Propositional Logic

Piyush Patil

August 24, 2017

The essence of propositional logic is to formally govern the methodology by which propositions are combined to form compound sentences. Informally, a *proposition* is a declarative sentence with a definite truth value, meaning it can be logically evaluated to be either true or false. Multiple propositions are combined (into compound sentences) using logical connectives; as such, propositions may be defined recursively using sub-propositions that are connected with a pre-defined logical connectives.

1 The Language

Let's begin the process of formalizing the intuitive notions of truth and propositions we introduced above. The atomic elements of propositions are a set of a pre-defined symbols that compose the logical language we use. We'll first specify the symbols of our language and then impose a syntax on it, which be designed so as to capture our intuitive notions of logical progression. There are two types of symbols we use: *logical symbols* and *propositional symbols*. Propositional symbols are the grounding building blocks of our language and are the most basic elements of our logical system which take on truth values, and we use logical symbols to stand for operators on propositional symbols, such as negating a symbol, grouping logically connected symbols, and linking symbols.

> **(Definition) Propositional symbol**: *The **propositional symbols** of our logical language are denoted A_n for $n \in \mathbb{N}$.*

> **(Definition) Logical symbol**: *The **logical symbols** of our logical langauge are*
>
> $$(,), \neg, \rightarrow$$
>
> *These symbols are used, in general, to connect compound statements.*

The language we use for propositional logic consists of finite sequences of symbols which adhere to a certain structure, which we'll soon define. These symbols, forming the basis for our logical language, are often referred to as composing our *alphabet*. We can combine such finite sequences as follows.

> **(Definition) Sum of finite sequences**: *Let $s = \langle s_1, \cdots, s_n \rangle$ and $t = \langle t_1, \cdots, t_m \rangle$ be sequences of symbols. Then we define*
>
> $$s + t := \langle s_1, \cdots, s_n, t_1, \cdots, t_m \rangle$$

In other words, we combine sequences by simply concatenating them. We can now formally define the logical language we've been alluding to thus far.

> **(Definition) Propositional language**: *The **propositional language** $\mathcal{L}_0$ is the smallest set of finite sequences of symbols satisfying the following.*
> 1. *$\mathcal{L}_0$ contains every propositional symbol, i.e. $\forall n \in \mathbb{N} : A_n \in \mathcal{L}_0$.*
> 2. *$\mathcal{L}_0$ is closed under negation, i.e. $\forall$ finite sequences $s : (\neg s) \in \mathcal{L}_0$.*
> 3. *$\mathcal{L}_0$ is closed under implication, i.e. $\forall$ finite sequences $s, t : (s \rightarrow t) \in \mathcal{L}_0$.*
>
> *We refer to elements of $\mathcal{L}_0$ as **propositional formulas**, or simply formulas for short.*

All we've done here is specify that the language we're using needs to contain our propositional symbols as a starting point, as well as contain compound sentences which are formed by operating on propositional symbols, which is done by either negating a symbol, inverting the truth value, or linking two symbols with an implication, chaining the truth values together unidirectionally (bidirectional chaining, of course, can be built from the implication using the logical operators we'll soon define). Of course, there are infinitely many sets satisfying these properties, but most of these will also contain superfluous elements which have nothing to do with our propositional language; what we really want is a set which satisfies the above axioms, and nothing more. We accomplish this by forcing our language to be the smallest such set.

To take a step back for a moment, note that everything we've done so far has involved what amounts to simple strings of symbols with no intrinsic meaning of their own. Our intention is to formally capture notions of mathematical proof, truth, and deduction by encoding these strings with the appropriate structure inherent to our way of doing mathematics. TODO

Intuitively, another way to think of $\mathcal{L}_0$ is as the intersection of all sets satisfying the above axioms. By intersecting every possible such set, we force our language to retain only those elements which are crucial to the condition that $\mathcal{L}_0$ satisfy our axioms, leaving behind the smallest such set which just barely meets our requirements.

1.1 Subformulas

We've already defined basic operations on our "strings" of symbols, such as concatenation. Another useful property to look at is formulas which contain other formulas. We define such notions of containment over sentences and symbolic sequences below.

(Definition) Block-subsequence: *Let $s = \langle s_1, \cdots, s_n \rangle$ be a sequence of symbols. Then sequence t is a **block-subsequence** of s if there exist $i, j \in \mathbb{N}$, with $i < j \leq n$, such that*

$$t = \langle s_i, \cdots, s_j \rangle$$

*If s and t are formulas, then we say t is a **subformula** of s.*

(Definition) Initial segment: *Let $s = \langle s_1, \cdots, s_n \rangle$. Then sequence t is an **initial segment** of s if there exists $k \in \mathbb{N}$, with $k \leq n$, such that*

$$t = \langle s_1, \cdots, s_k \rangle$$

In other words, t is a block-subsequence that starts at s_1.

(Definition) Occurrence: *Let s be a finite sequence of symbols and $\phi = \langle a_1, \cdots, a_n \rangle$ be a formula. An **occurrence** of s in ϕ is an interval (i, j) such that*
$$s = \langle a_i, \cdots, a_j \rangle$$

Notice that a formula is a subformula of another formula if and only if it occurs (ie the latter has an occurrence in the former) in the latter. The ideas of occurrence and being a subformula are important in formalizing the intuition that formulas in $\mathcal{L}_0$ are built or generated out of sub-formulas already in the propositional language. In fact, we can essentially prove as much.

(Theorem) Readability: *Let ϕ be any formula from $\mathcal{L}_0$. Then exactly one of the following is true:*
  1. $\phi = \langle A_n \rangle$ for some $n \in \mathbb{N}$
  2. $\phi = (\neg \psi)$ for some $\psi \in \mathcal{L}_0$
  3. $\phi = (\psi_1 \rightarrow \psi_2)$ for some $\psi_1, \psi_2 \in \mathcal{L}_0$

Proof: Consider the set L of finite sequences of symbols ϕ which satisfy the above conditions, so every element of L is either a single propositional symbol, the negation of a formula in $\mathcal{L}_0$ wrapped in parentheses, or an implication between two formulas in $\mathcal{L}_0$ wrapped in parentheses. Clearly, $L \subseteq \mathcal{L}_0$ since $\mathcal{L}_0$ is closed under the operations we used to define L. Notice that L satisfies the defining conditions of the propositional language, since it too is closed under the operations of negation and implication. Clearly, L contains every propositional symbol, by the first condition. For any $s \in L$, since $L \subseteq \mathcal{L}_0$ we have $s \in \mathcal{L}_0$ and hence by definition $(\neg s) \in L$. Similarly, for $s, t \in L$, since $s, t \in \mathcal{L}_0$ we have $(s \rightarrow t) \in L$. However, $\mathcal{L}_0$ was defined to be the smallest set satisfying these conditions, and therefore to have $L \subseteq \mathcal{L}_0$ we must have $L = \mathcal{L}_0$, and hence $\mathcal{L}_0$ satisfies the above conditions. $\square$

We've now shown that every formula in $\mathcal{L}_0$ is indeed, as our intuition suspected, generated from subformulas in $\mathcal{L}_0$. However, we haven't learned anything about the stronger statement of whether we can uniquely and completely create $\mathcal{L}_0$ by starting at propositional symbols and building new formulas by applying negation and implication operators, in an exponentially growing tree of formulas linked by operators. In such a way, every formula in $\mathcal{L}_0$ would be recursively decomposed through its path to the root of this tree to a sequence of generating operations, which would imply an inherently hierarchical structure to formulas. To prove this stronger intuition, we'll introduce a useful lemma followed by the theorem that the above ways of expressing formulas as operations of sub-formulas is unique.

