Mathematical Logic

A summary of "A Concise Introduction to Mathematical Logic", W. Rautenberg

Ari Feiglin

Contents

1 Propositional Logic

1.1 Semantics of Propositional Logic

Propositional logic is the study of logic removed from interpretation of individual variables and context. I will assume that the reader already has experience with propositional logic, as this is something an undergraduate will cover in one of their first courses. While this subsection will focus mainly on the semantics of propositional logic, we will begin by defining its *syntax*,

1.1.1 Definition
Let PV be an arbitrary set of **propositional variables** (which are regarded as arbitrary symbols). **Propositional formulas** are formulas defined recursively by the following rules,

(1) Propositional variables in PV are formulas, called **prime** or **atomic** formulas.

(2) If α and β are formulas, then so are $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, and $\neg\alpha$. $(\alpha \wedge \beta)$ is called the **conjunction** of α and β, $(\alpha \vee \beta)$ their **disjunction**, and $\neg\alpha$ the **negation** of α.

The set of all the formulas constructed in this matter is denoted $\mathcal{F}$.

We can generalize this definition; instead of utilizing only the symbols $\wedge$ and $\vee$, we can take a general *logical signature* σ consisting of logical connectives of differing arities. We then recursively define σ-formulas as following: if c is an n-ary logical connective in σ, and $\alpha_1, \dots, \alpha_n$ are formulas, then so is

$$(\mathsf{c}\alpha_1, \dots, \alpha_n)$$

Aleternatively, if we only consider binary and unary connectives, then if c is a unary connective, we define $\mathsf{c}\alpha$ to be a formula, and if $\circ$ is a binary connective, then $(\alpha \circ \beta)$ is a formula. But we don't have much need for such generalizations, as $\{\wedge, \vee, \neg\}$ is complete, in the sense that all connectives can be defined using them. This is a fact we will discuss soon.
We can define other connectives, for example $\rightarrow$ and $\leftrightarrow$ are used as shorthands:

$$(\alpha \rightarrow \beta) := \neg(\alpha \wedge \neg\beta), \qquad (\alpha \leftrightarrow \beta) := \big((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)\big)$$

We similarly define symbols for false and true:

$$\bot := (p_1 \wedge \neg p_1), \qquad \top = \neg\bot$$

For readability, we will use the following conventions when writing formulas (this is not a change to the definition of a formula, rather conventions for writing them in order to enhance readability)

(1) We will omit the outermost parentheses when writing formulas, if there are any.

(2) The order of operations for logical connectives is as follows, from first to last: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.

(3) We associate $\rightarrow$ from the right, meaning $\alpha \rightarrow \beta \rightarrow \gamma$ is to be read as $\alpha \rightarrow (\beta \rightarrow \gamma)$. All other connectives associate from the left, for example $\alpha \wedge \beta \wedge \gamma$ is to be read as $(\alpha \wedge \beta) \wedge \gamma$.

(4) Instead of writing $\alpha_0 \wedge \alpha_1 \wedge \cdots \alpha_n$, we write $\bigwedge_{i=0}^{n} \alpha_i$, similar for $\vee$.

Since formulas are constructed in a recursive manner, most of our proofs about them are inductive.

1.1.2 Principle (Principle of Formula Induction)
Let $\mathcal{E}$ be a property of strings which satisfies the following conditions:

(1) $\mathcal{E}\pi$ for all prime formulas π,

(2) If $\mathcal{E}\alpha$ and $\mathcal{E}\beta$, then $\mathcal{E}(\alpha \wedge \beta)$, $\mathcal{E}(\alpha \vee \beta)$, and $\mathcal{E}\neg\alpha$ for all formulas $\alpha, \beta \in \mathcal{F}$.

Then $\mathcal{E}\varphi$ is true for all formulas φ.

An example of this is that every formula $\varphi \in \mathcal{F}$ is either prime, or of one of the following forms

$$\varphi = \neg\alpha, \quad \varphi = (\alpha \wedge \beta), \quad \varphi = (\alpha \vee \beta)$$

The proof of this is straightforward: let $\mathcal{E}$ be this property. Then trivially, $\mathcal{E}\pi$ for all prime formulas π. And if $\mathcal{E}\alpha$ and $\mathcal{E}\beta$, then of course we have

$$\mathcal{E}\neg\alpha, \quad \mathcal{E}(\alpha \wedge \beta), \quad \mathcal{E}(\alpha \vee \beta)$$

This is the first step in showing the *unique formula reconstruction property.* Let us prove a lemma before proving the property itself,

1.1.3 Lemma
Proper initial segments of formulas are not formulas. Equivalently (by contrapositive), if α and β are formulas and $\alpha\xi = \beta\eta$ for arbitrary strings ξ and η, then $\alpha = \beta$.

Let us prove this by induction on α. If α is a prime formula, suppose that β is not a prime formula, then its first character is either (or $\neg$, but then $\alpha =$ (or $\alpha = \neg$, in contradiction. Thus β is a prime formula and so $\alpha = \beta$ as they are both a single character. Now if $\alpha = (\alpha_1 \circ \alpha_2)$, then the first character of β must too be (, so β is of the form $(\beta_1 * \beta_2)$. Thus

$$\alpha_1 \circ \alpha_2)\xi = \beta_1 * \beta_2)\eta$$

and so by our inductive assumption, $\alpha_1 = \beta_1$, and so $\circ = *$, and thus $\alpha_2 = \beta_2$ by our inductive assumption again. And so $\alpha = \beta$ as required. The proof for the case that $\alpha = \neg\alpha'$ is similar. ■

1.1.4 Proposition (Unique Formula Reconstruction Property)
Every compound formula $\varphi \in \mathcal{F}$ is of one of the following forms:

$$\varphi = \neg\alpha, \quad \varphi = (\alpha \wedge \beta), \quad \varphi = (\alpha \vee \beta)$$

For some formulas $\alpha, \beta \in \mathcal{F}$.

We have already shown existence. We will now show that this is unique, meaning that φ can be written uniquely as one of these strings. Using the lemma proven above, the proof for uniqueness of the reconstruction property is immediate. For example, if $\varphi = (\alpha_1 \wedge \beta_1)$ then obviously φ cannot be written as $\neg\alpha_2$ since $(\neq \neg$, and if $\varphi = (\alpha_2 \vee \beta_2)$ then by the lemma $\alpha_1 = \alpha_2$, and so we get that $\wedge = \vee$ in contradiction. And finally if $\varphi = (\alpha_2 \wedge \beta_2)$, then again by the lemma, $\alpha_1 = \alpha_2$ and $\beta_1 = \beta_2$ as required. The proof for $\neg$ and $\vee$ are similar. ■

Utilizing formula recursion, we can define functions on formulas. For example,

1.1.5 Definition
For a formula φ, we define $Sf\varphi$ to be the set of all subformulas of φ. This is done recursively:

$$Sf\pi = \{\pi\} \text{ for prime formulas } \pi,$$
$$Sf\neg\alpha = Sf\alpha \cup \{\alpha\}, \quad Sf(\alpha \circ \beta) = Sf\alpha \cup Sf\beta \cup \{(\alpha \circ \beta)\} \;\; \text{for a binary logical connective } \circ$$

Similarly, we can define the **rank** of a formula φ,

$$rank\pi = 0 \text{ for prime formulas } \pi,$$
$$rank\neg\alpha = rank\alpha + 1, \quad rank(\alpha \circ \beta) = \max\{rank\alpha, rank\beta\} + 1 \;\; \text{for a binary logical connective } \circ$$

And we can also define the set of variables in φ,

$$var\pi = \{\pi\} \text{ for prime formulas } \pi,$$
$$var\neg\alpha = var\alpha, \quad var(\alpha \circ \beta) = var\alpha \cup var\beta \;\; \text{for a binary logical connective } \circ$$

In all definitions $\circ$ is either $\wedge$ or $\vee$.

So now that we have discussed the syntax of propositional logic, it is time to discuss its semantics; how we assign to formulas truth values. Recall the truth tables for $\wedge$, $\vee$, and $\neg$:

α	β	$\alpha \wedge \beta$
1	1	1
1	0	0
0	1	0
0	0	0

α	β	$\alpha \vee \beta$
1	1	1
1	0	1
0	1	1
0	0	0

α	$\neg\alpha$
1	0
0	1

These define how the logical connectives function as functions on $\{0, 1\}$.

1.1.6 Definition
A **propositional valuation**, or a **propositional model**, is a function

$$w\colon PV \longrightarrow \{0,1\}$$

We can extend it to a function $w\colon PV \longrightarrow \mathcal{F}$ as follows:

$$w(\alpha\wedge\beta) = w\alpha\wedge w\beta,\quad w(\alpha\vee\beta) = w\alpha\vee w\beta,\quad w\neg\alpha = \neg w\alpha$$

Notice that we would need to define, for example, $w(\alpha\to\beta) = w\alpha\to w\beta$ had $\to$ been an element of our logical signature. But since $\to$ is defined using $\wedge$ and $\neg$, we must prove this identity:

$$w(\alpha\to\beta) = w\neg(\alpha\wedge\neg\beta) = \neg w(\alpha\wedge\neg\beta) = \neg(w\alpha\wedge\neg w\beta) = w\alpha\to w\beta$$

This is of course not a coincidence, but a result of the fact that $\alpha\to\beta = \neg(\alpha\wedge\neg\beta)$ (where $\alpha,\beta\in\{0,1\}$). Notice that furthermore,

$$w\top = 1,\quad w\bot = 0$$

1.1.7 Proposition
The valuation of a formula is dependent only on its variables. Meaning if φ is a formula and w and w' are two valuations where $w\pi = w'\pi$ for all $\pi\in var\varphi$, then $w\varphi = w'\varphi$.

We will prove this by induction on φ. For prime formulas, this is obvious as $var\varphi = \{\varphi\}$ and then $w\varphi = w'\varphi$ by the proposition's assumption. For $\varphi = \alpha\wedge\beta$, we have that

$$w\varphi = w\alpha\wedge w\beta = w'\alpha\wedge w'\beta = w'\varphi$$

where the second equality is our inductive assumption. The proof for $\varphi = \alpha\vee\beta$ and $\varphi = \neg\alpha$ is similar.

■

Let us suppose that $PV = \{p_1, p_2, \ldots, p_n, \ldots\}$, then we define $\mathcal{F}_n$ to be the set of formulas φ such that $var\varphi\subseteq\{p_1,\ldots,p_n\}$.

1.1.8 Definition
A **boolean function** is a function

$$f\colon\{0,1\}^n\longrightarrow\{0,1\}$$

for some $n\geq 0$. The set of boolean functions of arity n is denoted $\mathbf{B}_n$. A formula $\varphi\in\mathcal{F}_n$ **represents** a boolean function $f\in\mathbf{B}_n$ (similarly, f is represented by φ), if for all valuations w,

$$w\varphi = f(w\vec{p})\qquad (w\vec{p} = (wp_1,\ldots,wp_n))$$

So for example, $\alpha = p_1\wedge p_2$ represents the function $f(p,q) = p\wedge q$. This is since

$$f(wp_1, wp_2) = wp_1\wedge wp_2 = w(p_1\wedge p_2) = w\alpha$$

Since valuations of $\varphi\in\mathcal{F}_n$ are defined by their values on $p_1,\ldots,p_n$, φ represents at most a single function f. In fact, it represents the function

$$\varphi^{(n)}(x_1,\ldots,x_n) = w\varphi$$

where w is any valuation such that $wp_i = x_i$ (all of these valuations valuate φ the same). Now, notice that $\mathcal{F}_n\subset\mathcal{F}_{n+1}$ and $\mathbf{B}_n\subset\mathbf{B}_{n+1}$ and so $\varphi\in\mathcal{F}_n$ represents a function in $\mathcal{B}_{n+1}$ as well. But this function is not essentially in $\mathcal{B}_n$ in the sense that its last argument does not impact its value. Formally we say that given a function $f\colon M^n\longrightarrow M$, we call its ith argument *fictional* if for all $x_1,\ldots,x_i,\ldots,x_n\in M$ and $x_i'\in M$:

$$f(x_1,\ldots,x_i,\ldots,x_n) = f(x_1,\ldots,x_i',\ldots,x_n)$$

An *essentially n-ary* function is a function with no fictional arguments.

1.1.9 Definition
Two formulas α and β are **equivalent** if for every valuation w, $w\alpha = w\beta$. This is denoted $\alpha \equiv \beta$.

It is immediate that α and β are equivalent if and only if they represent the same function. A simple example of equivalence is $\alpha \equiv \neg\neg\alpha$. The following equivalences are easy to verify and the reader should already be familiar with them (α, β, and γ are formulas):

$$\begin{array}{ccl}
\alpha \wedge (\beta \wedge \gamma) \equiv \alpha \wedge \beta \wedge \gamma & \alpha \vee (\beta \vee \gamma) \equiv \alpha \vee \beta\gamma & \text{(associativity)} \\
\alpha \wedge \beta \equiv \beta \wedge \alpha & \alpha \vee \beta \equiv \beta \vee \alpha & \text{(commutativity)} \\
\alpha \wedge \alpha \equiv \alpha & \alpha \vee \alpha \equiv \alpha & \text{(idempotency)} \\
\alpha \wedge (\alpha \vee \beta) \equiv \alpha & \alpha \vee \alpha \wedge \beta \equiv \alpha & \text{(absorption)} \\
\multicolumn{2}{c}{\alpha \wedge (\beta \vee \gamma) \equiv \alpha \wedge \beta \vee \alpha \wedge \gamma} & (\wedge\text{-distributivity}) \\
\multicolumn{2}{c}{\alpha \vee \beta \wedge \gamma \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)} & (\vee\text{-distributivity}) \\
\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta & \neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta & \text{(de Morgan rules)}
\end{array}$$

Furthermore,

$$\alpha \vee \neg\alpha \equiv \top, \quad \alpha \wedge \neg\alpha \equiv \bot, \quad \alpha \wedge \top \equiv \alpha \vee \bot \equiv \alpha$$

Since $\alpha \to \beta = \neg(\alpha \wedge \neg\beta)$, by de Morgan rules, this is equivalent to

$$\equiv \neg\alpha \vee \neg\neg\beta \equiv \neg\alpha \vee \beta$$

Notice that

$$\alpha \to \beta \to \gamma \equiv \neg\alpha \vee (\beta \to \gamma) \equiv \neg\alpha \vee \neg\beta \vee \gamma \equiv \neg(\alpha \wedge \beta) \vee \gamma \equiv \alpha \wedge \beta \to \gamma$$

Inductively, we see that

$$\alpha_1 \to \cdots \to \alpha_n \to \gamma \equiv \alpha_1 \wedge \cdots \wedge \alpha_n \to \gamma$$

We could go on, but I assume you get the point.
$\equiv$ is obviously reflexive, symmetric, and transitive: therefore it is an equivalence relation on $\mathcal{F}$. But moreso it is a *congruence relation*, meaning it respects connectives. Explicitly, for all formulas $\alpha, \beta, \alpha', \beta' \in \mathcal{F}$:

$$\alpha \equiv \alpha',\, \beta \equiv \beta' \implies \alpha \wedge \beta \equiv \alpha' \wedge \beta',\, \alpha \vee \beta \equiv \alpha' \vee \beta',\, \neg\alpha \equiv \neg\alpha'$$

Congruence relations will be discussed in more generality in later sections. Inductively, we can prove the following result:

1.1.10 Theorem (The Replacement Theorem)

Suppose α and α' are equivalent formulas. Let φ be some other formula, and define φ' to be the result of replacing all occurences of α within φ by α'. Then $\varphi \equiv \varphi'$.

