1 A Propositional Language

The propositional language $L_0$ will be defined to be a set of words over the following alphabet $\Sigma$.

$$\Sigma = \{ \neg, \land, \lor, \rightarrow, ), (, p_1, p_2, \ldots \}$$

The $p_i$ are referred to as atomic formula symbols; $\neg, \land, \lor$, and $\rightarrow$ are called logical connective symbols; and $)$ and $( $ are grouping symbols.

Any member of $\Sigma^*$, the set of finite sequences of symbols from $\Sigma$, is called a word over $\Sigma$. The language $L_0$ is defined to be the smallest $X \subseteq \Sigma^*$ such that each of the following hold of $X$.

1. $X$ contains all the $p_i$.
2. If $A$ is in $X$, so is $\neg A$.
3. If $A$ and $B$ are in $X$, so are $(A \land B), (A \lor B)$, and $(A \rightarrow B)$.

The members of $L_0$, which are each finite sequences of symbols from $\Sigma$, are referred to as formulas.

In order to show that a formula $A$ is uniquely determined by the sequence of symbols $A \equiv \langle \sigma_1, \ldots, \sigma_n \rangle$, we want to show that there are not distinct ways of parsing $\langle \sigma_1, \ldots, \sigma_n \rangle$ that yield, say, $\langle \sigma_1, \ldots, \sigma_n \rangle \equiv (AcB) \equiv (A'c'B')$ with $A \neq A', B \neq B'$, or $c \neq c'$.

**Definition 1.1.** Define a map $W : \Sigma \rightarrow \{-1, 0, 1\}$ such that
1. \( W(p_i) = W(\ ) = -1 \)
2. \( W(\neg) = 0 \)
3. \( W(\rightarrow) = W(\lor) = W(\land) = W(\ ) = 1. \)

**Definition 1.2.** The weight of an expression \( A \equiv \langle \sigma_1, \ldots, \sigma_n \rangle \) is given by

\[
 w(A) = \sum_{i=1}^{n} W(\sigma_i).
\]

**Lemma 1.3.** If \( A \) is a formula, \( w(A) = -1 \). If \( A^- \) is a proper initial segment of \( A \), \( w(A^-) \geq 0 \).

Proof. If \( A \) is atomic, \( w(A) = -1 \). If \( A \equiv \neg B \), by the IH \( w(A) = 0 + -1. \) If \( A \equiv (B \cdot C) \), where \( c \) is a binary connective, \( w(A) = 1 + -1 + 1 - 1 - 1 = -1 \) by the IH. For the second part of the lemma, atomic formulas have no proper initial segments. If \( A \equiv \neg B \), for any proper initial segment \( A^- \), \( w(A^-) = 0 + m \), with \( m \geq 0 \) by the IH. If \( A \equiv (B \cdot C) \), the weight of proper initial segments have the form \( 1 + m \) or \( 1 + -1 + 1 + m \), where \( m \) is greater than or equal to zero by the IH.

**Theorem 1.4.** (Unique Readability) Suppose \( A, B, A', \) and \( B' \) are formulas.

1. If \( \langle \sigma_1, \ldots, \sigma_n \rangle \equiv \neg A \equiv \neg A' \), then \( A \equiv A' \)
2. If \( \langle \sigma_1, \ldots, \sigma_n \rangle \equiv (A \cdot c \cdot B) \equiv (A' \cdot c' \cdot B') \), then \( A \equiv A', B \equiv B' \), and \( c \equiv c' \).

Proof. (1) is clear. For (2), if \( A \not\equiv A' \), then one must be a proper initial segment of the other. Suppose \( A \) is a proper initial segment of \( A' \). Then \( A \) is not a formula by the lemma. But then neither is \( (A \cdot c \cdot B) \) contrary to our assumption. Therefore, \( A \equiv A' \). This forces \( c \equiv c' \) and \( B \equiv B' \).

## 2 Assignments and Validity

If \( \Gamma \subseteq L_0 \), we let \( \Gamma_{at} \) be the set of atomic formulas with occurrences in \( \Gamma \). An assignment for \( \Gamma_{at} \) is a function \( V : \Gamma_{at} \rightarrow \{0, 1\} \). An assignment \( V \) is extended to a valuation \( v \) for each of the formulas in \( \Gamma \) as follows.
1. If $A$ is atomic, then $v(A) = V(A)$.

2. If $A$ is $\neg B$, then $v(\neg B) = 1 - v(B)$.

3. If $A$ is $(B \land C)$, then $v(B \land C) = \min \{v(B), v(C)\}$.

4. If $A$ is $(B \lor C)$, then $v(B \lor C) = \max \{v(B), v(C)\}$.

5. If $A$ is $(B \rightarrow C)$, then $v(B \rightarrow C) = v(\neg B \lor C)$.

If $v(A) = 1$, we say that $v$ models $A$. If there is some valuation such that $v(A) = 1$, we say that $A$ is satisfiable. If every assignment for the atomic formulas with occurrences in $A$ extends to a valuation such that $v(A) = 1$, we say that $A$ is a tautology. This is written in symbols as $\models A$.

If $\Gamma$ and $\Delta$ are finite multisets of formulas and $\Theta$ is the set of atomic formulas with occurrences in $\Gamma \cup \Delta$, then $\Delta$ is a (semantic) consequence of $\Gamma$ if whenever all the formulas in $\Gamma$ evaluate to 1 under a valuation $v$ extending an assignment $V$ for $\Theta$, some $A \in \Delta$ evaluates to 1 under $v$ as well. We will use $\Gamma \vdash \Delta$ to indicate that $\Delta$ is a consequence of $\Gamma$. This is also expressed by saying that $\Delta$ follows (validly) from $\Gamma$.

**Proposition 2.1.** If $\Gamma$ and $\Delta$ are finite sets, then $\Gamma \vdash \Delta$ iff $\vdash \Lambda \Gamma \rightarrow \bigvee \Delta$, where $\Lambda \Gamma$ is a conjunction of all the formulas in $\Gamma$ and $\bigvee \Delta$ is a disjunction of all the formulas in $\Delta$.

**Proof.** $\Gamma \not\vdash \Delta$ iff there is a valuation such that $v(A) = 1$ for all $A \in \Gamma$ and $v(B) = 0$ for all $B \in \Delta$ iff $v(\Lambda \Gamma) = 1$ and $v(\bigvee \Delta) = 0$ iff $\not\vdash \Lambda \Gamma \rightarrow \bigvee \Delta$.\[1\]

**Theorem 2.2.** (Propositional Craig Interpolation) Suppose $\models A \rightarrow B$ and let $C \subseteq A_{at} \cap B_{at}$. Then there is some formula $I$—called an interpolant—composed of formulas from $C \cup \\{T\}$ such that $\models A \rightarrow I$ and $\models I \rightarrow B$.

