MATH 150: INTRODUCTORY LOGIC

Basic Definitions, Results, and Examples

By: Kevin Smith

Abstract:

This paper is designed to be a concise introduction and/or reference for the theory of Mathematical Logic. It was initially crafted from the definitions verbatim out of the text:

"A Mathematical Introduction to Logic, 2nd Edition"
By: Herbert B. Enderton

along with my own class notes from Math 150 at UC Irvine. It is equipped with many hyperlinks (listed in red) to help you navigate with ease.

From my experience, this theory finds utility mostly in proving homework problems or random theorems, but also in working with boolean statements in coding projects.

The deeper aspects of logic theory, according to wikipedia, can be broken down to four branches:

Set Theory,
Model Theory,
Recursion Theory, and
Proof Theory.

Here we do not focus so much on the separate branches, as we do a survey of the definitions in logic seen at the introductory level in the text aforementioned. For practical purposes, I would summarize our initial efforts as:

Creating a language and some grammatical rules for assembling the pieces into expressions and then declaring what it means to evaluate such expressions for truth. Once we have done this, we will focus on manipulation and generalization (together with associated notions).

TABLE OF CONTENTS (**Compressed**)

/<</ *Back to Top*

TABLE OF CONTENTS (Expanded)

1. Definitions in Sentential Logic

2. Definitions in First-Order Logic

/<</ *Back to Top*

---

3. Selected Theorems and Propositions

4. Selected Examples

/<</ *Back to Top*

/<</ *Back to Top*

SECTION 1: Definitions in Sentential Logic

We start by examining the traditional propositional (or sentential) logic and emulate the same build up later. Recall propositions are just statements that can be either true or false. They can be connected by "and", "or" etc.. For example "(it is cloudy) and (it is windy)". Connecting phrases like "therefore" are also frequently used. Deferring such statements and connecting words to symbols, whilst remembering their "English" translations, makes the logical manipulation easier.

1.1: Languages and Wff's of Sentential Logic

• Def: The **(formal) language of sentential logic** is the set of symbols:

$$\mathscr{L} = \{(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow, A_1, A_2, \dots, A_n, \dots\},$$

where each one in the set is unique in the sense that it cannot be created by concatenation of any of the other symbols in the set.

• Def: The elements {not, and, or, implies, dually implies}, respectively $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ of $\mathscr{L}$ are called *logical symbols*. These are more commonly referred to as **connectives**. Denote this set by C.

• Def: The elements $\{A_1, A_2, \dots, A_n, \dots\}$ of $\mathscr{L}$ are called **sentential symbols** or *propositions*. We'll denote this set by S.

• Def: An **expression** is a finite concatenation of symbols in $\mathscr{L}$. For example "$A_1 \vee A_2$". Such expressions can be given names for shorthand like α or other greek symbols. It is also common to put quotes around expressions as we do Strings in say Java.

Note that not all expressions are meaningful (e.g. "$) \rightarrow A_1$"). We now take time to dictate the grammar rules for $\mathscr{L}$, defined in terms of the following logical functions.

• Def: Let $\{\varepsilon_\neg, \varepsilon_\wedge, \varepsilon_\vee, \varepsilon_\rightarrow, \varepsilon_\leftrightarrow\}$ be a set of functions defined as follows:

$$\begin{aligned}
&\varepsilon_\neg(A_i) := \text{“}(\neg A_i)\text{”},\\
&\varepsilon_\wedge(A_i, A_j) := \text{“}(A_i \wedge A_j)\text{”},\\
&\varepsilon_\vee(A_i, A_j) := \text{“}(A_i \vee A_j)\text{”},\\
&\varepsilon_\rightarrow(A_i, A_j) := \text{“}(A_i \rightarrow A_j)\text{”},\\
&\varepsilon_\leftrightarrow(A_i, A_j) := \text{“}(A_i \leftrightarrow A_j)\text{”}.
\end{aligned}$$

The images of these functions are called *valid expressions* which we will denote as a set by E_V. We allow for recursive use of these functions in the obvious way:

$$\begin{aligned}
&\bar{\varepsilon}_\neg : E_V \rightarrow E_V; && \bar{\varepsilon}_\neg(\alpha_i) = \text{“}(\neg \alpha_i)\text{”},\\
&\bar{\varepsilon}_\wedge : E_V \times E_V \rightarrow E_V; && \bar{\varepsilon}_\wedge(\alpha_i, \alpha_j) = \text{“}(\alpha_i \wedge \alpha_j)\text{”},\\
&\bar{\varepsilon}_\vee : E_V \times E_V \rightarrow E_V; && \bar{\varepsilon}_\vee(\alpha_i, \alpha_j) = \text{“}(\alpha_i \vee \alpha_j)\text{”},\\
&\bar{\varepsilon}_\rightarrow : E_V \times E_V \rightarrow E_V; && \bar{\varepsilon}_\rightarrow(\alpha_i, \alpha_j) = \text{“}(\alpha_i \rightarrow \alpha_j)\text{”},\\
&\bar{\varepsilon}_\leftrightarrow : E_V \times E_V \rightarrow E_V; && \bar{\varepsilon}_\leftrightarrow(\alpha_i, \alpha_j) = \text{“}(\alpha_i \leftrightarrow \alpha_j)\text{”}.
\end{aligned}$$

whereby the images of these functions are still defined as valid expressions. Note: as a base case, sentence symbols are valid expressions. E_V is usually denoted by $\bar{\epsilon}$ in the text. I wouldn't worry about remembering the names of all the fancy sets we've created, just the result of what we've done. That is, we've defined "grammatically correct" expressions in terms of the symbols in the language $\mathscr{L}$.

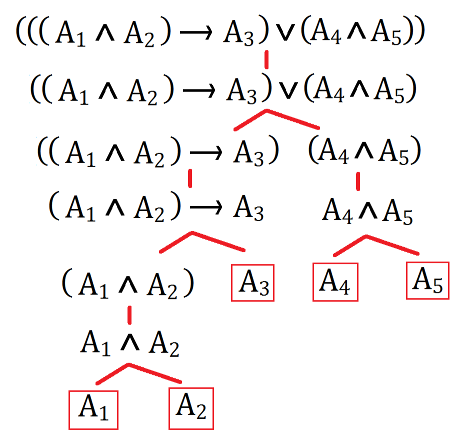
• <u>Def:</u> A **well-formed formula** (wff) is an element of E_V as defined above using our special functions (zero or more times) to concatenate symbols in $\mathscr{L}$. Wff's are usually denoted with greek letters.

• <u>Def:</u> We refer to the grammatical rules of the language as the **syntax**.

