Introduction to Logic

Carlos Martínez

November 18, 2025

Abstract

In this class we'll cover zeroth order logic (which is about *propositions*), and first order logic (which is about *predicates*). Developing these two logics will allow us to conclude very interesting (and philosophical) facts about mathematics itself. We'll also see some more mundane, but fun, applications in computing. Being fully formal in every proof that will follow (as is usually done in mathematical logic classes in university) would be a headache for everyone, so we'll often rely on informality, but always just a reasonable amount.

1. Why do we care about logic?

Problem 1.1. Solve the following 4×4 sudoku.

			3
			2
3			
4			

Figure 1.1: A humble 4×4 sudoku.

Of course, we have loftier goals than solving sudokus. The point is to notice that to solve it, we did a sort of reasoning that although it is careful, it is also informal. The point of logic is to *mechanize* this reasoning, so that even a computer can carry it out algorithmically, and allow us to study the properties of this reasoning. Then, more interestingly, we can wield this newly-found power over mathematics itself (since it is of course based on reasoning), to obtain very strange but enlightening statements such as the following:

Theorem 1.1. (Gödel's First Incompleteness Theorem, very informal formulation). Given any reasonable set of assumptions (satisfying *very* specific conditions) about the natural numbers, there are statements about the natural numbers that despite being true, are unprovable.

Unfortunately, we'll not be able to get all the way to proving this remarkable statement. A word of warning, though: this theorem has been incorrectly used in areas beyond mathematics to claim that in any set of axioms there are unprovable truths. Unfortunately, for this to be the case, the very specific conditions of the theorem have to be satisfied, and it's *very, very* rare for anything outside of mathematics to satisfy these conditions.

2. Atomic propositions and propositional functions

In what follows, we won't necessarily use the definitions that you'd find in a textbook; we would get so trapped in the details that we wouldn't be able to see the big picture, so we'll be slightly informal. Thankfully, we'll still be able to mostly prove the interesting things that logic has to tell us.

Definition 2.1. (Atomic proposition). Just a variable that can attain the value 0, meaning `False`, or the value 1, meaning `True`. We generally use $p, q, r, \cdots$ to represent it.

We'll call our **language** the set of all atomic propositions we'll use in a given context, and it may even be infinite – we could define a language $\mathbb{P} = \{x_1, x_2, x_3, \cdots\}$. However, it's crucial that this language is *countable*. We'll see why later.

Now, given our atomic propositions, we can make more complex propositions that can also be `True` or `False` by introducing the following five **propositional functions**:

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \to q$	$p \leftrightarrow q$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Table 1: Definitions of the propositional functions.

Important note: In what follows, we will use 1 and `True` basically interchangeably, and the same with 0 and `False`.

Note that we haven't even defined what a proposition is, only what an atomic proposition is:

Definition 2.2. (Proposition). Either an atomic proposition (that is, a variable), or an expression made out of finitely many applications of the five propositional functions above, starting with atomic propositions. If our language is $\mathbb{P}$, we denote the set of all possible propositions as $\mathrm{PF}_{\mathbb{P}}$.

Now, we can start to interpret the functions that we've just introduced in Table 1:

- Given a proposition p, the unary function $\neg$ will give us a new proposition $\neg p$, which means *"not p"*, or *"it is not the case that p"*.
- Given propositions p and q, the binary function $\wedge$ will give us a new proposition $p \wedge q$, which means *"p and q"*, or *"it is the case that both p and q"*.
- Given propositions p and q, the binary function $\vee$ will give us a new proposition $p \vee q$, which means *"p or q"*, or *"it is the case that either p or q, or even both could be true"*.
- Given propositions p and q, the binary function $\to$ will give us a new proposition $p \to q$, which means *"if p then q"*. If you look at the table, you may notice something interesting: if p is false, then $p \to q$ is true, because what it is just asserting is that *"it won't happen that q is true without p being true"*.

- Given propositions p and q, the binary function $\leftrightarrow$ will give us a new proposition $p \leftrightarrow q$, which means *"p if and only if q"*, or *"p and q are equivalent"*, or *"p and q have the same truth value"*.