**Proof.** If $v(A) = 0$ for all valuations and $A_{at} \cap B_{at} = \emptyset$, choose $I \equiv \neg T$. If this intersection is not empty, take some $p_i$ in it and let $I$ be $p_i \land \neg p_i$.

\[1\]To deal with the case when $\Gamma$ or $\Delta$ is empty, we define the valuation of the empty conjunction to be 1. The empty disjunction is always given the value 0.
There are $2^{|A_{at}|}$ possible assignments for the atomic formulas in $A$. If $v(A) = 1$ for a valuation extending one of these assignments $V$, define the following conjunction $A_v$. If $A_{at} \cap B_{at}$ is non-empty, let $p_i$ be any member of this intersection. If $V(p_j) = 1$, add $p_i \rightarrow p_i$ to the conjunction; if $V(p_j) = 0$ add $\neg(p_i \rightarrow p_i)$ to the conjunction. If $A_{at} \cap B_{at} = \emptyset$, replace $p_i$ everywhere with $\top$. Let $I$ be the disjunction of the $A_v$ for all valuations $v$ such that $v(A) = 1$.

By construction, $I$ is true iff $A$ is. Therefore, $\vdash A \rightarrow I$. If $\not\vdash I \rightarrow B$, $v(I) = 1$ and $v(B) = 0$ for some $v$. But then $v(A) = 1$ on this valuation, contradicting the fact that $\vdash A \rightarrow B$.

## 3 A Formal Proof Calculus

One question that one often wants answered when dealing with formal languages and valuations like the ones presented above is, “Is there an effective way to determine the tautologies?” An effective method for answering a question like this one is one which always gives the right answer in a finite number of steps. If a propositional formula $A$ contains $n$ distinct atomic formulas, we can always determine whether or not it is a tautology by checking its valuation for each of the $2^n$ possible assignments for its atomic formulas. Given proposition 2.1 and the fact that an argument consists of a finite number of premises and conclusions, we can also always determine in a finite number of steps whether $\Delta$ follows from some finite set of premises $\Gamma$ using this same method.

Checking $2^n$ assignments quickly becomes unfeasible, however. For example, if 60 different atomic formulas occur in $A$ and we are able to complete one of the $2^{60}$ rows in the required truth table per microsecond, it could take up to $\sim 366$ centuries to decide whether or not $A$ is a tautology. This is part of the motivation for introducing a deductive apparatus for our language. We’d like to come up with a way of generating tautologies and more complicated valid arguments from simpler ones like $p_1 \models p_1$.

Towards that end, we introduce the following so-called Gentzen-

---

2 See (Garey and Johnson 1979: 6-7).
3 Whether or not this generating procedure is strictly “fast” (i.e., can always be carried out in polynomial time) is an open question though. By (Cook and Reckhow 1979), there is some deductive
style apparatus, which will be referred to as \( G_P \).

**Structural Rules**

\[
\begin{align*}
\text{LW:} & \quad \frac{\Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta} \\
\text{LC:} & \quad \frac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta} \\
\text{RW:} & \quad \frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, A} \\
\text{RC:} & \quad \frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A} 
\end{align*}
\]

**Logical Connective Rules**

\[
\begin{align*}
\text{L\neg:} & \quad \frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta} \\
\text{R\neg:} & \quad \frac{A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A} \\
\text{L\land:} & \quad \frac{A, B, \Gamma \Rightarrow \Delta}{A \land B, \Gamma \Rightarrow \Delta} \\
\text{R\land:} & \quad \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \land B} \\
\text{L\lor:} & \quad \frac{A, \Gamma \Rightarrow \Delta}{A \lor B, \Gamma \Rightarrow \Delta} \\
\text{R\lor:} & \quad \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \lor B} \\
\text{L\rightarrow:} & \quad \frac{\Gamma \Rightarrow A, \Delta}{A \rightarrow B, \Gamma \Rightarrow \Delta} \\
\text{R\rightarrow:} & \quad \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \rightarrow B}
\end{align*}
\]

A proof in \( G_P \) is a finite rooted tree with leaves of the form \( A \Rightarrow A \), where \( A \) is atomic. Each non-leaf node of the tree is derived from one or more immediately preceding nodes by one of the rules listed above. The root of the tree is considered to be the sequent proved by the construction. \( \vdash \Gamma \Rightarrow \Delta \) will indicate that \( \Gamma \Rightarrow \Delta \) is derivable in \( G_P \).

### 4 Soundness and the Inversion Lemma

Since we’ve introduced this apparatus to generate valid arguments, we’d like to prove that the apparatus does just that. That is, we’d like to show that the moves we

apparatus in which all tautologies have fast proofs iff \( \text{NP} = \text{co-NP} \).

\( ^4 \) Other (possibly more familiar) proof systems such as the Hilbert-style systems with axioms and *modus ponens* as the inference rule and natural deduction systems involving assumptions and introduction and elimination rules can be shown to be capable of doing all the same work as a system like \( G_P \).
can make in our proof system correspond to true rules of reasoning, which never take us from a valid argument to an invalid one. Showing that our apparatus for proofs has this property is called showing it to be sound. (Note that a sequent \( \Gamma \Rightarrow \Delta \) is said to be valid iff \( \Gamma \models \Delta \).)

**Theorem 4.1. (Soundness Theorem)** \( \vdash \Gamma \Rightarrow \Delta \) implies \( \Gamma \models \Delta \).

*Proof.* The proof is by induction on the number of rules used in the derivation of \( \Gamma \Rightarrow \Delta \). If no inference rules are used, then the derivation consists of the single sequent \( A \Rightarrow A \), which is clearly valid. Suppose now that the first \( n \) applications of inference rules have produced valid sequents. There are 12 possible cases to consider for the \((n + 1)\)st rule applied.

If the \((n + 1)\)st rule applied is one of the structural rules, it is easy to see that lower sequent of the rule must be valid whenever the upper sequent is. The other cases are similarly straightforward. A few example cases illustrate the method.

Suppose the \((n + 1)\)st rule applied is L\( \neg \). Then we have derived a sequent of the form \( \neg A, \Gamma \Rightarrow \Delta \) from one of the form \( \Gamma \Rightarrow \Delta, A \). If all of the formulas in \( \Gamma \cup \{\neg A\} \) evaluate to 1 on \( v \), then \( v(A) = 0 \). By the validity of \( \Gamma \Rightarrow \Delta, A \), some formula in \( \Delta \) must evaluate to 1.

Suppose the \((n + 1)\)st rule applied is R\( \land \). Then this inference has the form

\[
\begin{array}{c}
\Gamma \Rightarrow \Delta, A \\
\Gamma \Rightarrow \Delta, B
\end{array} \quad \Rightarrow \\
\Gamma \Rightarrow \Delta, A \land B
\]

If all the formulas in \( \Gamma \) evaluate to 1 on \( v \), by the validity of the two upper sequents, one of the formulas in \( \Delta \cup \{A\} \) and one of the formulas in \( \Delta \cup \{B\} \) must evaluate to 1. If both \( v(A) = 1 \) and \( v(B) = 1 \), then \( v(A \land B) = 1 \). If one of \( A \) or \( B \) evaluates to 0, a formula in \( \Delta \) must evaluate to 1. In either case, the lower sequent is valid.