This will be proven more generally later.

1.1.11 Definition
Prime formulas and their negations are called **literals**. A formula of the form $\alpha_1 \vee \cdots \vee \alpha_n$ where each α_i is a conjunction of literals is called a **disjunctive normal form**. And similarly a formula of the form $\alpha_1 \wedge \cdots \wedge \alpha_n$ where each α_i is a disjunction of literals is called a **conjunctive normal form**. We will use the abreviations DNF and CNF for disjunctive and conjunctive normal forms, respectively.

So a DNF is a formula of the form

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} \ell_{i,j}$$

where for every i, j, $\ell_{i,j}$ is a literal: a formula of the form $p_{i,j}$ or $\neg p_{i,j}$ for some prime formula $p_{i,j}$. Similarly a CNF is a formula of the form

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} \ell_{i,j}$$

Let us temporarily introduce the following notation: for a prime formula p, let

$$p^1 := p, \qquad p^0 := \neg p$$

This allows us to more concisely state and prove the following theorem:

1.1.12 Theorem
Every boolean function $f \in \mathbf{B}_n$ for $n > 0$ is representable by the DNF

$$\alpha_f := \bigvee_{f(\vec{x})=1} p_1^{x_1} \wedge \cdots \wedge p_n^{x_n}$$

and a CNF

$$\beta_f := \bigwedge_{f(\vec{x})=0} p_1^{\neg x_1} \wedge \cdots \wedge p_n^{\neg x_n}$$

Let w be a valuation and $\vec{p} = (p_1, \dots, p_n)$ then

$$w\alpha_f = \bigvee_{f(\vec{x})=1} wp_1^{x_1} \wedge \cdots \wedge wp_n^{x_n}$$

Notice that wp^x is equal to 1 if and only if $wp = x$: suppose $x = 0$ then $wp^x = \neg wp$, which is equal to 1 if and only if $wp = 0 = x$, and similar for $x = 1$. Thus $w\alpha_f = 1$ if and only if there exists a $\vec{x}$ such that $f(\vec{x}) = 1$ and $wp_1^{x_1} \wedge \cdots \wedge wp_n^{x_n} = 1$, meaning that for each i, $wp_i = x_i$. This means that $\vec{x} = w\vec{p}$, and so $f(w\vec{p}) = f(\vec{x}) = 1$. Similarly if $f(w\vec{p}) = 1$ then let $\vec{x} = w\vec{p}$, and then $wp_1^{x_1} \wedge \cdots \wedge wp_n^{x_n} = 1$ and $f(\vec{x}) = 1$, so $w\alpha_f = 1$. So $w\alpha_f = f(w\vec{p})$ for all valuations w, which means that f is represented by α_f, as required. The proof for β_f is similar. ■

Notice that since every formula represents a boolean function, which by above can be represented by a DNF and a CNF, we get that every formula is equivalent to a DNF and a CNF.

1.1.13 Corollary
Every formula is equivalent to a DNF and a CNF.

1.1.14 Definition
A logical signature σ is **functional complete** if every boolean function is representable by a formula in this signature.

By corollary 1.1.13, $\{\neg, \wedge, \vee\}$ is functional complete. Since

$$\alpha \vee \beta \equiv \neg(\neg\alpha \wedge \neg\beta), \quad \alpha \wedge \beta \equiv \neg(\neg\alpha \vee \neg\beta)$$

$\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are both functional complete. Thus in order to show that a logical signature σ is functional complete, it is sufficient to show that $\neg$ and $\wedge$ or $\neg$ and $\vee$ can be represented by σ.

Note

If f is a function, instead of writing $f(x)$ or fx, many times we will instead write x^f. This is more concise and may reduce confusion in the case that x itself is a string wrapped in parentheses.

Let us define the function $\delta\colon \mathcal{F} \longrightarrow \mathcal{F}$ on formulas recursively by $p^\delta = p$ for prime formulas p and

$$(\neg\alpha)^\delta = \neg\alpha^\delta, \quad (\alpha \wedge \beta)^\delta = \alpha^\delta \vee \beta^\delta, \quad (\alpha \vee \beta)^\delta = \alpha^\delta \wedge \beta^\delta$$

Alternatively, α^δ is simply the result of swapping all occurences of $\wedge$ with $\vee$, and all occurences of $\vee$ with $\wedge$. α^δ is called the *dual formula* of α. Notice that the dual formula of a DNF is a CNF, and vice versa.
Now, suppose $f \in \mathbf{B}_n$, then let us define the *dual* of f,

$$f^\delta(\vec{x}) := \neg f(\neg\vec{x})$$

where $\neg\vec{x} = (\neg x_1, \dots, \neg x_n)$. Notice that δ is idempotent:

$$f^{\delta^2}(\vec{x}) = \neg f^\delta(\neg\vec{x}) = \neg\neg f(\neg\neg\vec{x}) = f(\vec{x})$$

1.1.15 Theorem (The Duality Principle for Two-Valued Logic)
If α represents the function f, then α^δ represents f^δ.

We will prove this by induction on α. If $\alpha = p$ is prime, then this is trivial. Now suppose that α and β represent f_1 and f_2 respectively. Then $\alpha \wedge \beta$ represents $f(\vec{x}) = f_1(\vec{x}) \wedge f_2(\vec{x})$, and $(\alpha \wedge \beta)^\delta = \alpha^\delta \vee \beta^\delta$ represents $g(\vec{x}) = f_1^\delta(\vec{x}) \vee f_2^\delta(\vec{x})$ by the induction hypothesis. Now,

$$f^\delta(\vec{x}) = \neg f(\neg\vec{x}) = \neg(f_1(\neg\vec{x}) \wedge f_2(\neg\vec{x})) = \neg f_1(\neg\vec{x}) \vee \neg f_2(\neg\vec{x}) = f_1^\delta(\vec{x}) \vee f_2^\delta(\vec{x}) = g(\vec{x})$$

So $f^\delta = g$, meaning that $(\alpha \wedge \beta)^\delta$ does indeed represent f^δ. The proof for $\alpha \vee \beta$ is similar. Now suppose α represents f, then $\neg\alpha$ represents $\neg f$, and α^δ represents f^δ by the induction hypothesis. And so $(\neg\alpha)^\delta = \neg\alpha^\delta$ represents $\neg f^\delta$, which is equal to $(\neg f)^\delta$ since

$$(\neg f)^\delta(\vec{x}) = (\neg\neg f)(\neg\vec{x}) = \neg(\neg f(\neg\vec{x})) = \neg f^\delta(\vec{x})$$

And so $(\neg\alpha)^\delta$ represents $(\neg f)^\delta$, as required. ∎

1.1.16 Definition
Suppose α is a formula and w is a valuation. Instead of writing $w\alpha = 1$, we now write $w \vDash \alpha$, and this is read as "*w satisfies α*". If X is a set of formulas, we write $w \vDash X$ if $w \vDash \alpha$ for all $\alpha \in X$, and w is called a **propositional model** for X. A formula α (respectively a set of formulas X) is **satisfiable** if there is some valuation w such that $w \vDash \alpha$ (respectively $w \vDash X$). $\vDash$ is called the **satisfiability relation**.

$\vDash$ has the following immediate properties:

$$\begin{array}{rlrl} w \vDash p \iff & wp = 1 \ (p \in PV) & w \vDash \alpha \iff & w \nvDash \alpha \\ w \vDash \alpha \wedge \beta \iff & w \vDash \alpha \text{ and } w \vDash \beta & w \vDash \alpha \vee \beta \iff & w \vDash \alpha \text{ or } w \vDash \beta \end{array}$$

These properties uniquely define $\vDash$, meaning we could have defined $\vDash$ recursively by these properties.
Notice that

$$w \vDash \alpha \to \beta \iff \text{if } w \vDash \alpha \text{ then } w \vDash \beta$$

This is due to the definition of $\to$ coinciding with our common usage of implication. Had we not defined $\to$, but instead added it to our logical signature, this above equivalence would have to be taken in the definition of the satisfiability relation (when axiomized by the above properties).

1.1.17 Definition
α is **logically valid**, or a **tautology**, if $w \vDash \alpha$ for all valuations w. This is abbreviated by $\vDash \alpha$. A formula which cannot be satisfied; ie. for all valuations w, $w \nvDash \alpha$; is called a **contradiction**.

For example, $\alpha \vee \neg\alpha$ is a tautology, while $\alpha \wedge \neg\alpha$ and $\alpha \leftrightarrow \neg\alpha$ are contradictions for all formulas α. Notice that the negation of a tautology is a contradiction and vice versa. $\top$ is a tautology and $\bot$ is a contradiction. The following are important tautologies of implication (keep in mind how $\to$ associates from the right):

$p \to p$	(self-implication)
$(p \to q) \to (q \to r) \to (p \to r)$	(chain rule)
$(p \to q \to r) \to (q \to p \to r)$	(exchange of premises)
$p \to q \to p$	(premise change)
$(p \to q \to r) \to (p \to q) \to (p \to r)$	(Frege's formula)
$((p \to q) \to p) \to p$	(Peirce's formula)

1.1.18 Definition

Suppose X is a set of formulas and α a formula, we say that α is a **logical consequence** if $w \vDash \alpha$ for every model w of X. In other words,

$$w \vDash X \implies w \vDash \alpha$$

This is denoted $X \vDash \alpha$.

Notice that $\vDash$ here is used for logical consequence (the consequence relation), and we used it before as the symbol for the satisfiability relation. Context will make it clear as to its meaning. We use the notation $\alpha_1, \dots, \alpha_n \vDash \beta$ to mean $\{\alpha_1, \dots, \alpha_n\} \vDash \beta$. This justifies the notation for tautologies: α is a tautology if and only if $\varnothing \vDash \alpha$ (since every valuation models $\varnothing$), which is shortened by the above notation to $\vDash \alpha$.
And we also use $X \vDash \alpha, \beta$ to mean $X \vDash \alpha$ and $X \vDash \beta$. And $X, \alpha \vDash \beta$ to mean $X \cup \{\alpha\} \vDash \beta$.
The following are examples of logical consequences

$$\alpha, \beta \vDash \alpha \wedge \beta, \quad \alpha \wedge \beta \vDash \alpha, \beta$$
$$\alpha, \alpha \to \beta \vDash \beta$$
$$X \vDash \bot \implies X \vDash \alpha \quad \text{for all formulas } \alpha$$
$$X, \alpha \vDash \beta,\, X, \neg\alpha \vDash \beta \implies X \vDash \beta$$

The final example is true because if $w \vDash X$ then either $w \vDash \alpha$ or $w \vDash \neg\alpha$, and in either case $w \vDash \beta$.
Let us now state some obvious properties of the consequence relation:

$$\begin{array}{ll} \alpha \in X \implies X \vDash \alpha & (\textit{reflexivity}) \\ X \vDash \alpha \text{ and } X \subseteq X' \implies X' \vDash \alpha & (\textit{monotonicity}) \\ X \vDash Y \text{ and } Y \vDash \alpha \implies X \vDash \alpha & (\textit{transitivity}) \end{array}$$

1.1.19 Definition
A **propositional substitution** is a mapping from prime formulas to formulas, $\sigma\colon PV \longrightarrow \mathcal{F}$, which is extended to a mapping between formulas $\sigma\colon \mathcal{F} \longrightarrow \mathcal{F}$ recursively:

$$(\alpha \wedge \beta)^\sigma = \alpha^\sigma \wedge \beta^\sigma, \quad (\alpha \vee \beta)^\sigma = \alpha^\sigma \vee \beta^\sigma, \quad (\neg\alpha)^\sigma = \neg\alpha^\sigma$$

If X is a set of formulas, we define
$$X^\sigma = \{\varphi^\sigma \mid \varphi \in X\}$$

Besides being intuitively important, the following proposition gives more insight into the usefulness of substitutions:

1.1.20 Proposition
Let X be a set of formulas, and α a formula. Then

$$X \vDash \alpha \implies X^\sigma \vDash \alpha^\sigma$$

Thus in a sense consequence is invariant under substitution.

Let w be a valuation, then we define w^σ as follows:

$$w^\sigma p = wp^\sigma$$

for prime formulas p. Now we claim that
$$w \vDash \alpha^\sigma \iff w^\sigma \vDash \alpha$$

We will prove this by induction on α. In the case that $\alpha = p$ is prime, then $w \vDash p^\sigma$ if and only if $wp^\sigma = w^\sigma p = 1$, and so this is if and only if $w^\sigma \vDash p$. Now by induction,

$$w \vDash (\alpha \wedge \beta)^\sigma \iff w \vDash \alpha^\sigma \text{ and } w \vDash \beta^\sigma \iff w^\sigma \vDash \alpha \text{ and } w^\sigma \vDash \beta \iff w^\sigma \vDash \alpha \wedge \beta$$

where the second equivalence is due to the induction hypothesis. The proof for formulas of the form $\alpha \vee \beta$ and $\neg\alpha$ proceed in a similar fashion.
Now, suppose $w \vDash X^\sigma$. This is if and only if $w \vDash \varphi^\sigma$ for all $\varphi \in X$, which is if and only if $w^\sigma \vDash \varphi$ by above. So $w \vDash X^\sigma$ if and only if $w^\sigma \vDash X$. And so if $X \vDash \alpha$ then let $w \vDash X^\sigma$, then $w^\sigma \vDash X$ meaning $w^\sigma \vDash \alpha$ and so $w \vDash \alpha^\sigma$ by above. So $X^\sigma \vDash \alpha^\sigma$ as required. ∎

These four properties,

$$\begin{array}{ll} \alpha \in X \implies X \vDash \alpha & (\textit{reflexivity}) \\ X \vDash \alpha \text{ and } X \subseteq X' \implies X' \vDash \alpha & (\textit{monotonicity}) \\ X \vDash Y \text{ and } Y \vDash \alpha \implies X \vDash \alpha & (\textit{transitivity}) \\ X \vDash \alpha \implies X^\sigma \vDash \alpha^\sigma & (\textit{substitution invariance}) \end{array}$$

are what define general consequence relations, and form the basis for a general theory of logical systems. Another property is

$$X \vDash \alpha \implies X_0 \vDash \alpha \text{ for some finite } X_0 \subseteq X \quad (\textit{finitary})$$

We will show in the next subsection that this is a property of our consequence relation.
Another property is the property

$$X, \alpha \vDash \beta \iff X \vDash \alpha \to \beta$$

termed the *semantic deduction theorem*. Let us prove the first direction: suppose w is a model of X, then if $w \vDash \alpha$ it is a model of $X \cup \{\alpha\}$ and so $w \vDash \beta$. So we have shown that if $w \vDash \alpha$, then $w \vDash \beta$, meaning $w \vDash \alpha \to \beta$ and so $X \vDash \alpha \to \beta$. end for the converse, if $w \vDash X, \alpha$ then it is a model of X and so $w \vDash \alpha, \alpha \to \beta$ and thus $w \vDash \beta$.
We can show by induction a generalization of this:

$$X, \alpha_1, \dots, \alpha_n \vDash \beta \iff X \vDash \alpha_1 \to \cdots \to \alpha_n \to \beta \iff X \vDash \alpha_1 \wedge \cdots \wedge \alpha_n \to \beta$$

The induction step is simple: take $X' = X \cup \{\alpha_1\}$ we get by our induction hypothesis,

$$\begin{aligned} X, \alpha_1, \dots, \alpha_n \vDash \beta \iff X', \alpha_2, \dots, \alpha_n \vDash \beta \iff X, \alpha_1 \vDash \alpha_2 \to \cdots \to \alpha_n \to \beta \\ \iff X \vDash \alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \beta \end{aligned}$$

as required. The deduction theorem makes proving many tautologies relating to implication much easier.