Example 2.3. Suppose that our language is $\{a, b, c\}$, so a, b, and c are all atomic propositions. Then, we can start to build new propositions, such as:

$$\neg b, a \vee (\neg b), (a \vee (\neg b)) \rightarrow c, \cdots$$

Note that sometimes when building propositions, the number of parentheses get unwieldy, so we'll sometimes omit them when it's clear from context exactly what we mean. And we can even give whichever meaning to our atomic propositions we want (given that the statements that we are making are unambiguously true or false). So for example, we can say:

- a stands for *"Carlos has an umbrella"*
- b stands for *"It is raining outside"*
- c stands for *"Carlos is playing the guitar outside"*

And then, the proposition $(a \vee (\neg b)) \rightarrow c$ makes sense: it means that *"if either Carlos has an umbrella or it isn't raining outside, then he must be playing the guitar outside"*. And the fact that this is a proposition doesn't necessarily make it true: I do have an umbrella, but instead of playing the guitar outside I'm presumably teaching logic in a room.

Problem 2.1. Prove (or at least intuitively understand why) the set of all possible propositions $\mathrm{PF}_{\mathbb{P}}$ must be countable (even with a countably infinite language $\mathbb{P}$).

3. Theories and Models

Further development of logic comes from the following question: suppose that we assume that some set of propositions (not necessarily atomic: they may be very complex) are all `True`. We may ask the following questions:

- Then, among all possible propositions we could construct, which others must also be necessarily `True`?
- Given any proposition φ, how can we determine whether φ is also necessarily `True` (a *consequence* of our set of propositions)?
- ...or necessarily `False` (*contradicting* our set of propositions)?
- ...or even the possibility that our set of propositions doesn't give us enough information to assert the truth value of φ (i.e. it depends; it is *independent* of our set of propositions)?
- Is there an algorithm or method to determine this?
- What does a proposition being a consequence of another even *mean*? This is where the idea of *models* comes in.

Definition 3.1. (Theory). Any set T of propositions: that is, $T \subseteq \mathrm{PF}_{\mathbb{P}}$. It may even be empty, or infinite (but always countable, by Problem 2.1). Each proposition $\varphi \in T$ will be called an **axiom**.

In short, a theory is any set of propositions that we want to assume to be `True`.

Example 3.2. Now, we can come back to the proposition we examined in Example 2.3. If our atomic propositions were a, b, and c, with the interpretations we gave them, there are many possible states of the world, for example:

- If $(a,b,c)=(0,0,1)$, we have a world where Carlos doesn't have an umbrella, it isn't raining outside, and Carlos is playing the guitar outside.
- If $(a,b,c)=(1,0,0)$, we have a world where Carlos has an umbrella, it isn't raining outside, and Carlos is not playing the guitar outside (interestingly, in this state of the world, the proposition $(a\vee(\neg b))\to c$ is `False`).

We can readily see that in fact there are 8 possible states of the world in this case, and in general, if we have exactly n propositional variables in our language, there will be 2^n possible states of the world.

Definition 3.3. (Model). A function $v:\mathbb{P}\to\{0,1\}$; that is, a truth assignment for each propositional variable in our language $\mathbb{P}$.

In short, a model is just a possible state of the world. In the example above, we had described two models for the language $\{a,b,c\}$.

As we can see, for a given proposition φ, there may be some models in which φ evaluates to `True` and there may be some where it evaluates to `False`. This motivates our next definition:

Definition 3.4. (Model of a theory). Given a proposition $\varphi\in\mathrm{PF}_{\mathbb{P}}$, we say that a model v is a **model of the proposition** φ if φ evaluates to `True` in the model v, and we write that $v\models\varphi$. More generally, given a theory $T\subseteq\mathrm{PF}_{\mathbb{P}}$, we say that v is a **model of the theory** T if v is a model of all elements of T; that is, the whole theory T evaluates to `True` in the model v. In that case, we write $v\models T$.

In other words, the set of all models of a theory T, which we usually denote by $M_{\mathbb{P}}(T)$, is precisely the set of all possible worlds where the whole theory T is true.

This framework gives us a very nice way of thinking about how a proposition can be a consequence of another, or even of a whole theory:

Definition 3.5. (Consequence of a theory). Given propositions $\varphi,\psi\in\mathrm{PF}_{\mathbb{P}}$, we say that ψ **is a consequence of** φ if every model v of φ is also a model of ψ; that is, if it is the case that for all models where φ holds, then ψ also holds. More generally, given a theory $T\subseteq\mathrm{PF}_{\mathbb{P}}$, we say that ψ **is a consequence of** T if every model v of T is also a model of ψ; that is, if for all models where the whole theory T holds, then ψ also holds.