The next theorem is something like an inverse of the soundness theorem: it shows that validity also propagates up proof trees.

**Lemma 4.2. (Inversion Lemma)** For any non-weakening inference (i.e., excluding the LW and RW rules), the validity of an inference’s lower sequent implies the validity of its upper sequent(s).
Proof. There are 10 inference rules to consider here. Again, the structural-rule cases are trivial. And, again, a few example cases will illustrate the full proof of the theorem.

Consider the R¬ rule. The lower sequent in this case is $\Gamma \Rightarrow \Delta, \neg A$. If all the formulas in $\Gamma \cup \{A\}$ evaluate to 1, by the validity of the lower sequent, a formula in $\Delta$ must evaluate to 1 as well. Therefore, $A, \Gamma \Rightarrow \Delta$ is valid.

Consider next the L∨ rule. The rule consists of an inference of the form

$$\frac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \lor B, \Gamma \Rightarrow \Delta}.$$

If all the formulas in $\Gamma \cup \{A\}$ evaluate to 1, then so do all the formulas in $\Gamma \cup \{A \lor B\}$. By the validity of the lower sequent, there is a formula in $\Delta$ that evaluates to 1. The validity of the other upper sequent follows by symmetry.

5 The Completeness of $G_P$

The soundness theorem tells us that any provable sequent is also valid. The next natural question to ask is whether any valid sequent is provable. Showing this to be the case is called showing the deductive apparatus to be complete—in other words, the system can prove everything we’d reasonably hope it could. This is a more difficult question to answer, but, since none of our proof rules completely eliminate a formula from a derivation, every formula that occurs in the derivation also occurs in the sequent proved. (A proof system for which this holds is said to have the subformula property.) Because all the material we need is there in the proved sequent we can at least hope to generate a proof of that sequent by growing a proof tree using the proved sequent as the root.

The following proof of the completeness theorem yields extra information about the length of the proof of the valid sequent in question. This proof and the formulation of the system $G_P$ given above is drawn from (Buss 1998).

**Theorem 5.1.** (Completeness Theorem) If $\Gamma$ and $\Delta$ are finite multisets, $\Gamma \vDash \Delta$ implies $\vdash \Gamma \Rightarrow \Delta$. This proof of $\Gamma \Rightarrow \Delta$ employs fewer than $2^n$ logical connective inferences, where $n$ is the number of connectives occurring $\Gamma \cup \Delta$. 7
Proof. The proof is by induction on the number of logical connectives in \( \Gamma \Rightarrow \Delta \).

If there are no logical connectives in this sequent, there can only be atomic formulas in \( \Gamma \) and \( \Delta \). In such a case, the sequent can only be valid if some atomic \( A \) is in both \( \Gamma \) and \( \Delta \). \( \Gamma \Rightarrow \Delta \) can then be proved from the initial sequent \( A \Rightarrow A \) using only applications of the weakening rule. No logical connective inferences are used, and \( 0 < 2^0 \).

There are now 8 cases to consider. Suppose a formula of the form \( \neg A \) occurs in \( \Gamma \) and let \( \Gamma^- = \Gamma - \{\neg A\} \). By the inversion lemma, the upper sequent below is valid. By the IH, we can prove this sequent in fewer than \( 2^{n-1} \) (connective) steps.

The lower sequent then follows by \( L\neg \).

\[
\begin{align*}
\Gamma^- &\Rightarrow \Delta, A \\
\neg A, \Gamma^- &\Rightarrow \Delta \\
\end{align*}
\]

This proof uses fewer than \( 2^{n-1} + 1 \leq 2^n \) connective inferences. A similar argument allows us to draw the analogous conclusion when \( \neg A \in \Delta \).

Suppose a formula of the form \( (A \lor B) \) is in \( \Gamma \), and again let \( \Gamma^- \) be \( \Gamma \) with this formula removed. Since initial sequents contain only atomic formulas, if \( (A \lor B) \) is in \( \Gamma \) it must’ve gotten there \( (i) \) by an application of a structural rule, in which case the sequent’s provability is clear, or \( (ii) \) as a result of an inference of the form

\[
\begin{align*}
A, \Gamma^- &\Rightarrow \Delta \\
B, \Gamma^- &\Rightarrow \Delta \\
\end{align*}
\]

By the inversion lemma, both of these upper sequents are valid. By the IH, they both have proofs using less than \( 2^{n-1} \) steps. Therefore, the proof of \( \Gamma \Rightarrow \Delta \) contains less than \( 2 \cdot (2^{n-1} - 1) + 1 = 2^n \) connective inferences.

Suppose \( (A \lor B) \in \Delta \), and let \( \Delta^- \) be as above. This formula got into \( \Delta \) by applying a structural rule or by the inference

\[
\begin{align*}
\Gamma &\Rightarrow \Delta^- , A, B \\
\Gamma &\Rightarrow \Delta^- , A \lor B \\
\end{align*}
\]

In the first case, the provability of the sequent is again clear. In the second, by the inversion lemma and the IH, the upper sequent is provable in \(< 2^{n-1} \) steps. So, the conclusion can be reached in fewer than \( 2^n \) steps. The cases for \( (A \land B) \) and \( (A \rightarrow B) \) are proved similarly.
6 The (Propositional) Compactness Theorem

So far, the (multi)sets we’ve been working with have all been finite. Suppose, however, that $\Theta$ is any countable set of formulas. We say that $\Theta$ implies the sequent $\Gamma \Rightarrow \Delta$ iff $\Theta \models \bigwedge \Gamma \rightarrow \bigvee \Delta$. If $\Theta$ implies $\Gamma \Rightarrow \Delta$, we’ll write $\Theta \models \Gamma \Rightarrow \Delta$.

If there is some valuation $v$ such that $v$ models every sentence in $\Gamma$, we say that $\Gamma$ has a model. The Compactness Theorem says that if $\Gamma$ is a set of formulas every finite subset of which has a model, then $\Gamma$ itself has a model.

The first-order version of this theorem is of central importance in the subject of model theory. We’ll use this first-order version later to prove things about the axiomatizability of theories and to prove one direction of the Löwenheim-Skolem Theorem. For now, however, we’ll just have a brief look at the theorem and its proof and then apply it in one instance before moving on to consider a different, model-based approach to the completeness theorem. The following lemma will be needed for our proof.

**Lemma 6.1. (König’s Lemma)** Let $T$ be a rooted, finitely-branching tree. $T$ is infinite iff it has an infinite branch.

**Proof.** From right-to-left is clear. Call the root node of our tree $v_0$. Suppose that $v_0$ has infinitely many descendants. Since $T$ is finitely-branching, $v_0$ only has finitely many children. At least one of these children, therefore, must also have infinitely many descendants. (Otherwise, the total number of nodes in the tree would be a finite sum of finite sets of nodes.) Choose one of these children with infinitely many descendants and call it $v_1$. In general, suppose we have constructed a branch $v_0, \ldots, v_n$, where each of the $v_i$ has infinitely many descendants. By assumption, $v_n$ only has finitely many children, and so it must have at least one child with infinitely many descendants by the reasoning employed above. We may, therefore, extend our branch with one of these children labeled $v_{n+1}$. This inductive construction generates an infinite branch.