(Theorem) No proper initial segments: *No proper initial segment of any formula in $\mathcal{L}_0$ is itself a formula in $\mathcal{L}_0$.*

Proof: Let $\phi \in \mathcal{L}_0$ so that by readability we have either $\phi = \langle A_n \rangle$ for some n, or $\phi = (\neg\psi)$ for some $\psi \in \mathcal{L}_0$, or $\phi = (\psi_1 \to \psi_2)$ for some $\psi_1, \psi_2 \in \mathcal{L}_0$. In the first case, the only proper initial segment of ϕ would be the empty sequence of symbols, which can't be in $\mathcal{L}_0$ by the definition of the propositional langauge. For the second and third cases, we can proceed by induction (the first case satisfies the base case) - take as the inductive hypothesis that for all sequences of length less than $l := $ *length of ϕ* the theorem holds. Then in the second case, no proper initial segment of ψ is a formula, and since any proper initial segment of ϕ must start with the symbols $(\neg$ (otherwise the proper initial segment would simply be (which is clearly not a formula), but cannot be followed by a valid formula since no proper initial segment of ψ is a formula; thus, it cannot be expressed as either a negation or implication of a formula and therefore contradicts readability. In the third case, since neither ψ_1 nor ψ_2 have proper initial segments that are formulas, and hence by the same logic, since any proper initial segment of ϕ starts with (, no proper initial segment can be a formula. $\square$

(Theorem) Unique readability: *Let ϕ be any formula in $\mathcal{L}_0$. Then exactly one of the following is true:*

1. *$\phi = \langle A_n \rangle$ for some $n \in \mathbb{N}$*

2. *$\phi = (\neg\psi)$ for some unique $\psi \in \mathcal{L}_0$*

3. *$\phi = (\psi_1 \to \psi_2)$ for some unique $\psi_1, \psi_2 \in \mathcal{L}_0$*

Proof: That such A_n, ϕ, or ψ_1, ψ_2 exist is guaranteed by readability, so we need only prove uniqueness. Suppose for the second case that

$$\phi = (\neg\psi) = (\neg\theta)$$

for $\psi, \theta \in \mathcal{L}_0$. If $\psi \neq \theta$ then prepending a negation symbol and wrapping in parentheses wouldn't make the formulas equal, and hence would contradict our assumption. Thus, $\psi = \theta$. For the second case, suppose

$$\phi = (\psi_1 \to \psi_2) = (\theta_1 \to \theta_2)$$

for $\psi_1, \psi_2, \theta_1, \theta_2 \in \mathcal{L}_0$. If $\psi_1 \neq \theta_1$ then for the expressions above to be equal, either ψ_1 is an initial segment of θ_1 or vice versa; both cases are impossible by the above theorem, and hence $\psi_1 = \theta_1$. From this it easily follows, if we simply remove symbols as in the second case, that $\psi_2 = \theta_2$. This proves uniqueness. $\square$

2 Truth Assignments

Thus far, we've been working purely with the symbolic manipulation of the propositional language and its contained sequences of symbols, which of course up to this point have been inherently meaningless. Of course, the purpose of introducing these ideas of formal languages is to rigorously codify some notion of proof, deduction, and truth. To do so, we need to formally incept a way of imbuing our symbols and sequences of symbols, which are meant to stand for propositions and boolean operators on these propositions, with notions of truth and falsity. To do this, and actually evaluate a given formula in $\mathcal{L}_0$, which of course represents a hierarchically built proposition, we interpret propositional symbols as boolean variables, standing for atomic propositions that take on their own truth values, and interpret logical symbols as operators that determine the truth of compound formulas based on the composing sub-formulas and propositions. This hints at a recursive way of determining the truth or falsity of a formula, by working down the hierarchy until propositional symbols are reached and then applying logical operators level by level. This also implies that determining the truth value of a formula is computable in a straightforward way, in a sense of computability that we will formalize later, which has its own philosophical implications.

(Definition) Truth assigment: *A truth assignment for $\mathcal{L}_0$ is a function $\nu : \{A_n, n \in \mathbb{N}\} \to \{\text{ T, F}\}$.*

This gives us a way to assign symbols truth values (simply by specifying a truth assignment); next, we want to make sure this notion of truth assignment conforms to how we want our logical operators to behave. Specifically, if formula ϕ has $\nu(\phi) = \text{ T}$ then we want $\nu(\neg\phi) = \text{ F}$, and vice versa, and moreover for formulas ϕ_1, ϕ_2 we want $\nu((\phi_1 \to \phi_2)) = \text{ T}$ if $\nu(\phi_2) = \text{ T}$ whenever $\nu(\phi_1) = \text{ T}$. Let's first prove that such a truth assignment exists in the first place.

(Theorem) Existence of truth assignment respecting logical operators: *Let ν be a truth assignment for $\mathcal{L}_0$. Then there exists a unique function $\overline{\nu} : \mathcal{L}_0 \to \{\text{ T, F}\}$ such that*

1. $\forall n \in \mathbb{N} : \overline{\nu}(A_n) = \nu(A_n)$

2. $\forall \phi \in \mathcal{L}_0 : \overline{\nu}((\neg\phi)) = \begin{cases} \text{T,} & \text{if } \overline{\nu}(\phi) = \text{ F} \\ \text{F,} & \text{otherwise} \end{cases}$

3. $\forall \phi_1, \phi_2 \in \mathcal{L}_0 : \overline{\nu}((\phi_1 \to \phi_2)) = \begin{cases} \text{F,} & \text{if } \overline{\nu}(\phi_1) = \text{ T } \text{and } \overline{\nu}(\phi_2) = \text{ F} \\ \text{T,} & \text{otherwise} \end{cases}$

