \wedge \neg G) \to \neg(F \vee G)$
(Iff/Implies)	A8	$(F \leftrightarrow G) \to \big((F \to G) \wedge (G \to F)\big)$
	A9	$\big((F \to G) \wedge (G \to F)\big) \to (F \leftrightarrow G)$
(Modus Ponens)	MP	$\{F, F \to G\} \models G$

In the above axioms and rule of inference, F, G and H can be any propositional formulas. Now let $\mathcal{S}$ be a set of formulas and let F be a formula. A **derivation of F from $\mathcal{S}$** is a list of formulas $F_1, F_2, \cdots, F_n$ with $F_n = F$ such that for each k, either

(Premise)	$F_k \in \mathcal{S}$, in which case F is called a **premise**,
(Axiom)	F_k is one of the axioms $A1 - A9$, or
(Modus Ponens)	$F_j = F_i \to F_k$ for some $i, j < k$ so that $\{F_i, F_j\} \models F_k$ by MP.

If such a derivation exists, we say that F **is derivable from $\mathcal{S}$** and we write

$$\mathcal{S} \vdash F.$$

6.6 Example: Let F and G be formulas. Make a derivation to show that $\{F, \neg F\} \vdash G$.

Solution:

1.	$\neg F$	Premise
2.	$\neg F \to (\neg G \to \neg F)$	A1
3.	$\neg G \to \neg F$	MP on 1,2
4.	$(\neg G \to \neg F) \to (F \to G)$	A2
5.	$F \to G$	MP on 3,4
6.	F	Premise
7.	G	MP on 6,5

6.7 Example: Make a derivation to show that $\{F \to G, G \to H\} \vdash F \to H$, where F, G and H are formulas.

Solution:

1.	$F \to G$	Premise
2.	$G \to H$	Premise
3.	$(G \to H) \to (F \to (G \to H))$	A1
4.	$F \to (G \to H)$	MP on 2,3

5.	$(F \to (G \to H)) \to ((F \to G) \to (F \to H))$	A3
6.	$(F \to G) \to (F \to H)$	MP on 4,5
7.	$F \to H$	MP on 1,6

6.8 Remark: If the list $F_1, F_2, \cdots, F_l$ is a derivation of F from $\mathcal{S}$, then note that $\mathcal{S} \models F_i$ for all i, and in particular $\mathcal{S} \models F_l$, that is $\mathcal{S} \models F$. Thus

$$\text{If } \mathcal{S} \vdash F \text{ then } \mathcal{S} \models F\,.$$

The remainder of this chapter is devoted to showing that the converse is also true.

6.9 Note: Let $\mathcal{S}$ be a set of formulas and let F, G and H be formulas. Then the following Derivation Rules all follow more or less immediately from the definition of a derivation.

(Premise)	If $F \in \mathcal{S}$ then $\mathcal{S} \vdash F$
(Adding Premises)	If $\big(\mathcal{T} \subseteq \mathcal{S}$ and $\mathcal{T} \vdash F\big)$ then $\mathcal{S} \vdash F$
(Axiom, A1-A9)	If F is one of the nine axioms then $\mathcal{S} \vdash F$
(Modus Ponens, MP)	If $\big(\mathcal{S} \vdash F$ and $\mathcal{S} \vdash F \to G\big)$ then $\mathcal{S} \vdash G$
(The Chain Rule)	If $\big(\mathcal{S} \vdash F$ and $\mathcal{S} \cup \{F\} \vdash G\big)$ then $\mathcal{S} \vdash G$
(Compactness)	If $\mathcal{S} \vdash F$ then there is a finite subset $\mathcal{T} \subseteq \mathcal{S}$ such that $\mathcal{T} \vdash F$
(And/Implies)	$\mathcal{S} \vdash (F \wedge G) \Longleftrightarrow \mathcal{S} \vdash \neg(F \to \neg G)$
(Or/And)	$\mathcal{S} \vdash \neg(F \vee G) \Longleftrightarrow \mathcal{S} \vdash \neg F \wedge \neg G$
(Iff/Implies)	$\mathcal{S} \vdash F \leftrightarrow G \Longleftrightarrow \mathcal{S} \vdash (F \to G) \wedge (G \to F)$

For example, if $\mathcal{S} \vdash F \wedge G$ then we have

1.	$\mathcal{S} \vdash F \wedge G$	Assumption
2.	$\mathcal{S} \vdash (F \wedge G) \to \neg(F \to \neg G)$	A4
3.	$\mathcal{S} \vdash \neg(F \to \neg G)$	MP on 1,2

6.10 Theorem: *(Derivation Rules) Let $\mathcal{S}$ be a set of formulas and let F, G and H be formulas. Then*

(Tautology)	$\mathcal{S} \vdash F \to F$
(Deduction)	$\mathcal{S} \vdash F \to G \Longleftrightarrow \mathcal{S} \cup \{F\} \vdash G$
(Contrapositive)	$\mathcal{S} \vdash \neg F \to \neg G \Longleftrightarrow \mathcal{S} \vdash G \to F$
(Implication)	*If* $\mathcal{S} \vdash F$ *then* $\mathcal{S} \vdash G \to F$
	If $\mathcal{S} \vdash \neg F$ *then* $\mathcal{S} \vdash F \to G$
(Double Negation)	$\mathcal{S} \vdash \neg\neg F \Longleftrightarrow \mathcal{S} \vdash F$
(Conjunction)	$\mathcal{S} \vdash F \wedge G \Longleftrightarrow \big(\mathcal{S} \vdash F$ *and* $\mathcal{S} \vdash G\big)$
(Contradiction)	*If* $\mathcal{S} \vdash F \wedge \neg F$ *then* $\mathcal{S} \vdash G$
	If $\mathcal{S} \cup \{\neg F\} \vdash G \wedge \neg G$ *then* $\mathcal{S} \vdash F$
(Implies/And)	$\mathcal{S} \vdash \neg(F \to G) \Longleftrightarrow \mathcal{S} \vdash F \wedge \neg G$

Proof: To prove the Tautology Rule, here is a derivation of $F \to F$ from any premise set $\mathcal{S}$.

1.	$F \to \big((F \to F) \to F\big)$	*A*1
2.	$\big(F \to ((F \to F) \to F)\big) \to \big((F \to (F \to F)) \to (F \to F)\big)$	*A*3
3.	$(F \to (F \to F)) \to (F \to F)$	MP on 1,2
4.	$F \to (F \to F)$	*A*1
5,	$F \to F$	MP on 4,3

Next we prove the Deduction Rule. In one direction, the proof is easy; suppose that $\mathcal{S} \vdash F \to G$. Then we have

1.	$\mathcal{S} \cup \{F\} \vdash F \to G$	Adding Premises
2.	$\mathcal{S} \cup \{F\} \vdash F$	Premise
3.	$\mathcal{S} \cup \{F\} \vdash G$	MP on 2,1

Suppose, conversely, that $\mathcal{S} \cup \{F\} \vdash G$. Let $F_1, F_2, \cdots, F_n$ be a derivation of G from $\mathcal{S} \cup \{F\}$. Note that F_1 is either a premise or an axiom. If F_1 is an axiom then we have

1.	$\mathcal{S} \vdash F_1$	Axiom
2.	$\mathcal{S} \vdash F_1 \to (F \to F_1)$	A1
3.	$\mathcal{S} \vdash F \to F_1$	MP on 1,2

If F_1 is a premise then either $F_1 \in \mathcal{S}$ or $F_1 = F$. If $F_1 \in \mathcal{S}$ then we have

1.	$\mathcal{S} \vdash F_1$	Premise
2.	$\mathcal{S} \vdash F_1 \to (F \to F_1)$	A1
3.	$\mathcal{S} \vdash F \to F_1$	MP on 1,2

and if $F = F_1$ then we have $\mathcal{S} \vdash F \to F_1$ by the Tautology Rule. In all cases we have $\mathcal{S} \vdash F \to F_1$. Now fix k and suppose, inductively, that $\mathcal{S} \vdash F \to F_i$ for all $i < k$. Note that either F_k is a premise or an axiom, or for some $i, j < k$ we have $F_j = F_i \to F_k$. If F_k is a premise or an axiom then as above we have $\mathcal{S} \vdash F \to F_k$. If $F_j = F_i \to F_k$ with $i, j < k$ then we have

1.	$\mathcal{S} \vdash F \to F_i$	Induction Hypothesis
2.	$\mathcal{S} \vdash F \to (F_i \to F_k)$	Induction Hypothesis
3.	$\mathcal{S} \vdash \big(F \to (F_i \to F_k)\big) \to \big((F \to F_i) \to (F \to F_k)\big)$	A3
4.	$\mathcal{S} \vdash (F \to F_i) \to (F \to F_k)$	MP on 2,3
5.	$\mathcal{S} \vdash F \to F_k$	MP on 1,4

In each case we have $\mathcal{S} \vdash F \to F_k$. By induction we have $\mathcal{S} \vdash F \to F_k$ for all k, and in particular, since $G = F_n$ we have $\mathcal{S} \vdash F \to G$.

The Contapositive Rule is easy to prove in one direction. Suppose that $\mathcal{S} \vdash \neg F \to \neg G$. Then we have

1.	$\mathcal{S} \vdash \neg F \to \neg G$	Assumption
2.	$\mathcal{S} \vdash (\neg F \to \neg G) \to (G \to F)$	A2
3.	$\mathcal{S} \vdash G \to F$	MP on 1,2

The other direction of the Contapositive Rule is more difficult, and we shall prove it after we have proven the Double Negation Rule.

To prove the first Implication Rule, suppose that $\mathcal{S} \vdash F$. Then

1.	$\mathcal{S} \vdash F$	Assumption
2.	$\mathcal{S} \vdash F \to (G \to F)$	A1
3.	$\mathcal{S} \vdash G \to F$	MP on 1,2

Now we use the first Implication Rule prove the second. Suppose that $\mathcal{S} \vdash \neg F$. Then

1.	$\mathcal{S} \vdash \neg F$	Assumption
2.	$\mathcal{S} \vdash \neg G \to \neg F$	Implication on 1
3.	$\mathcal{S} \vdash F \to G$	Contrapositive on 2

To prove the Double Negation Rule, suppose first that $\mathcal{S} \vdash \neg\neg F$. Then

1.	$\mathcal{S} \vdash \neg\neg F$	Assumption
2.	$\mathcal{S} \vdash \neg F \to \neg\neg\neg F$	Implication on 1
3.	$\mathcal{S} \vdash \neg\neg F \to F$	Contapositive on 2
4.	$\mathcal{S} \vdash F$	MP on 1,3

Conversely, suppose that $\mathcal{S} \vdash F$. Then using the portion of the Double Negation Rule that we have just proven, we have

1.	$\mathcal{S} \cup \{\neg\neg\neg F\} \vdash \neg\neg\neg F$	Premise
2.	$\mathcal{S} \cup \{\neg\neg\neg F\} \vdash \neg F$	Double Negation on 1
3.	$\mathcal{S} \vdash \neg\neg\neg F \to \neg F$	Deduction on 2
4.	$\mathcal{S} \vdash F \to \neg\neg F$	Contrapositive on 3
5.	$\mathcal{S} \vdash F$	Assumption
6.	$\mathcal{S} \vdash \neg\neg F$	MP on 5,4

Now we are ready to prove the other direction of the Contrapositive Rule. Suppose that $\mathcal{S} \vdash G \to F$. Then

1.	$\mathcal{S} \cup \{\neg\neg G\} \vdash G \to F$	Adding Premises
2.	$\mathcal{S} \cup \{\neg\neg G\} \vdash \neg\neg G$	Premise
3.	$\mathcal{S} \cup \{\neg\neg G\} \vdash G$	Double Negation on 2
4.	$\mathcal{S} \cup \{\neg\neg G\} \vdash F$	MP on 3,1
5.	$\mathcal{S} \cup \{\neg\neg G\} \vdash \neg\neg F$	Double Negation on 4
6.	$\mathcal{S} \vdash \neg\neg G \to \neg\neg F$	Deduction on 5
7.	$\mathcal{S} \vdash \neg F \to \neg G$	Contrapositive on 6

To prove the Conjunction Rule, suppose first that $\mathcal{S} \vdash F$ and $\mathcal{S} \vdash G$. Then

1.	$\mathcal{S} \cup \{F \to \neg G\} \vdash F$	Adding Premises
2.	$\mathcal{S} \cup \{F \to \neg G\} \vdash F \to \neg G$	Premise
3.	$\mathcal{S} \cup \{F \to \neg G\} \vdash \neg G$	MP on 1,2
4.	$\mathcal{S} \vdash (F \to \neg G) \to \neg G$	Deduction on 3
5.	$\mathcal{S} \vdash \neg\neg G \to \neg(F \to \neg G)$	Contrapositive on 4
6.	$\mathcal{S} \vdash G$	Assumption
7.	$\mathcal{S} \vdash \neg\neg G$	Double Negation on 6
8.	$\mathcal{S} \vdash \neg(F \to \neg G)$	MP on 7,5
9.	$\mathcal{S} \vdash F \wedge G$	And/Implies on 8

Conversely, suppose that $\mathcal{S} \vdash F \wedge G$. Then

1.	$\mathcal{S} \vdash \neg(F \to \neg G)$	And/Implies
2.	$\mathcal{S} \cup \{\neg F\} \vdash \neg F$	Premise
3.	$\mathcal{S} \cup \{\neg F\} \vdash F \to \neg G$	Implication on 2
4.	$\mathcal{S} \vdash \neg F \to (F \to \neg G)$	Deduction on 3
5.	$\mathcal{S} \vdash \neg(F \to \neg G) \to \neg\neg F$	Contrapositive on 4
6.	$\mathcal{S} \vdash \neg\neg F$	MP on 1,5
7.	$\mathcal{S} \vdash F$	Double Negation on 6

and

1.	$\mathcal{S} \vdash \neg(F \to \neg G)$	And/Implies
2.	$\mathcal{S} \vdash \neg G \to (F \to \neg G)$	A1
3.	$\mathcal{S} \vdash \neg(F \to \neg G) \to \neg\neg G$	Contrapositive on 2
4.	$\mathcal{S} \vdash \neg\neg G$	MP on 1,3
5.	$\mathcal{S} \vdash G$	Double Negation on 4