Whenever ψ is a consequence of φ, we write $\varphi\models\psi$; and similarly, whenever ψ is a consequence of φ, we write $T\models\psi$. Note that one naive but useful way to prove that $T\models\varphi$ is to brute-force it (if our language $\mathbb{P}$ is reasonably small): we go over all $2^{|\mathbb{P}|}$ models, determine which are models of T, and then, for each of these, verify that they are also models of φ. We'll later develop more clever ways to prove consequence.

Exercise 3.1. (Sanity check). Given a language $\{p,q\}$, prove that $\{p,p\to q\}\models q$.

4. Intermission: Bipartite Graphs

As a sanity check to make sure that what we've done is even remotely useful, let's consider bipartite graphs: recall that a graph G is *bipartite* if there exists a way to color its vertices with two colors (say, red and blue) in such a way that no two adjacent vertices have the same color.

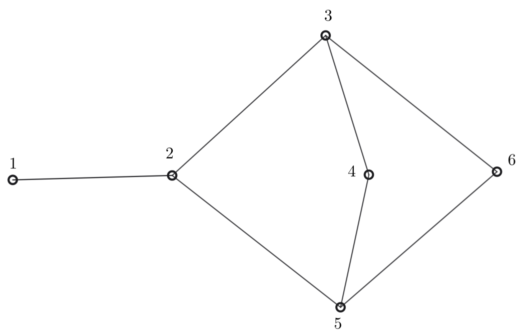

Figure 4.1: A cute graph.

Let's consider the particular graph G described in Figure 4.1. How may we describe the statement *"G is bipartite"* using our new framework? Here's how: we can consider the language $\mathbb{P} = \{R_1, R_2, R_3, R_4, R_5, R_6\}$, where the variable R_i stands for *"Vertex i is red"*. Furthermore, note that from this, we can also interpret $R_i = 0$ as meaning that *"Vertex i is blue"*. Then, whenever vertex i and vertex j are adjacent, we need the condition *"Vertex i and vertex j have a different color"*, which we can write as the proposition $\neg(R_i \leftrightarrow R_j)$. From this, we get that the statement *"G is bipartite"* can be expressed through the theory

$$T = \{\neg(R_1 \leftrightarrow R_2), \neg(R_2 \leftrightarrow R_3) \neg(R_2 \leftrightarrow R_5), \neg(R_3 \leftrightarrow R_4), \neg(R_4 \leftrightarrow R_5), \neg(R_3 \leftrightarrow R_6), \neg(R_5 \leftrightarrow R_6)\}$$

Now, we can make a few interesting observations:

- A model of our language $\mathbb{P}$ (representing a possible state of the world) in this case represents a *possible coloring* of the vertices. Note that such a coloring need not satisfy the condition that any two adjacent vertices must have different colors.
- However, a model of T must satisfy this condition (by definition).
- So, the set of all models $M_{\mathbb{P}}(T)$ represents all possible colorings that satisfy this condition.
- Furthermore, we expect that in all such colorings, vertices 3 and 5 have the same color, so somehow we should be able to prove that $T \models R_3 \leftrightarrow R_5$ (which we can indeed prove).

Exercise 4.1. How many models does T have? (Please do not check all 64 possible models).

Now, you may ask how could we express the notion of *bipartiteness* for all possible graphs G. It turns out that propositional logic doesn't have the expressive power for this. But thankfully, predicate logic, which we'll soon develop, will.

5. Analytic Tableaux (a.k.a. proof by "here's a cute tree")

Suppose that we have a theory T in a countable language $\mathbb{P}$, and some proposition $\varphi \in \mathrm{PF}_{\mathbb{P}}$. If we wanted to prove that $T \models \varphi$, one way we could do this is by *assuming that φ evaluates to* ***`False`****, and show that this can't happen in any model of T, by cleverly going through all the possible cases.*

To do this, we're going to build a rooted, labeled tree with entries of the form $T\varphi$ (standing for *"Proposition φ evaluates to* ***`True`****"*) and $F\varphi$ (standing for *"Proposition φ evaluates to* ***`False`****"*). Then, we'll progressively append one of ten **atomic tableaux** to some leaf of the tree, or entries of the form $T\alpha$ where $\alpha \in T$ (don't worry, we'll explain what's going on in a moment).