Proof: We can build this extended function by formalizing the recursive approach to looking up truth values we outlined above. First, in the base case, we simply set

$$\forall n \in \mathbb{N} : \overline{\nu}(\langle A_n \rangle) := \nu(A_n)$$

Next, we can define $\overline{\nu}$ inductively according to the conditions above. By unique readability, every $\phi \in \mathcal{L}_0$ is expressed uniquely as either a unit-length sequence consisting of a symbol, in which case we map it as specified in the base case, or as either $(\neg\psi)$ or $(\psi_1 \to \psi_2)$ for some $\psi, \psi_1, \psi_2 \in \mathcal{L}_0$. In the former case, we define

$$\overline{\nu}(\phi) = \overline{\nu}((\neg\psi)) = \begin{cases} \text{T,} & \text{if } \overline{\nu}(\psi) = \text{ F} \\ \text{F,} & \text{otherwise} \end{cases}$$

In the latter case, we define

$$\overline{\nu}(\phi) = \overline{\nu}((\psi_1 \to \psi_2)) = \begin{cases} \text{F,} & \text{if } \overline{\nu}(\phi_1) = \text{ T } \text{and } \overline{\nu}(\phi_2) = \text{ F} \\ \text{T,} & \text{otherwise} \end{cases}$$

If we inductively assume that formulas with length less than the length of ϕ map to well-defined images under $\overline{\nu}$, then so does ϕ, and hence it follows that $\overline{\nu}$ is well-defined. To prove uniqueness, suppose for the sake of contradiction that there's a function $\xi : \mathcal{L}_0 \to \{\text{ T, F,}\}$ which satisfies the above three conditions but is distinct from $\overline{\nu}$. Then there exist a set of $\phi \in \mathcal{L}_0$ for which $\overline{\nu}(\phi) \neq \xi(\phi)$; the lengths of these ϕ are obviously bound below by zero, and hence we can take the shortest formula ϕ. Hence, it follows that for any formula shorter than ϕ, the images through $\overline{\nu}$ and through ξ agree. Since ϕ is either a symbol, the negation of a shorter formula, or the implication between two shorter formulas, and the values of the functions at each of the shorter formulas agree, it follows that we must have $\overline{\nu}(\phi) = \xi(\phi)$, a contradiction and thereby proving uniquness. $\square$

What we're doing here is taking a truth assignment, which provides a way of looking up the truth values of the atomic elements of our propositions (which are expressed as formulas); it accounts for the base case in the recursive approach we outlined after. We then extend this base case to a full function on our propositional language, enabling us to recursively assign truth values to compound formulas in a way that respects our intuitive ideas of how logical operators ought to behave. Notice that from the inductive approach in the above proof, resulting from the inherently recursive structure of formulas, it follows that truth assignments which agree on propositional symbols have corresponding extensions to $\mathcal{L}_0$ (in the sense as defined in the above theorem) which also agree. In other words,

$$\text{if } \nu = \xi, \text{ i.e. } \forall n \in \mathbb{N} : \nu(A_n) = \xi(A_n), \text{ then } \overline{\nu} = \overline{\xi}$$

We can now define what it means to satisfy a formula, intuitively referring to the idea of meeting the boolean constraints imposed by the formula forcing it to evaluate to true.

(Definition) Satisfiability: *A truth assignment ν **satisfies** a formula ϕ if $\overline{\nu}(\phi) = \text{ T}$. Further, we say $\phi \in \mathcal{L}_0$ is a **tautology** if it's satisfied by every truth assignment, and is a **contradiction** if there's no truth assignment that satisfies it.*

We can use *truth tables* to systematically examine all possible truth assignments to a formula and see which satisfy it. If there are n propositional symbols in a formula, there are 2^n possible truth assignments; a truth table delineates each one and maps it to the corresponding truth value of the entire formula. Because the above theorems imply that a formula's truth value is completely determined by the symbols that compose it, a truth table fully characterizes a formula. Truth tables are useful not only as characterizations of formulas, but in computational paradigms for determining, efficiently through computation rather than simply non-constructively proving existence, the truth value of a formula.

(Theorem) Computability of satisfiability: *There exists an algorithm to determine whether a formula is a tautology, is satisfiable, or a contradiction.*

Proof: We can generate the truth table of the formula by iterating over the formula, extracting the (say, n) propositional symbols as we find them, and putting them in a table. We then iterate over each of the 2^n truth assignments and recursively compute (with the base case of looking up the truth value of a propositional symbol in the corresponding row) the truth assignment of the whole formula. This gives us the full truth table. We can now iterate over the truth values per assignment - if they are all true, the formula is a tautology, and if some are true, it's merely satisfiable, whereas if none are true, it's a contradiction. □

2.1 Truth Functions

In this section, we'll show that the propositional language we've defined is indeed as logically expressive as possible. To make this more precise, recall that over an infinite set of arbitrary placeholder propositional symbols, we only defined two logical symbols, meant to act as operators. From these two logical connectives we build hierarchically more and more formulas, but the question remains of whether we could have considered additional logical connectives that would have allowed us to build new formulas that would otherwise be inaccessible, rendering our choice of logical connectives arbitrary.

Truth functions formalize a very general notion of a logical connective. Intuitively, a truth function is simply any boolean function mapping an n-tuple of boolean values (i.e. T's and F's) to a single boolean value.

(Definition) Truth function: *An n-**place truth function** is a function from $\{$ T, F$\}^n$ to $\{$ T, F$\}$.*

In the most general case, a truth function is simply a lookup table arbitrarily mapping sequences of boolean value to a boolean value, and thus captures any notion of an n-ary logical operator we might consider. However, it's also clear that we immediately have a correspondence between formulas in $\mathcal{L}_0$ and truth functions, since a formula ϕ with n propositional symbols induces an n-place truth function, denoted f_ϕ, whose value at $\sigma \in \{$ T, F$\}^n$ is $\overline{\nu}(\phi)$, where ν is the truth assignment defined by $\nu(A_i) = \sigma_i$. Here we've assumed without loss of generality that the n propositional symbols appearing in ϕ are the first n indexed symbols. We can now prove the expressivity result we alluded to earlier.

(Theorem) $\mathcal{L}_0$ is maximally expressive: *Let f be an arbitrary truth function. Then there exists a formula $\phi \in \mathcal{L}_0$ such that $f = f_\phi$.*

Proof: Suppose f is an n-place truth function. First, given a sequence of truth values $\sigma \in \{$ T, F$\}^n$ we can build a formula ψ_σ, consisting of n propositional variables, that evaluates to T exactly when its ith propositional variable is assigned truth value σ_i. In other words, we'll build $\psi_\sigma \in \mathcal{L}_0$ such that for any truth assignment ν,

$$\overline{\nu}(\psi_\sigma) = \text{ T } \textit{if and only if } \forall i \in \{1, \cdots, n\} : \nu(A_i) = \sigma_i$$

where we've assumed without loss of generality that ψ_σ contains only the first n propositional symbols. Thus, ψ_σ correponds with σ in the sense that it's made true only by σ. The idea behind why we'd want to build such a ψ_σ is that we can then build the ϕ that we seek (i.e. where $f_\phi = f$) by looking at all σ for which $f(\sigma) = $ T and splicing the corresponding ψ_σ's together in such a way that the result is true whenever at least one of the ψ_σ's is true.