**Theorem 6.2. (Propositional Compactness Theorem)**

(i) $\Theta \models \Gamma \Rightarrow \Delta$ iff for some finite $\Theta' \subseteq \Theta$, $\Theta' \models \Gamma \Rightarrow \Delta$. 


(ii) $\Theta$ is satisfiable iff every finite subset of $\Theta$ is satisfiable.

Proof. (ii) implies (i) since

$$\Theta \not\models \Delta \iff \bigwedge \Gamma \not\models \bigvee \Delta \iff \Theta \cup \{ \neg \left( \bigwedge \Gamma \rightarrow \bigvee \Delta \right) \}$$

is satisfiable iff every finite subset is satisfiable iff for some finite $\Theta' \subseteq \Theta$, $\Theta' \not\models \Gamma \Rightarrow \Delta$.

So, we only have to prove (ii). The left-to-right direction is trivial. For the right-to-left direction, assume that $\Theta$ is not satisfiable. Since $\Theta$ is countable, the formulas in $\Theta$ can be enumerated as $A_1, A_2, \ldots$. If $\Theta$ is unsatisfiable, for every truth valuation $v$, for at least one of the $A_i$, $v(A_i) = 0$. Our plan will be to use this information to organize all the possible truth valuations of the $A_i$ into a finite, rooted, finitely-branching tree. By König’s lemma, this tree cannot have an infinite branch. The finite subset of formulas from $\Theta$ appearing on this tree will form our finite unsatisfiable subset of $\Theta$.

We construct the tree $T$ as follows. First, we’ll construct a larger complete binary-branching tree $T'$. Label the root of $T'$ $v$. Label $v$’s two children $A^R_1$ and $A^F_1$. In general, for any node $A^{(R/F)}_i$, label this node’s two children $A^R_{i+1}$ and $A^F_{i+1}$. Any branch through this complete tree gives a truth-valuation of all the $A_i$ in the obvious way. Since $\Theta$ is unsatisfiable, every branch of $T$ contains $A^F_i$ for some $i$. Let $T$ be $T'$ with the branch below $A^F_i$ removed for each such $A^F_i$. Every branch in $T$ is finite and $T$ is finitely-branching. Therefore, by König’s lemma, $T$ has only finitely-long branches. If we let $\Theta'$ be the set of $A_i$ such that $A^{(R/F)}_i$ appears as a node in $T$, $\Theta'$ is a finite subset of $\Theta$ that is not satisfiable.

We’ll now look at one simple application of the propositional form of the compactness theorem.

Definition 6.3. A graph $G$ can be represented by a pair of sets $V$ and $E$, where $V$ is a set of vertices (e.g., $\{v_1, \ldots, v_n\}$ or $\{v_1, v_2, \ldots\}$) and $E$ is a set of ordered pairs from $V$ such that $(v_j, v_i) \in E$ whenever $(v_i, v_j) \in E$ is (and, for present purposes, $(v_i, v_i) \notin E$). The elements of $E$ are called the edges of $G$. 

10
Example 6.4. The following is a pictorial representation of a graph.

The set of vertices for this graph \( V \) is \( \{v_1, \ldots, v_6\} \) and 

\[
E = \{(v_1, v_4), (v_1, v_5), (v_1, v_6), (v_4, v_1), (v_5, v_1), (v_6, v_1), \ldots\}.
\]

Definition 6.5. A graph \( G \) is said to be \( k \)-colorable if its vertices can be split up into \( k \) sets such that no two connected vertices are in the same set.

Example 6.6. The graph above is 2-colorable. Consider vertex \( v_1 \) and suppose we color it blue. Then \( v_4 \) can’t also be blue since it’s connected to \( v_1 \). So, color \( v_4 \) red. Then \( v_5 \) can’t be blue, but it can be red as well since it’s not connected to \( v_4 \).

In fact, if we color all the top vertices blue and all the bottom ones red, we’ll have partitioned the vertices into 2 sets (the red one and the blue one) such that no two connected vertices are in the same set.

We can use the compactness theorem to prove the following result, which holds even for infinite graphs (i.e., graphs where \( V \) is an infinite set).

Definition 6.7. If \( G = (V, E) \) is a graph and \( V' \subseteq V \) and \( E' = E \cap (V' \times V') \), we say that \( (V', E') \) is a subgraph of \( G \).

Proposition 6.8. Let \( G \) be a graph such that \( V \) is countable. Then if every finite subgraph of \( G \) is \( k \)-colorable, so is \( G \).

Proof. The whole challenge of proving this proposition via the compactness theorem is finding a way to characterize a graph’s \( k \)-colorability propositionally. Once we’ve done that properly, assuming that every finite subgraph of \( G \) is \( k \)-colorable
will guarantee that every finite subset of these formulas has a model. Therefore, by compactness, all the formulas together have a model. That is, $G$ is $k$-colorable.

Let $C = \{1, \ldots, k\}$ be the set names for our $k$ colors. 1 may correspond to ‘red’; 2 to ‘blue’; and so on. Further, let $p_{ij}$ be understood as saying “Vertex $v_i$ is colored $j$.” Now, if a given graph $G$ is $k$-colorable, then every vertex is one of the $k$ colors. So, the following formula should be true for every vertex of $G$: $p_{i1} \lor \ldots \lor p_{ik}$. Also, no vertex can be given more than one color, so we should also have $\neg(p_{i1} \land p_{i2}) \land \neg(p_{i1} \land p_{i3}) \land \ldots \land \neg(p_{i(k-1)} \land p_{ik})$ come out true for every vertex. Finally, in order to say that $G$ is $k$-colorable, we have to say that no connected vertices have the same color. This means that for each $(v_i, v_j) \in E$ we need a formula that says $\neg(p_{i1} \land p_{j1}) \land \ldots \land \neg(p_{ik} \land p_{jk})$.

Let $\Gamma$ be the set of all these formulas for the graph $G$. By assumption, any finite subset of these formulas has a model. By the compactness theorem $\Gamma$ has a model. In other words, $G$ is $k$-colorable.

7 The Extended Completeness Theorem

As a preview (with fewer moving parts) of a method commonly used to prove the completeness of first-order logics, we’ll consider the following model-theoretic proof of the “extended” completeness of a supplemented version of $G_P$ called $G_P + \text{Cut}$. We’ve already shown that $\Gamma \models \Delta$ implies $\vdash \Gamma \Rightarrow \Delta$, where $\Gamma$ and $\Delta$ are finite multisets. Our definition of a set of formulas $\Theta$ implying a sequent allowed for $\Theta$ to be infinite though. We’d like to show that whenever $\Theta \vdash \Gamma \Rightarrow \Delta$, there is some finite $\Theta' \subseteq \Theta$ such that $\vdash \Gamma \Rightarrow \Delta$ when formulas from $\Theta'$ can be freely used as axioms; that is, sequents of the form $\Rightarrow A$ can be used as leaves if $A \in \Theta'$.