To prove the first Contradiction Rule, suppose that $\mathcal{S} \vdash F \wedge \neg F$. Then

1.	$\mathcal{S} \vdash F$	Conjunction
2.	$\mathcal{S} \vdash \neg F$	Conjunction
3.	$\mathcal{S} \vdash F \to G$	Implication on 2
4.	$\mathcal{S} \vdash G$	MP on 1,3

To prove the second Contradiction Rule, suppose that $\mathcal{S} \cup \{\neg F\} \vdash G \wedge \neg G$. Then

1.	$\mathcal{S} \cup \{\neg F\} \vdash G \wedge \neg G$	Assumption
2.	$\mathcal{S} \cup \{\neg F\} \vdash \neg(G \to \neg\neg G)$	And/Implies on 1
3.	$\mathcal{S} \vdash \neg F \to \neg(G \to \neg\neg G)$	Deduction on 2
4.	$\mathcal{S} \vdash (G \to \neg\neg G) \to F$	Contrapositive on 3
5.	$\mathcal{S} \cup \{G\} \vdash G$	Premise
6.	$\mathcal{S} \cup \{G\} \vdash \neg\neg G$	Double Negation on 5
7.	$\mathcal{S} \vdash G \to \neg\neg G$	Deduction on 6
8.	$\mathcal{S} \vdash F$	MP on 7,4

Finally we prove the Implies/And Rule. This rule looks similar to the And/Implies Rule, but it is not identical. Suppose first that $\mathcal{S} \vdash \neg(F \to G)$. Then

1.	$\mathcal{S} \cup \{F \to \neg\neg G, F\} \vdash F$	Premise
2.	$\mathcal{S} \cup \{F \to \neg\neg G, F\} \vdash F \to \neg\neg G$	Premise
3.	$\mathcal{S} \cup \{F \to \neg\neg G, F\} \vdash \neg\neg G$	MP on 1,2
4.	$\mathcal{S} \cup \{F \to \neg\neg G, F\} \vdash G$	Double Negation on 3
5.	$\mathcal{S} \cup \{F \to \neg\neg G\} \vdash F \to G$	Deduction on 4
6.	$\mathcal{S} \vdash (F \to \neg\neg G) \to (F \to G)$	Deduction on 5
7.	$\mathcal{S} \vdash \neg(F \to G) \to \neg(F \to \neg\neg G)$	Contrapositive on 6
8.	$\mathcal{S} \vdash \neg(F \to G)$	Assumption
9.	$\mathcal{S} \vdash \neg(F \to \neg\neg G)$	MP on 8,7
10.	$\mathcal{S} \vdash F \wedge \neg G$	And/Implies on 9

Now suppose that $\mathcal{S} \vdash F \wedge \neg G$. Then by the And/Implies Rule we have $\mathcal{S} \vdash \neg(F \to \neg\neg G)$. Interchanging the roles of G and $\neg\neg G$ in the first 7 lines directly above, we obtain $\mathcal{S} \vdash \neg(F \to \neg\neg G) \to \neg(F \to G)$. Then Modus Ponens gives $\mathcal{S} \vdash \neg(F \to G)$.

6.11 Example: Make a derivation to show that $\neg F \vdash F \to G$.

Solution: Using the above theorem, we have

1.	$\neg F \vdash \neg F$	Premise
2.	$\neg F \vdash F \to G$	Implication on 1

This is not a derivation, it is merely a proof that a derivation exists. However, the proofs of the various parts of the above theorem can be used to expand this into a derivation. Using the proof of the Second Implication Rule, we expand the above 2 lines into the following.

1.	$\neg F \vdash \neg F$	Premise
2.	$\neg F \vdash \neg G \to \neg F$	Implication on 1
3.	$\neg F \vdash F \to G$	Contrapositive on 2

Then using the proofs of the First Implication Rule and the Contrapositive Rule, we further expand the above 3 lines to get the derivation

1.	$\neg F$	Premise
2.	$\neg F \to (\neg G \to \neg F)$	A1
3.	$\neg G \to \neg F$	MP on 1,2
4.	$(\neg G \to \neg F) \to (F \to G)$	A2
5.	$F \to G$	MP on 3,4

6.12 Example: Make a derivation to show that $F \vdash (F \to G) \to G$.

Solution: Using the above theorem we have

1.	$F, F \to G \vdash F$	Premise
2.	$F, F \to G \vdash F \to G$	Premise
3.	$F, F \to G \vdash G$	MP on 1,2
4.	$F \vdash (F \to G) \to G$	Deduction on 3

We use the inductive proof of the Deduction Rule to convert the first 3 of the above 4 lines into a derivation of $(F \to G) \to G$ from $\{F\}$.

1.	F	Premise
2.	$F \to ((F \to G) \to F)$	A1
3.	$(F \to G) \to F$	MP on 1,2
4.	$\big((F \to G) \to \big(((F \to G) \to (F \to G)) \to (F \to G)\big)\big)$ $\to \big(\big((F \to G) \to ((F \to G) \to (F \to G))\big) \to ((F \to G) \to (F \to G))\big)$	A3
5.	$(F \to G) \to \big(((F \to G) \to (F \to G)) \to (F \to G)\big)$	A1
6.	$\big((F \to G) \to ((F \to G) \to (F \to G))\big) \to ((F \to G) \to (F \to G))$	MP on 5,4
7.	$(F \to G) \to ((F \to G) \to (F \to G))$	A1
8.	$(F \to G) \to (F \to G)$	MP on 7,6
9.	$((F \to G) \to (F \to G)) \to \big(((F \to G) \to F) \to ((F \to G) \to G)\big)$	A3
10.	$((F \to G) \to F) \to ((F \to G) \to G)$	MP on 8,9
11.	$(F \to G) \to G$	MP on 3,10

6.13 Definition: Given a truth-evaluation α, the **set of true formulas under** α is

$$\mathcal{T}(\alpha) = \big\{F \,\big|\, \alpha(F) = 1\big\}.$$

Notice that $\mathcal{T}(\alpha)$ has the property that for every formula F, either $F \in \mathcal{T}(\alpha)$ or $\neg F \in \mathcal{T}(\alpha)$, but not both.

6.14 Remark: Let $\mathcal{S}$ be a satisfiable set of formulas with the property that for every formula F, either $F \in \mathcal{S}$ or $\neg F \in \mathcal{S}$. Then there is a unique truth-evaluation α which satisfies $\mathcal{S}$, and we have $\mathcal{S} = \mathcal{T}(\alpha)$.

Proof: Let α be any truth-evaluation which satisfies $\mathcal{S}$ so that for every $F \in \mathcal{S}$ we have $\alpha(F) = 1$. For any propositional variable P, since P is a formula, either we have $P \in \mathcal{S}$, in which case $\alpha(P) = 1$ or we have $\neg P \in \mathcal{S}$, in which case $\alpha(P) = 0$. Thus the truth-

evaluation α is uniquely determined and is given by

$$\alpha(P) = \begin{cases} 1 \text{ if } P \in \mathcal{S}, \\ 0 \text{ if } P \notin \mathcal{S}\,. \end{cases}$$

If $F \in \mathcal{S}$ then $\alpha(F) = 1$ so $F \in \mathcal{T}(\alpha)$, and so we have $\mathcal{S} \subseteq \mathcal{T}(\alpha)$. On the other hand, if $F \notin \mathcal{S}$ then $\neg F \in \mathcal{S}$ so $\neg F \in \mathcal{T}(\alpha)$ and hence $F \notin \mathcal{T}(\alpha)$, and this shows that $\mathcal{T}(\alpha) \subseteq \mathcal{S}$.

6.15 Definition: A set of formulas $\mathcal{S}$ is called **consistent** if there is no formula F such that $\mathcal{S} \vdash (F \wedge \neg F)$. If there is such a formula F, then $\mathcal{S}$ is called **inconsistent**.

6.16 Theorem: *Let $\mathcal{S}$ be a set of formulas and let F be a formula. Then*

(1) $\mathcal{S}$ is inconsistent if and only if $\mathcal{S} \vdash G$ for every formula G.

(2) $\mathcal{S} \vdash F \Longleftrightarrow \mathcal{S} \cup \{\neg F\}$ is inconsistent.

(3) If $\mathcal{S}$ is consistent and $\mathcal{S} \vdash F$ then $\mathcal{S} \cup \{F\}$ is consistent.

Proof: To prove part (1), suppose first that $\mathcal{S}$ is inconsistent, say $\mathcal{S} \vdash (F \wedge \neg F)$, and let G be any formula. Then by the first Contradiction Rule, we have $\mathcal{S} \vdash G$. Conversely, if $\mathcal{S} \vdash G$ for every formula G, then for every formula F we have $\mathcal{S} \vdash (F \wedge \neg F)$ so $\mathcal{S}$ is inconsistent.

To prove part (2), suppose first that $\mathcal{S} \vdash F$. Then $\mathcal{S} \cup \{\neg F\} \vdash F$ (Adding Premises) and $\mathcal{S} \cup \{\neg F\} \vdash \neg F$ (Premise) so $\mathcal{S} \cup \{\neg F\} \vdash F \wedge \neg F$ (Conjunction), and so $\mathcal{S} \cup \{\neg F\}$ is inconsistent. Conversely, suppose that $\mathcal{S} \cup \{\neg F\}$ is inconsistent, say $\mathcal{S} \cup \{\neg F\} \vdash G \wedge \neg G$. Then $\mathcal{S} \vdash F$ by the second Contradiction Rule.

Finally, to prove part (3), suppose that $\mathcal{S} \vdash F$ and that $\mathcal{S} \cup \{F\}$ is inconsistent, say $\mathcal{S} \cup \{F\} \vdash (G \wedge \neg G)$. Then $\mathcal{S} \vdash F \to (G \wedge \neg G)$ by the Implication Rule, and then $\mathcal{S} \vdash G \wedge \neg G$ by Modus Ponens, and so $\mathcal{S}$ is inconsistent. This proves part (3).

6.17 Theorem: *Let $\mathcal{S}$ be a consistent set of formulas. Then $\mathcal{S} \subseteq \mathcal{T}$ for some consistent set of formulas $\mathcal{T}$ with the property that for every formula F, either $F \in \mathcal{T}$ or $\neg F \in \mathcal{T}$.*

Proof: Make an infinite list of all formulas

$$F_1, F_2, F_3, \cdots$$

(one way to make such a list is to express formulas using a finite ordered symbol set, such as $\{\neg, \wedge, \vee, \to, \leftrightarrow, (,), P, 0, 1, \cdots, 9\}$, and then list all formulas lexicographically from short to long). Set $\mathcal{T}_0 = \mathcal{S}$ and note that $\mathcal{T}_0$ is consistent. For $k \geq 0$, having constructed a consistent set $\mathcal{T}_k$ with the property that for all $i \leq k$, either $F_i \in \mathcal{T}_k$ or $\neg F_i \in \mathcal{T}_k$, we define

$$\mathcal{T}_{k+1} = \begin{cases} \mathcal{T}_k \cup \{F_{k+1}\} & \text{if } \mathcal{T}_k \vdash F_{k+1}, \\ \mathcal{T}_k \cup \{\neg F_{k+1}\} & \text{if } \mathcal{T}_{k+1} \nvdash F_{k+1}\,. \end{cases}$$

By the above theorem, $\mathcal{T}_{k+1}$ is consistent; if $\mathcal{T}_k \vdash F_{k+1}$ then $\mathcal{T}_{k+1}$ is consistent by part (3), and if $\mathcal{T}_k \nvdash F_{k+1}$ then $\mathcal{T}_{k+1}$ is consistent by part (2). Thus we obtain consistent sets $\mathcal{T}_k$ with

$$\mathcal{S} = \mathcal{T}_0 \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \cdots\,.$$

such that each $\mathcal{T}_k$ has the property that for all $i \leq k$, either $F_i \in \mathcal{T}_k$ or $\neg F_i \in \mathcal{T}_k$. Now let

$$\mathcal{T} = \bigcup_{k=0}^{\infty} \mathcal{T}_k\,.$$

Clearly we have $\mathcal{S} \subseteq \mathcal{T}$. Also, given any formula F, either $F \in \mathcal{T}$ or $\neg F \in \mathcal{T}$; indeed we have $F = F_k$ for some k, and either $F_k \in \mathcal{T}_k$ or $\neg F_k \in \mathcal{T}_k$, and $\mathcal{T}_k \subseteq \mathcal{T}$. To finish

the proof, it remains only to show that $\mathcal{T}$ is consistent. Suppose, for a contradiction, that $\mathcal{T}$ is inconsistent, say $\mathcal{T} \vdash (K \wedge \neg K)$. Only finitely many formulas in $\mathcal{T}$ are used in any derivation of $(K \wedge \neg K)$, so we have $\{G_1, \cdots, G_l\} \vdash (K \wedge \neg K)$ for some $G_i \in \mathcal{T}$. For each i we have $G_i \in \mathcal{T}_{k_i}$ for some k_i. If we let $k = \max\{k_1, \cdots, k_l\}$ then for all i we have $G_i \in \mathcal{T}_k$, and so $\mathcal{T}_k \vdash (K \wedge \neg K)$. This contradicts the fact that $\mathcal{T}_k$ is consistent.

6.18 Theorem: *Let $\mathcal{S}$ be a consistent set of formulas with the property that for every formula F, either $F \in \mathcal{S}$ or $\neg F \in \mathcal{S}$. Then $\mathcal{S} = \mathcal{T}(\alpha)$, where α is the truth-evaluation*

$$\alpha(P) = \begin{cases} 1 \text{ if } P \in \mathcal{T} \\ 0 \text{ if } P \notin \mathcal{T}\,. \end{cases}$$

Proof: First note that if $F \in \mathcal{S}$ then certainly $\mathcal{S} \vdash F$, and if $\neg F \in \mathcal{S}$ then $\mathcal{S} \vdash \neg F$, and we cannot have both $F \in \mathcal{S}$ and $\neg F \in \mathcal{S}$ since if we did, then we would have $\mathcal{S} \vdash F$ and $\mathcal{S} \vdash \neg F$ so that $\mathcal{S} \vdash (F \wedge \neg F)$ (by the Conjunction Rule), but then $\mathcal{S}$ would be inconsistent. Thus for every formula F we have $F \in \mathcal{S} \Longleftrightarrow \mathcal{S} \vdash F$.