1.2 Gentzen Calculi

To begin this subsection, we will define a derivability relation $\vdash$ which axiomatizes the important properties of the consequence relation $\vDash$. Our goal is to show that by using these axioms, $\vdash$ is equivalent to $\vDash$, and this will allow us to prove important facts about $\vDash$, namely its finitaryness.

1.2.1 Definition
We define **Gentzen style sequent calculus** of $\vdash$ as follows: $X \vdash \alpha$ is to be read as "α is derivable from X" where α is a formula and X is a set of formulas. A pair $(X, \alpha) \in \mathcal{P}(\mathcal{F}) \times \mathcal{F}$, or more suggestively written $X \vdash \alpha$, is called a **sequent**. Gentzen-style rules have the form

$$\frac{X_1 \vdash \alpha_1 \mid \cdots \mid X_n \vdash \alpha_n}{X \vdash \alpha}$$

Which is to be understood as meaning that if for every i, $X_i \vdash \alpha_i$, then $X \vdash \alpha$.
Gentzen calculus has the following basic rules:

$$\begin{array}{llll} (\text{IS}) & \dfrac{}{\alpha \vdash \alpha} & (\text{MR}) & \dfrac{X \vdash \alpha}{X' \vDash \alpha} \quad (X \subseteq X') \\ (\wedge 1) & \dfrac{X \vdash \alpha, \beta}{X \vdash \alpha \wedge \beta} & (\wedge 2) & \dfrac{X \vdash \alpha \wedge \beta}{X \vdash \alpha, \beta} \\ (\neg 1) & \dfrac{X \vdash \alpha, \neg\alpha}{X \vdash \beta} & (\neg 2) & \dfrac{X, \alpha \vdash \beta \mid X, \neg\alpha \vdash \beta}{X \vdash \beta} \end{array}$$

(IS means "initial sequent", MR means monotonacity rule.)
Now we say that α is derivable from X, in short $X \vdash \alpha$, if $S_n = X \vdash \alpha$ and there exists a sequence of sequents $(S_0; \dots; S_n)$ where for every S_i, S_i is either an initial sequent (IS) or derivable using the basic rules from previous sequents in the sequence.

For example, we can derive $\alpha \wedge \beta$ from $\{\alpha, \beta\}$, meaning $\alpha, \beta \vdash \alpha \wedge \beta$. This can be done by the sequence:

$$\begin{pmatrix} \alpha \vdash \alpha \; ; \; \alpha, \beta \vdash \alpha \; ; \; \beta \vdash \beta \; ; \; \alpha, \beta \vdash \beta \; ; \; \alpha, \beta \vdash \alpha \wedge \beta \\ \text{IS} \; ; \; \text{MR} \; ; \; \text{IS} \; ; \; \text{MR} \; ; \; \wedge 1 \end{pmatrix}$$

Let us prove some more useful rules

$$\frac{X, \neg\alpha \vdash \alpha}{X \vdash \alpha}$$

(¬-elimination)

1	$X, \alpha \vdash \alpha$	(IS), (MR)
2	$X, \neg\alpha \vdash \alpha$	supposition
3	$X \vdash \alpha$	(¬2)

$$\frac{X, \neg\alpha \vdash \beta, \neg\beta}{X \vdash \alpha}$$

(reductio ad absurdum)

1	$X, \neg\alpha \vdash \beta, \neg\beta$	supposition
2	$X, \neg\alpha \vdash \alpha$	(¬1)
3	$X \vdash \alpha$	¬-elimination

$$\frac{X \vdash \alpha \to \beta}{X, \alpha \vdash \beta}$$

(→-elimination)

1	$X, \alpha, \neg\beta \vdash \alpha, \neg\beta$	(IS), (MR)
2	$X, \alpha, \neg\beta \vdash \alpha \wedge \neg\beta$	(∧1)
3	$X \vdash \neg(\alpha \wedge \neg\beta)$	supposition
4	$X, \alpha, \neg\beta \vdash \neg(\alpha \wedge \neg\beta)$	(MR)
5	$X, \alpha, \neg\beta \vdash \beta$	(¬1) on 2 and 4
6	$X, \alpha \vdash \beta$	¬-elimination

$$\frac{X \vdash \alpha \mid X, \alpha \vdash \beta}{X \vdash \beta}$$

(cut rule)

1	$X, \neg\alpha \vdash \alpha$	supposition, (MR)
2	$X, \neg\alpha \vdash \neg\alpha$	(IS), (MR)
3	$X, \neg\alpha \vdash \beta$	(¬1)
4	$X, \alpha \vdash \beta$	supposition
5	$X \vdash \beta$	(¬2) on 3 and 4

$$\frac{X, \alpha \vdash \beta}{X \vdash \alpha \to \beta}$$

(→-introduction)

1	$X, \alpha \wedge \neg\beta, \alpha \vdash \beta$	supposition, (MR)
2	$X, \alpha \wedge \neg\beta \vdash \alpha$	(IS), (MR), (∧2)
3	$X, \alpha \wedge \neg\beta \vdash \beta$	cut rule
4	$X, \alpha \wedge \neg\beta \vdash \neg\beta$	(IS), (MR), (∧2)
5	$X, \alpha \wedge \neg\beta \vdash \alpha \to \beta$	(¬1)
6	$X, \neg(\alpha \wedge \neg\beta) \vdash \alpha \to \beta$	(IS), (MR)
7	$X \vdash \alpha \to \beta$	(¬2) on 5 and 6

$$\frac{X \vdash \alpha, \alpha \to \beta}{X \vdash \beta}$$

(modus ponens)

1	$X \vdash \alpha \to \beta$	supposition
2	$X, \alpha \to \beta$	→-elimination
3	$X \vdash \alpha$	supposition
4	$X \vdash \beta$	cut rule

→-elimination and →-introduction give us the *syntactic deduction theorem*:

$$X, \alpha \vdash \beta \iff X \vdash \alpha \to \beta$$

Let R be a rule of the form

$$R\colon \frac{X_1 \vdash \alpha_1 \mid \cdots \mid X_n \vdash \alpha_n}{X \vdash \alpha}$$

Then we say that a property of sequents $\mathcal{E}$ is *closed under* R if $\mathcal{E}(X_1, \alpha_1), \ldots, \mathcal{E}(X_n, \alpha_n)$ implies $\mathcal{E}(X, \alpha)$.

1.2.2 Proposition (Principle of Rule Induction)

Let $\mathcal{E}$ be a property of sequents which is closed under all the basic rules of $\vdash$. Then $X \vdash \alpha$ implies $\mathcal{E}(X, \alpha)$.

We will prove this by induction on the length of the derivation of $S = X \vdash \alpha$, n. If $n = 1$ then $X \vdash \alpha$ must be an initial sequent and so by assumption $\mathcal{E}(X, \alpha)$. For the induction step, suppose the derivation is $(S_0; \ldots; S_n)$, so $S = S_n$. Then by our inductive hypothesis $\mathcal{E}S_i$ for all $i < n$. If S is an initial sequent then $\mathcal{E}S$ holds by assumption. Otherwise S is obtained by applying a basic rule on some of the sequents S_i for $i < n$. And since $\mathcal{E}S_i$ and $\mathcal{E}$ is closed under basic rules, we have that $\mathcal{E}S$ as required. ■

1.2.3 Lemma (Soundness of $\vdash$)

If $X \vdash \alpha$ then $X \vDash \alpha$. More suggestively,

$$\vdash \subseteq \vDash$$

Using the principle of rule induction, let $\mathcal{E}(X, \alpha)$ mean $X \vDash \alpha$ (formally this means $\mathcal{E} = \{(X, \alpha) \mid X \vDash \alpha\}$). Then we must show that $\mathcal{E}$ is closed under all the basic rules of $\vdash$. This means that we must show that

$$\alpha \vDash \alpha, \quad X \vDash \alpha \implies X' \vDash \alpha \text{ for } X \subseteq X', \quad X \vDash \alpha, \beta \iff X \vDash \alpha \wedge \beta,$$
$$X \vDash \alpha, \neg\alpha \implies X \vDash \beta, \quad X, \alpha \vDash \beta \text{ and } X, \neg\alpha \vDash \beta \implies X \vDash \beta$$

These are all readily verifiable (and some we have already shown). So $\mathcal{E}$ is indeed closed under all the basic rules of $\vdash$, and so $\mathcal{E}(X, \alpha)$ (meaning $X \vDash \alpha$) implies $X \vdash \alpha$. ■

The property above is called *soundness*, meaning $\vdash$ does not derive anything "incorrect".

1.2.4 Theorem

If $X \vdash \alpha$ then there exists a finite subset $X_0 \subseteq X$ such that $X_0 \vdash \alpha$.

Let $\mathcal{E}(X, \alpha)$ be the property that there exists a finite subset $X_0 \subseteq X$ such that $X_0 \vdash \alpha$. We will show that $\mathcal{E}$ is closed under the basic rules of $\vdash$. Trivially, $\mathcal{E}(X, \alpha)$ holds for $X = \{\alpha\}$, meaning $\mathcal{E}$ holds for (IS). And similarly if $\mathcal{E}(X, \alpha)$ and $X \subseteq X'$, since there exists a finite $X_0 \subseteq X$ such that $X_0 \vdash \alpha$, this same X_0 is a subset of X' and so $\mathcal{E}(X', \alpha)$ so $\mathcal{E}$ is closed under (MR).
Now if $\mathcal{E}(X, \alpha)$ and $\mathcal{E}(X, \beta)$ then suppose $X_1 \vdash \alpha$ and $X_2 \vdash \beta$ where $X_1, X_2 \subseteq X$ are finite. Then $X_0 = X_1 \cup X_2$ is finite, $X_0 \vdash \alpha, \beta$ and so $X_0 \vdash \alpha \wedge \beta$, and since $X_0 \subseteq X$ is finite, $\mathcal{E}(X, \alpha \wedge \beta)$ so $\mathcal{E}$ is closed under ($\wedge$1). Closure under the rest of the basic rules can be shown similarly. ■

1.2.5 Definition

A set of formulas X is **inconsistent** if $X \vdash \alpha$ for every formula α. If X is not inconsistent, it is termed **consistent**. X is **maximally consistent** if X is consistent but for every proper superset $X \subset Y$, Y is inconsistent.

Notice that X is inconsistent if and only if $X \vdash \bot$. Obviously if X is inconsistent, $X \vdash \bot$. Conversely, if $X \vdash \bot$ then $X \vdash p_1 \wedge \neg p_1$ and so by ($\wedge$2), $X \vdash p_1, \neg p_2$ and thus by ($\neg$1) for all formulas α, $X \vdash \alpha$.
Furthermore, if X is consistent it is maximally consistent if and only if for every formula α, either $\alpha \in X$ or $\neg\alpha \in X$ exclusively. If neither α nor $\neg\alpha$ are in X, then since X is maximally consistent, $X, \alpha \vdash \bot$ and $X, \neg\alpha \vdash \bot$ and therefore by ($\neg$2), $X \vdash \bot$ contradicting X's consistency. And if X contains α or $\neg\alpha$ for every formula α, then it is maximal: adding another formula α would mean that $\alpha, \neg\alpha \in X$ and so by (IS), (MR), and ($\neg$2), X would be inconsistent.
This means that maximally consistent sets X are *deductively closed*:

$$X \vdash \alpha \iff \alpha \in X$$

Obviously if $\alpha \in X$ then by (IS) and (MR), $X \vdash \alpha$. Now suppose that $X \vdash \alpha$, then since $\alpha \in X$ or $\neg\alpha \in X$, we cannot have $\neg\alpha \in X$ since X is consistent. Therefore $\alpha \in X$.

1.2.6 Lemma

The derivability relation has the following properties:

$$\texttt{C}^+: \quad X \vdash \alpha \iff X, \neg\alpha \vdash \bot, \qquad \texttt{C}^-: \quad X \vdash \neg\alpha \iff X, \alpha \vdash \bot$$

Meaning α is derivable from X if and only if $X \cup \{\neg\alpha\}$ is inconsistent. And similarly $\neg\alpha$ is derivable from X if and only if $X \cup \{\alpha\}$ is inconsistent.

We will prove $\mathtt{C}^+$. Suppose $X \vdash \alpha$, then $X, \neg\alpha \vdash \alpha$ by (MR) and $X, \neg\alpha \vdash \neg\alpha$ by (IS) and (MR). Thus by $(\neg 1)$, $X, \neg\alpha \vdash \beta$ for all formulas β by $(\neg 1)$ and in particular, $X, \neg\alpha \vdash \bot$. Now suppose $X, \neg\alpha \vdash \bot$ then by $(\wedge 2)$ and $(\neg 1)$, we have $X, \neg\alpha \vdash \alpha$ then by $\neg$-elimination, $X \vdash \alpha$. $\mathtt{C}^-$ is proven similarly. ■

1.2.7 Lemma (Lindenbaum's Theorem)
Every consistent set of formulas $X \subseteq \mathcal{F}$ can be extended to a maximally consistent set of formulas $X \subseteq X' \subseteq \mathcal{F}$.

Let us define the set

$$H = \{Y \subseteq \mathcal{F} \mid Y \text{ is consistent and } X \subseteq Y\}$$

This is partially ordered with respect to $\subseteq$, and since $X \in H$, H is not empty. Let $C \subseteq H$ be a chain, meaning that for every $Z, Y \in C$, either $Z \subseteq Y$ or $Y \subseteq Z$. Now we claim that $U = \bigcup C$ is an upper bound for C. So we must show that $U \in H$. Suppose not, then U is not consistent meaning $U \vdash \bot$. But then there must exist a finite $U_0 = \{\alpha_1, \dots, \alpha_n\} \subseteq U$ such that $U_0 \vdash \bot$. Now suppose $\alpha_i \in Y_i \in C$ then since C is linearly ordered, we can assume that every Y_i is contained within Y_n. But then by (MR), $Y_n \vdash \bot$ which conradicts $Y_n \in H$ being consistent.
So U is consistent and so $U \in H$, and obviously for every $Y \in C$, $Y \subseteq U$. So U is an upper bound for C, meaning that every chain in H has an upper bound in H, and so by Zorn's Lemma, H has a maximal element. This maximal element, call it X', is precisely a maximally consistent set containing X: it is consistent and contains X since it is in H, and it is maximal in H so for every $X \subseteq Y$, $Y \notin H$ so Y is inconsistent. ■

1.2.8 Lemma
A maximally consistent set of formulas X has the following property:

$$X \vdash \neg\alpha \iff X \nvdash \alpha$$

for all formulas α.