The cut rule to be added to the other rules for $G_P$ is the following.

\[
\text{Cut: } \frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}
\]

This rule was not necessary for proving the completeness theorem, but its use is necessary for proving the extended completeness theorem. To see this, consider the
fact that $A \land B$ implies $A$. But there can be no proof of $\Rightarrow A$ from $\Rightarrow A \land B$ in $GP$ since $GP$ has the subformula property.

The method for proving completeness discussed in this section derives from a proof given by Leon Henkin in 1949 and proceeds by proving the contrapositive of the completeness claim. That is, we’ll show that supposing that there is no formal proof of $\Gamma \Rightarrow \Delta$ using a only a finite set of axioms from $\Theta$ allows us to find a valuation that makes all the formulas of $\Theta \cup \Gamma$ true and all of the formulas in $\Delta$ false—or what amounts to the same thing, a model of $\Theta \cup \{\neg(\bigwedge \Gamma \rightarrow \bigvee \Delta)\}$.\footnote{The extended completeness theorem can also be proved using methods similar to those employed in the completeness proof. If $\Theta$ implies $\Gamma \Rightarrow \Delta$, some finite $\Theta' \subseteq \Theta$ also implies $\Gamma \Rightarrow \Delta$ by compactness. By the completeness theorem, $\vdash \Theta' \cup \Gamma \Rightarrow \Delta$. Each formula in $\Theta'$ can be taken as an axiom, so by $|\Theta'|$ cuts, we can conclude $\vdash \Gamma \Rightarrow \Delta$.}

**Definition 7.1.** A set of formulas $\Gamma$ is said to be *consistent* if for no finite $\Gamma' \subseteq \Gamma$, can the sequent $\Gamma' \Rightarrow$ be derived.

**Definition 7.2.** If $\Gamma$ is a set of formulas, we let $\neg \Gamma = \{\neg A : \text{ for } A \in \Gamma\}$.

**Definition 7.3.** If the sequent $\Gamma \Rightarrow \Delta$ is provable using only a finite set of axioms from $\Theta$, we say that $\Gamma \Rightarrow \Delta$ is $\Theta$-provable.

**Lemma 7.4.** Suppose $\Gamma \Rightarrow \Delta$ is not $\Theta$-provable. Then $\Theta \cup \Gamma \cup \neg \Delta$ is consistent. Similarly, if $\Gamma \Rightarrow \neg \Delta$ is not $\Theta$-provable, $\Theta \cup \Gamma \cup \Delta$ is consistent.

*Proof.* For the first part, suppose not. Then for some finite $\Gamma' \subseteq \Gamma$ and $\Delta \subseteq \Delta'$, $\Gamma' \cup \neg \Delta' \Rightarrow$ is $\Theta$-provable. For any $\neg A \in \neg \Delta'$, we can use the $R\neg$ to derive $\Gamma \cup \neg A \Rightarrow \neg \Delta'$, where $\Delta' = \Delta' - \{A\}$. The sequent $\neg \neg A \Rightarrow A$ is also easily provable for any $A \in \Delta'$. Therefore, by an application of the cut rule, $\Gamma' \cup \neg \Delta' \Rightarrow A$. After $|\Delta'|$ applications of this algorithm, we have $\Theta$-proofs of $\Gamma' \Rightarrow \Delta'$ and by weakening $\Gamma \Rightarrow \Delta$ contrary to the supposition. Next, suppose that $\Gamma' \cup \Delta' \Rightarrow$ is $\Theta$-provable for some finite subsets of $\Gamma$ and $\Delta$. Then $|\Delta'|$ applications of $R\neg$-proves $\Gamma' \Rightarrow \neg \Delta'$. Weakening yeilds a $\Theta$-proof of $\Gamma \Rightarrow \neg \Delta$.\]

**Definition 7.5.** A set of formulas $\Gamma$ is said to be *negation complete* if for every formula $A$, either $A$ or $\neg A$ is $\Gamma$-provable.
The next lemma provides a clue for a method of constructing a negation complete set of formulas containing $\Theta$, $\Gamma$, and $\neg \Delta$.

**Lemma 7.6.** $\Gamma$ is negation complete iff for every atomic formula $A$, either $A$ or $\neg A$ is $\Gamma$-provable.

**Proof.** From left-to-right is trivial. The other direction is proved by induction on the complexity of $A$.

If $A$ is atomic, $A$ or $\neg A$ is $\Gamma$-provable by assumption. If $A \equiv \neg B$, by the induction hypothesis either $B$ or $\neg B$ is $\Gamma$-provable. In the second case, there is nothing to show; in the first $\neg B$ is $\Gamma$-provable by the use of L$\neg$ and R$\neg$. If $A \equiv B \land C$, then by the IH either $B$ or $\neg B$ and $C$ or $\neg C$ are $\Gamma$-provable. If $B$ and $C$ are both $\Gamma$-provable, so is $B \land C$. If one of $\neg B$ or $\neg C$ is $\Gamma$-provable, then

$$
\frac{\Rightarrow \neg B \quad B \land C \Rightarrow B}{\neg B, B \land C \Rightarrow} \quad \frac{B \land C \Rightarrow}{\Rightarrow \neg (B \land C)}
$$

is a $\Gamma$-proof of $\neg (B \land C)$. The other cases are proved similarly. $\dashv$

This lemma suggests the following method for constructing a consistent, negation complete set containing $\Theta$, $\Gamma$, and $\neg \Delta$.

Start with the consistent set $\Phi_0 = \Theta \cup \Gamma \cup \neg \Delta$. If neither $p_1$ nor $\neg p_1$ has a $\Phi_0$-proof, let $\Phi_1 = \Phi_0 \cup \{p_1\}$. Otherwise, $\Phi_1 = \Phi_0$. In general, if $\Phi_n$ has been defined, $\Phi_{n+1} = \Phi_n \cup \{p_n\}$ if neither $p_n$ nor $\neg p_n$ is $\Phi_n$-provable. Otherwise, $\Phi_{n+1} = \Phi_n$.

Note that this process never takes us from a consistent $\Phi_n$ to an inconsistent $\Phi_{n+1}$. For suppose that for some finite $\Phi_{n+1}' \subseteq \Phi_{n+1}$ we have $\vdash \Phi_{n+1}' \Rightarrow$. Since $\Phi_n$ is consistent by assumption, if $\Phi_{n+1}$ is inconsistent, it must be $\Phi_n \cup \{p_{n+1}\}$. But by lemma 7.4, if $\Phi_n \cup \{p_{n+1}\}$ is inconsistent, then $\neg p_{n+1}$ is $\Phi_n$-provable. Contradicting the fact that our procedure had us add $p_{n+1}$ to $\Phi_n$ to get $\Phi_{n+1}$.