We show that $\mathcal{S} = \mathcal{T}(\alpha)$ by using induction on formulas to show that for all formulas F we have $F \in \mathcal{S} \Longleftrightarrow \alpha(F) = 1$. Note that when $F = P$, where P is a propositional variable, we do have $F \in \mathcal{S} \Longleftrightarrow \alpha(F) = 1$ by the definition of α. Suppose, inductively, that $G \in \mathcal{S} \Longleftrightarrow \alpha(G) = 1$ and that $H \in \mathcal{S} \Longleftrightarrow \alpha(H) = 1$.

For $F = \neg G$ we have $F = \neg G \in \mathcal{S} \iff G \notin \mathcal{S} \Longleftrightarrow \alpha(G) = 0 \Longleftrightarrow \alpha(F) = \alpha(\neg G) = 1$.

Now let $F = (G \wedge H)$. Using the Conjunction Rule, we have $F \in \mathcal{S} \Longleftrightarrow (G \wedge H) \in \mathcal{S}$ $\Longleftrightarrow \mathcal{S} \vdash (G \wedge H) \iff \big(\mathcal{S} \vdash G \text{ and } \mathcal{S} \vdash H\big) \iff \big(\alpha(G) = 1 \text{ and } \alpha(H) = 1\big) \Longleftrightarrow \alpha(G \wedge H) = 1$ $\Longleftrightarrow \alpha(F) = 1$.

Next let $F = (G \vee H)$. Using the Or/And Rule and the Conjuction Rule we have $F \notin \mathcal{S} \iff (G \vee H) \notin \mathcal{S} \iff \neg(G \vee H) \in \mathcal{S} \iff \mathcal{S} \vdash \neg(G \vee H) \iff \mathcal{S} \vdash (\neg G \wedge \neg H)$ $\Longleftrightarrow \big(\mathcal{S} \vdash \neg G \text{ and } \mathcal{S} \vdash \neg H\big) \iff \big(G \notin \mathcal{S} \text{ and } H \notin \mathcal{S}\big) \iff \big(\alpha(G) = 0 \text{ and } \alpha(H) = 0\big)$ $\Longleftrightarrow \alpha(G \vee H) = 0 \iff \alpha(F) = 0$.

Next let $F = (G \to H)$. Then using the Implies/And Rule and the Conjunction Rule, $F \notin \mathcal{S} \iff (G \to H) \notin \mathcal{S} \iff \neg(G \to H) \in \mathcal{S} \iff \mathcal{S} \vdash \neg(G \to H) \iff \mathcal{S} \vdash (G \wedge \neg H)$ $\Longleftrightarrow \big(\mathcal{S} \vdash G \text{ and } \mathcal{S} \vdash \neg H\big) \iff \big(G \in \mathcal{S} \text{ and } H \notin \mathcal{S}\big) \iff \big(\alpha(G) = 1 \text{ and } \alpha(H) = 0\big)$ $\Longleftrightarrow \alpha(G \to H) = 0 \iff \alpha(F) = 0$.

Finally, let $F = (G \leftrightarrow H)$. By the previous paragraph $\mathcal{S} \vdash (G \to H) \Longleftrightarrow \alpha(G \to H) = 1$ and similarly $\mathcal{S} \vdash (H \to G) \Longleftrightarrow \alpha(H \to G) = 1$. The Iff/Implies and Conjunction Rules give $F \in \mathcal{S} \iff (G \leftrightarrow H) \in \mathcal{S} \iff \mathcal{S} \vdash (G \leftrightarrow H) \iff S \vdash \big((G \to H) \wedge (H \to G)\big) \iff$ $\big(\mathcal{S} \vdash (G \to H) \text{ and } \mathcal{S} \vdash (H \to G)\big) \Longleftrightarrow \big(\alpha(G \to H) = 1 \text{ and } \alpha(H \to G) = 1\big) \iff \alpha(G \leftrightarrow H) = 1$ $\Longleftrightarrow \alpha(F) = 1$.

6.19 Theorem: *(The Completeness Theorem) Let $\mathcal{S}$ be a set of formulas and let F be a formula. Then*

$$\mathcal{S} \models F \Longleftrightarrow \mathcal{S} \vdash F\,.$$

Proof: We have already remarked that if $\mathcal{S} \vdash F$ then $\mathcal{S} \models F$. Suppose that $\mathcal{S} \nvdash F$. Then by part (2) of theorem 6.16, $\mathcal{S} \cup \{\neg F\}$ is consistent. By theorem 6.17, we can choose a consistent set of formulas $\mathcal{T}$, containing $\mathcal{S} \cup \{\neg F\}$, with the property that for every formula G, either $G \in \mathcal{T}$ or $\neg G \in \mathcal{T}$. By theorem 6.18, $\mathcal{T} = \mathcal{T}(\alpha)$ for some truth-evaluation α, and α satisfies $\mathcal{T}$. Since $\mathcal{S} \cup \{\neg F\} \subseteq \mathcal{T}$, the truth-evaluation α also satisfies $\mathcal{S} \cup \{\neg F\}$. Thus $\mathcal{S} \cup \{\neg F\}$ is satisfiable and so from the last part of theorem 2.15, $\mathcal{S} \not\models F$.

6.20 Theorem: *(the Compactness Theorem) Let $\mathcal{S}$ be a set of formulas and let F be a formula. If $\mathcal{S} \models F$ then there is a finite subset $\mathcal{T} \subseteq \mathcal{S}$ such that $\mathcal{T} \models F$.*

Proof: Suppose that $\mathcal{S} \models F$. Then by the Completeness Theorem, we also have $\mathcal{S} \vdash F$. Any derivation of F from $\mathcal{S}$ only uses finitely many premises, so there is a finite subset $\mathcal{T} \subseteq \mathcal{S}$ such that $\mathcal{T} \vdash F$. By the Completeness Theorem, we have $\mathcal{T} \models F$.

Chapter 7. First-Order Formulas

7.1 Remark: In this chapter we shall define terms and formulas in first-order languages, and we shall define the free variables in a formula. In the next chapter, we shall see that when a formula is placed in context, its terms will represent elements in a set, the formula will represent a meaningful mathematical statement about its free variables, and the truth of the statement will depend on the values which are assigned to its free variables. We shall provide one example now to motivate some of our definitions in this and the next chapter.

7.2 Example: The formula $F = \forall y\ x \times y \approx y \times x$ is a meaningless string of symbols until it is put into context. Its terms are the variable symbols x and y, and also the strings $x \times y$ and $y \times x$. The variable x is free, and the variable y is bound by the quantifier.

If these terms are taken to represent real numbers, where the symbol $\times$ represents multiplication, then the formula becomes the meaningful mathematical statement

"The real number x commutes with every real number."

This statement is true no matter which real number is assigned to the variable x.

If, on the other hand, the terms are taken to represent 2×2 matrices with real entries, then the formula becomes the meaningful mathematical statement

"The 2×2 matrix x commutes with all 2×2 matrices."

This statement can be true or false depending on the 2×2 matrix which is assigned to x. For example when x is taken to be the identity matrix, the statement is true, but when x is taken to be the matrix $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, the statement is false.

7.3 Definition: All **first-order languages** use symbols from the following **symbol set**:

$$\{\forall, \exists, \approx, \neg, \wedge, \vee, \rightarrow, \leftrightarrow,), (, x_1, x_2, x_3, \cdots\}$$

The symbols $\forall$ and $\exists$ are called **quantifiers**, specifically, $\forall$ is called the **universal quantifier**, and $\exists$ is called the **existential quantifier**. The symbol $\approx$ is the **equality** symbol. The symbols $\neg, \wedge, \vee, \rightarrow$ and $\leftrightarrow$ are called the **logic symbols**, and they have the same names as in propositional logic. The symbols x_i are called the **variable symbols**. We shall often use the symbols x, y, z, u, v, w to represent variable symbols.

In addition to the above symbols, a first-order language can use some or all of the following **special symbols**:

$$\{c_1, c_2, c_3, \cdots\} \cup \{f_1^1, f_2^1, f_3^1, \cdots f_1^2, f_2^2, f_3^2, \cdots\} \cup \{r_1^1, r_2^1, r_3^1, \cdots, r_1^2, r_2^2, r_3^2, \cdots\}$$

The symbols c_i are called the **constant symbols**, the symbols f_i^k are called the **function symbols**, and the symbols r_i^k are called the **relation symbols**. We shall often use a, b, c to represent constant symbols, f, g, h to represent function symbols, and p, q, r to represent relation symbols (in example 7.2, the symbol $\times$ is used as a binary function symbol).

The superscript k on the function symbol f_i^k is called its **arity**, and we say that f_i^k is a **k-ary** function symbol, and in particular, a 1-ary function symbol is called **unary**, a 2-ary function symbol is called **binary**, and a 3-ary function symbol is called **ternary**. When we avoid the use of superscripts and subscripts by using f, g and h to represent function

symbols, it will be necessary to specify the arity of each symbol. Similar definitions apply to the relation symbol r_i^k and its superscript k.

7.4 Definition: A **term** in a first-order language is a non-empty string, of variable, constant and function symbols, which can be obtained using finitely many applications of the following rules.

1. Every variable symbol is a term.
2. Every constant symbol is a term.
3. If $t_1, \cdots, t_n$ are terms and if f is an n-ary function symbol, then $ft_1 \cdots t_n$ is a term.

A term in **bracket notation** is defined similarly, but in the third rule, $ft_1 \cdots t_n$ is written as $f(t_1, \cdots, t_n)$. For certain binary function symbols f, we write (xfy) instead of fxy. The notations fxy and (xfy) are respectively known as **prefix notation**, and **infix notation**.

7.5 Theorem: *(Unique Readability of Terms) Terms are uniquely readable in the sense that every term t is either a variable symbol, or a constant symbol, or is of the form $t = ft_1 \cdots t_n$, where the terms $t_1 \cdots t_n$ are uniquely determined.*

Proof: As an exercise, you can try to prove this by imitating the proof of the Unique Readability Theorem for formulas in propositional logic.

7.6 Remark: If we used function symbols with undetermined arity, then we would lose unique readability. For example, if f and g are function symbols of unknown arity, and if $t = gfxy$ is a term, then it could be that f is unary and g is binary, in which case t would become $g(f(x), y)$ in bracket notation, or it could be that f is binary and g is unary, in which case t would become $g(f(x, y))$.

Similarly, if we used infix notation without brackets, that is if we wrote xfy instead of fxy, then we would lose unique readability. For example, if f is a binary function symbol used with infix notation without brackets, then the term $xfyfz$ could become either one of the two terms $((xfy)fz)$ or $(xf(yfz))$ when put into infix notation with brackets. In prefix notation, these two terms become $ffxyz$ and $fxfyz$ respectively.

In example 7.2, the symbol $\times$ is a binary function symbol used with infix notation without brackets.

7.7 Note: We can use the **tub algorithm** (which was used for propositional formulas in prefix notation in chapter 1) to determine whether a given string is a term, or a list of terms. In this new context, the algorithm works as follows.

Step 0: Let n be a counter.

Step 1: If the last symbol is not a variable or a constant, then the given string cannot be a list of terms. If the the last symbol is a variable or a constant, then set the counter to $n = 1$ and begin to work through the given string, from right to left.

Step 2: Look at the next symbol to the left: if it is a variable or constant then increase the value of n by 1; if it is a k-ary function symbol then reduce the value of n by $(k - 1)$.

Step 3: At this stage, if $n \leq 0$ then the given string cannot be a list of terms, and if $n > 0$ then the portion which has been examined so far is a list of n terms. If we have not yet examined every symbol in the given string, then go back to step 2.

7.8 Example: Let f, g and h be function symbols with f unary, g binary, and h ternary. Determine whether the string $fx\,h\,g\,h\,y\,fx\,z\,g\,a\,fb\,fz$ is a term or a list of terms.

Solution: The results of the tub algorithm are shown below. The value of the counter is shown beneath each symbol, and should be read from right to left.

$$\begin{array}{c} f\,x\,h\,g\,h\,y\,f\,x\,z\,g\,a\,f\,b\,f\,z \\ 0\,2\,3\,5\,4\,4\,3\,2\,3\,2\,2\,1\,1 \end{array}$$

At this stage, the value of the counter is 0, so we see that the string is not a list of terms.

7.9 Example: Again let f, g and h be function symbols with f unary, g binary and h ternary. Determine whether the string $g\,g\,x\,h\,a\,b\,f\,g\,a\,y\,x\,h\,c\,g\,f\,x\,f\,y\,g\,c\,x\,g\,f\,y\,z$ is a term or list of terms.

Solution: We perform the tub algorithm:

$$\begin{array}{c} g\,g\,x\,h\,a\,b\,f\,g\,a\,y\,x\,h\,c\,g\,f\,x\,f\,y\,g\,c\,x\,g\,f\,y\,z \\ 3\,4\,5\,4\,6\,5\,4\,4\,5\,4\,3\,2\,4\,3\,4\,4\,3\,3\,2\,3\,2\,1\,2\,2\,1 \end{array}$$

Since the final counter value is 3, this string is a list of three terms. When performing the tub algorithm by hand, it is convenient to draw tubs underneath the terms as we count them:

With the help of the tubs, it is easy to convert the list into bracket notation:

$$g\big(g(x,h(a,b,f(g(a,y)))),x\big)\ ,\ h\big(c,g(f(x),f(y)),g(c,x)\big)\ ,\ g(f(y),z)$$

7.10 Definition: A **formula** in a first-order language is a non-empty string which can be obtained using finitely many applications of the following rules.

1. If t_1 and t_2 are terms, then $t_1 \approx t_2$ is a formula.
2. If $t_1, \cdots, t_n$ are terms and r is an n-ary relation symbol, then $rt_1 \cdots t_n$ is a formula.
3. If F is a formula, then so is $\neg F$.
4. If F and G are formulas, and $\times \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, then $(F \times G)$ is a formula.
5. If F is a formula and x is a variable symbol, then $\forall x\, F$ and $\exists x\, F$ are formulas.