If $X \vdash \neg\alpha$ then $X \nvdash \alpha$ due to X's consistency. If $X \nvdash \alpha$ then $X \cup \{\neg\alpha\}$ is consistent in lieu of $\mathtt{C}^+$. But since X is maximal, $X \cup \{\neg\alpha\} = X$ meaning $\neg\alpha \in X$ and so by (IS) and (MR), $X \vdash \neg\alpha$. ■

1.2.9 Lemma
Maximally consistent sets are satisfiable.

Suppose X is maximally consistent, then let us define the valuation w by $w \vDash p \iff X \vdash p$. Then we claim that

$$X \vdash \alpha \iff w \vDash \alpha$$

This is trival for prime formulas. Now if $X \vdash \alpha \wedge \beta$:

$$X \vdash \alpha \wedge \beta \iff X \vdash \alpha, \beta \iff w \vDash \alpha, \beta \iff w \vDash \alpha \wedge \beta$$

where the second equivalence is due to the induction hypothesis. And if $X \vdash \neg\alpha$:

$$X \vdash \neg\alpha \iff X \nvdash \alpha \iff w \nvDash \alpha \iff w \vDash \neg\alpha$$

The first equivalence is due to the previous lemma, and the second is due to the induction hypothesis. And therefore $w \vDash X$, meaning X is satisfiable. ■

1.2.10 Theorem (The Completeness Theorem)
Let X and α be an arbitrary set of formulas and formula respectively. Then $X \vdash \alpha$ if and only if $X \vDash \alpha$. More suggestively,

$$\vdash \,=\, \vDash$$

We have already shown that $\vdash \,\subseteq\, \vDash$ and so all that remains is to show the converse. Suppose that $X \nvdash \alpha$, then $X, \neg\alpha$ is consistent by $\mathtt{C}^+$. Thus it can be extended to a maximally consistent set $X, \neg\alpha \subseteq X'$ which is satisfiable. Therefore so is $X, \neg\alpha$, which means that $X \nvDash \alpha$. ■

We get the following theorem as an immediate result from The Completeness Theorem and theorem 1.2.4:

1.2.11 Theorem
$X \vDash \alpha$ if and only if $X_0 \vDash \alpha$ for a finite $X_0 \subseteq X$.

1.2.12 Theorem (The Compactness Theorem)
A set $X \subseteq \mathcal{F}$ is satisfiable if and only if every finite $X_0 \subseteq X$ is satisfiable.

Obviously if X is satisfiable, so is $X_0 \subseteq X$. Now if X is not satisfiable, then $X \vdash \bot$ and so there exists a finite $X_0 \subseteq X$ such that $X_0 \vdash \bot$ (and so $X_0 \vDash \bot$) by the previous theorem. And so if X is not satisfiable, there exists a finite $X_0 \subseteq X$ which is not satisfiable. ■

Let us now give some examples of applications of the compactness theorem.

1.2.13 Proposition
Every set M can be linearly (also known as totally) ordered.

If M is finite, this is trivial: if $M = \{m_1, \dots, m_n\}$ simply define $m_1 < \cdots < m_n$. Now let M be any set, let us define the propositional variable (aka prime formula) p_{ab} for every $(a, b) \in M \times M$. This will represent $a < b$. So we define X to be the set of the following formulas, which represents M being linearly ordered,

$$\begin{aligned} \neg p_{aa} \quad & (a \in M), \\ p_{ab} \wedge p_{bc} \to p_{ac} \quad & (a, b, c \in M), \\ p_{ab} \vee p_{ba} \quad & (a \neq b \in M) \end{aligned}$$

If X is satisfiable, suppose $w \vDash X$, then we define the linear order $a < b$ if and only if $w \vDash p_{ab}$. Thus X is precisely the set of conditions necessary for $<$ to be a linear order: the first condition is irreflexivity, the second is transitivity, and the third totality (antisymmetry is gained through the combination of irreflexitivity and transitivity).
So if X is satisfiable, then M can be linearly ordered. By the compactness theorem, we need only to show that every finite $X_0 \subseteq X$ is satisfiable. If $X_0 \subseteq X$ is finite, then let us define M_0 to be the set of all symbols in M which occur in formulas in X_0. Since X_0 is finite, so is M_0 and therefore M_0 can be linearly ordered. Let us define $w_0 \vDash p_{ab} \iff a < b$ in M_0, then $w_0 \vDash X_0$. So by the compactness theorem X is satisfiable, as required. ■

Recall that showing that every set can be well-ordered (the well-ordering theorem) is equivalent to the axiom of choice. Since the compactness theorem is actually weaker than the axiom of choice, the linear ordering theorem (what we just showed) is weaker than the well-ordering theorem. Which is not surprising.

1.2.14 Proposition
A graph is k-colorable if and only if every finite subgraph is k-colorable.

A *graph* is a pair $G = (V, E)$ where V is a set of *vertices* and E is a set of *edges*. E is a subset of $\{\{v, u\} \mid v \neq u \in V\}$. The graph G is k-colorable if V can be partitioned into k *color classes*: $V = C_1 \cup \cdots \cup C_k$ such that if $a, b \in C_i$ then $\{a, b\} \notin E$, meaning two neighboring vertices do not have the same color.
Obviously if a graph is k-colorable, so is every subgraph. To show the converse, let $G = (V, E)$ be a graph, then let us define the set of formulas X, where prime formulas are of the form $p_{a,i}$ where $a \in V$ and $1 \leq i \leq k$:

$$\begin{aligned} & p_{a,1} \vee \cdots \vee p_{a,k} (a \in V) \\ & \quad \neg(p_{a,i} \wedge p_{a,j})(a \in V,\, 1 \leq i < j \leq k) \\ & \quad \neg(p_{a,i} \wedge p_{b,i})(\{a, b\} \in E,\, 1 \leq i \leq k) \end{aligned}$$

If X is satisfiable, $w \vDash X$, then we define $C_i = \{a \in V \mid w \vDash p_{a,i}\}$, ie. we color $a \in V$ with the color i if and only if $p_{a,i}$ is satisfied. Then $V = C_1 \cup \cdots \cup C_k$ since for every $a \in V$, $w \vDash p_{a,1} \vee \cdots \vee p_{a,k}$, so for every $a \in V$ there exists an $1 \leq i \leq k$ such that $w \vDash p_{a,i}$ so $a \in C_i$. And $C_i \cap C_j = \varnothing$ in lieu of $\neg(p_{a,i} \wedge p_{a,j})$. And if $\{a, b\} \in E$ then a and b cannot be in the same color class by $\neg(p_{a,i} \wedge p_{b,i})$. So the C_is give a valid k-coloring of G.
Let $X_0 \subseteq X$ be finite, then let us define $G_0 = (V_0, E_0)$ where V_0 is the set of vertices appearing in formulas in X_0, and E_0 be the edges connecting them. By assumption, G_0 is k-colorable since it is finite. Now we define the

valuation w_0 such that $w_0 \vDash p_{a,i}$ if and only if a is in the ith color class for $a \in V_0$. This must model X_0 since X_0 includes only statements saying that G_0 can be k-colored. So by the compactness theorem, X is satisfiable, as required. ■

There are more examples of applications of the compactness theorem. For example, the ultrafilter theorem, which we will visit later on.

1.3 Hilbert Calculi

In this subsection we will define another form of sequent calculus.

1.3.1 Definition

A define $\Lambda \subseteq \mathcal{F}$ to be a set of axioms, called the **logical axiom scheme**. Now, let Γ be a set of **rules of inference**, predicates of the form $R \in \Lambda^n \times \Lambda$ for $n > 0$, where $R((\varphi_1, \dots, \varphi_n), \varphi)$ which is to be understood as "*if* $\varphi_1, \dots, \varphi_n$ *then* φ".
If $X \subseteq \mathcal{F}$ is a set of formulas, then a **proof** is a sequence $\Phi = (\varphi_0, \dots, \varphi_n)$ where for every i, φ_i is either in $X \cup \Lambda$ or there exists a rule of inference $R \in \Gamma$ and indexes $i_1, \dots, i_n < i$ such that $R((\varphi_{i_1}, \dots, \varphi_{i_n}), \varphi)$.
In such a case, φ_n is termed **derivable** (or **provable**) from X, and is written $X \mathrel{|}\!\sim \varphi_n$ ($\mathrel{|}\!\sim$ to differentiate it from the derivability relation $\vdash$ from the previous subsection).

Hilbert-style calculi will use the following axiom scheme Λ:

$\Lambda1$	$(\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$	$\Lambda2$	$\alpha \to \beta \to \alpha \wedge \beta$
$\Lambda3$	$\alpha \wedge \beta \to \alpha, \quad \alpha \wedge \beta \to \beta$	$\Lambda4$	$(\alpha \to \neg\beta) \to \beta \to \neg\alpha$

And there is only a single rule of inference: $R((\alpha, \alpha \to \beta), \beta)$ called *modus ponens*, abbreviated MP. Essentially if α and $\alpha \to \beta$ then β.
The finiteness theorem for $\mathrel{|}\!\sim$ is immediate, since $X \mathrel{|}\!\sim \alpha$ requires a *finite* proof from X. And notice that

$$X \mathrel{|}\!\sim \alpha, \alpha \to \beta \implies X \mathrel{|}\!\sim \beta$$

Since if $\Phi_1 = (\varphi_0, \dots, \varphi_n)$ is a proof of α, and $\Phi_2 = (\varphi'_0, \dots, \varphi'_m)$ is a proof of $\alpha \to \beta$, then

$$\Phi = (\varphi_0, \dots, \varphi_n, \varphi'_0, \dots, \varphi'_m, \beta)$$

is a proof of $\alpha \to \beta$.

1.3.2 Proposition (Principle of Induction for $\mathrel{|}\!\sim$)

Let X be a set of formulas and $\mathcal{E}$ a property of formulas. Then if

(1) $\mathcal{E}\alpha$ is true for all $\alpha \in X \cup \Lambda$, and

(2) $\mathcal{E}\alpha$ and $\mathcal{E}\alpha \to \beta$ implies $\mathcal{E}\beta$ for all formulas α, β.

Then $X \mathrel{|}\!\sim \alpha$ implies $\mathcal{E}\alpha$.

We will prove this by induction on n, the length of the proof of α. If $n = 1$ then α is in $X \cup \Lambda$ and so by assumption $\mathcal{E}\alpha$. Now suppose $\Phi = (\varphi_0, \dots, \varphi_n)$ is a proof of $\alpha = \varphi_n$. If $\alpha \in X \cup \Lambda$ then by assumption $\mathcal{E}\alpha$. Otherwise Φ must contain formulas of the form α_i and $\alpha_i \to \alpha$. Since initial segments of proofs are themselves proofs, by our inductive hypothesis $\mathcal{E}\varphi_i$ for $i < n$. And thus $\mathcal{E}\alpha_i$ and $\mathcal{E}\alpha_i \to \alpha$ and so $\mathcal{E}\alpha$ as required. ■

This can obviously be generalized to a principle of induction for general rules of inferences, where the second condition is replaced with a general notion of closure under rules of inference.
Now we can show that $\mathrel{|}\!\sim \subseteq \vDash$, meaning if $X \mathrel{|}\!\sim \alpha$ then $X \vDash \alpha$ by defining the property $\mathcal{E}\alpha := X \vDash \alpha$. Since Λ contains only tautologies, for every $\alpha \in X \cup \Lambda$, $X \vDash \alpha$ meaning $\mathcal{E}\alpha$ for all $\alpha \in X \cup \Lambda$. And if $X \vDash \alpha$ and $X \vDash \alpha \to \beta$ then we know $X \vDash \beta$. So $\mathcal{E}$ satisfies the inductive properties stated above, meaning $X \mathrel{|}\!\sim \alpha$ implies $X \vDash \alpha$ as required.
Now, obviously $\mathrel{|}\!\sim$ is reflexive, monotonic, and transitive. Reflexivity follows directly from its definition. Monotonicity follows because a proof in X is also a proof in $X \subseteq X'$. And transitivity follows because if $X \mathrel{|}\!\sim Y$ and $Y \mathrel{|}\!\sim \alpha$, then by concatenating the proofs of $\varphi \in Y$ in X together with the proof of α in Y gives a proof of α in X.
Our goal for the remainder of this subsection is showing that $\mathrel{|}\!\sim = \vDash$, we will do this by showing that $\vdash \subseteq \mathrel{|}\!\sim$. As explained above, $\mathrel{|}\!\sim$ is reflexive and monotonic, meaning it satisfies (IS) and (MR) of the Gentzen-style calculus $\vdash$.

1.3.3 Lemma

(1) If $X \vdash \alpha \to \neg\beta$ then $X \vdash \beta \to \neg\alpha$

(2) $\vdash \alpha \to \beta \to \alpha$

(3) $\vdash \alpha \to \alpha$

(4) $\vdash \alpha \to \neg\neg\alpha$

(5) $\vdash \beta \to \neg\beta \to \alpha$

(1) By $\Lambda 4$, we have $X \vdash (\alpha \to \neg\beta) \to \beta \to \neg\alpha$, and since $X \vdash \alpha \to \neg\beta$ by modus ponens we get $X \vdash \beta \to \neg\alpha$.

(2) By $\Lambda 3$, $\vdash \beta \wedge \neg\alpha \to \neg\alpha$, and so by (1) we have $\vdash \alpha \to \neg(\beta \wedge \neg\alpha) = \alpha \to \beta \to \alpha$.

(3) Let $\gamma = \alpha$ and $\beta = \alpha \to \alpha$, then $\Lambda 1$ gives

$$\vdash (\alpha \to (\alpha \to \alpha) \to \alpha) \to (\alpha \to \alpha \to \alpha) \to \alpha \to \alpha$$

We know by (2), $\vdash \alpha \to (\alpha \to \alpha) \to \alpha$ and $\vdash \alpha \to \alpha \to \alpha$, so by applying modus ponens twice we get $\vdash \alpha \to \alpha$.

(4) Since $\vdash \neg\alpha \to \neg\alpha$ by (3), and applying (1) gives $\vdash \alpha \to \neg\neg\alpha$.

(5) By $\Lambda 3$, $\vdash \neg\beta \wedge \neg\alpha \to \neg\beta$, applying (1) gives $\vdash \beta \to \neg(\neg\beta \to \neg\alpha) = \beta \to \neg\beta \to \alpha$. ■

Since $\Lambda 3$ gives $\alpha \wedge \beta \to \alpha, \beta$, $\vdash$ satisfies ($\wedge 2$) of $\vdash$. $\Lambda 2$ gives $\alpha \to \beta \to (\alpha \wedge \beta)$ and so by applying MP twice, we get $\alpha, \beta \vdash \alpha \wedge \beta$ and so $\vdash$ satisfies ($\wedge 1$) of $\vdash$. Now by (5) of the above lemma, since $\vdash \alpha \to \neg\alpha \to \beta$, by applying MP twice we get that $X, \alpha, \neg\alpha \vdash \beta$ for all formulas β. By transitivity, this means that $X \vdash \alpha, \neg\alpha$ implies $X \vdash \beta$. Thus $\vdash$ satisfies (IS), (MR), ($\wedge 1$), ($\wedge 2$), and ($\neg 1$) of $\vdash$. We will now do a bit more work to show that it also satisfies ($\neg 2$).

1.3.4 Lemma (The Deduction Theorem)

$X, \alpha \vdash \gamma$ implies $X \vdash \alpha \to \gamma$.