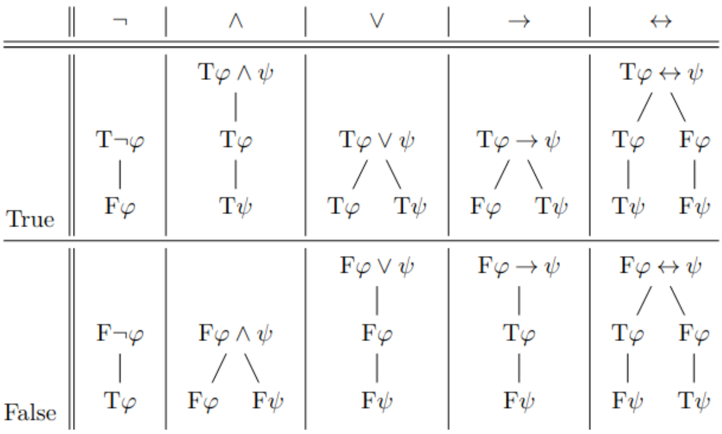

Figure 5.1: Atomic tableaux in propositional logic (Table 4.1 from Jakub Bulín's logic lecture notes).

Definition 5.1. (Finite tableau from theory T). A rooted, labeled tree defined inductively as follows:

1. For any proposition φ, the tree with a single node (the root) of the form $T\varphi$ or $F\varphi$ is a tableau from theory T.
2. Given any leaf L in branch B of a tableau from theory T, by appending the atomic tableau of any entry in the branch B to L, we obtain a tableau from theory T.
3. Given any leaf L of a tableau from theory T, by appending the entry $T\alpha$ for any $\alpha \in T$ to L, we obtain a tableau from theory T.

The power of tableaux comes from the following lemma:

Lemma 5.2. If a model v of a theory T agrees with the root of a finite tableau from theory T, then it must agree with all entries in some branch.

Proof. We can proceed by induction on the current "depth" of the branch. In the base case, we have only the root, which the model v agrees with. The model v will also agree with any entry of the form $T\alpha$ for $\alpha \in T$ (by definition). The last case is when we append the atomic tableaux of some entry in the branch; and for each of the ten atomic tableaux, we can verify that this is the case.

□

So, then, one way to prove that $T \models \varphi$ is to start with the entry $F\varphi$, and build a finite tableau from theory T until all branches are contradictory: that is, all branches have a pair of entries $T\psi$ and $F\psi$. This implies that $T \models \varphi$, since any model to agree with $F\varphi$, by the lemma, must agree with these two contradictory entries, which is impossible. This brings us to the **soundness theorem**:

Theorem 5.3. (Soundness of the tableau method) If φ is tableau provable from T (that is, there exists a finite tableau from theory T with root $F\varphi$ and with all its branches contradictory), then $T \models \varphi$.

As a note regarding notation, we usually denote that φ is tableau provable from T by writing $T \vdash \varphi$. So the soundness theorem can be summarized as *"if $T \vdash \varphi$ then $T \models \varphi$"*.

Example 5.4. (The contrapositive) We'll use the tableau method to prove that $\varphi \equiv (p \to q) \leftrightarrow (\neg q \to \neg p)$ is a **tautology**: that is, $\emptyset \models \varphi$.

6. Systematic Tableaux and Completeness

In this section, we'll end up proving something quite remarkable: if a proposition φ is valid in T, then a tableau proof of φ exists (and it can be found algorithmically!), even in the case where T has countably infinitely many axioms.

Definition 6.1. (Reduced entry with respect to branch B) We say that an entry E in a branch B of tableau τ is **reduced** if either: it's of the form Tp or Fp for some atomic proposition $p \in \mathbb{P}$, or, somewhere in the branch B, it appears as the root of an atomic tableau (meaning that it has been developed).