A formula obtained from rule 1 or rule 2 is called an **atomic** formula. For certain binary relation symbols, we shall write rxy in **infix notation** as xry. For example, the equality symbol $\approx$ is a binary relation symbol which we use in infix notation.

7.11 Note: We can make a **derivation** for a formula as we did in propositional logic to show explicitly how the formula can be obtained from the above rules.

7.12 Example: Let $F = \forall x\,(\neg\exists y\ rfgfxyfa \rightarrow fy \approx gfax)$, where f and g are function symbols with f unary and g binary, and r is a binary relation symbol. Make a derivation for F.

Solution: First, use the tub algorithm to verify that $fgfxy$, fa, fy and $gfax$ are terms. Alternatively, if you prefer, you could make a derivation for each of these terms. For example, a possible derivation for the term $gfax$ is as follows:

1.	x	rule 1, with the variable symbol x
2.	a	rule 2, with the constant symbol a
3.	fa	rule 3, with f, on term 2
4.	$gfax$	rule 3, with g, on terms 3 and 1

The above justifications refer to rules 1,2 and 3 from the definition of a term. Once we

have found the terms in F, we can make a derivation for the formula F:

1.	$rfgfxyfa$	rule 2, with r, on the terms $fgfxy$ and fa
2.	$\exists y\, rfgfxyfa$	rule 5, with $\exists y$, on line 1
3.	$\neg\exists y\, rfgfxyfa$	rule 3 on line 2
4.	$fy \approx gfax$	rule 1 on the terms fy and $gfax$
5.	$(\neg\exists y\, rfgfxyfa \rightarrow fy \approx gfax)$	rule 4, with $\rightarrow$, on lines 3 and 4
6.	$\forall x\,(\neg\exists y\, rfgfxyfa \rightarrow fy \approx gfax)$	rule 5, with $\forall x$, on line 5

The above justifications refer to rules 1-5 from the definition of a formula.

7.13 Note: Formulas are uniquely readable, in the sense that in any formula,

1. every occurrence of the symbol $\approx$ is followed by a uniquely determined term and preceded by a uniquely determined maximal term,

2. every n-ary relation symbol is followed by n uniquely determined terms,

3. every occurrence of the symbol $\neg$ is followed by a uniquely determined formula,

4. every occurrence of the symbol (is followed by a uniquely determined formula, which is followed by a unique binary connective $\times \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, which is followed, in turn, by another uniquely determined formula, which is followed by the symbol), and

5. every occurrence of a quantifier is followed by a variable symbol, which is followed by a uniquely determined formula. This formula is called the **scope** of (the occurrence of) the quantifier.

7.14 Example: Let f and g be function symbols with f unary and g binary, and let r be a binary relation symbol. Determine which of the following strings are formulas.
a) $\forall x\,(\exists y\; rxy \;\rightarrow\; gfxx \approx \neg ffy)$
b) $(\exists x\; rfxy \wedge (gbfy \vee x \approx fy))$
c) $\forall f\,\forall x\,\exists y\,(y \approx fx \wedge \forall z\,(z \approx fx \;\rightarrow\; z \approx y))$

Solution: None of these strings are formulas. The string in part a) is not a formula because the symbol $\approx$ must be followed by a term, but it is followed by the symbol $\neg$, which does not occur in terms. The string in part b) is not a formula because the second open bracket must be followed by a formula then by a binary connective, but it is followed by the *term* $gbfy$ (which is *not* a formula) then by the connective $\vee$. The string in part c) is not a formula because the first quantifier must be followed by a variable symbol, but it is followed by a function symbol (this string is a formula in **second-order** logic).

7.15 Definition: Let x be a variable symbol. For each occurrence of the symbol x, which does not immediately follow a quantifier, in a formula, we define whether the occurrence of x is **free** or **bound** inductively as follows.

1. In the formula $t_1 \approx t_2$, every occurrence of x is free, and no occurrence is bound.

2. In the formula $rt_1 \cdots t_n$, every occurrence of x is free, and no occurrence is bound.

3. In the formula $\neg F$, every occurrence of x is free or bound according to whether it was a free or bound occurrence in F.

4. In the formula $(F \times G)$, where $\times \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, each occurrence of the symbol x is either an occurrence in the formula F or an occurrence in the formula G, and each free (respectively, bound) occurrence of x in F remains free (respectivly, bound) in $(F \times G)$, and similarly for each free (or bound) occurrence of x in G.

5. For any variable symbol y other than x, each free (or bound) occurrence of x in F remains free (or bound) in the formula $\forall y\, F$, and similarly for the formula $\exists y\, F$. On the other hand, every free occurrence of x in F becomes bound in the formula $\forall x\, F$, and

every bound occurrence of x in F remains bound in the formula $\forall x\, F$, and similarly for the formula $\exists x\, F$.

A **free variable** of a formula F is any variable symbol that has at least one free occurence in F. A formula F which has no free variables is called a **sentence**.

When a quantifier in a given formula is followed by the variable symbol x and then by the formula F (so F is the **scope** of that quantifier), any free occurrence of x in F will become bound in the given formula (by an application of part 5 in the above definition), and we shall say that that occurrence of x **is bound by** (that occurrence of) the quantifier, or that (that occurrence of) the quantifier **binds** that occurrence of x.

7.16 Example: Let f and g be function symbols with f unary and g binary, and let r be a binary relation symbol. Let F be the formula

$$F = \exists x \left(\forall y \; gax \approx gyz \;\rightarrow\; (\exists x \neg \forall y \; rgxfyz \vee \exists z \neg rfgxyz) \right).$$

Determine which occurrences of the variable symbols in F are free and which are bound, and for each bound occurrence, indicate which quantifier binds it.

Solution: We indicate the free and bound occurrences and their binding quantifiers by placing integral labels under the relevant symbols: the free variables are given the label 0, each quantifier is given its own non-zero label, and each bound variable is given the same label as its binding quantifier:

$$\begin{array}{c} F = \exists x \left(\forall y \; gax \approx gyz \;\rightarrow\; (\exists x \neg \forall y \; rgxfyz \vee \exists z \neg rfgxyz) \right). \\ \quad 1 \quad 2 \quad\; 1 \quad\;\; 20 \quad\;\; 3 \quad 4 \quad\; 3 \;\; 40 \quad 5 \quad\;\; 105 \end{array}$$

7.17 Definition: Let x be a variable symbol, and let t be a term. Given any term s, we define $[s]_{x \mapsto t}$ inductively as follows:

1. $[y]_{x \mapsto t} = \begin{cases} t \text{ , if } y = x, \text{ and} \\ y \text{ , if } y \neq x, \end{cases}$ where y is a variable symbol,
2. $[c]_{x \mapsto t} = c$, for any constant symbol c, and
3. $[f t_1 \cdots t_n]_{x \mapsto t} = f[t_1]_{x \mapsto t} \cdots [t_n]_{x \mapsto t}$, for any n-ary function symbol f and any terms $t_1 \cdots t_n$.

As an exercise, you can prove, by induction on terms, that $[s]_{x \mapsto t}$ is a term; it is called the **term obtained by replacing all occurrences of x in s by t**. And now, for a formula F, we define $[F]_{x \mapsto t}$ inductively as follows:

1. $[t_1 \approx t_2]_{x \mapsto t} = [t_1]_{x \mapsto t} \approx [t_2]_{x \mapsto t}$,
2. $[r t_1 \cdots t_n]_{x \mapsto t} = r[t_1]_{x \mapsto t} \cdots [t_n]_{x \mapsto t}$,
3. $[\neg G]_{x \mapsto t} = \neg [G]_{x \mapsto t}$,
4. $\left[(G \times H) \right]_{x \mapsto t} = \left([G]_{x \mapsto t} \times [H]_{x \mapsto t} \right)$, where $\times \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, and
5. $[Ky\, G]_{x \mapsto t} = \begin{cases} Kx\, G & \text{, if } y = x, \\ Ky\, [G]_{x \mapsto t} & \text{, if } y \neq x \text{ and } y \text{ does not occur in } t, \\ Ku\, \left[[G]_{y \mapsto u} \right]_{x \mapsto t} & \text{, if } y \neq x \text{ and } y \text{ does occur in } t \end{cases}$

where $K \in \{\forall, \exists\}$, and u is the first variable symbol which is not x and does not occur in G or in t. As an exercise, you can verify that $[F]_{x \mapsto t}$ is a formula. It is called the **formula obtained by replacing the free occurrences of x in F by t** .

7.18 Example: Let f and g be function symbols with f unary and g binary, and let r be a binary relation symbol. Let $F = \big(\forall x\,\exists z\, z \approx gxy \vee \forall z\,(\neg fgxy \approx z \to \exists y\, r f x y)\big)$. Find $[F]_{y \mapsto fx}$ and $[F]_{x \mapsto gyx}$, using x, y, z and u, in that order, to represent the first 4 variable symbols.

Solution: We have

$$\begin{aligned}
[F]_{y\mapsto fx} &= \big[\big(\forall x\,\exists z\, z \approx gxy \vee \forall z\,(\neg fgxy \approx z \to \exists y\, r f x y)\big)\big]_{y\mapsto fx} \\
&= \big([\forall x\,\exists z\, z \approx gxy]_{y\mapsto fx} \vee [\forall z\,(\neg fgxy \approx z \to \exists y\, r f x y)]_{y\mapsto fx}\big) \\
&= \big(\forall u\,[\exists z\, z \approx guy]_{y\mapsto fx} \vee \forall z\,[\neg fgxy \approx z \to \exists y\, r f x y]_{y\mapsto fx}\big) \\
&= \big(\forall u\,\exists z\,[z \approx guy]_{y\mapsto fx} \vee \forall z\,([\neg fgxy \approx z]_{y\mapsto fx} \to [\exists y\, r f x y]_{y\mapsto fx})\big) \\
&= \big(\forall u\,\exists z\, z \approx gufx \vee \forall z\,(\neg fgxfx \approx z \to \exists y\, r f x y)\big)
\end{aligned}$$

and

$$\begin{aligned}
[F]_{x\mapsto gyx} &= \big[\big(\forall x\,\exists z\, z \approx gxy \vee \forall z\,(\neg fgxy \approx z \to \exists y r f x y)\big)\big]_{x\mapsto gyx} \\
&= \big([\forall x\,\exists z\, z \approx gxy]_{x\mapsto gyx} \vee \forall z\,\big([\neg fgxy \approx z]_{x\mapsto gyx} \to [\exists y\, r f x y]_{x\mapsto gyx}\big)\big) \\
&= \big(\forall x\,\exists z\, z \approx gxy \vee \forall z\,\big(\neg fggyxy \approx z \to \exists z\,[r f x z]_{x\mapsto gyx}\big)\big) \\
&= \big(\forall x\,\exists z\, z \approx gxy \vee \forall z\,(\neg fggyxy \approx z \to \exists z\, r f g y x z)\big)
\end{aligned}$$

Chapter 8. Interpretations

8.1 Definition: For a first-order language, an **interpretation** consists of

1. a non-empty set (or class) V, called the **universal set** (or the universal class),
2. a constant $c^V \in V$ for each constant symbol c,
3. an n-ary function $f^V : V^n \to V$ for each n-ary function symbol f, and
4. an n-ary relation $r^V \subseteq V^n$ for each n-ary relation symbol r.

We shall usually refer to the above interpretation simply as **the interpretation** V, and we shall often write c^V, f^V and r^V simply as c, f and r (so we will not distinguish notationally between the meaningless symbols and their meaningful counterparts).

8.2 Remark: Given an interpretation V, a formula becomes a meaningful mathematical statement. If the formula is a **sentence**, which means that it has no free variables, then the statement will either be true in V or false in V. If the formula does have free variables, then the statement will be a statement about those free variables, and its truth may depend on the values assigned to those free variables. In the next chapter, we shall give a formal definition for the truth of a formula in an interpretation under an assignment, but for now, we shall provide several fairly familiar examples of interpretations, in which we deal with the truth of formulas somewhat informally.

8.3 Definition: The language of **first-order number theory** is the first-order language with the special symbol set

$$\{0, 1, +, \times, <\}$$

where 0 and 1 are constant symbols, $+$ and $\times$ are binary function symbols which are both used with infix notation (often without brackets with the understanding that $\times$ is performed before $+$), and $<$ is a binary relation symbol which is used with infix notation.

8.4 Note: Unless otherwise stated, we shall *not* allow ourselves to use any other special symbols. In particular, we shall not allow ourselves to use the constant symbol 2, the unary function symbol $-$, the binary function symbol $-$, or the binary relation symbol $\leq$.

8.5 Note: The interpretations that we shall normally use for the language of first-order number theory, are the interpretations in which the universal set is the set of natural numbers $\mathbf{N} = \{0, 1, 2, \cdots\}$, the set of integers $\mathbf{Z}$, the set of rational numbers $\mathbf{Q}$, or the set of real numbers $\mathbf{R}$, and where the symbols 0, 1, $+$, $\times$ and $<$ are all given their usual meanings. If we wanted to use the interpretations in which the universal set is the set $\mathbf{Z}_n$ of integers modulo n, or the set of complex numbers $\mathbf{C}$, then we would probably avoid using the symbol $<$, which does not have a standard meaning for either of these two sets.

8.6 Example: Translate each of the following mathematical statements about integers into formulas in the language of first-order number theory.
a) $x|y$, that is, y is a multiple of x.
b) $x \equiv y \pmod{z}$.
c) $x = \min\{y, z\}$.
d) x is prime.
e) x is a power of 2.

Solution: Here are some possible translations.
a) $\exists z\ y \approx x \times z$
b) $\exists u\ x \approx y + z \times u$
c) $\big((y < z\ \rightarrow\ x \approx y) \wedge (\neg y < z\ \rightarrow\ x \approx z)\big)$
d) $\big(1 < x \wedge \big(\forall y\, \forall z\, ((0 < y \wedge 0 < z) \wedge x \approx y \times z)\ \rightarrow\ (y \approx 1 \vee z \approx 1)\big)\big)$
e) Note first that x is a power of 2 if and only if x is positive and every factor of x, which is greater than 1, is even. So we can translate this statement into the formula

$$\big(0 < x \wedge \forall y\, ((1 < y \wedge \exists z\ x \approx y \times z)\ \rightarrow\ \exists u\ y \approx u + u)\big)$$

8.7 Example: If we replace the symbol $\wedge$ by the symbol $\vee$ in our translation of part (c) in the above example, we obtain the formula $F = ((y < z\ \rightarrow\ x \approx y) \vee (\neg y < z\ \rightarrow\ x \approx z))$. Show that this formula is true in all interpretations, no matter what values are assigned to x, y and z (such a formula is called a *tautology*).