If we think instead in terms of deconstruction, one may recover all the sentence symbol parameters contained in wff's by stripping away outer parenthesis and delimiting by connectives sequentially. It so happens the steps in this sequence form a special type of graph.

• <u>Def:</u> An expression is said to be **uniquely readable** if it can be parsed (via some algorithm) into a binary tree whose terminal vertices are all sentential symbols — wff's are uniquely readable.

WFF PARSING EXAMPLE:



We are now ready to talk about the truth... in the language of sentential logic.

1.2: Truth Assignments, Tautological Implications, and Theories

Recall S was the set of sentence symbols. When translated to English, these amounted to propositions which held a value of truth. We want to declare such values for our wffs.

• Def: A map $\nu : S \to \{T, F\}$ is called a **truth assignment**, where T and F denote the values "true" and "false" respectively. We can extend this map to assign truth values to our wffs as follows. Let $\bar{\nu} : E_V \to \{T, F\}$ be such that:

1. $\bar{\nu}((\neg\alpha)) = \begin{cases} T \text{ if } \bar{\nu}(\alpha) = F, \\ F \text{ otherwise.} \end{cases}$

2. $\bar{\nu}((\alpha \wedge \beta)) = \begin{cases} T \text{ if } \bar{\nu}(\alpha) = T \text{ and } \bar{\nu}(\beta) = T, \\ F \text{ otherwise.} \end{cases}$

3. $\bar{\nu}((\alpha \vee \beta)) = \begin{cases} T \text{ if } \bar{\nu}(\alpha) = T \text{ or } \bar{\nu}(\beta) = T \text{ (or both)}, \\ F \text{ otherwise.} \end{cases}$

4. $\bar{\nu}((\alpha \to \beta)) = \begin{cases} F \text{ if } \bar{\nu}(\alpha) = T \text{ and } \bar{\nu}(\beta) = F, \\ T \text{ otherwise.} \end{cases}$

5. $\bar{\nu}((\alpha \leftrightarrow \beta)) = \begin{cases} T \text{ if } \bar{\nu}(\alpha) = \bar{\nu}(\beta), \\ F \text{ otherwise.} \end{cases}$

We've skipped the base step of this inductive definition with the understanding that sentential symbols are themselves elements of E_V.

• Def: A **truth table** is a useful tool for evaluating wff's by hand [see Example]. One will eventually come to remember definitions of connectives in terms of such truth tables. For instance, " $\leftrightarrow$ " is characterized by the fact that the T symbol only appears when both inputs are the same.

The following are some related definitions:

• Def: An **n-place boolean function realized by** α is a function $B_\alpha^n : \{F, T\}^n \to \{F, T\}$ defined as follows. Let $X = (X_1, \ldots, X_n)$ where each $X_i = \nu(A_i)$, for the sentence symbols A_i in α. Then $B_\alpha^n(X) := \bar{\nu}(\alpha \circ X)$.

In other words, given an n-tuple of truth assignments X, the extended evaluation of $\bar{\nu}(\alpha)$ given these truth assignments is the value of the boolean function. Just think of sentential symbols as variables in the expression for α, stick in the vector component arguments and evaluate from there.

• Def: A wff α is said to be **satisfiable** if there exists a B_α^n such that there is at least one n-tuple X for which $B_\alpha^n(X) = T$. A set Σ of wff's is satisfiable if each $\alpha \in \Sigma$ is satisfiable. A set Σ is *finitely satisfiable* if every finite subset of Σ is satisfiable.

The most basic non-satisfiable wff is called a **contradiction**: "$\alpha \wedge \neg\alpha$". This is a wff that has *all false* values in its truth table. Oppositely, a **tautology** is a wff with *all true* values in its table.

• Def: A set of wff's Σ **tautologically implies** τ (written $\Sigma \models \tau$) iff every truth assignment for the sentence symbols in Σ and τ that satisfies every member of Σ also satisfies τ.

• <u>Def:</u> α is **tautologically equivalent** to β iff $\alpha \vDash \beta$ and $\beta \vDash \alpha$. We denote this by $\alpha \,\mathrel{\text{⫤}}\!\!\vDash \beta$.

We utilize such equivalences between wff's to simplify expressions and hence perform less work in calculations. It should be noted that two wff's with such an equivalence also have the *exact same* truth tables. Observe that this is the case for : $(\alpha \leftrightarrow \beta) \,\mathrel{\text{⫤}}\!\!\vDash (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$.

• <u>Def:</u> A set of connectives is **complete** if we can write every possible wff in terms of the elements of the set. It is *incomplete* if there is some wff whose boolean function it realizes cannot be realized by a wff consisting of only connectives of the set. Note: the sets $\{\neg, \wedge\}$ and $\{\neg, \rightarrow\}$ are complete sets of connectives in the language of Sentential Logic. [See Example]

• <u>Def:</u> A set Σ of formulas (wff's) is **independent** if for every formula $\varphi \in \Sigma$, $\Sigma \diagdown \{\varphi\} \nvDash \{\varphi\}$.
The set is *dependent* otherwise.

The next few definitions are with regard to formal systems and point us in the direction of the section 2.

• <u>Def:</u> A **non-logical axiom** is a statement in an *instance* of a language that is taken to be true. A **logical axiom** is a statement in an abstract or formal language taken to be true. A **Rule of Inference** is a *logical implication* "known" from computing with truth tables, but stated as fact from then on out. **Theorems** are statements in a language deduced via rules of inference from axioms and other theorems (recursively defined). **Theories** are sets of theorems and axioms closed under rules of inference. A set of axioms together with a set of rules of inference is referred to as a **deductive system**. It should be noted that the set of axioms will inherently be independent. Here are some known rules of inference:

Rule of Inference	Tautology	Name
p p$\rightarrow q$ ——— $\therefore q$	$(\text{p} \wedge (p \rightarrow q)) \rightarrow q$	Modus Ponens
$\neg q$ p$\rightarrow q$ ——— $\therefore \neg p$	$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$	Modus Tollens
p$\rightarrow q$ q$\rightarrow r$ ——— $\therefore p \rightarrow r$	$((\text{p}\rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$	Hypothetical syllogism
$\neg p$ p$\vee q$ ——— $\therefore q$	$(\neg p \wedge (p \vee q)) \rightarrow q$	Disjunctive Syllogism
p ——— $\therefore (p \vee q)$	p$\rightarrow (p \vee q)$	Addition
$(\text{p}\wedge q) \rightarrow r$ ——— $\therefore p \rightarrow (q \rightarrow r)$	$((\text{p}\wedge q) \rightarrow r) \rightarrow (p \rightarrow (q \rightarrow r))$	Exportation
p$\vee q$ $\neg p \vee r$ ——— $\therefore q \vee r$	$((\text{p}\vee q) \wedge (\neg p \vee r)) \rightarrow q \vee r$	Resolution

>> https://www.geeksforgeeks.org/mathematical-logic-rules-inference/

• <u>Def:</u> A ***Quantifier*** is a symbol expressing the cases for which a statement applies. Exactly which cases the quantifier refers to divides up logic into **orders**. In propositional logic is a zeroth-order logic. Next we will see first-order logic where the quantifiers range over sets. Higher orders of logic quantify over sets of sets, sets of sets of sets, etc. A lot of mathematical languages are categorized with first-order logic, but there are higher orders needed in some cases...