Definition 6.2. (Systematic tableau) Starting with any entry E as a root, we can develop a tableau from theory $T = \{\alpha_1, \alpha_2, \cdots\}$ systematically using the following algorithm:

1. Let τ_0 be the single-entry tableau of root E.
2. For each $i \geq 0$:
3. Among all non-contradictory branches of τ_i, take the smallest level (that is, the one closest to the root) that contains at least one non-reduced entry. Then, choose the leftmost non-reduced entry in this level, say E. We define τ_i' as the tableau obtained by appending the atomic tableau of E to all the branches in which E is not reduced (if no non-reduced entries exist, we just define $\tau_i' = \tau_i$. **(Note how this makes E reduced in all the branches it's in).**
4. In all non-contradictory branches of τ_i', we append the entry $T\alpha_{i+1}$. We define the resulting tableau as τ_{i+1} (if it happens that $i+1 \geq |T|$, then we simply let $\tau_{i+1} = \tau_i'$). **(Note how this progressively makes each non-contradictory branch contain $T\alpha$ for all $\alpha \in T$).**

Then, we define the **systematic tableau** τ as the union of all of these tableaux $\tau_0, \tau_1, \tau_2, \cdots$, which may very well be infinite.

Exercise 6.1. Use tableaux (constructed haphazardly) or systematic tableaux to prove that $\{q \to p, r \to q, (r \to p) \to s\} \models s$.

Problem 6.1. Prove König's lemma for rooted trees: if a rooted tree has infinitely many vertices, then there exists an infinite branch.

We are now ready to prove the following remarkable statement:

Theorem 6.3. (Completeness of the tableau method). Given a theory T and a proposition φ, if $T \models \varphi$, then $T \vdash \varphi$.

Proof. We construct a systematic tableau from theory T of root $F\varphi$. If all the branches happen to be contradictory, we're done: this would mean that $T \vdash \varphi$. So suppose that a non-contradictory branch B exists. Then, we can actually construct a **canonical model of branch** B, called v, by defining it as follows:

$$v(p) = \begin{cases} 1 \text{ if the entry } Tp \text{ appears on } B \\ 0 \text{ otherwise} \end{cases}$$

for each $p \in \mathbb{P}$. This model is well-defined simply because the branch is non-contradictory, so we'll never encounter both Tp and Fp in this branch. The central claim is that v will agree with all entries of the branch, which we can prove inductively: to see this, we first note that v will agree with all those entries of the form Tp or Fp where p is atomic, and given any "compound entry", such as $T\varphi \vee \psi$, by the definition of a systematic tableau it must have been developed somewhere in the branch; and since, by induction hypothesis, v agreed with the decomposed entries, it must also agree with the compound entry.

In particular, by how we defined systematic tableaux, all axioms of T will appear in B, and so $v \models T$. But furthermore, it will agree with the root $F\varphi$, meaning that in at least one model of T, φ evaluates to `False`, which contradicts the assumption that $T \models \varphi$.

□

7. The Compactness Theorem for Propositional Logic

Maybe the most interesting application of the tableau method is the following remarkable statement, with many applications:

Theorem 7.1. (Compactness). A (possibly infinite) theory has a model if and only if every finite part has a model.

Proof. The forward direction is straightforward. For the backward direction, we can instead consider the contrapositive: we aim to prove that if a theory doesn't have a model, then some finite part won't have a model either.

So, suppose that a theory T doesn't have a model. Then, if p is some atomic proposition, we'll have that $T \models p \wedge \neg p$, precisely because for all models of T (that is, none), $p \wedge \neg p$ evaluates to `True`. But then, by the completeness theorem, $T \vdash p \wedge \neg p$, meaning that there exists a finite tableau from theory T with root $Fp \wedge \neg p$ whose branches are all contradictory.

Now, let T' be the (finite) theory consisting of all the axioms of T mentioned in this finite tableau. Then, this tableau will also be a finite tableau from theory T', so $T' \vdash p \wedge \neg p$. By soundness, this means that $T' \models p \wedge \neg p$, which can only happen if the finite subtheory T' has no model.

□

Example 7.2. (or exercise). Prove that an infinite graph is bipartite if and only if every finite part is bipartite.