To actually perform these constructions, we'll need to introduce two new abstractions on logical connectives. First, let's define the *conjunction* of formulas ψ_1, ψ_2 to be the formula $(\neg(\psi_1 \to (\neg\psi_2)))$. It's easily verifable (by examining the truth table) that a truth assignment satisfies the conjunction if and only if it satisfies both ψ_1 and ψ_2. We can extend the conjunction to countably many formulas inductively. Given this new construction, we can, for a fixed $\sigma \in \{$ T, F$\}^n$, build the ψ_σ we outlined (i.e. a formula which is designed to be satisfied precisely when

we follow σ in assigning truth values) by taking the conjunction of the $\theta_i(\sigma)$ over $1 \leq i \leq n$, where

$$\theta_i(\sigma) := \begin{cases} A_i, & \text{if } \sigma_i = \text{ T} \\ (\neg A_i) & \text{otherwise} \end{cases}$$

Notice that for the conjunction of the above $\theta_i(\sigma)$'s to be satisfied, each $\theta_i(\sigma)$ must be satisfied, and we specifically designed the $\theta_i(\sigma)$'s so that each is satisfied only by σ_i.

Next, we define the *disjunction* of formulas ψ_1, ψ_2 as $((\neg\psi_1) \to \psi_2)$. We've designed the disjunction to be satisfied when at least one of ψ_1 and ψ_2 is satisfied. As before, we can inductively define the disjunction of multiple formulas. We can now build ϕ by looking at every σ, of which there are finitely many choices (specifically, at most 2^n choices), for which $f(\sigma) = $ T. We define

$$\phi := \text{ disjunction over } \sigma \text{ s.t. } f(\sigma) = \text{ T of } \theta(\sigma) \text{ where } \theta(\sigma) := \text{ conjunction over } i \in \{1, \cdots, n\} \text{ of } \theta_i(\sigma)$$

Intuitively, we've designed ϕ to be satisfied by a boolean sequence whenever one of its conjunctions is satisfied by that boolean sequence, and since each conjunction is built so as to be satisfied by one unique boolean sequence that maps to T under f, it follows that ϕ is satisfied by exactly those boolean sequences which f maps to T. More formally, consider some $\tau \in \{ \text{ T, F}\}^n$. Let ν be the truth assignment given by $\nu(A_i) = \tau_i$. Then we have

$$f_\phi(\tau) = \overline{\nu}(\tau) = \text{ T} \iff \exists \sigma \text{ s.t. } \overline{\nu}(\theta_\tau) = \overline{\nu}(\theta_\sigma) = \text{ T} \iff \exists \sigma \text{ s.t. } \tau = \sigma \text{ where } f(\sigma) = \text{ T} \iff f(\tau) = \text{ T}$$

which follows from the way we've constructed disjunctions and conjunctions. It follows that $f_\phi = f$ as desired. $\square$

Whwat we've done in the above proof is something quite important worth noting. We defined extra logical connectives using the logical symbols we started with, allowing us to use them as an intermediate layer of abstraction to prove the above theorem. In fact, the two logical connectives we formed are extremely common and significant. **Conjunctions**, logical operators often denoted $\psi_1 \wedge \psi_2$ for formulas ψ_1, ψ_2, are defined so as to be satisfied only when both ψ_1 and ψ_2 are satisifed, equivalent to the linguistic "and", giving us a way to impose a logical condition that's dependent on both sub-conditions (ie ψ_1 and ψ_2) being true. **Disjunctions** are more relaxed, and are logical operators often denoted $\psi_1 \vee \psi_2$, and are satisfied when at least one of ψ_1 or ψ_2 is satisfied, allowing us to impose a logical condition that only requires some of its sub-conditions, but not necessarily all of them. In fact, the two operators correspond to necessity and sufficiency in mathematical jargon, and are thus extremely fundamental to logic. It's not uncommon to include them as logical symbols right from the start, but we chose to began with as minimal a set of symbols as we could in order to expedite proofs. As we saw above, this doesn't pose an issue since conjunctions and disjunctions can be defined as

$$\psi_1 \wedge \psi_2 := (\neg(\psi_1 \to (\neg\psi_2)))$$

$$\psi_1 \vee \psi_2 := ((\neg\psi_1) \to \psi_2)$$

We intuitively construct the former as follows: if ψ_1 and ψ_2 are inextricably linked as we desire, so that their conjunction depends on both of them being true, then since the formula $(\psi_1 \to \psi_2)$ is satisfied if ψ_1 is false or if ψ_1 is true and ψ_2 is true, the formula $(\psi_1 \to (\neg\psi_2))$ is always satisfied unless both ψ_1 and ψ_2 are true; since this is the opposite of what we want, we negate the expression. The latter is constructed with the idea that when ϕ_1 is true, we make it a false antecedent of the implication, forcing the expression to evaluate to true, and when ϕ_2 is true, we force it to be the true consequent, also forcing the expression to evaluate to true. This is made clear if we simply examine the truth tables.

3 Proof System for $\mathcal{L}_0$

Having formalized notions of propositions, logical operators, and satisfiability, we can now tackle the notion of proof. Given a subset Γ of $\mathcal{L}_0$, we'll intuitively define a *proof* as a finite sequence $\langle \phi_1, \cdots, \phi_n \rangle$ of formulas from $\mathcal{L}_0$ which satisfies certain conditions relevant to Γ. The idea here is that a proof is simply a progression of propositions, each deductively linked to the prior. To make these ideas formal, we must first introduce a set of *logical axioms*.

(Definition) Logical axiom: *Let ϕ_1, ϕ_2, ϕ_3 be propositional formulas. Then each of the following formulas is a logical axiom.*

1. *Group I axioms*

   (a) $((\phi_1 \to (\phi_2 \to \phi_3)) \to ((\phi_1 \to \phi_2) \to (\phi_1 \to \phi_3)))$

   (b) $(\phi_1 \to \phi_1)$

   (c) $(\phi_1 \to (\phi_2 \to \phi_1))$

2. *Group II axioms*

    (a) $(\phi_1 \to ((\neg\phi_1) \to \phi_2))$

3. *Group III axioms*

    (a) $(((\neg\phi_1) \to \phi_1) \to \phi_1)$

4. *Group IV axioms*

    (a) $((\neg\phi_1) \to (\phi_1 \to \phi_2))$

    (b) $(\phi_1 \to ((\neg\phi_2) \to (\neg(\phi_1 \to \phi_2))))$

Note that each logical axiom is a tautology. This is, of course, by design; our intention is to use axioms as a starting point in the iterative inference machine that is our proof system, and hence these axioms are meant to be taken as universally true. Proof systems that rely only on logical axioms are called *logic proof systems*, but more generally we distinguish between two kinds of axioms - *logical axioms* and *specific axioms*. The latter differ from the already defined former in that they are not simply a specified set of tautologies but rather a select subset of formulas from our propositional language that we choose to regard as universally true, a priori, so that formulas that deductively follow, in a sense we'll now make precise, from our axioms can also be regarded as tautologically, incontrovertibly true. A quick aside - if it wasn't already clear, by "true" we of course mean "satisfied", i.e. that a formula evaluates to T under some extended truth assignment.