SECTION 2: Definitions in First-Order Logic

2.1: Languages and Wff's of First-Order Logic

• Def: A **(formal) language in first-order logic** is given by the set of symbols:

$$\mathscr{L} = \{(,), \neg, \wedge, v_1, v_2, \dots, \exists, c_1, c_2, \dots, F^1, F^2, \dots, R^1, R^2, \dots\},$$

where each one in the set is unique in the sense that it cannot be created by concatenation of any of the other symbols in the set.

• Def: The elements $\{(,), \neg, \wedge, v_1, v_2, \dots, \exists\}$ of $\mathscr{L}$ are called **logical symbols** (denoted by S_L). The two particular subsets of logical symbols $\{\neg, \wedge\}$ and $\{v_1, v_2, \dots\}$ being the **connectives** and **variables** respectively (referenced K and V). The symbol $\{\exists\}$ being the **existential quantifier**.

Note: there is another quantifier called the **universal quantifier** ($\forall$), due to completeness of sets of connectives we can eliminate one or the other from our language. Technically $\exists v_i := \vee_{i=1}^n v_i$ and $\forall v_i := \wedge_{i=1}^n v_i$ (finite expressions in "or"/"and"). Using *DeMorgan's Law* we can use these interchangeably. Practically though, we use both.

• Def: The elements $\{c_1, c_2, \dots, F^1, F^2, \dots, R^1, R^2, \dots\}$ of $\mathscr{L}$ are called **non-logical symbols** (denoted by S_{NL}). We have the three subsets: $\{c_1, c_2, \dots\}$, $\{F^1, F^2, \dots\}$, and $\{R^1, R^2, \dots\}$ referred to by **constant symbols**, **function symbols**, and **relation symbols** (referenced by C, F, and R).

We now make our way to wffs as before, naming everything in between for later use.

• Def: An **expression** is a finite concatenation of symbols in $\mathscr{L}$. Reference the set of these by E.

• Def: A **term** is an expression of the form $F^i(x_1, \dots, x_n)$, where F^i is some function symbol and each x_j is some variable or constant symbol. We define constants and variables themselves to be terms. Denote this subset of E by T. Note that terms may also be of the form $F^i(t_1, \dots, t_n)$, where t_j's are terms.

• Def: An **atomic formula** is an expression of the form $R^i(t_1, \dots, t_n)$, where R^i is some relation symbol and each t_j is some term. Denote this set by A. We may use greek letters in place of atomic formulas. We do not define terms themselves to be atomic formulas however!

Examples: "$v_1 \dot{=} v_2$" and "$v_3 \dot{\leq} c_1$". In each case we have two basic terms formally related to create atomic formulas.

We now apply the logical functions to these atomic formulas:

• Def: Let $\{\varepsilon_\neg, \varepsilon_\wedge, \varepsilon_{\exists_i}\}$ be a set of functions defined as follows:

$$\begin{aligned}
\varepsilon_\neg : A \to E; \qquad & \varepsilon_\neg(\alpha) = \text{“}(\neg\alpha)\text{”}, \\
\varepsilon_\wedge : A \times A \to E; \qquad & \varepsilon_\wedge(\alpha, \beta) = \text{“}(\alpha \wedge \beta)\text{”}, \\
\varepsilon_{\exists_i} : A \to E; \qquad & \varepsilon_{\exists_i}(\alpha) = \text{“}(\exists v_i \alpha)\text{”}
\end{aligned}$$

The images of these functions are called **valid expressions** (E_V). We will admit extensions of our functions as before by letting them act on elements of E_V (or $E_V \times E_V$) which recursively increases the size of E_V. Note: as base cases, constants, variables, terms, and atomic formulas are valid expressions.

• Def: A **well-formed formula** (wff) is an element of E_V as defined above using our special functions (0 or more times) to concatenate symbols in $\mathscr{L}$. We also use greek letters in place of wff's.

Lastly, we have some relevant terminology to wffs:

• Def: A variable v_i is said to be of **free occurrence** in a wff α if:
Case (α is atomic): v_i is free in α iff v_i occurs (unquantified) in α,

Case ($\alpha := (\neg\beta)$): v_i is free in α iff v_i is free in β,

Case ($\alpha := (\beta_1 \wedge \beta_2)$): v_i is free in α iff v_i is free in β_1 or β_2, or

Case ($\alpha := (\exists v_j \beta)$): v_i is free in α iff v_i is free in β and if $v_i \neq v_j$.

If none of these cases apply, v_i is said to be **bounded**.

• Def: A wff α is called a **sentence** if there are no free occurences of variables in α.

• Def: A **tautology** is a wff obtainable from tautologies of sentential logic (having only the connectives in K) by replacing each sentence symbol by a wff of the first-order language.

Conventions: We have the following conventions for notational convenience:

1.) You don't always have to write the quotes " " around expressions if it is understood.
2.) The outermost parenthesis need not be listed. For example: $(A \wedge B)$ is $A \wedge B$.
3.) The negation symbol applies to as little as possible. For example: $\neg A \wedge B$ is $(\neg A) \wedge B$.
4.) All other logical symbols apply to as little as possible (*locality*).
5.) Where one symbol is used repeatedly, grouping is to the right. For example: $\alpha \wedge \beta \wedge \gamma$ is $\alpha \wedge (\beta \wedge \gamma)$.
6.) When we have *binary* function or relation symbols, the expression may be listed as such: write $x \dot{<} y$ instead of $\dot{<}(x, y)$.
7.) When it is clear, we may suppress the parenthesis and commas in the arguments of functions or relations. For example: write Fxy instead of $F(x, y)$. [Not too useful in my experience.]
8.) When listing variables, we won't necessarily use the proper names such as v_3, v_{10}, etc. We may just use x,y,z, or u,v, etc.
9.) If a variable v_i occurs free in φ, we use the notation $\varphi(v_i)$ in our meta-writing to remind us that this is the case. Formally however, this notation does not constitute a valid expression, so if we intend to use φ as a sub-formula in defining a larger one, we'll just leave out the (v_i). Similarly for multiple free variables $v_{i_1}, \ldots, v_{i_k}$ we write $\boldsymbol{\varphi(v_{i_1}, \ldots, v_{i_k})}$.
10.) (Read Section 2.2 to understand this).
Let $v_{i_1}, \ldots, v_{i_k}$ be variables which occur freely in φ and let $a_{i_1}, \ldots, a_{i_k}$ be their respective literal correspondents in the universe. We use the notation $\mathfrak{A} \vDash \boldsymbol{\varphi[[a_{i_1}, \ldots, a_{i_k}]]}$ to remind us that if $\mathfrak{A} \vDash \varphi$ with a particular evaluation, then the structure satisfies φ with any and all evaluations which agree on the set $v_{i_1}, \ldots, v_{i_k}$, (i.e. forall s, $s(v_{i_j}) = a_{i_j}$).