We will prove this using the principle of induction for $\vdash$. Let $\mathcal{E}\gamma$ mean $X \vdash \alpha \to \gamma$, we will show that $X, \alpha \vdash \gamma$ implies $\mathcal{E}\gamma$ by showing $\mathcal{E}$ is closed under the inductive properties stated in the Principle of Induction for $\vdash$. If $\gamma \in \Lambda \cup X \cup \{\alpha\}$, if $\gamma = \alpha$ then we showed above that $X \vdash \alpha \to \alpha$. Otherwise if $\gamma \in X \cup \Lambda$ then $X \vdash \gamma$ and $X \vdash \gamma \to \alpha \to \gamma$, so by MP $X \vdash \alpha \to \gamma$ meaning $\mathcal{E}\gamma$ as required.
Now, if $\mathcal{E}\beta$ and $\mathcal{E}\beta \to \gamma$, meaning $X \vdash \alpha \to \beta$ and $X \vdash \alpha \to \beta \to \gamma$. Then by $\Lambda 1$, applying MP twice gives $X \vdash \alpha \to \gamma$ as required. ■

1.3.5 Lemma

$\vdash \neg\neg\alpha \to \alpha$

By $\Lambda 3$ and MP, we have $\neg\neg\alpha \wedge \neg\alpha \vdash \neg\alpha, \neg\neg\alpha$. Let τ be any formula where $\vdash \tau$, then since we have already verified rule ($\neg 1$), $\neg\neg\alpha \wedge \neg\alpha \vdash \neg\tau$. And so by the deduction theorem, $\vdash \neg\neg\alpha \wedge \neg\alpha \to \neg\tau$. We showed above that this means $\vdash \tau \to \neg(\neg\neg\alpha \wedge \neg\alpha)$, and since $\vdash \tau$ by MP we get $\vdash \neg(\neg\neg\alpha \wedge \neg\alpha) = \neg\neg\alpha \to \alpha$ as required. ■

1.3.6 Lemma

$\vdash$ also satisfies rule ($\neg 2$) of the Gentzen-style calculus $\vdash$.

This is the rule that $X, \alpha \vdash \beta$ and $X, \neg\alpha \vdash \beta$ implies $X \vdash \beta$. If $X, \alpha \vdash \beta$ and $X, \neg\alpha \vdash \beta$ then $X, \alpha \vdash \neg\neg\beta$ and $X, \neg\alpha \vdash \neg\neg\beta$. By the deduction theorem, this means $X \vdash \alpha \to \neg\neg\beta, \neg\alpha \to \neg\neg\beta$. And thus $X \vdash \neg\beta \to \neg\alpha, \neg\beta \to \neg\neg\alpha$. Thus MP yields $X, \neg\beta \vdash \neg\alpha, \neg\neg\alpha$. So let $\vdash \tau$ and so by ($\neg 1$), we get $X, \neg\beta \vdash \neg\tau$, and so again by the deduction theorem, $X \vdash \neg\beta \to \neg\tau$, meaning $X \vdash \tau \to \neg\neg\beta$. Since $\vdash \tau$ by MP we get $X \vdash \neg\neg\beta$ and since $X \vdash \neg\neg\beta \to \beta$, by MP we get $X \vdash \beta$ as required. ■

1.3.7 Theorem (The Completeness Theorem)
$X \mathrel{|\!\sim} \alpha$ if and only if $X \vDash \alpha$. More suggestively,

$$\mathrel{|\!\sim} \;=\; \vDash$$

We have already shown $\mathrel{|\!\sim} \subseteq \vDash$. Since $\mathrel{|\!\sim}$ satisfies all the basic rules of $\vdash$, $\vdash \subseteq \mathrel{|\!\sim}$ (by $\vdash$'s principle of induction). Now since $\vdash = \vDash$, we get that $\vDash \subseteq \mathrel{|\!\sim} \subseteq \vDash$, and so $\mathrel{|\!\sim} = \vDash$. ■

It is important to note that Λ is sufficient to obtain all tautologies only because $\to$ was defined via $\neg$ and $\wedge$. Had it been taken as just another connective, we would've needed to add axioms to Λ stating the relation between $\to$ and $\neg$ and $\wedge$.

2 First Order Logic

2.1 Mathematical Structures

Our first step in studying first order logic (which will be defined later) is defining the general notion of a *mathematical structure*. Mathematical structures (also known as first order structures) give a useful generalization of many of the algebraic and relational objects mathematicians study.

2.1.1 Definition
An **extralogical signature** is a set σ of symbols of three types: function symbols, relational symbols, and constant symbols. Function symbols and relational symbols are also given an **arity**, a positive integer. Formally, we can view σ as a tuple: $\sigma = (\sigma_f, \sigma_r, \sigma_c, \mathsf{ar})$, where σ_f is a set of function symbols, σ_r is a set of relational symbols, and σ_c is a set of constant symbols (meaning that they are all just sets of symbols). Further assume that σ_f, σ_r, and σ_c are all disjoint. ar is a function mapping symbols in σ_f and σ_r to positive integers.

2.1.2 Definition
Let σ be an extralogical signature (for short, a signature), **mathematical structure** over σ (for short, a σ-structure) is a pair $\mathcal{A} = (A, \sigma^{\mathcal{A}})$ where A is some set, called the **domain** of the structure, and $\sigma^{\mathcal{A}}$ is an **interpretation** of σ. This means that for every function symbol $f \in \sigma$, $\sigma^{\mathcal{A}}$ consists of an operation $f^{\mathcal{A}}\colon A^{\mathsf{ar}(f)} \longrightarrow A$, for every relational symbol $r \in \sigma$, $\sigma^{\mathcal{A}}$ contains a relation $r^{\mathcal{A}} \subseteq A^{\mathsf{ar}(r)}$, and for every constant symbol $c \in \sigma$, $\sigma^{\mathcal{A}}$ contains a constant $c^{\mathcal{A}} \in A$.
Constants may be viewed as 0-ary operations.
The domain of a mathematical structure $\mathcal{A}$ will always be denoted by A.

We now define some general notions relating to structures.
Suppose $A \subseteq B$, and f is an n-ary operation on B. Then A is *closed under* f if $f(A^n) \subseteq A$, meaning that for every $\vec{a} \in A^n$, $f\vec{a} \in A$. If $n = 0$, ie. if f is a constant c, then this simply means that $c \in A$. It is obvious that the intersection of a family of sets closed under f is itself closed under f, and thus we can discuss the smallest set closed under f. For example, $\mathbb{N}$ is closed under $+$ (when viewed as a binary operation of $\mathbb{N}$, $\mathbb{Q}$, etc.), but not under $-$.
Suppose $A \subseteq B$ again, and r^B is an n-ary relation on B. Then the *restriction* of r^B to A is the n-ary relation $r^A = r^B \cap A^n$. For example the restriction of $<^{\mathbb{Z}}$, the standard order of $\mathbb{Z}$, to $\mathbb{N}$ is $<^{\mathbb{N}}$, the standard order of $\mathbb{N}$.
If f^B is an n-ary operation on B and $A \subseteq B$ is closed under f^B, then we define f^B's restriction to A to be the operation $f^A\vec{a} = f^B\vec{a}$.
So if $\mathcal{B}$ is a σ-structure and $A \subseteq B$ is closed under all operations (including constants), then A can be given the structure of a σ-structure naturally: define $\mathcal{A} = (A, \sigma^{\mathcal{A}})$ where for $f \in \sigma_f$ take $f^{\mathcal{A}} = f^A$ the restriction of $f^{\mathcal{B}}$ to A, for $r \in \sigma_r$ take $r^{\mathcal{A}} = r^A$ the restriction of $r^{\mathcal{B}}$ to A, and for $c \in \sigma_c$ take $c^{\mathcal{A}} = c^{\mathcal{B}}$. $\mathcal{A}$ is called a *substructure* of $\mathcal{B}$, denoted $\mathcal{A} \subseteq \mathcal{B}$.
Note that not every subset $A \subseteq B$ can be extended to a substructure of $\mathcal{B}$. For example, $\{1\} \subseteq \mathbb{Z}$ but if the signature σ is taken to include the constant 0, then since $\{1\}$ does not contain $0^{\mathbb{Z}} = 0$ it cannot be extended to a substructure. And similarly if σ includes $+$, then since $\{+\}$ is not closed under $+$, it cannot be extended to a substructure.
Suppose $\mathcal{A}$ is a σ-structure and $\sigma_0 \subseteq \sigma$ is another extralogical signature (meaning $\sigma_{0_x} \subseteq \sigma_x$ for $x = f, r, c$ and $\mathsf{ar}_0(\mathsf{s}) = \mathsf{ar}(\mathsf{s})$ for all relational and function symbols $\mathsf{s} \in \sigma_0$). Then we define the σ_0-structure $\mathcal{A}_0$ where the interpretation of each symbol $\mathsf{s} \in \sigma_0$ is $\mathsf{s}^{\mathcal{A}_0} = \mathsf{s}^{\mathcal{A}}$. $\mathcal{A}_0$ is called the σ_0-*reduct* of $\mathcal{A}$, and conversely $\mathcal{A}$ is called the σ-*expansion* of $\mathcal{A}_0$.
Many times, if σ is a signature consisting of the symbols $\mathsf{s}_1, \mathsf{s}_2, \ldots,$, we will write a σ-structure as $(A, \mathsf{s}_1, \mathsf{s}_2, \ldots)$ instead of writing out the signature. And further, we will often write the signature as a set instead of as a tuple of sets and an arity function. What symbols are functions, relational, and constants, and their arities are to be understood from context.
Mathematical structures defined over a signature without relational symbols are termed *algebraic structures*, while structures defined over a signature without function or constant symbols are termed *relational structures*. For example, mathematical structures of the form $\mathcal{A} = (A, \circ)$ where $\circ$ is a binary operation are called *magmas*. If $\circ$ is associative, $\mathcal{A}$ is a *semigroup*, if it is invertible in each argument then it is a *group*, etc. These are examples of very common algebraic structures. Another common algebraic structure are *rings* and *fields*: both are structures of the form $\mathcal{A} = (A, +, \cdot, 0, 1)$ which satisfy certain axioms. Notice that a structure of this form is not necessarily a ring, but all rings are structures of this form.

A *semilattice* is another type of algebraic structure, and is a special case of a magma where $\circ$ is associative, commutative, and idempotent (meaning $a \circ a = a$ for all $a \in A$). For example $(\{0,1\}, \wedge)$ is a semilattice. We can define the partial order $\leq$ by $a \leq b \iff a \circ b = a$. This is reflexive since $\circ$ is, anticommutative since $\circ$ is commutative, and if $a \leq b$ and $b \leq c$ then $a = a \circ b = a \circ (b \circ c) = (a \circ b) \circ c = a \circ c$ so $a \leq c$. And a *lattice* is an algebraic structure of the form $\mathcal{A} = (A, \cap, \cup)$ where $(A, \cap)$ and $(A, \cup)$ are both semilattices and the following absorption laws hold: $a \cap (a \cup b) = a$ and $a \cup (a \cap b) = a$. A *distributive lattice* is a lattice which satisfies the distributive properties: $x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$ and $x \cup (y \cap z) = (x \cup y) \cap (z \cup z)$. For example if M is a set, then $(\mathcal{P}(M), \cap, \cup)$ is a lattice.

A *boolean algebra* is an algebraic structure $\mathcal{A} = (A, \cap, \cup, \neg)$ where the reduct $(A, \cap, \cup)$ is a distributive lattice and

$$\neg\neg x = x, \quad \neg(x \cap y) = \neg x \cup \neg y, \quad x \cap \neg x = y \cap \neg y$$

The standard example is the boolean algebra $\mathit{2} = (\{0,1\}, \wedge, \vee, \neg)$.

A relational structure $\mathcal{A} = (A, \lhd)$ where $\lhd$ is a binary relation is often called a *graph* (this coincides with the definition of a directed graph). If $\lhd$ is irreflexive and transitive, this is a *(strict) partially ordered set*, or poset for short, and we generally write $<$ for $\lhd$. A *partially ordered set* is when $\lhd$ is reflexive, transitive, and antisymmetric, then we usually write $\leq$ for $\lhd$. Each partially ordered set gives rise to a strict partially ordered set and vice versa, by defining $a \leq b \iff a < b \vee a = b$

2.1.3 Definition

Let σ be some signature, and $\mathcal{A}$ and $\mathcal{B}$ be σ-structures. Then a map $h\colon A \longrightarrow B$ (though we will generally write $h\colon \mathcal{A} \longrightarrow \mathcal{B}$) is called a **homomorphism** provided that for every function symbol f, relational symbol r, and constant symbol c in σ, and $\vec{a} \in A^n$:

$$h(f^{\mathcal{A}}(\vec{a})) = f^{\mathcal{B}}(h(\vec{a})), \qquad h(c^{\mathcal{A}}) = c^{\mathcal{B}}, \qquad r^{\mathcal{A}}(\vec{a}) \implies r^{\mathcal{B}}(h(\vec{a}))$$

where $h(\vec{a}) = (h(a_1), \ldots, h(a_n))$.

A **strong homomorphism** is a homomorphism where the third condition on relations is replaced by the stronger $r^{\mathcal{B}}(h(\vec{a}))$ if and only if there exists a $\vec{b} \in A^n$ such that $h(\vec{a}) = h(\vec{b})$ and $r^{\mathcal{A}}(\vec{b})$ (thus we need not require that every $\vec{b}$ with the same image as $\vec{a}$ under h satisfy $r^{\mathcal{A}}$, only that one does). In other words, the condition is replaced with

$$r^{\mathcal{B}}(h(\vec{a})) \iff (\exists \vec{b} \in A^n)\big(h(\vec{a}) = h(\vec{b}) \wedge r^{\mathcal{A}}(\vec{b})\big)$$

An injective strong homomorphism $\mathcal{A} \longrightarrow \mathcal{B}$ is called an **embedding** of $\mathcal{A}$ into $\mathcal{B}$. If further the embedding is surjective, it is termed a **isomorphism**. If there exists an isomorphism between $\mathcal{A}$ and $\mathcal{B}$, the two structures are called **isomorphic**, and this is denoted $\mathcal{A} \cong \mathcal{B}$. Similarly if $\mathcal{A} = \mathcal{B}$ then an isomorphism is called a **automorphism**.

We will sometimes dispense of parentheses and write $f\vec{a}$ instead of $f(\vec{a})$.

Notice that for algebraic structures, strong and "weak" homomorphisms are one and the same. Furthermore, if $h\colon \mathcal{A} \longrightarrow \mathcal{B}$ is an embedding, the condition on h being a strong isomorphism is simply

$$r^{\mathcal{A}}\vec{a} \iff r^{\mathcal{B}}h\vec{a}$$

as $(\exists \vec{b} \in A^n)(h\vec{a} = h\vec{b} \wedge r^{\mathcal{A}}\vec{a})$ is equivalent to $r^{\mathcal{A}}\vec{a}$ as $h\vec{a} = h\vec{b}$ implies $\vec{a} = \vec{b}$.