8. Predicate Logic: Variables, Relations, Functions

In predicate logic, our language will be somewhat more complicated, with the following components:

1. **(Variables).** These are **not** propositional variables; they are just the sort of variables you'd use in math as placeholders, to denote particular objects. We'll introduce (countably) infinitely many of them: $x_1, x_2, x_3, \cdots$. This component is necessary, regardless of whether we introduce any functions or relations.
2. **(Function symbols).** We'll introduce finitely many, or countably infinitely many of them, as many as we wish: $f_1, f_2, f_3, \cdots$. Each function symbol f_i will have a certain **arity** which we must specify: the number of arguments it can take. We'll later see that function symbols represent what we would expect them to represent.
3. **(Relation symbols).** Again, we'll introduce as many as we wish: $R_1, R_2, R_3, \cdots$, and for each, we'll specify their arity.
4. **(Quantifiers).** For each variable x, now we'll have the symbols $(\forall x)$ (meaning: "for all x") and $(\exists x)$ (meaning: "there exists an x") at our disposal.
5. **(All other connectives)**. We'll also introduce those connectives we're familiar with from propositional logic: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.

We'll be allowed to have function symbols of arity 0, which we'll call **constants**.

We haven't yet explained what these things mean, right now we're only concerned about the things we're allowed to write. But we'll get there in a bit.

9. Terms and Formulas

Definition 9.1. (Term). Given a language L, we define a **term** inductively:

1. Variables and constant symbols are terms.
2. If $t_1, t_2, \cdots t_n$ are terms, and f is a function symbol of arity n, then $f(t_1, t_2, \cdots, t_n)$ is a term.

In short, a term is anything that we can sensibly interpret as an object (we'll talk a bit more about interpretations later).

Example 9.2. If c is a function of arity 0, f a function of arity 2, g a function of arity 3, and x, y variables, then $g(x, y, f(c, y))$ is a term.

Definition 9.3. (Formula). Given a language L, we define a **formula** inductively:

1. Given terms $t_1, t_2, \cdots, t_n$ and a relation symbol R of arity n, $R(t_1, \cdots, t_n)$ is a formula (and such a formula we call **atomic**).
2. If φ, ψ are formulas, then any combination of them with the connectors $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ is also a formula (very similar to how we construct propositions in propositional logic).

3. If φ is a formula and x a variable, then $(\exists x)(\varphi)$ and $(\forall x)(\varphi)$ are formulas.

In short, a formula is anything that we can sensibly interpret as a statement with a truth value.

Example 9.4. If $+$ and $\cdot$ are function symbols of arity 2, $=$ is a relation symbol of arity 2, and $1,2$ are constant symbols, then $=(+(1,1),2)$ is a formula (better known as $1+1=2$). Maybe more interestingly, $(\forall x)(\forall y)(=(+(x,y),+(y,x)))$ is also a formula.

10. Models in Predicate Logic (a.k.a. Structures)

Given a language L, we can now look at what a model in predicate logic looks like: it's an *interpretation* of the symbols of the language.

So, given a language L, we define:

Definition 10.1. (L-structure). Let $\mathcal{R}$ and $\mathcal{F}$ be the set of relation symbols and the set of function symbols in L, respectively. Then, an L-structure is a tuple $(A,\mathcal{R}^A,\mathcal{F}^A)$, where:

- A is a non-empty set, called the domain (that is, the set where all the variables, and all terms, reside).
- $\mathcal{R}^A$ is the set of interpretations of all the relation symbols in $\mathcal{R}$: in particular, if R is a relation symbol of arity n, then $R^A \in \mathcal{R}^A$ is an n-ary relation (i.e. a subset of A^n).
- $\mathcal{F}^A$ is the set of interpretations of all the function symbols in $\mathcal{F}$: in particular, if f is a function symbol of arity n, then f^A is an n-ary function of type $A^n \to A$ (and if c is a constant, $c^A \in A$).

From this, the interpretation of the terms is what you'd expect; for example, given constants c_1,c_2 and a binary function symbol f, the value of $f(c_1,c_2)$ would be $f^A(c_1^A,c_2^A)$. Note also that in general, for any term t, we'll use t^A to represent its interpretation in structure $\mathcal{A}$.

What about terms that contain variables? In that case, we talk about the value of a term with respect to some **assignment of variables** e; that is, a function from the set of all variables to the domain (and by the way, whenever we talk about an assignment we mean an assignment of variables in particular), and in such cases we can denote the value by $t^A[e]$ to prevent ambiguity.

Now, given a structure $\mathcal{A}$, how do we define the truth value of a formula? It's a bit more complicated than in propositional logic (also note, just as we did in propositional logic, we'll use the convention φ to always mean a formula).