    (Definition) Γ-Proof: *Let $\Gamma \subseteq \mathcal{L}_0$ be a collection of formulas and $s = \langle \phi_1, \cdots, \phi_n \rangle$ be a sequence of formulas. We say s is a Γ-**proof** if, for each $i \le n$, at least one of the following holds:*

    1. *$\phi_i \in \Gamma$*

    2. *ϕ_i is a logical axiom*

    3. *$\exists j_1, j_2 < i$ s.t. $\phi_{j_2} = (\phi_{j_1} \to \phi_i)$*
       Further, for any formula $\phi \in \mathcal{L}_0$, we write $\Gamma \vdash \phi$ if there exists a finite sequence $s = \langle \phi_1, \cdots, \phi_n \rangle$ such that s is a Γ-proof and $\phi_n = \phi$.

Let's unpack this definition. Essentially, we are taking Γ to be our set of specific axioms, and regarding a proof to be a sequence of propositions. The conditions we impose on this sequence are meant to cement what it means for a proof to be logical and deductive; namely, we require that each proposition in the sequence, corresponding intuitively to a step in a proof, be either an axiom (logical or specific - these are the first two conditions), or both follow from a previous proposition in a valid way. The latter condition is what the third point in the definition formalizes. We require first that $(\phi_{j_1} \to \phi_i)$ so that the "current" proposition ϕ_i in the proof follows from a previous one, and further that $\phi_{j_2} := (\phi_{j_1} \to \phi_i)$ is itself in the proof and hence is deductively valid and can be taken as true (otherwise, simply requiring that ϕ_i follows from a previous statement from the proof isn't enough, since we don't know - or more formally, haven't yet proved - that the implicatory link between ϕ_{j_1} and ϕ_i is valid within our proof). The notation $\Gamma \vdash \phi$ just means that ϕ can be "proven" from the axiomatic system Γ (with logical axioms of course), i.e. that ϕ follows from Γ.

We'll now prove a series of useful lemmas concerning our proof system and the nature of proof.

    (Theorem) Inference lemma: *For $\Gamma \subseteq \mathcal{L}_0$, let ϕ, ψ be formulas with $\Gamma \vdash \psi$ and $\Gamma \vdash (\psi \to \phi)$. Then $\Gamma \vdash \phi$.*
    Proof: Let $\langle \psi_1, \cdots, \psi_n \rangle$ be a Γ-proof of ψ and $\langle \omega_1, \cdots, \omega_m \rangle$ be a Γ-proof of $(\psi \to \phi)$. It follows easily that

$$\langle \psi_1, \cdots, \psi_n, \omega_1, \cdots, \omega_m, \phi \rangle$$

is a Γ-proof for ϕ, since clearly the sequence up to its last element satisfies the conditions for a Γ-proof and for the last element, ϕ, we have $\omega_m = (\psi \to \phi)$ where $\psi_n = \psi$ is also in the sequence. Thus, $\Gamma \vdash \phi$. $\square$

This lemma makes concrete the foundational basis of our deduction, and makes formal the way we mean to use the implication operator. It essentially guarantees that if we can prove both an implication and the antecedent of the implication, then we can prove the consequent. Philosophically, this is related to the idea of modus ponens, the idea that if both of the propositions P and $(P \to Q)$ are true, then Q must be true as well, but is actually a stronger statement as it concerns not truth but provability, which implies truth.

Next, we'll link the two ideas or provability and truth (which we formalized as satisfiability), by showing that if a formula can be proven, then it's truth, or equivalently its satisfiability, depends completely and inextricably on the specific axioms (and not the logical axioms, since these are always tautologies) that its proof relies on.

(Theorem) Soundness lemma: *Let $\Gamma \subseteq \mathcal{L}_0$ be a subset of formulas and ϕ be a Γ-provable formula, i.e. so that $\Gamma \vdash \phi$. Suppose we have the truth assignment*

$$\nu : \{A_n, n \in \mathbb{N}\} \to \{\text{ T},\ \text{F}\} \text{ s.t. } \forall \psi \in \Gamma : \overline{\nu}(\psi) = \text{ T}$$

Then we must have $\overline{\nu}(\phi) = \text{ T}$.

Proof: TODO fix this. We proceed by induction on $n := |\Gamma|$. Suppose $n = 0$, so that we have no specific axioms and only logical axioms. Then for ϕ to be Γ-proven indicates that ϕ is the last element in a sequence of formulas that are all logical axioms, and hence is either equivalent to a logical axiom or the consequent of an implication with a logical axiom as the antecedent. Logical axioms are tautologies, and so in either case, ϕ is satisfied by every truth assignment, proving the base case.

Next, assume as the inductive hypothesis the theorem for subsets with at most n elements, and consider a subset Γ with $n + 1$ elements. If every Γ-proof of ϕ uses n or less formulas from Γ, then by considering the subset of Γ composed of those formulas and applying the inductive hypothesis it follows that $\overline{\nu}(\phi) = \text{ T}$. For Γ-proofs using all $n + 1$ formulas in Γ, the Γ-proof must contain an element of the form $(\psi \to \phi)$ for some $\psi \in \Gamma$, and since $\overline{\nu}(\psi) = \text{ T}$ and $\overline{\nu}((\psi \to \phi)) = \text{ T}$ we must have $\overline{\nu}(\phi) = \text{ T}$. $\square$

This theorem expresses what it really means for a formula to be Γ-proven; the truth of the specific axioms composing Γ are sufficient to determine the truth of the formula, which is exactly what we'd want from a notion of proof. It essentially states that if we can satisfy every axiom in Γ, and we know that those axioms are sufficient to prove the formula ϕ then the same satisfying truth assignment must also satisfy ϕ. After all, this direction of sufficiency in satisfiability and truth is what we'd expect from proof; to say ϕ "follows from" the formulas in Γ is to say that when the latter are true, ϕ must be too.

Let's move on to the final lemma we'll consider with regards to the ideas of provability and proof from axioms we just introduced. Taken together, these three lemmas form the basis of our "laws of thought" which govern exactly what it means for a logical statement to rigorously follow from another, and what our underlying logical framework for truth is.