The composition of homomorphisms is itself a homomorphism: if $h_1\colon \mathcal{A} \longrightarrow \mathcal{B}$ and $h_2\colon \mathcal{B} \longrightarrow \mathcal{C}$ are homomorphisms then

$$h_2 \circ h_1(f^{\mathcal{A}}\vec{a}) = h_2(f^{\mathcal{B}}h_1\vec{a}) = f^{\mathcal{C}}h_2 \circ h_1(\vec{a})$$
$$h_2 \circ h_1(c^{\mathcal{A}}) = h_2 c^{\mathcal{B}} = c^{\mathcal{C}}$$
$$r^{\mathcal{A}}\vec{a} \implies r^{\mathcal{B}}h_1\vec{a} \implies r^{\mathcal{C}}h_2h_1\vec{a}$$

And if h_1 and h_2 are strong homomorphisms, and h_1 is surjective, then $h_2 \circ h_1$ is also a strong homomorphism:

$$r^{\mathcal{C}}h_2 \circ h_1\vec{a} \iff (\exists \vec{b} \in B^n)\big(h_2\vec{b} = h_2h_1\vec{a} \wedge r^{\mathcal{B}}\vec{b}\big)$$

Since h_1 is surjective, suppose $h_1\vec{a}_0 = \vec{b}$ then

$$\iff (\exists \vec{a}_0 \in A^n)\big(h_2h_1\vec{a}_0 = h_2h_1\vec{a} \wedge r^{\mathcal{B}}h_1\vec{a}_0\big)$$

Since $r^{\mathcal{B}}h_1\vec{a}_0$ if and only if there exists an a_1 such that $h_1\vec{a}_0 = h_1\vec{a}_1$ and $r^{\mathcal{A}}\vec{a}_1$, so this is equivalent to

$$\iff (\exists \vec{a}_1 \in A^n)\big(h_2h_1\vec{a}_1 = h_2h_1\vec{a} \wedge r^{\mathcal{A}}\vec{a}_1\big)$$

As required.

2.1.4 Definition
Let σ be a signature and $\mathcal{A}$ be a σ-structure. Then a **congruence** on $\mathcal{A}$ is an equivalence relation on A, $\approx$, such that for all function symbols $f \in \sigma$ with arity $n > 0$,

$$\vec{a} \approx \vec{b} \implies f^{\mathcal{A}}\vec{a} \approx f^{\mathcal{A}}\vec{b}$$

where $\vec{a} \approx \vec{b}$ means $a_i \approx b_i$ for $i = 1, \ldots, n$ where $\vec{a} = (a_1, \ldots, a_n)$ and $\vec{b} = (b_1, \ldots, b_n)$.
Let us denote $a/{\approx}$ to be the equivalence class of a under $\approx$, and $\vec{a}/{\approx} = (a_1/{\approx}, \ldots, a_n/{\approx})$ for $\vec{a} \in A^n$. Let $f \in \sigma$ be a function symbol, $r \in \sigma$ be a relational symbol, and $c \in \sigma$ be a constant symbol, then let us define the σ-structure $\mathcal{A}'$ over the domain partition $A/{\approx}$ by

$$f^{\mathcal{A}'}(\vec{a}/{\approx}) := (f^{\mathcal{A}}(\vec{a}))/{\approx}, \qquad r^{\mathcal{A}'}(\vec{a}/{\approx}) \iff (\exists \vec{b} \approx \vec{a})(r^{\mathcal{A}}\vec{b}), \qquad c^{\mathcal{A}'} = (c^{\mathcal{A}})/{\approx}$$

These are well-defined as they are independent of the choice of representative from an equivalence class (only the first definition, for $f^{\mathcal{A}'}$, is not true for general equivalence relations). $\mathcal{A}'$ is the **quotient structure** of $\mathcal{A}$ modulo $\approx$, also denoted by $\mathcal{A}/{\approx}$ (the use of $\mathcal{A}'$ was to make it more readable in superscripts).

Let G be a group with the identity e and $\approx$ be a congruence on G. Then let us define $N = \{g \in G \mid g \approx e\}$, and N is a normal subgroup: if $g \in N$ and $h \in G$ then $hgh^{-1} \approx heh^{-1} = e$, and so $hgh^{-1} \in N$. And if N is a normal subgroup, let us define $a \approx_N b$ if and only if $ab^{-1} \in N$, then if $a_1 \approx_N a_2$ and $b_1 \approx_N b_2$ then

$$a_1 b_1 \approx_N a_2 b_2 \iff a_1 b_1 b_2^{-1} a_2^{-1} \in N \iff a_1(b_1 b_2^{-1} a_2^{-1} a_1) a_1^{-1} \in N$$

since $b_1 b_2^{-1} \in N$ and $a_2^{-1} a_1 \in N$, and since N is normal, this is indeed correct. So $\approx_N$ is a congruence on G. This relation is deeper: recall that normal groups are simply kernels of group homomorphisms. So we can define the kernel of general homomorphisms:

2.1.5 Definition
Let $h\colon \mathcal{A} \longrightarrow \mathcal{B}$ be a homomorphism of σ-structures. Then h's **kernel** is the congruence on $\mathcal{A}$ defined by

$$a \approx_h b \iff h(a) = h(b)$$

This is indeed a congruence on $\mathcal{A}$: if $\vec{a} \approx_h \vec{b}$ and $f \in \sigma$ then

$$f^{\mathcal{A}}\vec{a} \approx_h f^{\mathcal{A}}\vec{b} \iff hf^{\mathcal{A}}\vec{a} = hf^{\mathcal{A}}\vec{b} \iff f^{\mathcal{B}}h\vec{a} = f^{\mathcal{B}}h\vec{b}$$

which is true since $h\vec{a} = h\vec{b}$ as $\vec{a} \approx_h \vec{b}$.
Let h be a group homomorphism, and K be its kernel (viewed as a normal subgroup) then $\approx_h = \approx_K$ where $\approx_K$ is defined for groups as previously: $h(a) = h(b)$ if and only if $h(ab^{-1}) = e$ if and only if $ab^{-1} \in K$ if and only if $a \approx_K b$. So this definition of a kernel is natural, and generalizes much nicer than the group-theoretic definition.

2.1.6 Theorem (The Isomorphism Theorem)

(1) Let $\mathcal{A}$ be a σ-structure, and $\approx$ a congruence on $\mathcal{A}$. Then $k\colon a \mapsto a/{\approx}$ is a strong homomorphism from $\mathcal{A}$ onto $\mathcal{A}/{\approx}$.

(2) Conversely, if $h\colon \mathcal{A} \longrightarrow \mathcal{B}$ is a surjective strong homomorphism of σ-structures, then $\iota\colon a/{\approx_h} \mapsto h(a)$ is an isomorphism between $\mathcal{A}/{\approx_h}$ and $\mathcal{B}$. Furthermore, $h = \iota \circ k$.

Let $f, r, c \in \sigma$ be function, relational, and constant symbols respectively. For readability, we will ignore superscripts.

(1) We do this directly:

$$k(f\vec{a}) = (f\vec{a})/{\approx} = f\big(\vec{a}/{\approx}\big) = f(k\vec{a})$$
$$(\exists \vec{b} \in A^n)\big(k\vec{a} = k\vec{b} \wedge r\vec{b}\big) \iff (\exists \vec{b} \approx \vec{a})(r\vec{b}) \iff r(\vec{a}/{\approx}) \iff rk\vec{a}$$
$$k(c) = c/{\approx} = c^{\mathcal{A}/\approx}$$

So k is indeed a strong homomorphism.

(2) The definition of ι is obviously sound (ie. it is well-defined) and injective by the definition of $\approx_h$:

$$\iota(a/{\approx_h}) = \iota(b/{\approx_h}) \iff h(a) = h(b) \iff a \approx_h b \iff a/{\approx_h} = b/{\approx_h}$$

It is surjective since if $b \in \mathcal{B}$, since h is surhjective there exists an $a \in \mathcal{A}$ such that $h(a) = b$ and so $\iota(a/{\approx_h}) = h(a) = b$. Now, ι is a strong homomorphism:

$$\begin{gathered}\iota f(\vec{a}/{\approx_h}) = \iota(f\vec{a})/{\approx_h} = h(f\vec{a}) = f(h\vec{a}) = f\iota(a/{\approx_h}) \\ r\iota(\vec{a}/{\approx_h}) \iff rh(\vec{a}) \iff (\exists\vec{b} \approx_h \vec{a})(r(\vec{b})) \iff r(\vec{a}/{\approx_h}) \\ \iota c/{\approx_h} = h(c) = c\end{gathered}$$

By the definitions of ι and k, $h = \iota \circ k$. ■

We need not require h be surjective: instead we alter the claim and ι becomes an isomorphism between $\mathcal{A}$ and the image of $\mathcal{A}$ under h (denoted $h\mathcal{A}$), which is a substructure of $\mathcal{B}$ (this is easy to verify). This corollary is a direct result of the above theorem, as h is a strong homomorphism from $\mathcal{A}$ to $h\mathcal{A}$.

2.1.7 Definition

Let $\{A_i\}_{i\in I}$ be a family of sets, then we define their **direct product** to be the set of function $I \longrightarrow \bigcup_{i\in I} A_i$ such that for every $i \in I$, $i \mapsto a_i$ where $a_i \in A_i$. Such a function is denoted $(a_i)_{i\in I}$ (similar to how a sequence is denoted $(a_n)_{n=1}^\infty$ as it represents a function $\mathbb{N} \longrightarrow \mathbb{R}$ which maps $n \mapsto a_n$). So the direct product is defined as, in set-theoretic terms:

$$\prod_{i\in I} A_i = \left\{ f\colon I \longrightarrow \bigcup_{i\in I} A_i \;\middle|\; (\forall i \in I)(f(i) \in A_i) \right\}$$

Where the function f is written as $(f(i))_{i\in I}$ (this is generally more readable).
If $\{\mathcal{A}_i\}_{i\in I}$ is a family of σ-structures, we define their **direct product** to be a σ-structure $\mathcal{B} = \prod_{i\in I} \mathcal{A}_i$ whose domain is the direct product of the domains of $\mathcal{A}_i$ (so if A_i is the domain of $\mathcal{A}_i$, the domain is $B = \prod_{i\in I} A_i$) and for every function symbol f, relational symbol r, and constant symbol c in σ we define

$$f^{\mathcal{B}}\vec{a} = (f^{\mathcal{A}_i}\vec{a}_i)_{i\in I}, \qquad r^{\mathcal{B}}\vec{a} \iff r^{\mathcal{A}_i}\vec{a}_i \text{ for all } i \in I, \qquad c^{\mathcal{B}} = (c^{\mathcal{A}_i})_{i\in I}$$

Where $\vec{a} = \big((a_i^1)_{i\in I}, \dots, (a_i^n)_{i\in I}\big) \in B^n$ and $\vec{a}_i = (a_i^1, \dots, a_i^n) \in A_i^n$ is obtained by looking at the components of $\vec{a}$ at a specific $i \in I$ (take care, this is not the ith component of $\vec{a}$).
If all the structures are the same, $\mathcal{A}_i = \mathcal{A}$ for all $i \in I$, then $\prod_{i\in I} \mathcal{A}_i$ is called the **direct power** of $\mathcal{A}$ and is denoted $\mathcal{A}^I$. If $I = \{1, \dots, n\}$ then $\prod_{i\in I} \mathcal{A}_i$ is also written $\mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ and $\prod_{i\in I} \mathcal{A}$ is written $\mathcal{A}^n$.

Notice that our concept of $\mathbb{R}^n$ as an abelian group corresponds with the above definition. But here we have also defined the coordinate-wise product of vectors in $\mathbb{R}^n$.
We can define the *projection homomorphism* from a direct product to one of its components:

$$\pi_j\colon \prod_{i\in I} \mathcal{A}_i \longrightarrow \mathcal{A}_j, \qquad (a_i)_{i\in I} \mapsto a_j$$

where $j \in I$. This is indeed a homomorphism, let $\vec{a} = \big((a_i^1)_{i\in I}, \dots, (a_i^n)_{i\in I}\big)$, then $f\vec{a} = \big((fa_i^1)_{i\in I}, \dots, (fa_i^n)_{i\in I}\big)$ and so

$$\pi_j f\vec{a} = (fa_j^1, \dots, fa_j^n) = f(a_j^1, \dots, fa_j^n) = f\pi_j\vec{a}$$

As required, and $r\vec{a}$ if and only if $r(a_i^1, \dots, a_i^n)$ for all $i \in I$, which implies $r(a_j^1, \dots, a_j^n) = r\pi_j\vec{a}$. The case for constants is implied by the proof for functions.
But it is not necessarily strong: the condition for strongness is that $r\pi_j\vec{a}$ must be equivalent to

$$(\exists\vec{b})(\pi_j\vec{b} = \pi_j\vec{a} \wedge r\vec{b}) \iff (\exists\vec{b})\big(\pi_j\vec{b} = \pi_i\vec{a} \wedge (\forall i)(r\pi_i\vec{b})\big)$$

Since the definition of $r^{\mathcal{B}}\vec{a}$ is literally $r^{\mathcal{A}_i}\pi_i\vec{a}$ for all $i \in I$. So this is clearly stronger than $r\pi_j\vec{a}$, and unless we know that for every i, $r^{\mathcal{A}_i}$ can be satisfied, it is strictly stronger. But if we know that for all $i \in I$ (except for potentially j), there exists a $\vec{a}_i \in \mathcal{A}_i$ such that $r^{\mathcal{A}_i}\vec{a}_i$, then this is equivalent.

2.2 Syntax of First-Order Languages

First-order logic allow us to discuss precise concepts relating to mathematical structures. Unlike propositional logic, first-order logic has the ability to discuss individual variables within a mathematical structure, and it can *quantify* them as well. Like propositional logic, we must first discuss the syntax of first-order logic, which is more involved.

Let us define a set of variables, which is taken to be a countably infinite set of distinct symbols: $Var = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots\}$. Like any language, we must first define the *alphabet* over which we define the language of first-order logic. First-order logic over an extralogical signature σ has the alphabet consisting of: the extralogical symbols of σ; the variables in *Var*; the logical connectives $\wedge$ and $\neg$; the quantifier $\forall$ (*for all*); the equality sign $\boldsymbol{=}$ (in boldface to distinguish it from the metalogical symbol $=$); and parentheses (and). Other logical connectives, like $\vee$, $\leftrightarrow$, and $\rightarrow$ can be defined via $\wedge$ and $\neg$, as discussed before. Similarly other quantifiers like $\exists$ (*there exists*) and $\exists!$ (*there exists a unique*) can be defined as well, which will be discussed later.

From the set of all strings over this alphabet are many meaningless ones, for example $)\forall\wedge$ would be a string over this alphabet, but it has no useful meaning. Like what we did in the previous section, we will recursively define meaningful strings from this alphabet.

2.2.1 Definition

We first define **terms** in this language. Terms are defined recursively as:

(1) Variables and constant symbols in σ, are **prime terms**.

(2) If $f \in \sigma$ is an n-ary function symbol, and $t_1, \dots, t_n$ are terms, then $ft_1 \cdots t_n$ is a term as well.

The set of all terms (ie. all strings constructed in this matter) is denoted $\mathcal{T}$.

Notice that we do not use parentheses with terms, as this simplifies syntax and parentheses turn out to be unnecessary. Despite this, when we actually need to write terms (ie. outside of proofs about terms), we may add parentheses for readability. Also note that all of these definitions (and all the coming definitions) are dependent on the choice of extralogical signature σ, so we may speak of *terms over* σ, or σ*-terms*.

Notice that we can view $\mathcal{T}$ as a σ'-structure where σ' is the signature obtained from σ after removing all relational symbols. This is as for $f \in \sigma'$ we define $f^{\mathcal{T}}(t_1, \dots, t_n) = ft_1 \cdots t_n$ (the right hand side is a term, a string, and is to be read literally) and for $c \in \sigma'$ we define $c^{\mathcal{T}} = c$. So $\mathcal{T}$ is also sometimes called the *term algebra*.

2.2.2 Proposition (Principle of Term Induction)

Let $\mathcal{E}$ be a property of strings (over this language) such that $\mathcal{E}$ is true for all prime terms, and for all $n > 0$ and each n-ary function symbol $f \in \sigma$, $\mathcal{E}t_1, \dots, \mathcal{E}t_n$ implies $\mathcal{E}ft_1 \cdots t_n$. Then $\mathcal{E}$ holds for all terms.