Definition 10.2. (Truth value of formula φ in $\mathcal{A}$ under the assignment e). We'll denote it by $\mathrm{TVal}^A(\varphi)[e]$. We define it inductively:

1. If φ is an atomic formula of the form $R(t_1,t_2,\cdots,t_n)$, then the truth value of φ under assignment e will be `True` if and only if $(t_1^A[e],\cdots,t_n^A[e]) \in R^A$.
2. Given formulas φ and ψ involving the logical connectives, their truth value is precisely what you'd expect: for example, the truth value of $\neg\varphi$ under assignment e will be `True` if and only if the truth value of φ is `False`.

3. ...and when the formulas are of the form $(\forall x)\varphi$ or $(\exists x)\varphi$, we *tweak* the value of x in the assignment e through all its possible values to obtain the truth value; for example, $(\forall x)\varphi$ will evaluate to `True` under the assignment e if, no matter how we tweak x in the assignment e, φ still evaluates to `True`.

11. Theories and Validity of Formulas

Definition 11.1. (Validity of formula φ in structure $\mathcal{A}$). We say that $\mathcal{A} \models \varphi$ if under all possible assignments, the truth value of φ in $\mathcal{A}$ is `True`.

And finally, we are ready to talk about theories and consequences, which are in a way basically identical to the notions in propositional logic:

Definition 11.2. (Theory in language L). A set of formulas in language L (once again, may be infinite).

So then, we say that structure $\mathcal{A}$ is a **model of theory** T if all of the axioms of T are valid in $\mathcal{A}$; and $T \models \varphi$ if for all models of T, φ is valid.

12. Tableaux in Predicate Logic

First, one useful type of formula:

Definition 12.1. (Closed formula; or sentence). A formula whose truth value in some structure is independent of the assignment of its variables: in particular, every variable appearing in the formula must be *bound* by some quantifier.

So as to not repeat things unnecessarily, we'll notice that sentences work very similar to good old propositions, so we can just import some of the work done with tableaux in propositional logic. In particular, the atomic tableaux for the five logical connectives will look the same as before.

However, we now also have quantifiers, so we must deal with them by adding four brand new atomic tableaux:

	$\forall$	$\exists$
True	$\mathrm{T}(\forall x)\varphi(x)$ | $\mathrm{T}\varphi(x/t_i)$	$\mathrm{T}(\exists x)\varphi(x)$ | $\mathrm{T}\varphi(x/c_i)$
False	$\mathrm{F}(\forall x)\varphi(x)$ | $\mathrm{F}\varphi(x/c_i)$	$\mathrm{F}(\exists x)\varphi(x)$ | $\mathrm{F}\varphi(x/t_i)$

Figure 12.1: Atomic tableaux for quantifiers; t_i is a term that doesn't contain any variables, and c_i is a new auxiliary constant we haven't used before in the branch (this is Table 7.2 of J. Bulín's logic lecture notes)

Note that in the table, we can find notation of the form $\varphi(x/y)$; which represents the *reasonable* replacement (substitution) of all relevant mentions of x with whatever y is. And by the way, defining what a reasonable replacement even means would require us to delve into substitutability, instances and variants; which is honestly not really important for the intuition and would only make things more difficult. Intuitively, we can just say that we're only allowed to make substitutions that will not change the fundamental meaning of the formula.

Exercise 12.1. Given unary relation symbols P and Q, prove that $(\forall x)(P(x)\to Q(x))\to((\forall x)P(x)\to(\forall x)Q(x))$ is a tautology (that is, it follows from the empty theory) using tableaux.

We must also note that we are right now dealing with languages without equality; however, some very interesting facts will still be provable.

13. Soundness and Completeness

For soundness, we'll first look at a lemma that may look familiar:

Lemma 13.1. If a structure $\mathcal{A}$ agrees with the root of a tableau from theory T, then this structure can be expanded with some constants to agree with all entries of some branch B.

This would allow us to prove soundness:

Theorem 13.2. (Soundness of the tableau method in predicate logic). If $T \vdash \varphi$, then $T \models \varphi$.

and we would also prove:

Theorem 13.3. (Completeness of the tableau method in predicate logic). If $T \models \varphi$, then $T \vdash \varphi$.

We are reaching a very technical part of the subject, so as to avoid boredom and despair, we will not go over the proofs of these facts, although in many ways they are quite similar to the ones in propositional logic.