Solution: Let V be any set, let $<$ be any binary relation on that set, and let x, y and z be any elements of V. The formula $y < z$ must either be true or false. If $y < z$ is false, then the formula $(y < z\ \rightarrow\ x \approx y)$ will be true, and hence F will be true. If $y < z$ is true, then the formula $\neg y < z$ will be false, so the formula $(\neg y < z\ \rightarrow x \approx z)$ will be true, and hence F will again be true.

8.8 Example: Add the unary function symbol f to the language of first-order number theory, and translate the following statements, about real numbers and the function f, into formulas.
a) Every real real number has a unique cubed root.
b) $0 < |x - a| < b$.
c) f is increasing.
d) $\lim\limits_{x \to a} f(x) = b$.

Solution: Here are some possible translations.
a) $\forall x\, \exists y\, (y \times y \times y \approx x \wedge \forall z\, (z \times z \times z \approx x\ \rightarrow\ z \approx y))$
b) Note first that we have $0 < |x - a| < b$ if and only if ($x \neq a$ and $-b < x - a < b$) if and only if ($x \neq a$ and $-b < x - a$ and $x - a < b$), so we can translate this into the formula

$$(\neg x \approx a \wedge (a < x + b \wedge x < a + b))$$

c) $\forall x\, \forall y\, (x < y\ \rightarrow fx < fy)$
d) Let ϵ and δ be variable symbols. Then $\lim\limits_{x \to a} f(x) = b$ translates as

$$\forall \epsilon\, \big(0 < \epsilon\ \rightarrow\ \exists \delta\, \big(0 < \delta \wedge \forall x\, \big((\neg x \approx a \wedge (a < x + \delta \wedge x < a + \delta))\ \rightarrow\ (b < fx + \epsilon \wedge fx < b + \epsilon)\big)\big)\big)$$

8.9 Remark: The following example illustrates that when a mathematical statement is written in English, it can sometimes be vague and ambiguous, but when it is expressed as a first-order formula, its meaning will be precise and unambiguous.

8.10 Example: Translate the statement "there is a real number between any two real numbers" into a formula in first-order number theory.

Solution: The meaning of this statement is unclear; it can be interpreted in several different ways. We shall assume that the word "between" means "strictly between", but we can still find several possible meanings for this statement.

The statement "there exists a real number x such that for all real numbers y and z, x is between y and z" can be translated as $\exists x\,\forall y\,\forall z((y < x \wedge x < z) \vee (z < x \wedge x < y))$. This statement is false. Indeed, given any x we could choose $y = x + 1$ and $z = x + 2$ to get a counterexample.

Perhaps the statement was intended to mean "given any real numbers y and z, it is possible to find a real number x which is between y and z". This version of the statement can be translated as $\forall y\,\forall z\,\exists x\,((y < x \wedge x < z) \vee (z < x \wedge x < y))$. This is also false, since it could be that $y = z$.

Perhaps the statement was supposed to be the true statement "given any two distinct real numbers y and z, there exists a real number x which is between y and z. This can be translated as $\forall y\,\forall z\,\big(\neg y \approx z \;\rightarrow\; \exists x\,((y < x \wedge x < z) \vee (z < x \wedge x < y))\big)$, or it can be translated a little more efficiently as $\forall y\,\forall z\,\big(y < z \;\rightarrow\; \exists x\,(y < x \wedge x < z)\big)$.

8.11 Example: For each of the following sentences, determine which of the interpretations **N**, **Z**, **Q** and **R** make the statement true.

a) $F = \exists x\,\forall y\;\neg y < x$
b) $G = \forall x\,\forall y\,((x < y \vee x \approx y) \;\leftrightarrow\; x < y + 1)$
c) $H = \forall x\,(\neg x \approx 0 \;\rightarrow\; \exists y\; x \times y \approx 1)$
d) $K = \exists x\;(1 + 1) \times x \approx 1 \wedge \neg\,\exists x\; x \times x \approx 1 + 1$

Solution: In all four interpretations, the formula F means "there exists a number which is less than or equal to every number", that is "there exists a smallest number". This is true in **N** but false in the other three interpretations.

The formula G means "for all numbers x and y we have $x \leq y$ if and only if $x < y+1$". This is true in **N** and in **Z** but false in **Q** and **R**.

The formula H means "every non-zero number has a multiplicative inverse". This is true in **Q** and in **R** but false in **N** and in **Z**.

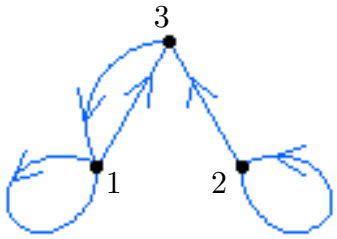
The formula K means "the number 2 has a multiplicative inverse, but it has no square root". This is true in **Q** but false in the other three interpretations.

8.12 Definition: A **directed graph** G consists of a non-empty set V called the **vertex set**, and a set $E \subseteq V^2$ called the **edge set**. When $(x, y) \in E$ we say that (x, y) is an **edge from** x **to** y and we say that x is the **initial point** of the edge (x, y) and that y is the **final point** of the edge (x, y). When V has n elements, we say that G is a graph **on** n **vertices** and, in this case, we shall usually take $V = \{1, 2, \cdots, n\}$.

8.13 Note: We can draw a picture to represent a given graph G by drawing one point in the plane, labeled by x, for each element $x \in V$, and by drawing a line with an arrow on it from the point x to the point y for each pair $(x, y) \in E$.

8.14 Example: Draw a picture of the graph with vertex set $V = \{1, 2, 3\}$ and edge set $E = \{(1, 1), (1, 3), (2, 2), (2, 3), (3, 1)\}$.

Solution:



8.15 Remark: We shall not draw a picture in which there are *two* lines with arrows from one vertex to another, since that would correspond to an ordered pair being listed twice in the edge set E. Similarly, we shall not draw a picture with two loops at one point. For a loop, it does not make any difference which way the arrow is pointing.

8.16 Definition: The language of **first-order directed graph theory** is the first-order language with the special symbol set

$$\{r\}$$

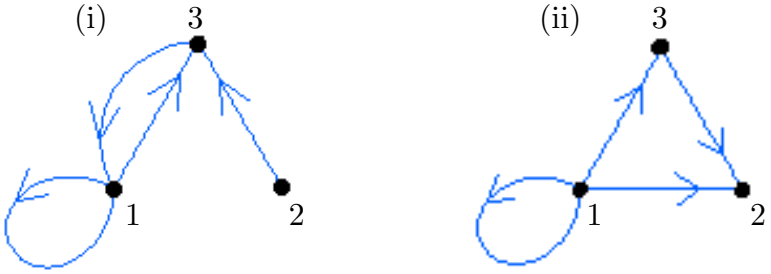
where r is a binary relation symbol, which we shall use with infix notation.

8.17 Note: For the first-order language of directed graph theory, each interpretation can be considered to be a graph: the universal set V is the vertex set of the graph, and the relation r^V associated to the relation symbol r is the edge set, $r^V = E \subseteq V^2$, so that the formula xry will be true when there is an edge from x to y.

8.18 Example: Translate the following statements about directed graphs into formulas in the language of first-order directed graph theory.
a) The graph has exactly two vertices.
b) The graph has exactly three vertices.
c) The graph has exactly one edge.
d) There is an edge from the vertex x to every vertex but one.

Solution: Here are some possible translations.
a) $\exists x\, \exists y\, (\neg x \approx y \wedge \forall z\, (z \approx x \vee z \approx y))$
b) $\exists x\, \exists y\, \exists z\, \big(((\neg x \approx y \wedge \neg x \approx z) \wedge \neg y \approx z) \wedge \forall u\, ((u \approx x \vee u \approx y) \vee u \approx z)\big)$
c) $\exists x\, \exists y\, \big(xry \wedge \forall z\, \forall u\, (zru \to (z \approx x \wedge u \approx y))\big)$
d) The statement "there is an edge from x to every vertex but one, and there may or may not be an edge from x to that one vertex" can be translated as $\exists y\, \forall z\, (\neg z \approx y \to xrz)$, while the statement "there is an edge from x to every vertex but one, and there is no edge from x to that one vertex" can be translated as $\exists y\, \forall z\, (xrz \leftrightarrow \neg z \approx y)$.

8.19 Example: Let $F = \forall x\, \forall y\, (xry \to \exists z\ zrx)$ Determine which of the following two graphs make the formula F true.



Solution: We provide two solutions to this problem. The first method involves some understanding of the meaning of the formula. The formula can be translated as "every vertex which is the initial point of an edge, is also the final point of an edge". Using this translation, we can see by inspection that the graph (i) makes F false (when $x = 2$ and $y = 3$, xry is true but $\exists z\ zrx$ is false), and the graph (ii) makes F true (indeed for every value assigned to x, $\exists z\ zrx$ is true).

A more systematic method, requiring less understanding, involves making a table in which we list possible values which can be assigned to the variables in V, and then we list the truth-values for various subformulas of F. We do this for each graph:

For the graph (i)

x	y	xry	$\exists z\, zrx$	$(xry \to \exists z\, zrx)$
1	1	1	1	1
1	2	0	1	1
1	3	1	1	1
2	1	0	0	1
2	2	0	0	1
2	3	1	0	0
3	1	1	1	1
3	2	0	1	1
3	3	0	1	1

For the graph (ii)

x	y	xry	$\exists z\, zrx$	$(xry \to \exists z\, zrx)$
1	1	1	1	1
1	2	1	1	1
1	3	0	1	1
2	1	0	1	1
2	2	0	1	1
2	3	0	1	1
3	1	0	1	1
3	2	1	1	1
3	3	0	1	1

From the 6^{th} row in the first table, in which $x = 2$ and $y = 3$, we can see that F is false for the graph (i). On the other hand, since all the entries in the last column of the second table are equal to 1, we see that F is true for graph (ii).

8.20 Example: Let $F = \forall x\, (\forall y\ xry \to \exists z\ zrx)$ (compare this to the formula of the previous example). Show that F is true for *every* graph G (and hence in every interpretation).

Proof: Let V be any non-empty vertex set, and let $E \subseteq V^2$ be any edge set $\big[$we must show that F is true$\big]$. Let $x \in V$ be arbitrary $\big[$we must show that $(\forall y\ xry \to \exists z\ zrx)$ is true$\big]$. Suppose that the formula $\forall y\ xry$ is true $\big[$we must show that $\exists z\ zrx$ is true$\big]$. Then in particular, taking $y = x$, the formula xrx is true. Thus the formula $\exists z\ zrx$ is true, since we can take $z = x$. Since x was arbitrary, we have shown that F is true.

8.21 Remark: Mathematical Logic and Set Theory originated in about 1900. At that time, several **paradoxes** were discovered within mathematics. An example of a paradox outside the realm of mathematics is the well known self-referential statement "This statement is false". This amusing statement is true if and only if it is false. Such paradoxes are quite fun to play around with, and they pose no threat to the discipline of mathematics.

An example of a paradox within the realm of mathematics is the following famous paradox called **Russel's paradox**. Let X be the set of all sets, and let $S = \big\{A \in X \big| A \notin A\big\}$ (for example, since $\mathbf{Z} \notin \mathbf{Z}$ we have $\mathbf{Z} \in S$, and since $X \in X$ we have $X \notin S$). Then we have $S \in S \iff S \notin S$, so the statement $S \in S$ is true if and only if it is false.

A paradox in mathematics allows us to mathematically prove anything we wish, using a proof by contradiction. For example, we can use Russel's paradox to prove that $0 = 1$ as follows. Suppose, for a contradiction, that $0 \neq 1$. Let X be the set of all sets and let $S = \big\{A \in X \big| A \notin A\big\}$. Then we have $S \in S$ if and only if $S \notin S$, which gives us a contradiction. Thus $0 = 1$.

This is not a desirable state of affairs if one wants to be confident that statements which have been mathematically proven must be true.

One possible solution to this problem is to allow for the possibility that statements can be neither true nor false. If we make this allowance, then the above proof by contradiction would become an unacceptable proof. This solution weakens the power of mathematics by eliminating some of the acceptable methods of proof. This is not the solution which has been adopted.

The solution which has actually been adopted by mathematicians, is to make a precise set of rules by which mathematical sets can be produced, and by then defining all mathematical objects to be certain sets. These rules are the **axioms of set theory**. It is not possible to construct the set of all sets using the axioms of set theory, and so the collection of all sets is not a mathematical set, and it cannot be used in any mathematical proof as we used it above, in our proof that $0 = 1$.

8.22 Definition: The language of **first-order set theory** is the first-order language with the special symbol set

$$\{ \in \}$$

where $\in$ is a binary connective symbol which we write using infix notation.

8.23 Note: When we use the language of first-order set theory, the interpretation we have in mind is the one in which the universal class V is the class of all sets (as mentioned in the above remark, the collection of all sets is not as a mathematical set, but we can refer to it as the *class* of all sets), and the binary relation $\in$ is the relation for which $x \in y$ is true when x is an element of y.

8.24 Remark: Essentially all of mathematics can be done in the language of first-order set theory. The manner in which this is done is often presented in a course on set theory. We shall not do this here, but we shall list a few of the axioms of set theory, and we shall indicate how a few familiar mathematical objects can be defined to be certain sets.

8.25 Example: Translate each of the following statements about sets into formulas in first-order set theory.
a) $x = \emptyset$, that is, x is the empty set.
b) $z = x \cup y$.
c) $x \cap y \subseteq z$.