**Proposition 7.7.** Let $\Phi^* = \bigcup_{i \in \mathbb{N}} \Phi_i$. Then $\Phi^*$ is consistent and negation complete.
Proof. $\Phi^*$ is negation complete by lemma 7.6 and its construction. Suppose now that $\Phi^*$ is inconsistent. Then for some $\Phi^* \subseteq \Phi^*$, the formulas from $\Phi^*$ can be used as axioms to give a $\Phi^*$ proof of the empty sequent. All of these formulas must be in $\Phi_n$ for some $n$, which contradicts the consistency of all the $\Phi_i$. ¬

The fact that a consistent set of formulas can always be extended to a consistent negation complete set in this way is often called Lindenbaum’s Lemma. We’ll see another version of this lemma when we discuss the completeness of first-order logic.

Since $\Theta \cup \Gamma \cup \neg \Delta \subseteq \Phi^*$, any model of $\Phi^*$ is also a model of $\Theta \cup \Gamma \cup \neg \Delta$, so any such model shows $\Theta \not\models \Gamma \Rightarrow \Delta$.

Lemma 7.8. $\Phi^*$ has a model.

Proof. In order to demonstrate this, we start by defining an assignment $V$ for $\Phi^*$. Define $V$ as follows. If $A$ is atomic and $\Phi^*$-provable, let $V(A) = 1$; otherwise, $V(A) = 0$. We’ll show that with this definition $v(A) = 1$ iff $A$ is $\Phi^*$-provable.

If $A$ is atomic and $v(A) = 1$, then by the way $V$ was defined $A$ has a $\Phi^*$-proof. If $A$ is $\Phi^*$-provable, by the definition of $V$, $v(A) = 1$.

Suppose that $A \equiv \neg B$ and $v(A) = 1$. Then $v(B) = 0$ and by the induction hypothesis, $B$ has no $\Phi^*$ proof. Since $\Phi^*$ is negation complete, $\neg B$ does. In the other direction, if $\neg B$ is $\Phi^*$-provable, the consistency of $\Phi^*$ implies that $B$ is not. So, by the IH, $v(B) = 0$. Therefore, $v(\neg B) = 1$.

Suppose $A \equiv B \lor C$ and $v(A) = 1$. Then $v(B) = 1$ or $v(C) = 1$. So, by the IH, $B$ or $C$ has a $\Phi^*$-proof. In either case, we may conclude that $B \lor C$ is $\Phi^*$-provable by RW and RV. For the other direction, suppose $B \lor C$ has a $\Phi^*$-proof. If neither $B$ nor $C$ is $\Phi^*$-provable, then since $\Phi^*$ is negation complete, $\neg B$ and $\neg C$ both have $\Phi^*$-proofs. By using the cut rule on the $\Phi^*$-provable sequents $B \Rightarrow \neg \neg B$ and $\neg \neg B \Rightarrow$, we can derive $\Phi^*$-proofs of $B \Rightarrow$ and $C \Rightarrow$. So, $\neg B \Rightarrow$ and $\neg C \Rightarrow$, and $\neg B \Rightarrow$ and $\neg C \Rightarrow$, we derive $\Phi^*$-proofs of $B \Rightarrow$ and $C \Rightarrow$.

Now, for all $A \in \Phi^*$, $A$ is $\Phi^*$-provable, so $v(A) = 1$ for all $A \in \Phi^*$. That is, $\Phi^*$ has a model. ¬
All of these results combine to yield a proof of the extended completeness theorem.

**Theorem 7.9.** (Extended Completeness Theorem) If $\Theta \models \Gamma \Rightarrow \Delta$, then $\Gamma \Rightarrow \Delta$ using only a finite set of premises from $\Theta$.

**Proof.** We prove the contrapositive. Suppose $\Gamma \Rightarrow \Delta$ is not $\Theta$-provable. Then by lemma 7.4, $\Theta \cup \Gamma \cup \neg\Delta$ is consistent. We can therefore extend this set to a consistent negation complete set $\Phi^*$ by the method described above. By lemma 7.8, $\Phi^*$ has a model. That is, there is some $v$ such that $v(A) = 1$ for all $A \in \Phi^*$. $\Theta \cup \Gamma \cup \neg\Delta$ is a subset of $\Phi^*$ so $v$ makes all of the formulas in $\Theta$, $\Gamma$, and $\neg\Delta$ true. But then $\Theta \not\models \Gamma \Rightarrow \Delta$. 

## 8 First-order Languages

We’re now ready to move on to the study of first-order languages. Many of the same concepts and proof methods employed for the propositional languages considered above will find a place here as well. But, our languages now will be far more expressive, and the results we prove will often be more interesting and, as a result, more difficult to prove.

Again, in order to define a formal language, we’ll need an alphabet to form words over.

**Definition 8.1.** A first-order language’s alphabet $\Sigma$ consists of the following.

1. **variables:** $v_1, v_2, \ldots$
2. **constant symbols:** $c_1, c_2, \ldots$
3. for each $n$, **$n$-ary function symbols:** $f_1^n, f_2^n, \ldots$

---

6 A first-order language is distinguished from higher-order languages by only allowing quantification over the individuals of our domain. That is, if our domain of quantification is the natural numbers, “Every number has a successor” is expressible in a first-order language, but, “Every set of natural numbers has a least element” is not since it quantifies over sets of elements.

7 For example, in a propositional language, an argument like “All men are mortal; Socrates is a man; therefore Socrates is mortal” has the form $p_1; p_2; \therefore p_3$, which is not valid. We certainly want our language to provide us with the means to capture the validity of arguments like this one though.

8 Aside from the variables and logical symbols, any of the symbols referred to in this definition may be omitted.
4. for each \( n \), \( n \)-ary predicate symbols: \( P^n_1, P^n_2, \ldots \)

5. logical symbols: \( \lor, \land, \neg, \to, \bot, \exists, \forall, = \)

6. grouping symbols: \( (), (\) \)

Members of \( \Sigma^* \) are again referred to as words over \( \Sigma \). When we eventually relate these first-order languages to various structures, we’ll want to have certain words that allow us to refer to particular objects. These expressions are called terms. The set of terms \( T \) is defined to be the smallest set \( X \) such that each of the following holds.

1. The variables and constant symbols of \( \Sigma \) are in \( X \).
2. if \( t_1, \ldots, t_n \) are in \( X \) and \( f \) is an \( n \)-ary function symbol, then \( f(t_1, \ldots, t_n) \) is in \( X \).

We can now define our language \( L \) to be the smallest \( X \subseteq \Sigma^* \) such that each of the following holds of \( X \).

1. If \( t_1, \ldots, t_n \) are terms, and \( P \) is an \( n \)-ary predicate symbol, \( P(t_1, \ldots, t_n) \) and \( t_1 = t_2 \) are in \( X \). (These formulas are said to be atomic.)
2. If \( A \) and \( B \) are in \( X \) and \( x \) is a variable, then \( \neg A \), \( (A \lor B) \), \( (A \land B) \), \( (A \to B) \), \( (\exists x A) \), and \( (\forall x A) \) are also in \( X \).