Now that we have the expressions to play with, we cover some technicalities, and then talk truth...

2.2: Structure, Evaluation, Satisfaction, and Theories

Previously in Sentential Logic, we dealt with a formal language, but never really came face to face with that formality. The reason being that the only symbols "formalized" were the propositions themselves (which we assumed to always come from English). Technically, they could have come from another language, say German for example. In the case of our abstract first-order language, there are many more symbols being formalized and the translation is not just to English, but rather to a mathematical language such as Calculus or Group Theory. We attempt to capture this notion rigorously with *structures*. Once we have done this, we may speak of truth of wff's as they exist instantially.

Recall,

$$\mathscr{L} = \{(,), \neg, \wedge, v_1, v_2, \dots, \exists, \boldsymbol{c_1}, \boldsymbol{c_2}, \boldsymbol{\dots}, \boldsymbol{F^1}, \boldsymbol{F^2}, \boldsymbol{\dots}, \boldsymbol{R^1}, \boldsymbol{R^2}, \dots\},$$

• <u>Def:</u> A **structure** $\mathfrak{A}$ is an instance of a formal language $\mathscr{L}$, together with a **universe of discourse**, denoted: $\boldsymbol{U \equiv |\mathfrak{A}|}$. More precisely, a structure is an assignment of (abstract) non-logical symbols $\{c_i, F^i, R^i\}$ in $\mathscr{L}$ to corresponding (concrete) instances $\{c_i^{\mathfrak{A}}, F^{i,\mathfrak{A}}, R^{i,\mathfrak{A}}\}$, as well as an assignment of a universe, U, for variables v_i to be quantified over.

Each structure is identified by its universe and its deviation from the original language, since all the logical symbols stay the same each time. This identification is summarized in a so called **signature** for $\mathfrak{A}$:

$$\boldsymbol{\mathfrak{A} \equiv \{|\mathfrak{A}|;\ c_1^{\mathfrak{A}}, c_2^{\mathfrak{A}},\ \dots\ , F^{1,\mathfrak{A}}, F^{2,\mathfrak{A}},\ \dots\ , R^{1,\mathfrak{A}}, R^{2,\mathfrak{A}}, \dots\}}.$$

• <u>Def:</u> We speak of **interpretations** of symbols in $\mathscr{L}$ as the literal correspondents, $X^{\mathfrak{A}}$, given above (the image of the structure as a map). Coupled with **evaluation** functions, which are assignments $\epsilon : V \to |\mathfrak{A}|$ of variables to elements in the universe, we obtain **interpretations of wffs**.

We can summarize both interpretation and evaluation in an **extended evaluation** map:

$$\boldsymbol{\bar{\epsilon} : \mathscr{L} \to \mathfrak{A}} \qquad \text{such that:}$$

1.) $\bar{\epsilon}$ restricts to the identity on logical symbols $\{(,), \neg, \wedge, v_1, v_2, \dots, \exists\}$,

2.) $\bar{\epsilon}|_V = \epsilon$, i.e. $\forall v_i \in V$, $\bar{\epsilon}(v_i) := \epsilon(v_i)$,

3.) $\bar{\epsilon}|_C = \mathfrak{A}$, i.e. $\forall c_i \in C$, $\bar{\epsilon}(c_i) := \mathfrak{A}(c_i) = c_i^{\mathfrak{A}}$,

4.) $\bar{\epsilon}|_F = \mathfrak{A}$, i.e. forall terms t_j, $\bar{\epsilon}(F^i(t_1, \dots, t_n)) := \mathfrak{A}(F^i(t_1, \dots, t_n)) = F^{i,\mathfrak{A}}(\bar{\epsilon}(t_1), \dots, \bar{\epsilon}(t_n))$,

5.) $\bar{\epsilon}|_R = \mathfrak{A}$, i.e. forall terms t_j, $\bar{\epsilon}(R^i(t_1, \dots, t_n)) := \mathfrak{A}(R^i(t_1, \dots, t_n)) = R^{i,\mathfrak{A}}(\bar{\epsilon}(t_1), \dots, \bar{\epsilon}(t_n))$.

Now let's return to the notion of determining truth, within individual structures.

• Def: We define **satisfaction** of wff's, φ, with respect to a structure $\mathfrak{A}$ and extended evaluation $\bar{\epsilon}$ by:

Case (φ is a Constant, Variable, or Term): Trivially satisfied.

Case (φ is an Atomic Formula): Say $\varphi = R^i(t_1, \ldots, t_n)$, then φ is satisfied if $(\bar{\epsilon}(t_1), \ldots, \bar{\epsilon}(t_n)) \in R^{i,\mathfrak{A}}$,

Case $\big(\varphi = (\neg\alpha)\big)$: φ is satisfied if α is not satisfied,

Case $(\varphi = (\alpha \to \beta))$: φ is not satisfied if α is and β is not. φ is satisfied otherwise, and

Case $(\varphi = (\forall v_i \alpha))$: φ is satisfied if for any element $x \in |\mathfrak{A}|$, we can substitute x in for the interpretation of v_i everywhere it appears in the interpretation of α and the resulting formulas are satisfied.

Note: In the above cases, we write $\boldsymbol{\mathfrak{A} \models \varphi[\bar{\epsilon}]}$ and say "*the structure* $\mathfrak{A}$ *satisfies* φ *with respect to* $\bar{\epsilon}$". If $\bar{\epsilon}$ is understood from context, we may simplify the notation to: $\boldsymbol{\mathfrak{A} \models \varphi}$, etc. Now, if Σ is a *set of* $\mathscr{L}$*-sentences*, that is, $\Sigma = \{\varphi \mid \varphi \text{ has all bounded variables}\}$, then $\boldsymbol{\mathfrak{A} \models \Sigma[\bar{\epsilon}] \leftrightarrow \forall \varphi \in \Sigma, \mathfrak{A} \models \varphi[\bar{\epsilon}]}$.