Solution: Here are some possible translations.
a) $\forall y \, \neg y \in x$
b) $\forall u \, (u \in z \ \leftrightarrow \ (u \in x \vee u \in y))$
c) $\forall u \, ((u \in x \wedge u \in y) \ \rightarrow \ u \in z)$

8.26 Example: The following statements are all axioms in set theory (the name of each axiom is given in brackets). Translate them to formulas in first-order set theory.
a) (Equality) Two sets are equal if and only if they have the same elements.
b) (Pair) If x and y are sets, then $\{x, y\}$ is a set.
c) (Union) If x and y are sets, then the union $x \cup y$ is a set.
d) (Power Set) If x is a set, then the set of all subsets of x is a set.
e) (Separation) Let F be a formula in first-order set theory with one free variable z. Then if x is a set, then $y = \{z \in x | F \text{ is true}\}$ is a set.
f) (Infinity) There exists a non-empty set x with the property that for every $y \in x$ there is a $z \in x$ with $z \neq y$ such that $y \subseteq z$.

Solution: Here are some possible translations.
a) $\forall x \, \forall y \, (x \approx y \ \leftrightarrow \ \forall z \, (z \in x \ \leftrightarrow \ z \in y))$
b) $\forall x \, \forall y \, \exists z \, \forall u \, (u \in z \ \leftrightarrow \ (u \approx x \vee u \approx y))$
c) $\forall x \, \forall y \, \exists z \, \forall u \, (u \in z \ \leftrightarrow \ (u \in x \vee u \in y))$
d) $\forall x \, \exists y \, \forall z \, (z \in y \ \leftrightarrow \ \forall u \, (u \in y \ \rightarrow \ u \in x))$
e) $\forall x \, \exists y \, \forall z \, (z \in y \ \leftrightarrow \ (z \in x \wedge F))$
f) $\exists x \, \big(\exists y \ y \in x \wedge \forall y \, (y \in x \ \rightarrow \ \exists z \, ((z \in x \wedge \neg z \approx y) \wedge \forall u \, (u \in y \ \rightarrow u \in z)))\big)$

8.27 Remark: There are a few other axioms, including the Replacement Axiom and the Axiom of Choice, and with these axioms it is possible to build up essentially all of mathematics in the language of first-order set theory. When this is done, all mathematical objects, including numbers and functions, are taken to be sets. Let us illustrate informally how one can begin to build up mathematics from these axioms.

To begin with, we can construct the **empty set**. To do this, choose any set x (a set exists by the Axiom of Infinity), then use the Axiom of Separation to form the set $y = \{z \in x | \neg z \approx z\}$, which is a set with no elements. The Axiom of Equality implies that there is only one set with no elements, so we can call it *the* empty set, and denote it by $\emptyset$. Using the empty set , we can construct the set of **natural numbers N** by defining $0 = \emptyset$, $1 = \{0\} = \{\emptyset\}$, $2 = \{0, 1\} = \{\emptyset, \{\emptyset\}\}$, $3 = \{0, 1, 2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$ and so on. With this definition of $\mathbf{N}$, the natural number x is a set of x elements, and we have $x+1 = x \cup \{x\}$.

Once the natural numbers have been defined, it is possible to define the set of **integers Z** by defining each integer to be an ordered pair in $\mathbf{N} \times 2 = \mathbf{N} \times \{0, 1\}$. For a natural number n we also write n for the ordered pair $n = (n, 0)$, and we write $-n$ for the ordered pair $-n = (n, 1)$. In order to perform this construction of $\mathbf{Z}$ from $\mathbf{N}$, we first need to define an **ordered pair**. Given sets x and y we can define the ordered pair (x, y) to be the set $(x, y) = \{\{x\}, \{x, y\}\}$ (this is either a set with two elements, when $x \neq y$, or it is a set with one element, when $x = y$). The statement $z = (x, y)$ can be translated into the first-order formula $\forall u \, \big(u \in z \ \leftrightarrow \ (\forall v \, (v \in u \ \leftrightarrow \ v \approx x) \vee \forall v \, (v \in u \ \leftrightarrow \ (v \approx x \vee v \approx y)))\big)$.

Once we have defined ordered pairs, we can then define the **product** of two sets by $x \times y = \{(u, v) | u \in x, v \in y\}$. To be more precise, writing $p(x)$ to denote the set of all subsets of x, we have $x \times y = \{w \in p(p(x \cup y)) | w = (u, v) \text{ for some } u \in x, v \in y\}$. Notice that the axioms Pair, Union, Power Set and Separation are all involved in the construction of the product $x \times y$. A **function** $f : x \to y$ is defined to be a subset $f \subseteq x \times y$ with the property that for every $u \in x$ there exists a unique $v \in y$ such that $(u, v) \in f$. In this definition, a function is a actually defined to be equal to what is normally called its graph. If the statement $f : x \to y$ were translated into a first-order formula, the resulting formula would be very long.

Chapter 9. Assignments

9.1 Definition: Let V be an interpretation. An **assignment in** V, or more precisely an **assignment of values in** V **to all the variable symbols**, is a map

$$\alpha : \{ \text{ variable symbols } \} \to V \, .$$

An **assignment in** V **on** $\{x_1, \cdots, x_k\}$ is a map $\alpha : \{x_1, \cdots, x_k\} \to V$.

Given an assignment α in V, a variable symbol x, and an element $v \in V$, let $[\alpha]_{x \mapsto v}$ denote the assignment in V which is defined by

$$[\alpha]_{x \mapsto v}(y) = \begin{cases} \alpha(y) \text{ if } y \neq x, \text{ and} \\ v \qquad \text{if } y = x, \end{cases}$$

where y is any variable symbol.

An assignment α in V induces an **assignment of values in** V **to all terms**, that is a map

$$\alpha : \{ \text{ terms } \} \to V$$

as follows: given a term t, we define $\alpha(t) \in V$ inductively by

1. $\alpha(x)$ is already known when x is any variable symbol,
2. $\alpha(c) = c^V$ for any constant symbol, and
3. $\alpha(f t_1 \cdots t_n) = f^V(\alpha(t_1), \cdots, \alpha(t_n))$, where f is n-ary and $t_1, \cdots t_n$ are terms.

Also, the assignment α now induces an **assignment of truth-values to all formulas**, that is a map,

$$\alpha : \{ \text{ formulas } \} \to \{1, 0\} \, ,$$

as follows: given a formula F, we define the **truth-value** $\alpha(F) \in \{1, 0\}$ inductively by

1. $\alpha(t_1 \approx t_2) = \begin{cases} 1 \text{ if } \alpha(t_1) = \alpha(t_2) \text{ in } V, \text{ and} \\ 0 \text{ if } \alpha(t_1) \neq \alpha(t_2) \text{ in } V \end{cases}$
2. $\alpha(r t_1 \cdots t_n) = \begin{cases} 1 \text{ if } (\alpha(t_1), \cdots, \alpha(t_n)) \in r^V \subseteq V^n \text{ , and} \\ 0 \text{ if } (\alpha(t_1), \cdots, \alpha(t_n)) \notin r^V \subseteq V^n \end{cases}$
3. $\alpha(\neg G) \quad = \begin{cases} 1 \text{ if } \alpha(G) = 0 \text{ , and} \\ 0 \text{ if } \alpha(G) = 1 \end{cases}$, as in propositional logic,
4. $\alpha((G \times H))$ is defined as in propositional logic, when $\times \in \{\wedge, \vee, \to, \leftrightarrow\}$, and
5. $\alpha(\forall x \, G) = \begin{cases} 1 \text{ if } [\alpha]_{x \mapsto v}(G) = 1 \text{ for every } v \in V, \text{ and} \\ 0 \text{ if } [\alpha]_{x \mapsto v}(G) = 0 \text{ for some } v \in V \end{cases}$

   $\alpha(\exists x \, G) = \begin{cases} 1 \text{ if } [\alpha]_{x \mapsto v}(G) = 1 \text{ for some } v \in V, \text{ and} \\ 0 \text{ if } [\alpha]_{x \mapsto v}(G) = 0 \text{ for every } v \in V. \end{cases}$

For an interpretation V, an assignment α in V, and a formula F, if $\alpha(F) = 1$ then we say that F is **true** in the interpretation V under the assignment α, and if $\alpha(F) = 0$ then we say that F is **false** in the interpretation V under the assignment α. When F is true in V under every assignment α, we say that F is **true in** V, and when F is false in V under every assignment α, we say that F is **false in** V.

9.2 Example: Let a be a constant symbol, let f be a binary function symbol, and let r be a binary relation symbol. Let F be the formula $F = \forall y\,(fxy \approx a \to \exists x\, rxy)$. Let V be the interpretation given by $V = \{2, 3\}$, $a^V = 2$, $f^V(u, v) = |u - v| + 2$ and $r^V = \{(3, 2)\}$. Let α be an assignment with $\alpha(x) = 2$ and $\alpha(y) = 3$. Determine whether $\alpha(F) = 1$.

Solution: First we note that F has one free variable, namely x, so the truth-value of F will only depend on $\alpha(x)$; the fact that $\alpha(y) = 3$ is irrelevant. Now let us painstakingly wade our way through the definition step by step. Write $G = (fxy \approx a \to \exists x\, rxy)$ so $F = \forall y\, G$. We have $\alpha(F) = 1 \iff \alpha(\forall y\, G) = 1 \iff [\alpha]_{y\mapsto v}(G) = 1$ for every $v \in V$. There are two possible values for $v \in V$, so we consider two cases.

Case 1: suppose that $v = 2$. Write $\beta = [\alpha]_{y\mapsto v}$ so we have $\beta(x) = \alpha(x) = 2$ and $\beta(y) = v = 2$. We wish to find $\beta(G) = \beta(fxy \approx a \to \exists x\, rxy)$, so we shall find $\beta(fxy \approx a)$ and $\beta(\exists x\, rxy)$. We have $\beta(fxy \approx a) = 1 \iff \beta(fxy) = \beta(a) \iff f^V(\beta(x), \beta(y)) = a^V \iff f^V(2, 2) = 2$. Since $f^V(2, 2) = |2 - 2| + 2 = 2$, we have $\beta(fxy \approx a) = 1$. Next, let us find $\beta(\exists x\, rxy)$. We have $\beta(\exists x\, rxy) = 1 \iff [\beta]_{x\mapsto u}(rxy) = 1$ for some $u \in V$.

We consider each of the two possible values for $u \in V$. Case 1(a): suppose that $u = 2$ and write $\gamma = [\beta]_{x\mapsto u}$ so that $\gamma(x) = u = 2$ and $\gamma(y) = \beta(y) = 2$. We have $[\beta]_{x\mapsto u}(rxy) = 1 \iff \gamma(rxy) = 1 \iff (\gamma(x), \gamma(y)) \in r^V \iff (2, 2) \in r^V$. Since $(2, 2) \notin r^V$ we have $[\beta]_{x\mapsto u}(rxy) = 0$. Case 1(b): suppose that $u = 3$ and write $\gamma = [\beta]_{x\mapsto u}$ so that $\gamma(x) = u = 3$ and $\gamma(y) = \beta(y) = 2$. Then $[\beta]_{x\mapsto u}(rxy) = 1 \iff \gamma(rxy) = 1 \iff (\gamma(x), \gamma(y)) \in r^V \iff (3, 2) \in r^V$. Since $(3, 2) \in r^V$, we have $[\beta]_{x\mapsto u}(rxy) = 1$.

By the result of case 1(b), we see that $[\beta]_{x\mapsto u}(rxy) = 1$ for some $u \in V$ (namely $u = 3$) and hence $\beta(\exists x\, rxy) = 1$. Since $\beta(fxy \approx a) = 1$ and $\beta(\exists x\, rxy) = 1$, we have $\beta(fxy \approx a \to \exists x\, rxy) = 1$. Thus when $v = 2$ we have $[\alpha]_{y\mapsto v}(G) = \beta(G) = 1$.

Case 2: suppose that $v = 3$. Write $\beta = [\alpha]_{y\mapsto v}$ so we have $\beta(x) = \alpha(x) = 2$ and $\beta(y) = v = 3$. Then $\beta(fxy \approx a) = 1 \iff f^V(\beta(x), \beta(y)) = a^V \iff f^V(2, 3) = 2$. But $f^V(2, 3) = |2 - 3| + 2 = 3 \neq 2$ and so $\beta(fxy \approx a) = 0$. Since $\beta(fxy \approx a) = 0$ we have $[\alpha]_{y\mapsto v}(G) = \beta(fxy \approx a \to \exists x\, rxy) = 1$; it is unnecessary to find $\beta(\exists x\, rxy)$.

Merely for the sake of extra practice, let us find $\beta(\exists x\, rxy)$. We have $\beta(\exists x\, rxy) = 1 \iff [\beta]_{x\mapsto u}(rxy) = 1$ for some $u \in V$. Case 2(a): suppose that $u = 2$ and write $\gamma = [\beta]_{x\mapsto u}$ so that $\gamma(x) = u = 2$ and $\gamma(y) = \beta(y) = 3$. Then $[\beta]_{x\mapsto u}(rxy) = 1 \iff \gamma(rxy) = 1 \iff (\gamma(x), \gamma(y)) \in r^V \iff (2, 3) \in r^V$. Since $(2, 3) \notin r^V$ we have $[\beta]_{x\mapsto u}(rxy) = 0$. Case 2(b): suppose that $u = 3$ and write $\gamma = [\beta]_{x\mapsto u}$ so that $\gamma(x) = u = 3$ and $\gamma(y) = \beta(y) = 3$. Then $[\beta]_{x\mapsto u}(rxy) = 1 \iff \gamma(rxy) = 1 \iff (\gamma(x), \gamma(y)) \in r^V \iff (3, 3) \in r^V$. Since $(3, 3) \notin r^V$, we have $[\beta]_{x\mapsto u}(rxy) = 0$. For both possible values of u we have found that $[\beta]_{x\mapsto u}(rxy) = 0$, and so $\beta(\exists x\, rxy) = 0$.

From the results of cases 1 and 2, we see that $[\alpha]_{y\mapsto v}(G) = 1$ for every $v \in V$. Thus $\alpha(F) = \alpha(\forall y\, G) = 1$.