Note that some of the formulas in \( L \) contain free variables. These formulas don’t “say” any one thing, so they can’t be counted as being simply true or false. Consider, for instance, the difference between the two formulas \( \forall v_2(v_2 = 1) \) and \( (v_2 = 1) \). If our universe of discourse is taken to be the natural numbers, the first expression is simply false—not every number is equal to 1. However, we can’t say that the second expression is simply true or simply false. If we treat \( v_2 \) as temporarily picking out 1, which we’ll write as \( (v_2 = 1)[1] \), then the formula is true. Yet, \( (v_2 = 1)[5] \) is false.

Each formula that is neither simply true nor simply false will, like the example above, have a free variable in it. The true or false formulas—also called sentences—contain only bound variables. The distinction between bound and free variables is
familiar, but we can formally define the set of free variables in a formula $A$, written $FV(A)$, inductively as follows.

1. If $A$ is atomic, then $FV(A)$ is the set of all the variables occurring in $A$.
2. If $A \equiv (\neg B)$, then $FV(A) = FV(B)$.
3. If $A \equiv (B \land C)$ or $(B \lor C)$ or $(B \rightarrow C)$, then $FV(A) = FV(B) \cup FV(C)$.
4. If $A \equiv (\forall x B)$ or $(\exists x B)$, then $FV(A) = FV(B) - \{x\}$.
5. Nothing else is in $FV(A)$.

Example 8.2. In the language of arithmetic, the following are terms: $v_1$, 27, $3 + 5$, $4 \times (3 + 101)$. The following are not terms: $v_1v_2$, 2+, $-1 \times 2$. Some examples of formulas of this language are $2 + 4 = 19$, $2 = 2$, $3 + 1 = 4 \land v_4 = v_4$, $\forall v_5(v_5 + 1 = 2)$. The following are not formulas of this language: $\exists v_2 = 1$, $x = y$, $v_1 \lor v_2$, $3 + (\forall v_7(v_7 = v_7))$.

In order to prove that the terms and formulas of these new languages are unambiguous, we may proceed with the same basic methodology employed when proving unique readability of propositional formulas.

Definition 8.3. Define a map $W : \Sigma \rightarrow \{-1, 0, 1\}$ such that

1. $W(v_i) = W = (c_i) = W(\ ) = -1$
2. $W(\neg) = 0$

$w(A)$ for an expression $A = \langle s_1, \ldots, s_n \rangle$ is defined as before by summing $W(s_i)$ from 1 to $n$.

Lemma 8.4. If $t$ is a term, $w(t) = -1$. If $t^-$ is a proper initial segment, $w(t^-) \geq 0$.

---

9 According to our official definition, $3 + 5$ should be written as $+(3, 5)$, but we’ll generally use the normal “infix” form of these expressions and leave it understood what their real form is.
Proof. If $t$ is a variable or a constant symbol, $w(t) = -1$. If $t \equiv f(t_1 \ldots t_n)$, then $w(t) = n + 1 - 1 - n - 1 = -1$ by the IH. Variables and constant symbols have no proper initial segments, and for any proper initial segment $t^-$ of $f(t_1 \ldots t_n)$, $w(t^-) = n - k$ where $k < n$.

**Theorem 8.5. (Unique Readability of Terms)** For terms $t_1, \ldots, t_n, t'_1, \ldots, t'_n$,

$$\text{if } \langle s_1, \ldots, s_p \rangle \equiv f(t_1 \ldots t_n) \equiv f'(t'_1 \ldots t'_n), \text{ then } f \equiv f' \text{ and } t_i \equiv t'_i \text{ for } 1 \leq i \leq n.$$ 

Proof. That $f \equiv f'$ is clear. If $t_i \not\equiv t'_i$ for any $i$, then one is a proper initial segment of the other. But then one is not a term contrary to assumption.  

Using the fact that every term has weight $-1$, the analogue of the previous lemma for formulas follows easily. Further, the proof that the atomic formulas of $L$ are uniquely readable is essentially the same as the proof of the unique readability of terms. The full unique readability of formulas follows from the propositional case along with the fact that the result is immediate when $\langle s_1, \ldots, s_p \rangle \equiv QxA \equiv Q'xA'$, where $Q$ and $Q'$ are quantifiers.

9 Structures for First-order Languages

When discussing propositional languages, we were able to determine the truth or falsity of any formula by simply assigning each atomic formula the value 1 or 0. Things are a bit more complicated in the first-order case. Here, we’ll have to give a more complicated map from our language to a structure for the language.

**Definition 9.1.** Let $L$ be a first-order language. Then a structure $M$ for $L$ consists of the following.

1. a non-empty set $U$ called the universe of $M$
2. for each constant symbol $c$, an element $c^M \in U$.
3. for each $n$-ary function symbol $f$, a $n$-ary function $f^M : U^n \rightarrow U$
4. for each $n$-ary predicate symbol $P$, a set $P^M \subseteq U^n$
Example 9.2. Consider again the language of arithmetic. For now, in order to be especially careful about distinguishing between the symbols of our formal language and the ordinary operations of arithmetic, I’ll put a dot over the function symbols from our language. Just as $1$ is a symbol of our language, while $1$ is not, $\hat{+}$ is a function symbol from $\Sigma$, while $+$ indicates the actual function of addition on the natural numbers. With this convention in place, we can define the following language for arithmetic. We’ll take $0$ and $1$ as our only constant symbols, $\hat{P}$ as a binary function symbol, and $\hat{=} \equiv$ as a binary predicate symbol. (Let’s ignore variables for the moment.) The following is one possible structure for this language. Let $U = \mathbb{N}, 0^M = 0, 1^M = 1, \hat{+}^M$ be the $+$ function that takes $m+\text{n to } m + n \in \mathbb{N}$, and let $\hat{=}^M$ be the following subset of $\mathbb{N}^2$, $\{(n, n) : n \in \mathbb{N}\}$.

The next obvious step, which we’ll turn to in the next section, is to define what it is for a formula $A$ of our language to be true in a structure $M$. The definition will look familiar for the most part, but, because we have both formulas and sentences to deal with, some subtlety is necessary.

10 Satisfaction and Truth

The guiding idea behind the definition of truth according to an interpretation in what follows is that we want to say that, for example, $P(a)$ is true in a structure $M$ if the object named by $a$, i.e., $a^M$, is a member of $P^M \subseteq U$.

The way we’ll deal with open formulas was first worked out by Alfred Tarski. We’ll say that any formula can be satisfied or not by certain ways of substituting objects from our universe into the places of variables. So, for example, we’ll say that the open formula $v_1 = 2 + 3$ is satisfied by the assignment of the object $5$ as the denotation of the variable $v_1$. The idea is, again, if we take $v_1$ to function as a temporary name for $5$, we get a true sentence.

Definition 10.1. If $t$ is a term and $\overline{a}$ is a list that includes all the variables that occur in $t$, we’ll say that $\overline{a} = (a_1, \ldots, a_n)$, where all the $a_i$ are elements of $U$, is appropriate for $t$.

\[ ^{10} \text{We’re taking } n \text{ to abbreviate the sum of } n \text{ 1s. That is, } n \equiv 1 + \ldots + 1. \]
Now we’ll define the object of $U$ picked out by each term given an appropriate assignment, which we’ll write $t^M[\overline{a}]$.