(Theorem) Deduction lemma: *For $\Gamma \subseteq \mathcal{L}_0$ and formulas $\phi, \psi \in \mathcal{L}_0$, suppose that $\Gamma \cup \{\phi\} \vdash \psi$. Then $\Gamma \vdash (\phi \to \psi)$.*

Proof: TODO $\square$

As the name implies, the deduction lemma formalizes what it means for one formula to imply another, in terms of provability. The idea is that given a set of axioms, if we can add in a formula to that set, effectively giving it a temporary status of self-evident, axiomatic truth that we imbue the elements of Γ with, and prove ψ, then clearly ϕ is sufficient to prove ψ (in addition to the axioms, of course), and hence it should follow that we can prove that ϕ implies ψ.

These three lemmas - the inference lemma, soundness lemma, and deduction lemma - cement the connection between our formal notion of proof from axiomatic formulas and truth, which we formalized using truth functions and their extensions. Hence, they will allow us to rigorously reason about provability and truth within our logical framework in a way that aligns with our intuition regarding logic and mathematical proof. Let's now take a look at the nature of the axioms themselves, and what we can say about them. First, we need to answer the question of whether our axiomatic system is indeed a valuable system to consider. Given that axioms exist for us to use as a foundation for deriving truth, we ought to require that our axioms follow some notion of self-consistency that's consistent with our basic ideas of what truth is. For example, built into our definition of truth functions is that the formal languages we work with obey the law of the excluded middle - every formula is either true or false. Next, our axioms should be consistent with the law that both a proposition and its negation cannot both be true at the same time. Let's define this notion of consistency.

(Definition) Consistent: *We say that $\Gamma \subseteq \mathcal{L}_0$ is **inconsistent** if*
$$\exists \phi \in \mathcal{L}_0 \text{ s.t. } \Gamma \vdash \phi \text{ and } \Gamma \vdash (\neg \phi)$$

*Otherwise, we say Γ is **consistent**.*

One might question even our fundamental assumption for why consistency is desirable - why, ultimately, is it a problem if our axiomatic system is inconsistent? Intuitively, we require consistency because lack thereof causes our axiomatic system to lose all meaning, and to lose any semblance of truth. The theorem below states this formally, by showing that inconsistent axioms can prove any possible formula, completely devaluing our notion of proof.

(Theorem) Inconsistency can prove everything: *Suppose $\Gamma \subseteq \mathcal{L}_0$ is inconsistent. Then*
$$\forall \phi \in \mathcal{L}_0 : \Gamma \vdash \phi$$

Proof: Since Γ is inconsistent, we can find a ψ such that $\Gamma \vdash \psi$ and $\Gamma \vdash (\neg\psi)$. Then,

$$\Gamma \vdash (\psi \to ((\neg\psi) \to \phi))$$

since that formula is a logical axiom. Then, by the inference lemma,

$$\Gamma \vdash ((\neg\psi) \to \phi)$$

and hence by the inference lemma again, we have

$$\Gamma \vdash \phi$$

which proves the theorem. $\square$

Thus, it's desirable for us to choose consistent axioms. It would appear that the larger our set of axioms, the greater the risk of inconsistency, as the number of logical dependencies between the axioms and the full scope of what can be proven grows exponentially, and so does the chance of a formula appearing for which we could prove both it and its negation. Nonetheless, larger axiomatic systems are more expressive and powerful, so a natural question to ask is what the largest axiomatic system we can find is without sacrificing consistency. The next definition solidifies what we're looking for.

(Definition) Maximally consistent: *Consider a consistent $\Gamma \subseteq \mathcal{L}_0$. Γ is **maximally consistent** if*
$$\forall \phi \in \mathcal{L}_0 : \text{ if } \Gamma \cup \{\phi\} \text{ is consistent, then } \phi \in \Gamma$$

In other words, Γ is both consistent and so large that the addition of any new formula sacrifices its consistency. Let's look at how we can go about building a maximally consistent axiomatic system. We begin with the following intuitive building block.

(Theorem) Possibility of adding formulas and preserving consistency: *Suppose $\Gamma \subseteq \mathcal{L}_0$ is consistent. Then for any $\phi \in \mathcal{L}_0$, at least one of $\Gamma \cup \{\phi\}$ and $\Gamma \cup \{(\neg\phi)\}$ is consistent.*
Proof: We'll prove that if $\Gamma \cup \{(\neg\phi)\}$ is inconsistent, then $\Gamma \cup \{\phi\}$ is consistent. From this and its contrapositive, the theorem follows. So, suppose that $\Gamma \cup \{(\neg\phi)\}$ is inconsistent. The idea behind the proof is that given that the addition of the negation of ϕ forces Γ to be inconsistent, despite Γ being consistent, then because inconsistent systems prove everything, we can conclude that $\Gamma \cup \{(\neg\phi)\}$ proves ϕ, ie

$$\Gamma \cup \{(\neg\phi)\} \vdash \phi$$

Intuitively, $(\neg\phi)$ shouldn't help prove ϕ, and hence if $\Gamma \cup \{(\neg\phi)\}$ can prove ϕ, we'd expect Γ alone to be able to prove ϕ. Formally, this follows from the deduction lemma, which tells us that

$$\Gamma \vdash ((\neg\phi) \to \phi)$$

and since $(((\neg\phi) \to \phi) \to \phi)$ is a logical axiom, by the inference lemma $\Gamma \vdash \phi$. Now that we've shown that Γ proves ϕ, which we'd intuitively expect if $\Gamma \cup \{(\neg\phi)\}$ is inconsistent (since this would be the case if Γ already "contained" ϕ in its provable statements, which is what's causing the addition of $(\neg\phi)$ to kill consistency), we can proceed by contradiction.
Suppose for the sake of contradiction that $\Gamma \cup \{\phi\}$ is also inconsistent. Then, again, since inconsistent systems prove everything, we have $\Gamma \cup \{\phi\} \vdash \psi$ for every $\psi \in \mathcal{L}_0$. By the deduction lemma, we have $\Gamma \vdash (\phi \to \psi)$. Since we just showed that $\Gamma \vdash \phi$, by the inference lemma we have $\Gamma \vdash \psi$. We've thus proven that Γ can prove arbitrary formulas from $\mathcal{L}_0$, and hence that Γ must be inconsistent, a contradiction. $\square$

Notice that a corollary is that for a maximally consistent $\Gamma \subseteq \mathcal{L}_0$, for any formula ϕ, either (1) Γ contains ϕ or $(\neg\phi)$ or (2) if Γ can prove ϕ, then Γ must contain ϕ. Intuitively, every propositional formula is associated with a maximally consistent axiom set - either the formula or its negation is an axiom itself, or the provability of the formula within our axiom set implies that it was an axiom to begin with. Since maximally consistent axiomatic sets are as large as possible, if they don't contain a formula, it must be because it already contains the negation of the formula and hence adding the formula would cause inconsistency. Notice that this first condition is in fact a biconditional, since if a consistent axiomatic set contains every formula or its negation, then the only formulas it doesn't contain are the negations of formulas that are contained (assuming double negation cancels out, which is easy to prove), and hence the addition of any new formula would break the consistency.