We summarize the results of our case-by-case analysis in the following table of values and truth-values.

x	y	fxy	$fxy \approx a$	rxy	$\exists x\, rxy$	G	F
2	2	2	1	0	1	1	1
2	3	3	0	0	0	1	1
3	2	3	0	1	1	1	0
3	3	2	1	0	0	0	0

9.3 Definition: Let $V = \{v_1, \cdots, v_n\}$ be a finite interpretation. A **table of values and truth-values on** $\{x_1, \cdots, x_k\}$ **for** V is a table with the following properties.

(1) The leading header row is of the form $x_1\ x_2\ \cdots\ x_k\ t_1\ t_2\ \cdots\ t_l\ F_1\ F_2\ \cdots\ F_m$ where $x_1, \cdots, x_k, t_1, \cdots, t_l$ is a derivation of terms (using only variable symbols from $\{x_1, \cdots, x_k\}$), and $F_1, \cdots, F_m$ is a derivation of formulas (using only the terms previously derived).

(2) There are n^k rows (not counting the header row); for each of the n^k assignments α in V on $\{x_1, \cdots, x_k\}$, there is a row of the form $\alpha(x_1) \cdots \alpha(x_n)\ \alpha(t_1) \cdots \alpha(t_l)\ \alpha(F_1) \cdots \alpha(F_m)$. Note that each $\alpha(x_i) \in V$ and each $\alpha(t_i) \in V$ but each $\alpha(F_i) \in \{1, 0\}$.

(3) The rows are ordered so that in the first k columns (headed by $x_1, \cdots, x_k$), the rows list all the elements in V^k in lexicographical order.

9.4 Example: Let a be a constant symbol, let f be a unary function symbol, let g be a binary function symbol and let r be a binary relation symbol. Let V be the interpretation given by $V = \{1, 2, 3\}$ with $a^V = 2$ and with f^V, g^V and r^V determined by the following tables of values and truth-values on $\{x, y\}$.

x	fx
1	3
2	3
3	2

x	y	gxy
1	1	3
1	2	1
1	3	2
2	1	3
2	2	3
2	3	1
3	1	1
3	2	2
3	3	1

x	y	rxy
1	1	0
1	2	1
1	3	0
2	1	0
2	2	1
2	3	1
3	1	0
3	2	1
3	3	0

Let $F = \exists x \left(\exists y\, gfxy \approx a \wedge \forall x \left(rxfy \rightarrow x \approx y\right)\right)$. Find all the assignments α on $\{x, y\}$ such that $\alpha(F) = 1$.

Solution: We write $G = \exists y\, gfxy \approx a$, $H = \forall x \left(rxfy \rightarrow x \approx y\right)$ so that $F = \exists x \left(G \wedge H\right)$, and then we make a table of values and truth-values for the formula F.

x	y	a	fx	fy	$gfxy$	$gfxy \approx a$	G	$rxfy$	$x \approx y$	$rxfy \rightarrow x \approx y$	H	$G \wedge H$	F
1	1	2	3	3	1	0	1	0	1	1	0	0	0
1	2	2	3	3	2	1	1	0	0	1	1	1	1
1	3	2	3	2	1	0	1	1	0	0	0	0	0
2	1	2	3	3	1	0	1	1	0	0	0	0	0
2	2	2	3	3	2	1	1	1	1	1	1	1	1
2	3	2	3	2	1	0	1	1	0	0	0	0	0
3	1	2	2	3	3	0	0	0	0	1	0	0	0
3	2	2	2	3	3	0	0	0	0	1	1	0	1
3	3	2	2	2	1	0	0	1	1	1	0	0	0

Notice that the only free variable in G is x, and so the truth-values of G only depend on $\alpha(x)$. On the other hand, in H and in F the only free variable is y, so the truth-values of H and F only depend on $\alpha(y)$. From the final column, we see that there are three assignments on $\{x, y\}$ such that $\alpha(F) = 1$. They are the assignments α with $\alpha(y) = 2$ and $\alpha(x)$ arbitrary.

9.5 Definition: Let F and G be formulas, and let $\mathcal{S}$ be a set of formulas.

We say that F is a **tautology**, and we write $\models F$, when for every interpretation V and every assignment α in V we have $\alpha(F) = 1$.

We say that F and G are **truth-equivalent**, and we write F treq G, when for every interpretation V and every assignment α in V we have $\alpha(F) = \alpha(G)$.

We say that $\mathcal{S}$ is **satisfiable** when there exists an interpretation V and an assignment α in V such that $\alpha(F) = 1$ for every $F \in \mathcal{S}$. Such an assignment α is said to **satisfy** $\mathcal{S}$. When $\mathcal{S} = \{F\}$, we omit the set-brackets and say that F is satisfiable.

We say that the argument "$\mathcal{S}$ therefore G" is **valid**, and we write $\mathcal{S} \models G$, when for every interpretation V and every assignment α in V, if $\alpha(F) = 1$ for every $F \in \mathcal{S}$ then $\alpha(G) = 1$. When $\mathcal{S}$ is finite we shall often omit the set-brackets. The formulas in $\mathcal{S}$ are called the **premises** of the argument, and the formula G is called the **conclusion** of the argument.

9.6 Example: For any term t, the formula $x \approx t$ is satisfiable, and the formula $t \approx t$ is a tautology. If f is a unary function symbol, then the formula $y \approx fx$ is satisfiable, and the formula $\exists y\, y \approx fx$ is a tautology. For any terms s and t, we have $s \approx t$ treq $t \approx s$. For any formula F, we have $\forall x \forall y\, F$ treq $\forall y \forall x\, F$, and also $\exists x \exists y\ F$ treq $\exists y \exists x\, F$. For any terms t_1, t_2 and t_3 we have $\{t_1 \approx t_2, t_2 \approx t_3\} \models t_1 \approx t_3$.

9.7 Example: Let 0 and 1 be constant symbols. Show that the formula $0 \approx 1$ is satisfiable.

Solution: Although in everyday mathematics we would normally use the symbols 0 and 1 to denote two distinct objects, it is very easy to invent an interpretation in which they denote the same object. For example, we could take $V = \{2, 3\}$ and we could set $0^V = 1^V = 2$.

9.8 Example: Show that the formula $x + 0 \approx x$, in first-order number theory, is not a tautology.

Solution: Although in the interpretations we would normally consider in number theory, the formula $x + 0 \approx x$ is true, no matter which value is assigned to x, it is easy to make up an interpretation and an assignment under which the formula is not true. For example, we could take $V = \mathbf{Z}$, the set of integers, and let $+^V$ be the usual addition function, and then we could take 0^V to be equal to the integer 1 instead of the integer 0. Then under any assignment α in V, the formula $x + 0 \approx x$ is false.

9.9 Theorem: *(Relationships Between Tautologies, Truth-Equivalence, Satisfiability and Validity) Let F, G and F_i be formulas, and let $\mathcal{S}$ be a set of formulas. Write $(F_1 \wedge \cdots \wedge F_n)$ for $(\cdots((F_1 \wedge F_2) \wedge F_3) \wedge \cdots \wedge F_n)$. Then*

(1) $\models F \iff \emptyset \models F \iff \neg F$ *is unsatisfiable.*

(2) $F \models G \iff \models F \to G \iff F \wedge \neg G$ *is unsatisfiable.*

(3) F treq $G \iff \models (F \leftrightarrow G) \iff \big(F \models G$ *and* $G \models F\big)$.

(4) $\{F_1, \cdots, F_n\} \models G \iff (F_1 \wedge \cdots \wedge F_n) \models G \iff \models (F_1 \wedge \cdots \wedge F_n) \to G$
$\iff (F_1 \wedge \cdots \wedge F_n) \wedge \neg G$ *is unsatisfiable* $\iff \{F_1, \cdots, F_n, \neg G\}$ *is unsatisfiable.*

(5) $\mathcal{S} \models G \iff \mathcal{S} \cup \{\neg G\}$ *is not satisfiable.*

Proof: The proof is similar to the proof of the analogous theorem from propositional logic (theorem 2.15). The proof is left as an exercise.

9.10 Example: Let $F = \forall x \exists y\,(\neg y \approx fx \to ryfx)$, where f is a unary function symbol and r is a binary relation symbol. Show that F is a tautology.

Solution: Informally, but inaccurately, we can think of x and y as being elements in some universal set V, and think of f as an actual unary function, and argue as follows: given any $x \in V$ we can choose $y = fx \in V$ and then the formula $y \approx fx$ will be true and so the formula $\neg y \approx fx \to ryfx$ will also be true.

The above argument is inaccurate because it is not x and y which are elements of V, but rather it is $\alpha(x)$ and $\alpha(y)$, where α is an assignment, and furthermore, we cannot choose to have $y = fx$ because y is a string of length 1 while fx is a string of length 2. We now provide an accurate version of the above argument.

Let V be an interpretation and let α be an assignment in V. We must show that $\alpha\big(\forall x \exists y\,(\neg y \approx fx \to ryfx)\big) = 1$. Let $u \in V$ be arbitrary, and let $\beta = [\alpha]_{x \mapsto u}$. We must show that $\beta\big(\exists y\,(\neg y \approx fx \to ryfx)\big) = 1$. Choose $v = f^V(u) \in V$ and let $\gamma = [\beta]_{y \mapsto v}$. We must show that $\gamma(\neg y \approx fx \to ryfx) = 1$. Since $\gamma(y) = v = f^V(u) = f^V(\gamma(x)) = \gamma(fx)$, we have $\gamma(y \approx fx) = 1$. Since $\gamma(y \approx fx) = 1$, we also have $\gamma(\neg y \approx fx \to ryfx) = 1$, as required.

9.11 Example: Let $F = \forall x\,(\exists y\,\neg rxy \vee \exists y\,ryx)$, where r is a binary relation symbol. Show that F is a tautology.

Solution: First, we think of x and y as being elements of a universal set V, and think of r as a binary relation, and argue informally, but inaccurately, as follows. Let $x \in V$ be arbitrary. Suppose that $\exists y\,\neg rxy$ is false. Then $\forall y\,rxy$ must be true. In particular, taking $y = x$, we have rxx true and so $\exists y\,ryx$ is true. We now translate this into a rigorous argument.

Let V be an interpretation and let α be an assignment in V. We must show that $\alpha\big(\forall x\,(\exists y\,\neg rxy \vee \exists y\,ryx)\big) = 1$. Let $u \in V$ be arbitrary and let $\beta = [\alpha]_{x \mapsto u}$. We must show that $\beta(\exists y\,\neg rxy \vee \exists y\,ryx) = 1$. Suppose that $\beta(\exists y\,\neg rxy) = 0$. Then for every $v \in V$ we have $[\beta]_{y \mapsto v}(\neg rxy) = 0$ and hence $[\beta]_{y \mapsto v}(rxy) = 1$. In particular, we have $[\beta]_{y \mapsto u}(rxy) = 1$. Write $\gamma = [\beta]_{y \mapsto u}$. Then we have $\gamma(x) = \gamma(y) = u$ and we have $\gamma(rxy) = 1$, which means that $\big(\gamma(x), \gamma(y)\big) \in r^V$. Also, $\big(\gamma(y), \gamma(x)\big) = (u, u) = \big(\gamma(x), \gamma(y)\big) \in r^V$, and so $\gamma(ryx) = 1$, that is $[\beta]_{y \mapsto u}(ryx) = 1$. Under the assumption that $\beta(\exists y\,\neg rxy) = 0$ we have shown that $\beta(\exists y\,ryx) = 1$ and this proves that $\beta(\exists y\,\neg rxy \vee \exists y\,ryx) = 1$, as required.

9.12 Example: Let $F = \forall x\,(\exists y\,rxy \to \forall y\,\neg fx \approx y)$, where f is a unary function symbol and r is a binary relation symbol. Show that F is satisfiable.

Solution: Informally, but inaccurately, we think of x and y as elements of a set V, and we think of f as a unary function and r as a binary relation. Note that the formula $\forall y\,\neg fx \approx y$ is false, since given any $x \in V$ we can choose $y = fx \in V$. So the only way to make the formula $\exists y\,rxy \to \forall y\,\neg fx \approx y$ true is to make the formula $\exists y\,rxy$ false. Thus for all $x \in V$ and all $y \in V$ we need to have rxy false, and to do this, the relation r must be the empty relation.

To make this precise, let V be any non-empty set, let $f^V : V \to V$ be any unary function, let $r^V = \emptyset$, the empty set, and let α be any assignment in V. We claim that $\alpha(F) = 1$. Let $u \in V$ be arbitrary, and let $\beta = [\alpha]_{x \mapsto u}$. We need to show that $\beta(\exists y\,rxy \to \forall y\,\neg fx \approx y) = 1$. To do this, we shall show that $\beta(\exists y\,rxy) = 0$. Let $v \in V$ be arbitrary, and let $\gamma = [\beta]_{y \mapsto v}$. We must show that $\gamma(rxy) = 0$. Since $r^V = \emptyset$, we have $\big(\gamma(x), \gamma(y)\big) = (u, v) \notin r^V$, and so $\gamma(rxy) = 0$, as required.

9.13 Example: Let F and G be formulas. Show that $\forall x\,(F \wedge G)$ treq $\forall x\,F \wedge \forall x\,G$.

Solution: Let V be an interpretation and let α be an assignment in V.

Suppose first that $\alpha\big(\forall x\,(F \wedge G)\big) = 1$. We claim that $\alpha(\forall x\,F \wedge \forall x\,G) = 1$. Let $u \in V$ be arbitrary. Since $\alpha\big(\forall x(F \wedge G)\big) = 1$, we have $[\alpha]_{x\mapsto u}(F \wedge G) = 1$, and so $[\alpha]_{x\mapsto u}(F) = 1$ and $[\alpha]_{x\mapsto u}(G) = 1$. Since u was arbitrary and $[\alpha]_{x\mapsto u}(F) = 1$, we have $\alpha(\forall x\,F) = 1$. Since u was arbitrary and $[\alpha]_{x\mapsto u}(G) = 1$, we have $\alpha(\forall x\,G) = 1$. Since $\alpha(\forall x\,F) = 1$ and $\alpha(\forall x\,G) = 1$, we have $\alpha(\forall x\,F \wedge \forall x\,G) = 1$, as claimed. Thus $\forall x\,(F \wedge G) \models \forall x\,F \wedge \forall x\,G$.