This is true since $\mathcal{T}$ is taken as the smallest set obtained by the two rules (that it contains all prime terms, and if $t_1, \dots, t_n$ are terms then so is $ft_1 \cdots t_n$). So $\mathcal{E}$ must then contain $\mathcal{T}$.

2.2.3 Lemma

Let t be a term, then no proper intial segment of t is a term.

This follows from the principle of term induction: let $\mathcal{E}t$ be the property "*no proper initial segment of t is a term, and t is not a proper initial segment of some other term*". Then $\mathcal{E}$ holds for prime terms, as these are atomic characters from the alphabet and thus have no proper initial segments. And let p be a prime term since all other terms are either prime, or of the form $ft_1 \cdots t_n$, since $p \neq f$ are distinct symbols, p cannot be a proper initial segment of another term. Now if $\mathcal{E}t_1, \dots, \mathcal{E}t_n$ and $f \in \sigma$ is n-ary then any proper initial segment of $ft_1 \cdots t_n$ is of the form $ft_1 \cdots t_k\xi$, where ξ is a proper initial segment of t_{k+1} (it may also be empty). But in order for $ft_1 \cdots t_k\xi$ to be a term, it must be equal to $fs_1 \cdots s_n$ for other terms s_i, but by $\mathcal{E}t_1$, s_1 can not be an initial segment of t_1 nor can t_1 be an initial segment of s_1, so $t_1 = s_1$. Continuing inductively, we have that $t_i = s_i$ for $i \leq k$, and so we get that $\xi = s_{k+1} \cdots s_n$, but this implies s_{k+1} is an initial segment of ξ, but then s_{k+1} is a proper initial segment of t_{k+1}, and so it cannot be a term by $\mathcal{E}t_{k+1}$ in contradiction.

So no proper initial segment of $ft_1 \cdots t_n$ is a term. And if $ft_1 \cdots t_n$ is the proper initial segment of some other term $fs_1 \cdots s_n$, then by induction we see that $t_i = s_i$ (since t_1 is either an initial segment of s_1 or vice versa) contradicting it being proper. ■

2.2.4 Proposition (Unique Term Concatenation Property)
Suppose t_i and s_j are terms, then if $t_1 \cdots t_n = s_1 \cdots s_m$ then $n = m$ and $t_i = s_i$ for all $1 \leq i \leq n$.

If $t_1 \cdots t_n = s_1 \cdots s_m$ then t_1 is either a initial segment of s_1 or vice versa, but by the lemma above, this cannot be proper, so $t_1 = s_1$. Thus $t_2 \cdots t_n = s_2 \cdots s_m$ and so inducting on n, we get $t_i = s_i$ and $n = m$ as required. ■

Using the unique term concatenation property, we can recursively define functions on terms without worrying about them being well-defined:

2.2.5 Definition
Let t be a term, then we define its **set of variables** recursively as follows:

$$\mathit{var}c = \varnothing \text{ for constant symbols } c, \qquad \mathit{var}x = \{x\} \text{ for } x \in \mathit{Var}, \qquad \mathit{var}ft_1 \cdots t_n = \mathit{var}t_1 \cup \cdots \cup \mathit{var}t_n$$

Alternatively we could simply define it as the set of all symbols in *Var* which occur in t.

2.2.6 Definition
We now define **formulas** in our language (again, these are defined with respect to a specific signature, and may be called *formulas over* σ or σ*-formulas*). These are strings defined recursively by the rules

(1) If s and t are terms, then $s = t$ is a formula, called an **equation**.

(2) If $t_1, \dots, t_n$ are terms and $r \in \sigma$ is an n-ary relational symbol, then $rt_1 \cdots t_n$ is a formula.

(3) If α and β are formulas and x is a variable, then $(\alpha \wedge \beta)$, $\neg\alpha$, and $\forall x\alpha$ are formulas.

Formulas defined by the first two rules are **prime formulas**. Formulas which do not contain any quantifiers (no occurences of $\forall$, and since other quantifiers like $\exists$ are defined using $\forall$, this includes all other quantifiers) are called **quantifier-free**.
The set of all formulas over a signature σ is denoted $\mathcal{L}\sigma$. In the case that σ contains only a single symbol, $\sigma = \{\mathsf{s}\}$, we may write $\mathcal{L}_{\mathsf{s}}$ instead. And the set of all formulas over the signature $\varnothing$ is denoted $\mathcal{L}_{=}$ and is called the **language of pure identity**.

If $\circ$ is a binary operation, we will often write $t \circ s$ instead of $\circ ts$ as dictated by the definition of compound terms. Similarly if $\triangleleft$ is a binary relation, we will often write $t \triangleleft s$ instead of $\triangleleft ts$. Formally these are abbreviations which refer to the correct form of writing the terms and formulas.
We define the following abbreviations:

$$(\alpha \vee \beta) := \neg(\neg\alpha \wedge \neg\beta), \qquad (\alpha \to \beta) := \neg(\alpha \wedge \neg\beta), \qquad (\alpha \leftrightarrow \beta) := ((\alpha \to \beta) \wedge (\beta \to \alpha))$$
$$\exists x\alpha := \neg\forall x\neg\alpha$$

The first line of definitions should be familiar; they are the same as those defined in the previous section. The definition on the second line should make sense intuitively: "*there exists an x such that α*" if and only if not all xs don't satisfy α. The symbols $\forall$ and $\exists$ are called *quantifiers*. $\forall$ is also called the *universal quantifier*, and $\exists$ is the *existential quantifier*.

2.2.7 Proposition (Principle of Formula Induction)
If $\mathcal{E}$ is a property of strings such that $\mathcal{E}$ holds for all prime formulas and $\mathcal{E}\alpha$ and $\mathcal{E}\beta$ implies $\mathcal{E}(\alpha \wedge \beta)$, $\mathcal{E}\neg\alpha$, and $\mathcal{E}\forall x\alpha$, then $\mathcal{E}$ holds for all formulas in $\mathcal{L}$.

Again, this is directly due to the definition of $\mathcal{L}$.

2.2.8 Proposition (Unique Formula Reconstruction Property)
Every formula $\alpha \in \mathcal{L}$ is either prime or can be written uniquely as $(\alpha \wedge \beta)$, $\neg\alpha$, or $\forall x\alpha$ for $\alpha, \beta \in \mathcal{L}$ and

$x \in Var$.

The proof of this is similar to all similar previous propositions: first show that no proper initial segment of a formula is itself a formula, then this follows immediately.
Now, instead of discussing σ-structures and σ-terms, we will refer to them as $\mathcal{L}$-structures and $\mathcal{L}$-terms where $\mathcal{L}$ is a first-order language (which is itself defined over a signature σ. This is simply accepted terminology.)

2.2.9 Definition
We define the set of variables in a formula φ recursively. First we define it on equations: $var s = t = var s \cup var t$, then we define it on prime formulas which are not equations: $var t_1 \cdots t_n = var t_1 \cup \cdots var t_n$. Now for the recursive par:

$$var(\alpha \wedge \beta) = var\alpha \cup var\beta, \quad var\neg\alpha = var\alpha, \quad var\forall x\alpha = var\alpha \cup \{x\}$$

Alternatively we could define it as all the variables which occur in φ.
As before, we define the **rank** of a formula recursively as follows:

$$rank\pi = 0 \text{ for prime formulas } \pi, \quad rank(\alpha \wedge \beta) = \max\{rank\alpha, rank\beta\} + 1, \quad rank\neg\alpha = rank\alpha + 1,$$
$$rank\forall x\alpha = rank\alpha + 1$$

And we similarly define the **quantifier rank** of a formula, which measures the maximum nesting depth of a quantifier in the formula:

$$qrank\pi = 0 \text{ for prime formulas } \pi, \quad qrank(\alpha \wedge \beta) = \max\{qrank\alpha, qrank\beta\}, \quad qrank\neg\alpha = qrank\alpha,$$
$$qrank\forall x\alpha = qrank\alpha + 1$$

The set of **subformulas** of a formula is defined similar to before:

$$Sf\pi = \{\pi\} \text{ for prime formulas } \pi, \quad Sf(\alpha \wedge \beta) = Sf\alpha \cup Sf\beta \cup \{(\alpha \wedge \beta)\}, \quad Sf\neg\alpha = Sf\alpha \cup \{\alpha\},$$
$$Sf\forall x\alpha = Sf\alpha \cup \{\forall x\alpha\}$$

2.2.10 Definition
A string of the form $\forall x$ (and by extension $\exists x$) is called a **prefix**. And given a subformula of the form $\forall x\alpha$, α is called the **scope** of $\forall x$. Occurrences of x within the scope of an occurrence of $\forall x$ are termed **bound occurrences** of x, all other occurrences of x are termed **free occurrences** of x. In general we say that a variable x **occurs bound** in a formula φ if the prefix $\forall x$ occurs in φ.
We define $bnd\varphi$ to be the set of all variables which occur bound in φ, and $free\varphi$ to be the set of all variables which have free occurrences in φ.

$bnd\varphi$ and $free\varphi$ can also be defined recursively:

$$bnd\pi = \varnothing \text{ for prime formulas } \pi, \quad bnd(\alpha \wedge \beta) = bnd\alpha \cup bnd\beta, \quad bnd\neg\alpha = bnd\alpha,$$
$$bnd\forall x\alpha = bnd\alpha \cup \{x\}$$
$$free\pi = var\pi \text{ for prime formulas } \pi, \quad free(\alpha \wedge \beta) = free\alpha \cup free\beta, \quad free\neg\alpha = free\alpha,$$
$$free\forall x\alpha = free\alpha \setminus \{x\}$$

Notice that a variable can occur both free and bound in a formula, for example in the below formula x occurs both free and bound

$$\forall x(x = y) \wedge (x = y)$$

This will generally be avoided, but it can happen. We could strengthen our definitions of formulas to ensure that this does not occur, but there is no need to do so.

2.2.11 Definition

Let us define $\mathcal{L}^k$ to be the set of all formulas φ such that $\mathit{free}\varphi \subseteq \{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{k-1}\}$. Thus $\mathcal{L}^0 \subseteq \mathcal{L}^1 \subseteq \cdots$ and $\mathcal{L} = \bigcup_{k=0}^{\infty} \mathcal{L}^k$. $\mathcal{L}^0$ is the set of all formulas which contain no free variables, formulas belonging to $\mathcal{L}^0$ are called **sentences** or **closed formulas**.

Note

If φ is a formula, we write $\varphi(\vec{x})$ to mean that $\mathit{free}\varphi \subseteq \{x_1, \ldots, x_n\}$, where $\vec{x} = (x_1, \ldots, x_n)$ and x_i are all arbitrary and distinct. Similarly if t is a term, $t(\vec{x})$ means $\mathit{var}t \subseteq \{x_1, \ldots, x_n\}$.
And we write $f\vec{t}$ to mean the compound term $ft_1 \cdots t_n$ where $\vec{t} = (t_1, \ldots, t_n)$ where t_i are terms. Similarly we write $r\vec{t}$ to mean the prime formula $rt_1 \cdots t_n$.

Now we would like to define a notion of substitution, which is a natural concept to have. But importantly we'd only like to substitute variables at their free occurrences, why? Suppose you have the formula $\varphi(y) = \exists x(x+x = y)$ (in the context of integers, this means y is even). We could substitute y for 2 and get $\exists x(x + x = 2)$. But now say we wanted to substitute x for 2, then should we get $\exists 2(2 + 2 = y)$ (which is not a valid formula), or $\exists x(2 + 2 = y)$? Well, neither, because neither really makes sense. This is since bound occurrences already have meaning associated with them by their quantifier, so it makes little sense to substitute them.

2.2.12 Definition

A **substitution** (also called a *global* substitution) is a function which assigns to every variable a term, meaning it is a function $\sigma\colon \mathit{Var} \longrightarrow \mathcal{T}$, where $x \mapsto x^\sigma$. We first extend it to a substitution of terms, ie. a function $\sigma\colon \mathcal{T} \longrightarrow \mathcal{T}$ as follows:

$$c^\sigma = c \text{ for constant symbols } c, \quad (ft_1 \cdots t_n)^\sigma = ft_1^\sigma \cdots t_n^\sigma$$

and now we extend it to a substitution of formulas, ie. a function $\sigma\colon \mathcal{L} \longrightarrow \mathcal{L}$ as follows:

$$(s = t)^\sigma = s^\sigma = t^\sigma, \quad (rt_1 \cdots t_n)^\sigma = rt_1^\sigma \cdots t_n^\sigma, \quad (\alpha \wedge \beta)^\sigma = \alpha^\sigma \wedge \beta^\sigma, \quad (\neg\alpha)^\sigma = \neg\alpha^\sigma, \quad (\forall x\alpha)^\sigma = \forall x\alpha^\tau$$

where τ is the substitution $y^\tau = y^\sigma$ for variables y distinct from x, and $x^\tau = x$ as we'd like to substitute only at free occurrences of variables.
A **simultaneous substitution** is a substitution σ such that there exist variables $x_1, \ldots, x_n \in \mathit{Var}$ and terms $t_1, \ldots, t_n \in \mathcal{T}$ such that

$$x^\sigma = \begin{cases} t_i & x = x_i \\ x & \text{else} \end{cases}$$

So we substitute only x_is with t_is, and leave all other variables the same. Instead of φ^σ we write instead $\varphi\frac{t_1 \cdots t_n}{x_1 \cdots x_n}$. In the case that $n = 1$ (we only substitute a single variable), this is called a **simple substitution**.

Notice that while by definition there is no significance in the order of writing the variables and their substitutions in a simultaneous substitution (meaning there is no difference between $\varphi\frac{t_1 \cdots t_n}{x_1 \cdots x_n}$ and $\varphi\frac{t_{\sigma 1} \cdots t_{\sigma n}}{x_{\sigma 1} \cdots x_{\sigma n}}$ where σ is a permutation), it is not true in general that

$$\varphi\frac{t_1 t_2}{x_1 x_2} = \varphi\frac{t_1}{x_1}\frac{t_2}{x_2} \left(= \left(\varphi\frac{t_1}{x_1}\right)\frac{t_2}{x_2}\right)$$

For example, let $\varphi = x_1 < x_2$, then $\varphi\frac{x_2 x_1}{x_1 x_2} = x_2 < x_1$, while $\varphi\frac{x_2}{x_1}\frac{x_1}{x_2} = x_2 < x_2\frac{x_1}{x_2} = x_1 < x_1$. Though it is the case that

$$\varphi\frac{\vec{t}}{\vec{x}} = \varphi\frac{y}{x_n}\frac{t_1 \cdots t_{n-1}}{x_1 \cdots x_{n-1}}\frac{t_n}{y}$$

where y is a variable not in $\mathit{var}\varphi \cup \mathit{var}\vec{x} \cup \mathit{var}\vec{t}$. This should make sense, as we substitute x_n first with y, which remains unchanged by the next simultaneous substitution, and then substitute y for t_n. Thus inductively we see that every simultaneous substitution can be written as a composition of simple substitutions.

2.3 Semantics of First-Order Languages

Similar to how in propositional logic we defined models to give meaning to propositional formulas, we do the same for first-order logic.