However, we'll make a few comments on the proof of completeness: it hinges on the construction of a particular canonical model that agrees with a non-contradictory branch, and the idea is very interesting: we use the set of all terms in our language that do not contain variables as our domain (seeing them as finite strings).

Just like in propositional logic, we'll require some sort of systematic tableaux, which we'll not cover here, but it's essentially the same idea as in propositional logic, we'll just be adding an extra step to make sure that entries containing quantifiers are also reduced.

14. Consequences: Compactness and Löwenheim-Skolem

Assuming the truth of the soundness and completeness theorems for predicate logic, we prove compactness, and the proof is actually identical to the one in propositional logic, so we'll not repeat it:

THEOREM 14.1. (Compactness theorem in predicate logic) A theory T has a model if and only if every finite part has a model.

And finally, we can talk about two versions of the Löwenheim-Skolem theorem:

THEOREM 14.2. (Löwenheim-Skolem for languages without equality). If L is a countable language without equality and theory T in this language has a model, then T has a countably infinite model.

PROOF. Consider some sentence σ that is contradiction (such as, for example, $\varphi \wedge \neg\varphi$ for some atomic formula φ). Then, construct the systematic tableau from T of $F\sigma$. Given that T has a model, a proof of σ cannot be obtained, so some branch must be non-contradictory, and we can consider the canonical model of this branch (which is, by the way, countably infinite).

□

The main idea of the version where we introduce equality is this: equality (in the sense of identity) may cause the size of the canonical model of the non-contradictory branch to collapse into a finite size, so we can only talk of countable models, not countably infinite ones:

THEOREM 14.3. (Löwenheim-Skolem for languages with equality). If L is a countable language with equality and theory T has a model, then it has a countable model.

Now we can start playing around with these theorems to find rather counterintuitive consequences.

15. Disturbing Consequence #1: Non-Standard Model of the Natural Numbers

Consider the structure N whose domain is the natural numbers (including 0), equipped with the unary function S (the successor function), the binary function $+$ (addition), the binary function $\cdot$ (multiplication), the constant 0 (literally zero), and the binary relation $\leq$ (less than or equal to).

Let $\mathrm{Th}(\mathbb{N})$ be the set of all sentences valid in this structure, and let $\mathbf{n}$ be the term consisting of applying S to 0 n times, where $n \in \mathbb{N}$.

Now, let c be a new constant symbol, and consider the theory

$$T = \mathrm{Th}(\mathbb{N}) \cup \{\mathbf{n} < c \mid n \in \mathbb{N}\}$$

Every finite part of this theory has a model (the natural numbers, and where our interpretation of constant c is anything larger than all of the numerals mentioned in the axioms of the right set). So by the compactness theorem, T has a model. But what does this model represent? It's some sort of structure that behaves exactly like the natural numbers (in the sense of all of the statements of $\mathrm{Th}(\mathbb{N})$ being true), but has some element bigger than anything we would consider a natural number.

16. Disturbing Consequence #2: Countable Algebraically Closed Fields

We call a field F algebraically closed if every non-zero polynomial has a root (an example of such a field are the complex numbers $\mathbb{C}$.

We can express the notion of a field using a first-order theory (and it would be a good exercise to try to write out this theory), let's call it T. But then, to this theory we can append countably infinitely many axioms of this form:

$$(\forall x_{n-1}) \cdots (\forall x_0)(\exists y)(y^n + x_{n-1} y^{n-1} + \cdots + c_1 y + c_0) = 0$$

This new theory, let's call it T', will capture the notion of an algebraically closed field; so in particular, $\mathbb{C} \models T'$. But then, by Löwenheim-Skolem, this theory of algebraically closed fields has a countable model.

17. Disturbing Consequence #3: Skolem's Paradox

The foundations of mathematics are usually expressed in set theory, and the most usual set of axioms for set theory that we like to use is called ZFC (Zermelo-Fraenkel set theory with the axiom of choice). Crucially, this theory is a first order theory, and we like to assume that it has a model. And if this is the case, then by Löwenheim-Skolem, it has a countable model.

If you think about this, this means that there's a model of ZFC where there are countably many sets. But wait – aren't there some sets with uncountably many elements, and all of whose elements are sets?

There are many ways we can think of this apparent paradox – one way of seeing it is that any first-order theory will, inevitably, only be able to capture the "setness" of countably many sets.