Particularly,

• Def: A **model ($\mathfrak{M}$) of a theory (T)** is a structure that satisfies all sentences in T $\big(\mathfrak{M} \models T[\bar{\epsilon}]\big)$. The set of all models of a theory T is denoted by $\boldsymbol{Mod(T)}$, (in this case $\mathfrak{M} \in Mod(T)$).

The remaining definitions in this section refer to structures.

• Def: Two structures $\mathfrak{A}$ and $\mathfrak{B}$ are called **elementarily equivalent** (denoted $\mathfrak{A} \equiv \mathfrak{B}$) if they satisfy the same exact set of $\mathscr{L}$-sentences.

• Def: Let $\mathfrak{A}, \mathfrak{B}$ be structures for some F.O.L. ($\mathscr{L}$). A **structure homormorphism** $h : \mathfrak{A} \to \mathfrak{B}$ is a function with the following properties:

1.) For each n-place relation symbol R and each n-tuple $(a_1, \ldots, a_n)$ of elements in $|\mathfrak{A}|$, $(a_1, \ldots, a_n) \in R^{\mathfrak{A}}$ iff $(h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}$,
2.) For each n-place function symbol F and each such n-tuple, $h(F^{\mathfrak{A}}(a_1, \ldots, a_n)) = F^{\mathfrak{B}}(h(a_1), \ldots, h(a_n))$.

If in addition to being a homomorphism, h is 1-1, then it is called an **isomorphism** (or an *isomorphic embedding*) of $\mathfrak{A}$ into $\mathfrak{B}$. If it is also onto, we write $\mathfrak{A} \cong \mathfrak{B}$.

• Def: A **substructure** $\mathfrak{B}$ of $\mathfrak{A}$ is a structure such that there exists an isomorphic embedding $f : \mathfrak{B} \to \mathfrak{A}$ for which $|\mathfrak{B}| \subseteq |\mathfrak{A}|$.

We covered a lot here. See the examples below for some clarification. When you are "satisfied", let's go back to the concept of theories again...

>> Structure and Evaluation Example
>> Model of a Theory Example

2.3: Logical Implication and Provability

Recall the discussion at the end of Section 1.2. There we discussed *axioms*, *rules of inferences*, *deductive systems*, and *theories*. We wish to revisit these concepts with a few other relevant definitions.

• Def: Given a variable v_i, a wff φ, and a term t in some language $\mathscr{L}$, if we replace every free occurence of v_i in φ by t, then we denote this new formula by $\varphi(v_i/t)$. Alternatively we can denote this by $\varphi_t^{v_i}$. We say that t is **substitutable** for v_i in φ if no occurence of a variable in t becomes bounded in $\varphi(v_i/t)$. This notation is only for convenience, it is *not* a valid expression.

• Def: The set of **logical axioms** Λ, includes all generalizations of wff's of the following forms:
1.) Tautologies,
2.) $\forall v_i \alpha \to \alpha_t^{v_i}$, where t is substitutable for v_i in α,
3.) $\forall v_i(\alpha \to \beta) \to (\forall v_i \alpha \to \forall v_i \beta)$
4.) $\alpha \to \forall v_i \alpha$, where v_i does not occur free in α,
5.) $v_i \doteq v_i$,
6.) $v_i \doteq v_j \to (\alpha \to \alpha')$, where α is atomic and α' is obtained from α by replacing v_i in zero or more (but not necessarily all) places by v_j.

Notes: (i) A **generalization** of a wff ψ is a new wff $\varphi = \forall v_{i_1} \dots \forall v_{i_n} \psi$ and
(ii) numbers (5) and (6) are only defined if our language includes a binary relation who's interpretation is equality.

• Def: Let Σ be a set of wffs, φ a wff. Then Σ **logically implies** φ, written $\boldsymbol{\Sigma \models \varphi}$, iff for every $\mathscr{L}$-structure $\mathfrak{A}$ and every evaluation $\epsilon : V \to |\mathfrak{A}|$ such that $\forall \psi \in \Sigma$, $\mathfrak{A} \models \psi$, then $\mathfrak{A} \models \varphi$.
◦ Let Σ be a set of sentences, σ a sentence. Then $\Sigma \models \sigma$ iff $\forall \mathfrak{A}$ such that $\forall \varphi \in \Sigma$, $\mathfrak{A} \models \varphi$, then $\mathfrak{A} \models \sigma$.

• Def: Two wff's α and β are said to be **logically equivalent** iff $\alpha \models \beta$ and $\beta \models \alpha$, denoted $\alpha \models\!\!\!\!\!\dashv \beta$.

• Def: An **$\mathscr{L}$-theory** T is a set of $\mathscr{L}$-sentences closed under logical implication. That is, T is a theory iff T is a set of sentences such that for any sentence σ of the language,

$$T \models \sigma \implies \sigma \in T.$$

• Def: An **$\mathscr{L}$-proof** or *logical deduction* of φ from Σ is a finite sequence $< \alpha_0, \dots, \alpha_n >$ of formulas such that $\alpha_n = \varphi$ and for each $k \leq n$, either:
1.) $\alpha_k \in \Sigma \cup \Lambda$, or
2.) α_k is obtained by one of our rules of inference using earlier formulas in the sequence.

• Def: We say φ is **provable** from Σ (and of course from Λ) if there is an $\mathscr{L}$-proof of φ from Σ. If Σ proves φ in the manner described above, we write: $\boldsymbol{\Sigma \vdash \varphi}$.

• Def: A set of wff's Σ is said to be **deductively consistent** if there is a sentence σ such that $\Sigma \nvdash \sigma$. Σ is *deductively inconsistent* otherwise. This says a set of formulas is deductively consistent if there is some sentence (such as a contradiction) which is not provable from the set.

• Def: A set Σ of expressions is **decidable** iff there exists an effective procedure that, given an expression α, will decide whether or not $\alpha \in \Sigma$. Some theories exist that are undecidable!

That's all for the introductory definitions. Next we have a collection of results and examples.

SECTION 3: Selected Theorems and Propositions

• **Compactness for Sentential Logic:**
Let Σ be an infinite set of wff's. Then Σ is satisfiable iff Σ is finitely satisfiable.

• **Compactness for First-Order Logic:** If Σ is a set of $\mathscr{L}$-sentences in a first-order language, then the following are equivalent:
(i) Σ has a model and
(ii) Every finite subset $\Delta \subseteq \Sigma$ has a model.