The second statement tells us that maximally consistent axiom sets can't prove any formulas they haven't taken as axioms to begin with; they are, indeed, so maximal that their "proving power" has saturated, and they contain every formula

they can prove (unless the formula, or its negation, is already contained). This, too, is quite clear, since if a maximally consistent set could prove a formula it didn't contain, then intuitively it wouldn't quite be maximal, as clearly the addition of a provable formula isn't going to shatter the set's consistency. Quite trivially, any set of axioms can prove the axioms it contains; therefore, the second part of this corollary tells us that for maximally consistent axiomatic sets, provability and containment are one and the same.

We'll conclude the section by proving two useful lemmas followed by the theorem that consistent sets can extended so as to become maximally consistent, followed by the an initial version of the completeness theorem, which we'll further explore in the next section.

(Theorem) Condition for containment of implications in maximally consistent sets: *Let $\Gamma \subseteq \mathcal{L}_0$ be maximally consistent. For any formulas $\phi_1, \phi_2 \in \mathcal{L}_0$, $(\phi_1 \to \phi_2) \in \Gamma$ if and only if either $\phi_1 \notin \Gamma$ or $\phi_2 \in \Gamma$.*
Proof: Here's the intuition for the proof, which we'll then formalize and fill in the details of. If $\phi_1 \notin \Gamma$, then since Γ is maximally consistent we must have $(\neg\phi_1) \in \Gamma$ and therefore $\Gamma \vdash (\neg\phi_1)$. Intuitively, this means that under the axiomatic system of Γ, the formula ϕ_1 is false, and hence any implication with ϕ_1 as the antecedent should be true and therefore provable by Γ, and hence, by our assumption of maximal consistency, contained in it. Formally, we can now conclude that $\Gamma \vdash (\phi_1 \to \phi_2)$ by the inference lemma because $((\neg\phi_1) \to (\phi_1 \to \phi_2))$ is a logical axiom and therefore we must have $\Gamma \vdash (\phi_1 \to \phi_2)$.
Next, suppose that instead we have $\phi_2 \in \Gamma$. With similar intuitive reasoning as above, our proof sketch is to say that if ϕ_2 is considered true under the axiomatic system of Γ then any implication with ϕ_2 as the consequent ought to be true. As before, since $\phi_2 \in \Gamma$ we have $\Gamma \vdash \phi_2$, and since $(\phi_2 \to (\phi_1 \to \phi_2))$ is a logical axiom and therefore provable by Γ, we can conclude from the inference lemma that $\Gamma \vdash (\phi_1 \to \phi_2)$.
This proves one direction of the theorem. To prove the other condition, we'll prove the contrapositive. Hence, we begin by assuming that $\phi_1 \in \Gamma$ and $\phi_2 \notin \Gamma$, with the intent of proving that $(\phi_1 \to \phi_2) \notin \Gamma$. The idea here is that if ϕ_1 is true but ϕ_2 is false, under Γ, then their implication must also be false under any consistent (let alone maximally consistent) Γ. We can conclude from our assumptions that $(\neg\phi_2) \in \Gamma$ and therefore $\Gamma \vdash \phi_1$ and $\Gamma \vdash (\neg\phi_2)$. Formalizing our intuition that implications with true antecedents and false consequents must be false, we assert that since $(\phi_1 \to ((\neg\phi_2) \to (\neg(\phi_1 \to \phi_2))))$ is a logical axiom and hence provable by Γ, by the inference lemma, $\Gamma \vdash ((\neg\phi_2) \to (\neg(\phi_1 \to \phi_2)))$. But $((\neg\phi_2) \to (\neg(\phi_1 \to \phi_2)))$ is also a logical axiom, and therefore by the inference lemma again, we conclude that $\Gamma \vdash (\neg(\phi_1 \to \phi_2))$, from which it follows that $(\phi_1 \to \phi_2) \notin \Gamma$. $\square$

The goal we're working towards through these lemmas is to show that consistency is a sufficient condition for satisfiability; to do so, we start with the special case of maximal consistency.

(Theorem) Maximal consistency is sufficient for satisfiability: *Suppose $\Gamma \subseteq \mathcal{L}_0$ is maximally consistent. Then Γ is satisfiable.*
Proof: We wish to find a truth assignment ν which satisfies every formula in Γ. We'll constructively design such a ν in the most obvious way possible: let

$$\forall n \in \mathbb{N} : \nu(A_n) = \begin{cases} \text{T} & \text{if } \langle A_n \rangle \in \Gamma \\ \text{F} & \text{otherwise} \end{cases}$$

We want to prove that $\forall \phi \in \mathcal{L}_0 : \overline{\nu}(\phi) = \text{T}$ *if and only if* $\phi \in \Gamma$. This is quite an intuitive claim, as one would expect truth assignments constructed specifically for satisfiability in Γ over propositional symbols to extend to all formulas built out of those symbols. We'll prove this by induction on the length n of ϕ. The base case with $n = 1$ is trivial and follows by definition. Now take as the inductive hypothesis that $\overline{\nu}$ behaves as we want it to for formulas with length n or shorter. Consider now a formula ϕ of length $n + 1$.
Then by readability we'll consider two cases: either $\phi = (\neg\psi)$ for some ψ or $\phi = (\psi_1 \to \psi_2)$ for some ψ_1, ψ_2. In the first case, $\phi \in \Gamma$ if and only if $\psi \notin \Gamma$, by maximum consistency. This means that $\overline{\nu}(\phi) = \text{T}$ if and only if $\overline{\nu}(\psi) = \text{F}$. Thus, if $\phi \in \Gamma$ then $\psi \notin \Gamma$ which means, by the inductive hypothesis, that $\overline{\nu}(\psi) = \text{F}$ which means $\overline{\nu}(\phi) = \text{T}$. Conversely, if $\phi \notin \Gamma$ then by a similar argument we have $\overline{\nu}(\phi) = \text{F}$.
In the second case, we have $\phi \in \Gamma$ if and only if either $\psi_1 \notin \Gamma$ or $\psi_2 \in \Gamma$, and hence $\overline{\nu}(\phi) = \text{T}$ if and only if either $\overline{\nu}(\psi_1) = \text{F}$ or $\overline{\nu}(\psi_2) = \text{T}$. By the inductive hypothesis, $\overline{\nu}(\phi) = \text{T}$ if and only if $\psi_1 \notin \Gamma$ or $\psi_2 \in \Gamma$, which proves that $\overline{\nu}(\phi) = \text{T}$ if and only if $\phi \notin \Gamma$. $\square$

Finally, let's show that the notion of maximal consistency meets what we intuitively expect from a maximal object, in an algebraic context; namely, that it ought to contain all sub-objects that meet its criteria.