Next suppose that $\alpha(\forall x\,F \wedge \forall x\,G) = 1$, so that $\alpha(\forall x\,F) = 1$ and $\alpha(\forall x\,G) = 1$. We claim that $\alpha\big(\forall x\,(F \wedge G)\big) = 1$. Let $u \in V$ be arbitrary. Since $\alpha(\forall x\,F) = 1$ we have $[\alpha]_{x\mapsto u}(F) = 1$, and since $\alpha(\forall x\,G) = 1$ we have $[\alpha]_{x\mapsto u}(G) = 1$. Since $[\alpha]_{x\mapsto u}(F) = 1$ and $[\alpha]_{x\mapsto u}(G) = 1$, we have $[\alpha]_{x\mapsto u}(F \wedge G) = 1$, and since u was arbitrary, this implies that $\alpha\big(\forall x(F \wedge G)\big) = 1$, as claimed. Thus $\forall x\,F \wedge \forall x\,G \models \forall x\,(F \wedge G)$.

9.14 Theorem: *Let V be an interpretation, let α and β be assignments in V, and let F be a formula. If $\alpha(x) = \beta(x)$ for every variable symbol x which has a free occurrence in F, then $\alpha(F) = \beta(F)$. In particular, if x is not free in F then for any $u \in V$ we have $[\alpha]_{x\mapsto u}(F) = \alpha(F)$.*

Proof: We omit the proof, which uses induction on terms and induction on formulas.

9.15 Example: Let F and G be formulas, and suppose that x is not free in G. Show that $\forall x\,(F \vee G)$ treq $\forall x\,F \vee G$.

Solution: Let V be an interpretation and let α be an assignment in V. Suppose that $\alpha\big(\forall x\,(F \vee G)\big) = 1$. We claim that $\alpha(\forall x\,F \vee G) = 1$. Suppose that $\alpha(G) = 0$. We need to show that $\alpha(\forall x\,F) = 1$. Let $u \in V$ be arbitrary, and let $\beta = [\alpha]_{x\mapsto u}$. We must show that $\beta(F) = 1$. Since $\alpha\big(\forall x\,(F \vee G)\big) = 1$, we have $\beta(F \vee G) = 1$. Since x is not free in G, we have $\beta(G) = \alpha(G) = 0$ by the above theorem. Since $\beta(G) = 0$ and $\beta(F \vee G) = 1$ we have $\beta(F) = 1$, as required. Thus we have $\forall x\,(F \vee G) \models \forall x\,F \vee G$.

Conversely, suppose that $\alpha(\forall x\,F \vee G) = 1$. We claim that $\alpha\big(\forall x\,(F \vee G)\big) = 1$. Let $u \in V$ be arbitrary, and let $\beta = [\alpha]_{x\mapsto u}$. We need to show that $\beta(F \vee G) = 1$, so we suppose that $\beta(G) = 0$, and we claim that $\beta(F) = 1$. Since x is not free in G, we have $\alpha(G) = \beta(G) = 0$, by the above theorem. Since $\alpha(\forall x\,F \vee G) = 1$ and $\alpha(G) = 0$ we have $\alpha(\forall x\,F) = 1$, and so $[\alpha]_{x\mapsto u}(F) = 1$, that is $\beta(F) = 1$, as required. Thus we have $\forall x\,F \vee G \models \forall x\,(F \vee G)$.

9.16 Example: Find formulas F and G such that $\forall x\,(F \vee G) \not\models \forall x\,F \vee G$, and find formulas F and G such that $\forall x\,F \vee G \not\models \forall x\,(F \vee G)$.

Solution: By the previous example, we know that to obtain examples of such formulas F and G we must have x free in G.

First, let $F = x \approx y$ and $G = \neg x \approx y$, let V be any interpretation with at least 2 elements, and let α be any assignment in V with $\alpha(x) = \alpha(y) \in V$. Then $\alpha\big(\forall x\,(F \vee G)\big) = 1$, indeed $\forall x\,(F \vee G)$ is a tautology. Also $\alpha(\forall x\,F) = 0$ since V has at least 2 elements, and $\alpha(G) = 0$ since $\alpha(x) = \alpha(y) \in V$, and so $\alpha(\forall x\,F \vee G) = 0$. Thus for these formulas F and G we have $\forall x\,(F \vee G) \not\models \forall x\,F \vee G$.

Now let $F = G = x \approx y$, let V be any interpretation with at least 2 elements, and let α be any assignment in V with $\alpha(x) = \alpha(y) \in V$. Then $\alpha(G) = 1$ since $\alpha(x) = \alpha(y) \in V$, so $\alpha(\forall x\,F \vee G) = 1$, but $\alpha\big(\forall x(F \vee G)\big) = 0$ since V has at least 2 elements. Thus for these formulas F and G we have $\forall x\,F \vee G \not\models \forall x\,(F \vee G)$.

Chapter 10. Derivation of Valid Arguments

10.1 Theorem: *(Basic Truth-Equivalences) For any formulas F and G, and any variable symbols x and y, in addition to the 24 Basic Truth-Equivalences from propositional logic, we have the following* **Basic Truth-Equivalences**.

(Double Quantifier)	*25.*	$\forall x\,\forall y\,F$ treq $\forall y\,\forall x\,F$
	26.	$\exists x\,\exists y\,F$ treq $\exists y\,\exists x\,F$
(Negating Quantifier)	*27.*	$\neg\forall x\,F$ treq $\exists x\neg F$
	28.	$\neg\exists x\,F$ treq $\forall x\,\neg F$
(Separating Quantifier)	*29.*	$\forall x\,(F\wedge G)$ treq $\forall x\,F\wedge\forall x\,G$
	30.	$\exists x\,(F\vee G)$ treq $\exists x\,F\vee\exists x\,G$
	31.	$\exists x\,(F\to G)$ treq $\forall x\,F\to\exists x\,G$
(Quantifying Unused Variable)	*32.*	$\forall x\,F$ treq F *if* x *is not free in* F
	33.	$\exists x\,F$ treq F *if* x *is not free in* F
(Changing Variables)	*34.*	$\forall x\,F$ treq $\forall y\,[F]_{x\mapsto y}$ *if* y *is not free in* F
	35.	$\exists x\,F$ treq $\exists y\,[F]_{x\mapsto y}$ *if* y *is not free in* F

Proof: The first Separating Quantifier rule was proven in Example 9.13. The rest of the proof is left as an exercise.

10.2 Theorem: *For any formulas F, G and H, for any set of formulas $\mathcal{S}$, for any variable symbols x and y, and for any terms s, t, s_i and t_i, in addition to the 36 Basic Validity Rules from propositional logic, we have the following* **Basic Validity Rules**.

(Equality)	*37.*	$\mathcal{S}\models t\approx t$
	38.	*If* $\mathcal{S}\models s\approx t$ *then* $\mathcal{S}\models t\approx s$
	39.	*If* $\mathcal{S}\models t_1\approx t_2$ *and* $\mathcal{S}\models t_2\approx t_3$ *then* $\mathcal{S}\models t_1\approx t_3$
	40.	*If* $S\models s\approx t$ *and* $\mathcal{S}\models[F]_{x\mapsto s}$ *then* $\mathcal{S}\models[F]_{x\mapsto t}$
(Universality)	*41.*	*If* $S\models[F]_{x\mapsto y}$ *where* y *is not free in* $\mathcal{S}\cup\{\forall x\,F\}$*, then* $\mathcal{S}\models\forall x\,F$
	42.	*If* $\mathcal{S}\models\forall x\,F$ *then* $\mathcal{S}\models[F]_{x\mapsto t}$
	43.	*If* $\mathcal{S}\cup\{[F]_{x\mapsto t}\}\models G$ *then* $\mathcal{S}\cup\{\forall x\,F\}\models G$
(Existentiality)	*44.*	*If* $\mathcal{S}\models[F]_{x\mapsto t}$ *then* $\mathcal{S}\models\exists x\,F$
	45.	*If* $\mathcal{S}\cup\{[F]_{x\mapsto y}\}\models G$ *where* y *is not free in* $\mathcal{S}\cup\{G,\exists x\,F\}$*, then* $\mathcal{S}\cup\{\exists x\,F\}\models G$
	46.	*If* $\mathcal{S}\cup\{\exists x\,F\}\models G$ *then* $\mathcal{S}\cup\{[F]_{x\mapsto t}\}\models G$

Proof: The proof is left as an exercise.

10.3 Definition: Let $\mathcal{S}$ be a set of formulas and let F be a formula. Suppose that $\mathcal{S}\models F$. A **derivation** of the valid argument $\mathcal{S}\models F$ is a finite list of valid arguments $\mathcal{S}_1\models F_1,\mathcal{S}_2\models F_2,\cdots,\mathcal{S}_l\models F_l$ with each $\mathcal{S}_k$ finite, such that $F_l=F$, $\mathcal{S}_l\subseteq\mathcal{S}$ and each valid argument $\mathcal{S}_k\models F_k$ is obtained from zero or more previous valid arguments $\mathcal{S}_i\models F_i$ with $i<k$ using one of the Basic Validity Rules; we shall only use the Truth-Equivalence Rule in the case that the truth-equivalence is obtained by substitution from one of the Basic Truth-Equivalences, and we shall only use the Tautology Rule in the case that the tautology is of the form $F\to F$.

10.4 Example: Make a derivation to show that $\models \forall x\, \exists y\, (\neg y \approx fx \to ryfx)$ where f is unary and r is binary (see example 9.10).

Solution:

1.	$\models fx \approx fx$	Basic Validity Rule 37
2.	$\models \neg\neg fx \approx fx$	Double Negation on line 1
3.	$\models \neg fx \approx fx \to rfxfx$	Implication on line 2
4.	$\models \exists y\, (\neg y \approx fx \to ryfx)$	Basic Validity Rule 44 on line 3
5.	$\models \forall x\, \exists y\, (\neg y \approx fx \to ryfx)$	Basic Validity Rule 41 on line 4

10.5 Example: Make a derivation to show that $\models \forall x\, (\exists y\, \neg rxy \vee \exists y\, ryx)$ where r is binary (see example 9.11).

Solution:

1.	$\neg\exists y\, \neg rxy \models \neg\exists y\, \neg rxy$	Premise
2.	$\neg\exists y\, \neg rxy \models \forall y\, \neg\neg rxy$	Basic Truth-Equivalence 28 on line 1
3.	$\neg\exists y\, \neg rxy \models \forall y\, rxy$	Double-Negation on line 2
4.	$\neg\exists y\, \neg rxy \models rxx$	Basic Validity Rule 42 on line 3
5.	$\neg\exists y\, \neg rxy \models \exists y\, ryx$	Basic Validity Rule 44 on line 4
6.	$\models \exists y\, \neg rxy \vee \exists y\, ryx$	Disjunction on line 5
7.	$\models \forall x\, (\exists y\, \neg rxy \vee \exists y\, ryx)$	Basic Validity Rule 41 on line 6

10.6 Example: Let F and G be formulas, and suppose that x is not free in G. Show that $\forall x\, (F \vee G)$ treq $\forall x\, F \vee G$ (see example 9.15).

Solution: We shall make two derivations; one to show that $\forall x\, (F \vee G) \models \forall x\, F \vee G$ and the other to show that $\forall x\, F \vee G \models \forall x\, (F \vee G)$. We have

1.	$\forall x\, (F \vee G) \models \forall x\, (F \vee G)$	Premise
2.	$\forall x\, (F \vee G) \models F \vee G$	Basic Validity Rule 42 on 1
3.	$\forall x\, (F \vee G), \neg G \models F$	Disjunction on 2
4.	$\forall x\, (F \vee G), \neg G \models \forall x\, F$	Basic Validity Rule 41 on 3, (x is not free in $\neg G$)
5.	$\forall x\, (F \vee G) \models \forall x\, F \vee G$	Disjunction on 4

and we have

1.	$\forall x\, F \vee G \models \forall x\, F \vee G$	Premise
2.	$\forall x\, F \vee G\, , \neg G \models \forall x\, F$	Disjunction on 1
3.	$\forall x\, F \vee G\, , \neg G \models F$	Basic Validity Rule 42 on 2
4.	$\forall x\, F \vee G \models F \vee G$	Disjunction on 3
5.	$\forall x\, F \vee G \models \forall x\, (F \vee G)$	Basic Validity Rule 41 on 4 (x is not free in $\forall x\, F \vee G$)

10.7 Definition: A **group** is a set G with an element $1 \in G$, called the **identity** element, and a binary operation $\times$ such that

1. for all $x, y, z \in G$ we have $(x \times y) \times z = x \times (y \times z)$,
2. for all $x \in G$ we have $x \times 1 = 1 \times x = x$, and
3. for all $x \in G$ there exists $y \in G$ such that $x \times y = y \times x = 1$.

The above three requirements are called the **axioms** of group theory. Note that the axioms can be expressed as first order formulas.

10.8 Example: One of the first results that we can prove about groups is that the identity element of a group G is unique in the sense that for all $u \in G$, if u has the property that $x \times u = u \times x = x$ for all $x \in G$, then we must have $u = 1$. We can prove this as follows. Let $u \in G$. Suppose that for all $x \in G$ we have $x \times u = u \times x = x$. Since $u \times x = x$ for all x, by taking $x = 1$ we obtain $u \times 1 = 1$. Since $x \times 1 = x$ for all x, by taking $x = u$ we obtain $u \times 1 = u$. Since $u \times 1 = u$ and $u \times 1 = 1$ we must have $u = 1$, as required.

Formalize the above proof by making a derivation of valid arguments to show that

$$\Big\{\forall x\,(x\times 1\approx x\ \wedge\ 1\times x\approx x)\Big\}\models\forall u\big(\forall x\,(x\times u\approx x\ \wedge\ u\times x\approx x)\to u\approx 1\big)\,.$$

Solution: Here is a derivation which formalizes the above proof.

1. $\forall x\,(x\times 1\approx x\ \wedge\ 1\times x\approx x)\ ,\ \forall x\,(x\times u\approx x\ \wedge\ u\times x\approx x)\ \models\ \forall x\,(x\times 1\approx x\ \wedge\ 1\times x\approx x)$ ∎