• **DeMorgan's Law:** Let $A_1, \ldots, A_n$ be sentence symbols, then:

$$\neg(A_1 \wedge \cdots \wedge A_n) = \neg A_1 \vee \cdots \vee \neg A_n$$
$$\neg(A_1 \vee \cdots \vee A_n) = \neg A_1 \wedge \cdots \wedge \neg A_n.$$

• **Gordel's Soundness and Completness Thorem:**
Suppose Σ is a set of $\mathscr{L}$-sentences and σ is an $\mathscr{L}$-sentence. Then the following are equivalent:
(i) $\Sigma \vdash \sigma$
(ii) $\Sigma \vDash \sigma$.
Note: (i) $\Longrightarrow$ (ii) is referred to as the *Soundness Theorem* and (ii) $\Longrightarrow$ (i) is referred to as the *Completeness Theorem*.

• **Important Tautologies in Sentential Logic:** Let P and Q be sentence symbols, then:
(i) Associative and Commutative Laws for $\wedge$, $\vee$, and $\leftrightarrow$.
(ii) Distributive Laws:

$$\vDash (A \wedge (B \vee C)) \leftrightarrow ((A \wedge B) \vee (A \wedge C))$$
$$\vDash (A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C))$$

(iii) Contradiction: $\vDash \neg(P \to Q) \Leftrightarrow (P \wedge \neg Q)$
(iv) Contraposition: $\vDash (P \to Q) \Leftrightarrow (\neg Q \to \neg P)$
(v) Excluded Middle: $\vDash (A \vee (\neg A))$

• **Model Existence Theorem:**
Let Σ be a set of $\mathscr{L}$-sentences. Then Σ is deductively consistent iff Σ has a model (i.e. there is an $\mathscr{L}$-structure $\mathfrak{A}$ such that $\mathfrak{A} \vDash \Sigma$).

• **Post's Theorem:**
If G is an n-place Boolean function such that $n \geq 1$, then there exists a wff α such that $G = B_\alpha^n$ (i.e. such that α realizes G).

• **Post's Corollary:**
For any wff α, we can find a tautologically equivalent wff β in disjunctive normal form.

• **Statements Regarding Deductive Consistency:** Given a set of $\mathscr{L}$-formulas in a first-order language,
then the following are equivalent:
(i) Σ is deductively inconsistent,
(ii) For any $\mathscr{L}$-formula φ, $\Sigma \vdash \varphi \wedge \neg\varphi$, and
(iii) For some formula φ, $\Sigma \vdash \varphi \wedge \neg\varphi$.

• **Tautological Implications and Boolean Functions:**
Suppose α and β are wff's whose sentential symbols are amongst $\{A_1, \ldots, A_n\}$. Then:
(i) $\alpha \vDash \beta$ iff for every $X \in \{T, F\}^n$, $B^n_\alpha(X) \leq B^n_\beta(X)$,
(ii) $\alpha \,=\!\!|\!\!\vDash \beta$ iff for every $X \in \{T, F\}^n$, $B^n_\alpha(X) = B^n_\beta(X)$, and
(iii) $\vDash \alpha$ iff for every $X \in \{T, F\}^n$, $B^n_\alpha = T$.

• **Well Definition of Evaluations in First-Order Logic:**
If φ is a formula of a fixed first-order language, $\mathfrak{A}$ is an $\mathscr{L}$-structure, and s is an evaluation, then:
(i) $\mathfrak{A} \vDash \varphi[s]$ is defined iff s evaluates all variables with free occurences in φ and
(ii) $\mathfrak{A} \vDash \varphi[s]$ depends only on the variables with free occurence in φ.

• **Well Definition Corollary:**
A sentence φ is either *true* or *false* in $\mathfrak{A}$ but not both (i.e. either $\mathfrak{A} \vDash \varphi$ **or** $\nvDash \varphi$).

SECTION 4: Selected Examples

1.) Evaluating of a Wff (*back to def*)

Q: Given the language of sentential logic, define a wff by $\alpha := ((p \wedge (\neg q)) \leftrightarrow (p \to r))$, where p, q, and r are sentence symbols. Determine under what conditions $\bar{\nu}(\alpha) = T$.

A: Using the truth assignments (on pg. 2), we may create a table that evaluates α in parts:

p	q	r	$\neg q$	$(p \wedge (\neg q))$	$(p \to r)$	$((p \wedge (\neg q)) \leftrightarrow (p \to r))$
T	T	T	F	F	T	F
T	T	F	F	F	F	**T**
T	F	T	T	T	T	**T**
T	F	F	T	T	F	F
F	T	T	F	F	T	F
F	T	F	F	F	T	F
F	F	T	T	F	T	F
F	F	F	T	F	T	F

From here it is easy to see the conditions that yield true for α are when p, q, and r are respectively either T, T, F or T, F, T. ■

2.) Negating an Expression

Q: Negate the following sentence (which is written in a model of the language of Topology):

"$\forall\varepsilon\Bigg((\varepsilon > 0) \to \Bigg(\exists\delta\Big((\delta > 0) \to \Big(\forall y\big((d(x,y) < \delta) \to (\rho(f(x), f(y)) < \varepsilon)\big)\Big)\Big)\Bigg)\Bigg)$".

A: We may use repeated application of DeMorgan's Law and the tautology involving implication (see page10):

$$\neg\Bigg(\forall\varepsilon\Bigg((\varepsilon > 0) \to \Bigg(\exists\delta\Big((\delta > 0) \to \Big(\forall y\big((d(x,y) < \delta) \to (\rho(f(x), f(y)) < \varepsilon)\big)\Big)\Big)\Bigg)\Bigg)\Bigg)$$

$$= \exists\varepsilon\Bigg((\varepsilon > 0) \wedge \neg\Bigg(\exists\delta\Big((\delta > 0) \to \Big(\forall y\big((d(x,y) < \delta) \to (\rho(f(x), f(y)) < \varepsilon)\big)\Big)\Big)\Bigg)\Bigg)$$

$$= \exists\varepsilon\Bigg((\varepsilon > 0) \wedge \Bigg(\forall\delta\Big((\delta > 0) \wedge \neg\Big(\forall y\big((d(x,y) < \delta) \to (\rho(f(x), f(y)) < \varepsilon)\big)\Big)\Big)\Bigg)\Bigg)$$

$$= \exists\varepsilon\Bigg((\varepsilon > 0) \wedge \Bigg(\forall\delta\Big((\delta > 0) \wedge \Big(\exists y\big((d(x,y) < \delta) \wedge \neg(\rho(f(x), f(y)) < \varepsilon)\big)\Big)\Big)\Bigg)\Bigg)$$

$$= \exists\varepsilon\Bigg((\varepsilon > 0) \wedge \Bigg(\forall\delta\Big((\delta > 0) \wedge \Big(\exists y\big((d(x,y) < \delta) \wedge (\rho(f(x), f(y)) \geq \varepsilon)\big)\Big)\Big)\Bigg)\Bigg)$$ ■

3.) **Determining if a Set of Connectives is Complete** (*back to def*)

Q: Given the set $C = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, show that the subset $K = \{\neg, \wedge\}$ is complete.