(Theorem) Extensibility of consistent sets: *Suppose $\Gamma \subseteq \mathcal{L}_0$ is consistent. Then there exists maximally consistent $\Gamma^* \subseteq \mathcal{L}_0$ with $\Gamma \subseteq \Gamma^*$.*

Proof: We'll construct Γ^* using the theorem we proved earlier on adding formulas while preserving consistency. Namely, recall that we proved that if $\Delta \subseteq \mathcal{L}_0$ is consistent, then for any formula ϕ, we can either add ϕ or $(\neg\phi)$ to Δ and preserve consistency. We'll use this idea to construct Γ^* one formula at a time, iteratively. This will require us to enumerate We'll use this idea to construct Γ^* one formula at a time, iteratively.

This will require us to enumerate the formulas of $\mathcal{L}_0$, which can be done since there are countably many propositional symbols and, since each formula consists of finitely many propositional symbols, hence there are countably many formulas. One such enumeration, for example, might be to start with the (finitely many) formulas of length 1 which only use the propositional symbol A_1, followed by the (finitely many) formulas of length 2 which only use either A_1 or A_2 (or both), followed by the (finitely many) formulas of length 3 which use no symbols other than A_1, A_2, A_3, and so on. Thus, we can enumerate

$$\mathcal{L}_0 = \{\phi_n, n \in \mathbb{N}\}$$

We now iteratively consider every single formula in $\mathcal{L}_0$ and grow Γ's size with each one while preserving consistency. At the end of this process, because we've considered every formula in the propositional language, it'll follow that if the result is consistent, which it should be, then it's maximally consistent.

Let's formally prove this. Define the sequence of sets $\{\Gamma_n, n \in \mathbb{N}\}$ recursively as follows. Let $\Gamma_0 = \Gamma$, and for $n > 0$, let

$$\Gamma_n = \begin{cases} \Gamma_{n-1} \cup \{\phi_{n-1}\} & \text{if } \Gamma_{n-1} \cup \{\phi_{n-1}\} \text{ is consistent} \\ \Gamma_{n-1} \cup \{(\neg\phi_{n-1})\} & \text{otherwise} \end{cases}$$

Notice that this forms an ascending chain, where $\forall n \in \mathbb{N} : \Gamma_n \subseteq \Gamma_{n+1}$. It's easy to use the theorem we alluded to in the beginning of the proof to prove, by induction, that each Γ_n is consistent. We now take Γ^* to be the "last" set in the above sequence, by defining

$$\Gamma^* = \bigcup_{n \in \mathbb{N}} \Gamma_n$$

Then Γ^* must be consistent, since if it weren't, it would prove a formula and its negation, which means Γ_n is inconsistent for some n, which is a contradiction. Further, Γ^* must be maximally consistent, since for every $\phi \in \mathcal{L}_0$ there's an n for which $\phi = \phi_n$ which means $\phi \in \Gamma^*$ or $(\neg\phi) \in \Gamma^*$ and hence the only formulas not in Γ^* are the negations of formulas already in Γ^* (assuming that double negation cancels, which is easy to prove), which exactly satisfies the definition of maximal consistency, since this means adding new formulas breaks the consistency of Γ^*. $\square$

We can now prove the central result, which follows as a straightforward corollary of the last two lemmas.

(Theorem) Consistency is sufficient for satisfiability: *Suppose that $\Gamma \subseteq \mathcal{L}_0$ is consistent. Then Γ is satisfiable.*

Proof: The proof is simple - simply use the last theorem to extend Γ to a maximally consistent set, and then define the truth assignment ν as defined in one of the lemmas we used, i.e.

$$\forall n \in \mathbb{N} : \nu(A_n) = \begin{cases} \text{T}, & \text{if } \langle A_n \rangle \in \Gamma \\ \text{F}, & \text{otherwise} \end{cases}$$

to show that the result is satisfiable. Since Γ is contained in the maximally consistent set we constructed, which itself is satisfiable, Γ must be satisfiable as well, proving the theorem. $\square$

4 Logical Implication and Compactness

(Definition) Logical implication: *For $\Gamma \subseteq \mathcal{L}_0$ and $\phi \in \mathcal{L}_0$, we say Γ **logically implies** ϕ if*

$$\forall \text{ truth assignments } \nu : \text{ if } \nu \text{ satisfies } \Gamma \text{ then } \nu \text{ satisfies } \phi$$

where by ν satisfying Γ we mean that $\forall \psi \in \Gamma : \overline{\nu}(\psi) = \text{T}$. We denote logical implication by Γ with $\Gamma \vDash \phi$.

Intuitively, logical implication simply means that satisfying axioms forces satisfying of a formula, so that in a sense the formula "follows from" the axioms. It's simply a different way of codifying the idea of deduction from axioms, and in fact, as we'll show, is equivalent to Γ-proof. In fact, right off the bat we notice that the soundness lemma tells us that if $\Gamma \vdash \phi$ then $\Gamma \vDash \phi$, so that provability entails logical implication.

(Definition) Finite satisfiability: $\Gamma \subseteq \mathcal{L}_0$ *is **finitely satisfiable** if every finite subset of Γ is satisfiable.*

This is a useful definition in proving the main result of this section, compactness.

(Theorem) Compactness for $\mathcal{L}_0$: *Let $\Gamma \subseteq \mathcal{L}_0$ logically imply $\phi \in \mathcal{L}_0$. Then there exists a finite subset $\Gamma_0 \subseteq \Gamma$ that logically implies ϕ.*

Proof: Because $\Gamma \vDash \phi$, we know that $\Gamma \cup \{(\neg\phi)\}$ is not satisfiable, and hence by the completeness theorem that $\Gamma \cup \{(\neg\phi)\}$ is inconsistent. This means that there's a formula which is Γ-provable such that it's negation is also Γ-provable; both Γ-proofs are finite sequences, and hence there's a finite subset $\Gamma_1 \subseteq \Gamma\{(\neg\phi)\}$ that's also inconsistent. We can decompose this subset as $\Gamma_0 \cup \{(\neg\phi)\}$ if $(\neg\phi) \in \Gamma_1$, and if not, we can simply add it in; in either case, $\Gamma_0 \cup \{(\neg\phi)\}$ is inconsistent as well. This means that $\Gamma_0 \cup \{(\neg\phi)\}$ can prove anything, including ϕ, and hence it can logically imply anything, including ϕ. $\square$

We can reformulate the completeness theorem to state that if $\Gamma \vdash \phi$ then $\Gamma \vDash \phi$. TODO