2.3.1 Definition

Suppose $\mathcal{L}$ is a first-order language, then an **$\mathcal{L}$-model** (or an **$\mathcal{L}$-interpretation**) is a pair $\mathcal{M} = (\mathcal{A}, w)$ where $\mathcal{A}$ is an $\mathcal{L}$-structure and w is a **valuation function**, $w\colon \mathit{Var} \longrightarrow A$, $x \mapsto x^w$. We denote $f^{\mathcal{A}}$, $r^{\mathcal{A}}$, $c^{\mathcal{A}}$, and x^w also by $f^{\mathcal{M}}$, $r^{\mathcal{M}}$, $c^{\mathcal{M}}$, and $x^{\mathcal{M}}$ respectively.

We can extend valuations to $\mathcal{T}$ in an obvious manner:

$$c^{\mathcal{M}} = c \text{ for constant symbols } c, \qquad (ft_1\cdots t_n)^{\mathcal{M}} = f^{\mathcal{M}} t_1^{\mathcal{M}} \cdots t_n^{\mathcal{M}}$$

In place of $t^{\mathcal{M}}$ we may write $t^{\mathcal{A},w}$ or simply t^w if the structure is understood. But we will usually stick with $t^{\mathcal{M}}$. Notice that the valuation of a term t depends only on the valuation of the variables and extralogical symbols occurring in t:

2.3.2 Proposition

Suppose t is an $\mathcal{L}$-term, and $\mathcal{M}$ and $\mathcal{M}'$ are two $\mathcal{L}$-models. Let V be a set of variables where $\mathit{vart} \subseteq V$. Now suppose that $\mathcal{M}$ and $\mathcal{M}'$ agree on their valuations of V and extralogical symbols in t: for every $x \in V$, $x^{\mathcal{M}} = x^{\mathcal{M}'}$ and for every extralogical symbol s occuring in t, $\mathsf{s}^{\mathcal{M}} = \mathsf{s}^{\mathcal{M}'}$. Then $t^{\mathcal{M}} = t^{\mathcal{M}'}$.

This is done by term induction. If t is a prime term, then $t = c$ for some constant or $t = x$ for some variable. In either case the proposition is satisfied by its assumption (that $\mathcal{M}$ and $\mathcal{M}'$ agree on variables and extralogical symbols occurring in t). Now suppose $t = ft_1\cdots t_n$ then by the assumption of the proposition, $f^{\mathcal{M}} = f^{\mathcal{M}'}$ and by the induction hypothesis $t_i^{\mathcal{M}} = t_i^{\mathcal{M}'}$. Thus

$$t^{\mathcal{M}} = f^{\mathcal{M}} t_1^{\mathcal{M}} \cdots t_n^{\mathcal{M}} = f^{\mathcal{M}'} t_1^{\mathcal{M}'} \cdots t_n^{\mathcal{M}'} = t^{\mathcal{M}'}$$

as required. ■

2.3.3 Definition

We now define the **satisfiability relation** for first-order models. Let $\mathcal{M} = (\mathcal{A}, w)$ be a model. For every $a \in \mathcal{A}$ and $x \in \mathit{Var}$ let us define the model $\mathcal{M}_x^a = (\mathcal{A}, w')$ where $y^{w'} = y^w$ for variables y distinct from x, and $x^{w'} = a$. Meaning

$$y^{\mathcal{M}_x^a} = \begin{cases} a & y = x \\ y^w & \text{else} \end{cases}$$

So now we define the satisfiability relation $\vDash$ recursively as follows:

$$\begin{aligned} \mathcal{M} \vDash s = t \quad &\iff s^{\mathcal{M}} = t^{\mathcal{M}}, & \mathcal{M} \vDash r\vec{t} \quad &\iff r^{\mathcal{M}} \vec{t}^{\mathcal{M}}, \\ \mathcal{M} \vDash (\alpha \wedge \beta) &\iff \mathcal{M} \vDash \alpha \text{ and } \mathcal{M} \vDash \beta, & \mathcal{M} \vDash \neg\alpha &\iff \mathcal{M} \nvDash \alpha, \end{aligned}$$
$$\mathcal{M} \vDash \forall x\alpha \iff \mathcal{M}_x^a \vDash \alpha \text{ for all } a \in \mathcal{A}$$

If $\mathcal{M} \vDash \varphi$, then $\mathcal{M}$ is said to model φ. And if $X \subseteq \mathcal{L}$ is a set of formulas, we write $\mathcal{M} \vDash X$ if for all $\varphi \in X$, $\mathcal{M} \vDash \varphi$, and we similarly say $\mathcal{M}$ models X.

We can generalize $\mathcal{M}_x^a$ to $\mathcal{M}_{\vec{x}}^{\vec{a}}$ where the underlying structure remains the same and

$$y^{\mathcal{M}_{\vec{x}}^{\vec{a}}} = \begin{cases} a_i & y = x_i \\ y^{\mathcal{M}} & \text{else} \end{cases}$$

Notice that $\mathcal{M}_{\vec{x}}^{\vec{a}} = (\mathcal{M}_{x_1}^{a_1})_{x_2}^{a_2}\ldots$. It follows immediately that if we use $\forall\vec{x}$ as an abbreviation for $\forall x_1 \forall x_2 \cdots \forall x_n$, then we get

$$\mathcal{M} \vDash \forall\vec{x}\alpha \iff \mathcal{M}_{\vec{x}}^{\vec{a}} \vDash \alpha \text{ for all } \vec{a} \in \mathcal{A}^n$$

It is easily verifiable that

$$\begin{aligned} \mathcal{M} \vDash (\alpha \vee \beta) &\iff \mathcal{M} \vDash \alpha \text{ or } \mathcal{M} \vDash \beta & \mathcal{M} \vDash (\alpha \to \beta) &\iff \text{if } \mathcal{M} \vDash \alpha \text{ then } \mathcal{M} \vDash \beta \\ \mathcal{M} \vDash (\alpha \leftrightarrow \beta) &\iff \mathcal{M} \vDash \alpha \text{ if and only if } \mathcal{M} \vDash \beta \end{aligned}$$

And also $\mathcal{M} \vDash \exists x\alpha = \neg\forall x\neg\alpha$ if and only if $\mathcal{M} \nvDash \forall x\neg\alpha$ so there exists an $a \in \mathcal{A}$ such that $\mathcal{M}_x^a \nvDash \neg\alpha$, meaning there exists an $a \in \mathcal{A}$ such that $\mathcal{M}_x^a \vDash \alpha$. This chain of reasoning is readily seen to be reversible. So we have shown

$$\mathcal{M} \vDash \exists x\alpha \iff \text{there exists an } a \in \mathcal{A} \text{ such that } \mathcal{M}_x^a \vDash \alpha$$

2.3.4 Definition
A formula or set of formulas is said to be **satisfiable** if it has a model. $\varphi \in \mathcal{L}$ is called a **tautology** (or **generally/logically valid**), denoted $\vDash \varphi$, if $\mathcal{M} \vDash \varphi$ for every model $\mathcal{M}$. Two formulas α and β are said to be **logically equivalent**, denoted $\alpha \equiv \beta$, if for every model $\mathcal{M}$,

$$\mathcal{M} \vDash \alpha \iff \mathcal{M} \vDash \beta$$

Now, say $\mathcal{A}$ is an $\mathcal{L}$-structure, then we write $\mathcal{A} \vDash \varphi$ for a formula φ if $(\mathcal{A}, w) \vDash \varphi$ for all valuations $w\colon \mathit{Var} \longrightarrow A$. Similarly one writes $\mathcal{A} \vDash X$ for a set of formulas X if $\mathcal{A} \vDash \varphi$ for all $\varphi \in X$.

2.3.5 Definition
Finally we define the **consequence relation** for first-order logic. Suppose X is a set of formulas and φ is a formula, then we write $X \vDash \varphi$ if every model of X models φ. Meaning $\mathcal{M} \vDash X \implies \mathcal{M} \vDash \varphi$.

Again, $\vDash$ is used to denote both the satisfaction and consequence relations. The meaning of the notation is to be understood from context. Moreso, $\vDash$ is also used for the satisfaction relation of structures. And again we write $\varphi_1, \dots, \varphi_n \vDash \varphi$ in place of $\{\varphi_1, \dots, \varphi_n\} \vDash \varphi$ and all the usual shorthands.
Notice that while by definition if $\mathcal{M}$ is a model then $\mathcal{M} \vDash \varphi$ or $\mathcal{M} \vDash \neg\varphi$ for all formulas φ. But if $\mathcal{A}$ is a structure, then it is possible for $\mathcal{A}$ to satisfy neither φ nor $\neg\varphi$ (but if it does satisfy one, it cannot satisfy the other obviously). Take for example the formula $x = y$, then if $\mathcal{A}$ is a structure with at least two elements, suppose $a \neq b \in \mathcal{A}$, then we can define a valuation which satisfies $x = y$ and one which does not. And so $\mathcal{A}$ satisfies neither $x = y$ nor $\neg x = y = x \neq y$.
Now suppose φ is a formula and let $x_1, \dots, x_n$ be an enumeration of $\mathit{free}\varphi$ (according to some accepted total order of Var, for example by index), then we define the *generalized of* φ or its *universal closure* to be the sentence

$$\varphi^g := \forall x_1 \cdots \forall x_n \varphi$$

From the definitions provided above, it is immediate that if $\mathcal{A}$ is a structure then

$$\mathcal{A} \vDash \varphi \iff \mathcal{A} \vDash \varphi^g$$

And in general $\mathcal{A} \vDash X \iff \mathcal{A} \vDash X^g := \{\varphi^g \mid \varphi \in X\}$.

2.3.6 Theorem (The Coincidence Theorem)
Let φ be a formula, and V be a set of variables such that $\mathit{free}\varphi \subseteq V$. Let $\mathcal{M}$ and $\mathcal{M}'$ be two models over the same domain A such that $x^{\mathcal{M}} = x^{\mathcal{M}'}$ for all variables $x \in V$, and $\mathsf{s}^{\mathcal{M}} = \mathsf{s}^{\mathcal{M}'}$ for all extralogical symbols s occurring in φ. Then $\mathcal{M} \vDash \varphi$ if and only if $\mathcal{M}' \vDash \varphi$.

We prove this by induction on φ. If φ is a prime formula of the form $rt_1 \cdots t_n$, by the assumptions of the theorem and proposition 2.3.2, $t_i^{\mathcal{M}} = t_i^{\mathcal{M}'}$ for all $1 \leq i \leq n$, and $r^{\mathcal{M}} = r^{\mathcal{M}'}$, so $r^{\mathcal{M}}\vec{t}^{\mathcal{M}} \iff r^{\mathcal{M}'}\vec{t}^{\mathcal{M}'}$ as required. This proof holds for equations as well. Now by the inductive hypothesis we get

$$\mathcal{M} \vDash \alpha \wedge \beta \iff \mathcal{M} \vDash \alpha \text{ and } \mathcal{M} \vDash \beta \iff \mathcal{M}' \vDash \alpha \text{ and } \mathcal{M}' \vDash \beta \iff \mathcal{M}' \vDash \alpha \wedge \beta$$

Similar for formulas of the form $\neg\alpha$.
Now, let $a \in \mathcal{A}$ and suppose $\mathcal{M}_x^a \vDash \varphi$. Then let $V' = V \cup \{x\}$ then $\mathit{free}\varphi \subseteq V'$ (since $\mathit{free}\varphi \subseteq \mathit{free}\forall x\varphi \cup \{x\} \subseteq V \cup \{x\}$) and $\mathcal{M}_x^a$ and $\mathcal{M}'^a_x$ coincide for all $y \in V'$ (though it is possible that $x^{\mathcal{M}} \neq x^{\mathcal{M}'}$). Thus by our inductive hypothesis $\mathcal{M}_x^a \vDash \varphi$ if and only if $\mathcal{M}'^a_x \vDash \varphi$. Thus

$$\mathcal{M} \vDash \forall x\varphi \iff \mathcal{M}_x^a \vDash \varphi \text{ for all } a \in \mathcal{A} \iff \mathcal{M}'^a_x \vDash \varphi \text{ for all } a \in \mathcal{A} \iff \mathcal{M}' \vDash \forall x\varphi$$

as required. ■

Let $\sigma \subseteq \sigma'$ be two signatures, and $\mathcal{L} \subseteq \mathcal{L}'$ be their respective first-order languages. Now, if $\mathcal{M} = (\mathcal{A}, w)$ is an $\mathcal{L}$-model, it can be arbitrarily extended to an $\mathcal{L}'$-model $\mathcal{M}' = (\mathcal{A}', w)$, where $\mathcal{A}'$ is the σ'-expansion of $\mathcal{A}$, by arbitrarily setting $\mathsf{s}^{\mathcal{M}'}$ for $\mathsf{s} \in \sigma' \setminus \sigma$. Now, let us set $V = \mathit{Var}$ and by the coincidence theorem we get that for every $\varphi \in \mathcal{L}$ since $\mathcal{M}$ and $\mathcal{M}'$ agree on the extralogical symbols (as $\mathcal{A}'$ is an expansion of $\mathcal{A}$) and variables n V (since the valuation remains the same), we get that

$$\mathcal{M} \vDash \varphi \iff \mathcal{M}' \vDash \varphi$$

If we denote the consequence relation of $\mathcal{L}$ by $\vDash_{\mathcal{L}}$, then it follows that if $\mathcal{L} \subseteq \mathcal{L}'$, $\vDash_{\mathcal{L}'}$ is a *conservative* extension of $\vDash_{\mathcal{L}}$: for every $\varphi \in \mathcal{L}$ and $X \subseteq \mathcal{L}$, $X \vDash_{\mathcal{L}'} \varphi$ if and only if $X \vDash_{\mathcal{L}} \varphi$. Indeed: if $\mathcal{M}'$ is an $\mathcal{L}'$-model then let $\mathcal{M}$ be the $\mathcal{L}$-reduct of $\mathcal{M}'$ and so $\mathcal{M} \vDash_{\mathcal{L}} X$ if and only if $\mathcal{M}' \vDash_{\mathcal{L}'} X$, and same for φ.
So the satisfiability of φ depends only on the symbols occurring in φ, we need not the subscripts in $\vDash$.
Another consequence of the coincidence theorem is the *omission of superfluous quantifiers*:

$$\forall x\varphi \equiv \varphi \equiv \exists x\varphi \text{ if } x \notin \mathit{free}\varphi$$

To see this, let $\mathcal{M}$ be a model and $a \in \mathcal{A}$ be arbitrary. Then let $V = \mathit{free}\varphi$ and $\mathcal{M}' = \mathcal{M}^a_x$, and by the coincidence theorem since $y^{\mathcal{M}} = y^{\mathcal{M}'}$ for all $y \in V$ (since $x \notin \mathit{free}\varphi$) we have that $\mathcal{M} \vDash \varphi$ if and only if $\mathcal{M}^a_x \vDash \varphi$. So $\mathcal{M} \vDash \forall x\varphi$ if and only if $\mathcal{M}^a_x \vDash \varphi$ for all $a \in \mathcal{A}$, which is if and only if $\mathcal{M} \vDash \varphi$, which is if and only if $\mathcal{M}^a_x \vDash \varphi$ for some $a \in \mathcal{A}$, which is by definition $\mathcal{M} \vDash \exists x\varphi$.
This fact should be intuitive, for example $\forall x\exists x(x > 0)$ is the same as $\exists x(x > 0)$ and $\exists x\exists x(x > 0)$ since the outermost quantifier is superfluous.

3 Index