A: The most straightforward way to approach this problem is to go through and *inductively* show that any formula involving connectives in $C \backslash K$ has a tautologically equivalent formula written in terms of only connectives in K. Accordingly:

(Case $\vee$:) Let $\alpha := \beta \vee \gamma$ for two subwff's that contain only connectives in K. Clearly using DeMorgan's Law we have the equivalent: $\alpha' := \neg(\neg\beta \wedge \neg\gamma)$.

(Case $\rightarrow$:) Let $\alpha := \beta \rightarrow \gamma$ as before. From the Contradiction Tautology (p.10) we have: $\alpha' := \neg(\beta \wedge \neg\gamma)$.

(Case $\leftrightarrow$:) Since $\alpha := \beta \leftrightarrow \gamma = (\beta \rightarrow \gamma) \wedge (\gamma \rightarrow \beta)$, we're done by the previous case. ■

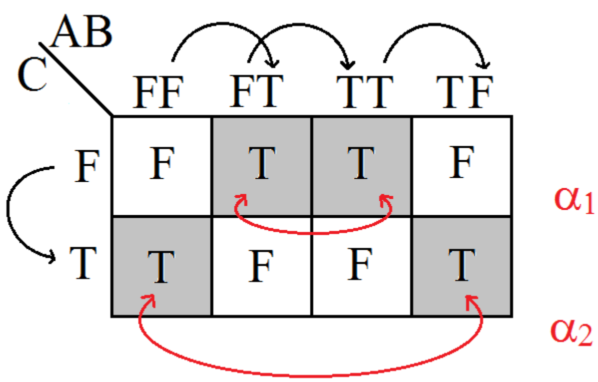
4.) **Finding a Wff That Realizes a Boolean Function**

Q: Let a 3-place boolean function $G : \{T, F\}^3 \rightarrow \{T, F\}$ be such that:

$G(F, T, F) = T$;
$G(T, T, F) = T$;
$G(F, F, T) = T$;
$G(T, F, T) = T$;
$G(A, B, C) = F$ otherwise.

Find a wff α which satisfies the above conditions when evaluated with each triplet.

A: We will be using what is known as the "Karnaugh Method (or Map)" for 3 parameters.
1.) List the possible combinations of truth values for A and B across the columns and for C across the rows by starting with one combination and then subsequently altering one value at a time (black arrows).
2.) Cover all T's in the table (gray).
3.) Group adjacent blocks of powers of 2 (i.e. 1,2,4,8,...).
<u>Note:</u> In this construction outer edges count as adjacent to their opposite side.
4.) Encode the common truth values of the rectangles into sub wff's by: either their respective parameter (in the case of T) or the negation of the parameter (for F).
5.) Lastly, disjoin ($\vee$) all sub-wff's.



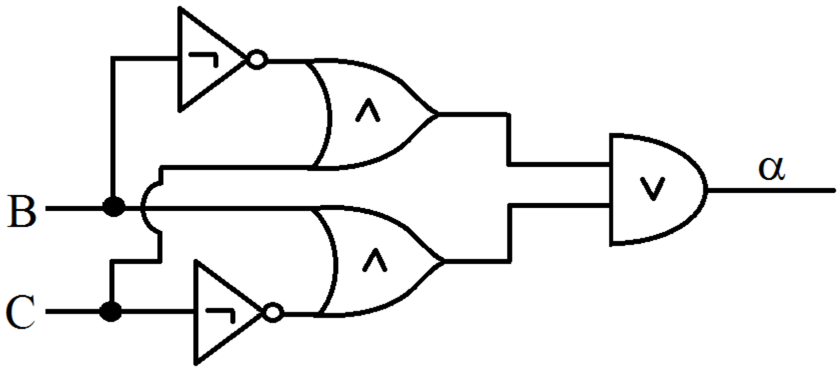
Following this procedure yields: $\alpha_1 = (B \wedge (\neg C))$ (as the first block has two common T's in the second position) and $\alpha_2 = ((\neg B) \wedge C)$. From which we have: $\alpha = (\alpha_1 \vee \alpha_2) = ((B \wedge (\neg C)) \vee ((\neg B) \wedge C))$.

One can construct the truth table and see that the above conditions are satisfied. ■

5.) **Drawing a Switching Circuit for a Wff**

Q: Draw the switching circuit corresponding to the wff α given in the previous problem:

A: See diagram below, notice the labels in the circuit elements.



6.) **Finding $\mathscr{L}$-formulas in First-Order Logic**

Q: Consider the Language af Arithmetic: $\mathscr{L} = \{\dot{0}, \dot{S}, \dot{+}, \dot{\times}, \dot{<}\}$, then let $N_{std} = (\mathbb{N}, 0, S, +, \times, <)$ be the standard structure where $\mathbb{N} = \{0, 1, 2, \dots\}$, 0 is ordinary zero, plus and times are ordinary, S is the "successor function" given by: $S(n) = n + 1$, and $<$ is the ordinary less than relation symbol. Express the following statements as $\mathscr{L}$-formulas in F.O.L.:

a.) "u, v are relative primes".
b.) "There are infinitely many pairs of relative primes."

A: For part (a) we know from a previous course that if two numbers are relatively prime then their G.C.D. is 1 (or equivalently if there exist numbers x and y such that $ux + vy = 1$).
Hence we have $\varphi(u, v) := \exists x \exists y (u \dot{\times} x \dot{+} v \dot{\times} y \doteq \dot{S}\dot{0})$, or even more properly:

$$\varphi(u, v) := \Bigg(\exists x \Bigg(\exists y \Bigg(\doteq \Big(\dot{+} \big(\dot{\times}(u, x), \dot{\times}(v, y) \big), \dot{S}\dot{0} \Big) \Bigg) \Bigg) \Bigg).$$

For (b) we have: $$\sigma := \forall v_i \exists v_j \Big(\varphi(v_i, v_j) \wedge \Big(\exists v_k \exists v_l \big(\varphi(v_k, v_l) \wedge (v_i \dot{<} v_k) \wedge (v_j \dot{<} v_l) \big) \Big) \Big),$$

which informally says "forall numbers there is a corresponding one for which the two are relatively prime and given such a pair, we may always find relatively prime pairs which are larger than our original pair". ■

• <u>Def:</u> (Recall notation convention 10 from Section 2.1). Let $\mathfrak{A}$ be an $\mathscr{L}$-structure with corresponding universe $|\mathfrak{A}|$. Furthermore let $A \subseteq |\mathfrak{A}|^n$ for some $n \in \mathbb{N}$. We say the set A is **definable** in $\mathfrak{A}$ if there is an $\mathscr{L}$-formula $\varphi(v_{i_1}, \ldots, v_{i_n})$ such that:

$$A = \{(a_1, \ldots, a_n) \in |\mathfrak{A}|^n : \mathfrak{A} \models \varphi[[a_1, \ldots, a_n]]\}.$$

This says that A is definable if it is the set of all n-tuples for which the interpretation of some φ in $\mathfrak{A}$ is satisfied (with respect to any evaluation that agrees on A).

This definition can be extended to include parameters as follows. If $v_{i_{n+1}}, \ldots, v_{i_{n+m}}$ are also variables that occur free in φ and if forall evaluations that agree on A and $\{v_{i_{n+1}}, \ldots, v_{i_{n+m}}\}$, then given the images of these new variables $b_{n+1}, \ldots, b_{n+m} \in |\mathfrak{A}|$, we say A is *definable with parameters* if:

$$A = \{(a_1, \ldots, a_n) \in |\mathfrak{A}|^n : \mathfrak{A} \models \varphi[[a_1, \ldots, a_n, b_{n+1}, \ldots, b_{n+m}]]\}.$$

7.) Defining Sets in a Structure

Q: Then for the set up given in the previous problem:

a.) Define the set of numbers u which are divisible by 3.
b.) Define the complement of the set described in part (a).

A: For (a) we may list the defining formula as: $\varphi(u) := \exists k(u \dot{=} \dot{S}\dot{S}\dot{S}\dot{0} \dot{\times} k)$. It should be clear that we thus have a set $A \subseteq \mathbb{N}$ given by $A = \{u \in \mathbb{N} | \varphi(u)\}$. Then for (b) we have:

$$\varphi^c(u) := \neg\varphi(u) = \neg(\exists k(u \dot{=} \dot{S}\dot{S}\dot{S}\dot{0} \dot{\times} k)) = \forall k(\neg(u \dot{=} \dot{S}\dot{S}\dot{S}\dot{0} \dot{\times} k)).$$ ■

8.) Structure and Evaluation (*back to section*)

Q: Give an example of a first order language together with a structure and evaluation.
A: This comes from p.69 and p.81 of the text.
Take the *language of set theory* $\mathscr{L} := \{S_L, c_1, R^1\}$, where we write $c_1 \equiv \dot{\emptyset}$, $R^1(x, y) \equiv \dot{\in}$.

Now, let $\mathfrak{A} \equiv \{|\mathfrak{A}| \; ; \; \dot{\emptyset}^{\mathfrak{A}} \; , \; \dot{\in}^{\mathfrak{A}}\} := \{\mathbb{N} \; ; \; 0 \; , \; <\}$. Then the evaluation is given by assigning variables to natural numbers. *[It seems as though evaluations are redundant technicalities.]* ■

9.) Theory Model (*back to section*)

Q: Give an example of a wff in a F.O.L. modeled by a structure.
A: With the situation of the previous problem. Define $\boldsymbol{\varphi :=}$ **“**$\boldsymbol{(\exists v_1(\neg(\exists v_2(\dot{\epsilon} v_2 v_1))))}$**”**.
Within the structure, coupled with conventions, this translates to $\boldsymbol{\varphi' =}$ **“**$\boldsymbol{\exists x \forall y(\neg(y < x))}$**”**.
Which when read in English says that “there is a natural number for which no other natural number is smaller than.” This is true and refers to $x = 1$, so we say the structure models the sentence.

10.) A Logical Deduction Proof

Q: Assume the following statement holds: “For all formulas φ and ψ, $\vdash \forall x(\varphi \to \psi) \to (\forall x\varphi \to \forall x\psi)$”.
Show that for all formulas φ, $\vdash \exists x\forall y\varphi \to \forall y\exists x\varphi$.

A: Here is an abbreviated deduction.

1.)	$\forall y\varphi \to \varphi$	Quantifier Axiom
2.)	$\neg\varphi \to \neg\forall y\varphi$	1; Contrapositive
3.)	$\forall x(\neg\varphi \to \neg\forall y\varphi)$	2; Generalization
4.)	$\forall x(\neg\varphi \to \neg\forall y\varphi) \to (\forall x\neg\varphi \to \forall x\neg\forall y\varphi)$	Assumption
5.)	$\forall x\neg\varphi \to \forall x\neg\forall y\varphi$	3,4; Modus Ponens
6.)	$\neg\forall x\neg\forall y\varphi \to \neg\forall x\neg\varphi$ $[\exists x\forall y\varphi \to \exists x\varphi]$	5; Contrapositive
7.)	$\exists x\forall y\varphi \to \forall y\exists x\varphi$	6; Quantifier Rule ■

The following was answered by me on Math Stack Exchange HERE.

11.) Advanced (Modal) Logic Deduction Proof

Want to show:

$$\left[\left[\Box(p \to q)\right] \wedge \left[\Diamond\Box\neg q\right]\right] \vdash_{\Lambda} \left[\neg\Diamond p\right].$$

Where deduction is in $\Lambda := \langle KTB \cup PL \rangle$

(K): $\Box(p \to q) \to (\Box p \to \Box q)$

(T): $p \to \Diamond p$

(B): $p \to \Box\Diamond p$

$(Dual)$: $\Diamond p \leftrightarrow \neg\Box\neg p$

$(Modus\_Ponens)$: $[(\varphi \in \Lambda) \wedge (\varphi \to \psi \in \Lambda)] \implies (\psi \in \Lambda)$

$(Uniform\_Substitution)$: $(\varphi \in \Lambda) \implies (\varphi[..subs..] \in \Lambda)$

$(Generalization)$: $(\varphi \in \Lambda) \implies (\Box\varphi \in \Lambda)$

Proof:

$0:$ $\Box(p \to q)$ (Hyp)

$1:$ $\Diamond\Box\neg q$ (Hyp)

—

$2:$ $\Box(\neg q \to \neg p)$ (Contrapositive (0))

$3:$ $\Box(\neg q \to \neg p) \to \big(\Box\neg q \to \Box\neg p\big)$ (Unif. Sub (2) into (K))

$4:$ $\Box\neg q \to \Box\neg p$ $(M.P.(2,3))$

—

$5-6:$ $\neg\Box\Diamond q \to \neg q$ (Contrapositive of (B) with Sub.)

$7:$ $\neg\Box\Diamond q$ (Dual of (1))

$8:$ $\neg q$ (M.P.(7,6))

$9:$ $\Box\neg q$ (Generalize (8))

$10:$ $\Box\neg p$ (M.P.(9,4))

$11:$ $\neg\Diamond p$ (Dual of (10)) ■

Reference for this answer was *Modal Logic* by Blackburn et al. (Ch.4, pg.190).

This was a nice question that also hints at further reading!