1. If $t$ is a variable $v_i$, then $t^M[\overline{a}] = a_i$.
2. If $t$ is a constant symbol $c$, then $t^M[\overline{a}] = c^M$.
3. If $t$ is of the form $f(s_1, \ldots, s_n)$, where an $f$ is an $n$-ary function symbol and $s_1, \ldots, s_n$ are terms, then $t^M[\overline{a}] = f^M(s_1^M[\overline{a}], \ldots, s_n^M[\overline{a}])$.

Again, the idea here is to think of an appropriate assignment as telling us to treat the variable $v_1$ as a name for $a_1$, $v_2$ as a name for $a_2$, and so on. When thought of in this way, even a term containing variables will pick out a particular object of our universe given an assignment.

**Example 10.2.**

$((v_1 + 2) \times v_2)^M[1, 7] \Rightarrow (\times (+ (v_1, 2), v_2))^M[1, 7] \Rightarrow \times^M (+^M(v_1^M[1, 7], 2^M[1, 7]), v_2^M[1, 7]) \Rightarrow \times^M (3, 7) \Rightarrow$

We can now use this way of determining an object of the universe given an assignment to define what it is for an assignment to satisfy a formula in a structure $M$, which we’ll write $M \models A[\overline{a}]$.

**Definition 10.3.** If $\overline{a} = (a_1, \ldots, a_n)$ is an appropriate assignment for a term $t$ and $v_i$ is a variable, we say that $\overline{a}'$ is an $i$-variant of $\overline{a}$ iff $\overline{a}'$ differs from $\overline{a}$ in at most the $i^{th}$ place.

**Example 10.4.** $(1, 2, 4, 6, 7)$ and $(1, 2, 3, 6, 7)$ are 3-variants of $(1, 2, 3, 6, 7)$.

**Definition 10.5.** Definition of $M \models A[\overline{a}]$

1. If $A$ is an atomic formula, then it’s of the form (i) $P(t_1, \ldots, t_n)$ or (ii) $t_1 = t_2$.
   
   (i) $M \models A[\overline{a}]$ iff $(t_1^M[\overline{a}], \ldots, t_n^M[\overline{a}]) \in P^M$;
   
   (ii) $M \models A[\overline{a}]$ iff $t_1^M[\overline{a}] = t_2^M[\overline{a}]$.  

21
2. If \( A \equiv (\neg B) \), then \( M \models A[\overline{a}] \) iff \( M \not\models B[\overline{a}] \).

3. If \( A \equiv (B \land C) \), then \( M \models A[\overline{a}] \) iff \( M \models B[\overline{a}] \) and \( M \models C[\overline{a}] \).

4. If \( A \equiv (B \lor C) \), then \( M \models A[\overline{a}] \) iff \( M \not\models B[\overline{a}] \) or \( M \models C[\overline{a}] \).

5. If \( A \equiv (B \rightarrow C) \), then \( M \models A[\overline{a}] \) iff \( M \not\models B[\overline{a}] \) or \( M \models C[\overline{a}] \).

6. If \( A \equiv (\exists v_i B) \), then \( M \models A[\overline{a}] \) iff for some \( i\)-variant \( \overline{a}' \), \( M \models B[\overline{a}'] \).

7. If \( A \equiv (\forall v_i B) \), then \( M \models A[\overline{a}] \) iff for every \( i\)-variant \( \overline{a}' \), \( M \models B[\overline{a}'] \).

With this definition of satisfaction in place, we can say when a formula is true in a structure \( M \) and when it is valid.

**Definition 10.6.** A is true in a structure \( M \) iff it is satisfied by every assignment \( \overline{a} \). A is valid if it is true in every structure.

**Proposition 10.7.** Suppose \( t \) is a term and that \( \overline{x} \) is a list containing all the variables that occur in \( t \). If \( \overline{a} \) and \( \overline{b} \) are assignments appropriate for \( t \) such that if \( v_i \) occurs in \( t \), \( a_i = b_i \), then \( t^M[\overline{a}] = t^M[\overline{b}] \). \( ^{11} \)

**Proof.** If \( t \) is the variable \( v_i \), then by our definition, \( t^M[\overline{a}] = a_i \) and \( t^M[\overline{b}] = b_i \). By assumption, \( a_i = b_i \). Next, if \( t \) is a constant symbol \( c \), then, again according to our definition, \( t^M[\overline{a}] = c^M \) and \( t^M[\overline{b}] = c^M \). So, \( t^M[\overline{a}] = t^M[\overline{b}] \).

Finally, if \( t \) is of the form \( f(s_1, \ldots, s_n) \), where the \( s_i \) are terms, we have

\[
\begin{align*}
  f(s_1, \ldots, s_n)^M[\overline{a}] &= f^M(s_1^M[\overline{a}], \ldots, s_n^M[\overline{a}]) \\
  &= f^M(s_1^M[\overline{b}], \ldots, s_n^M[\overline{b}]) \\
  &= f(s_1, \ldots, s_n)^M[\overline{b}],
\end{align*}
\]

where the first equality uses our definition, the second is the induction hypothesis, and the third uses our definition again. \( \square \)

**Proposition 10.8.** Suppose \( A \) is a formula and that \( \overline{x} \) is a list containing all the variables that occur in \( A \). If \( \overline{a} \) and \( \overline{b} \) are assignments appropriate for \( t \) such that if \( v_i \) occurs free in \( t \), \( a_i = b_i \), then \( M \models A[\overline{a}] \) iff \( M \models A[\overline{b}] \). \( ^{12} \)

\( ^{11} \) In other words, the \( i \)th place in an assignment only matters if \( v_i \) actually occurs in \( t \).

\( ^{12} \) In other words, the \( i \)th place in an assignment only matters if \( v_i \) is free \( A \).
Proof. If \( A \) is atomic, then it is of the form \( P(t_1, \ldots, t_n) \) or \( t_1 = t_2 \), where the \( t_i \) are terms. Then \( M \models P(t_1, \ldots, t_n) [\overline{a}] \) iff \( (t_1^M[\overline{a}], \ldots, t_n^M[\overline{a}]) \in P^M \) and \( M \models t_1 = t_2[\overline{a}] \) iff \( t_1^M[\overline{a}] = t_2^M[\overline{a}] \). Since in an atomic formula all occurring variables are free, we can use Proposition 10.7 to conclude that \( t_i[\overline{a}] = t_i[\overline{b}] \).

So, \( (t_1^M[\overline{a}], \ldots, t_n^M[\overline{a}]) \in P^M \) iff \( (t_1^M[\overline{b}], \ldots, t_n^M[\overline{b}]) \in P^M \) iff \( M \models A[\overline{b}] \), and \( t_1^M[\overline{a}] = t_2^M[\overline{a}] \) iff \( t_1^M[\overline{b}] = t_2^M[\overline{b}] \) iff \( M \models t_1 = t_2[\overline{a}] \).

If \( A \equiv \neg B \), then \( M \models A[\overline{a}] \) iff \( M \not\models B[\overline{a}] \) iff \( M \not\models B[\overline{b}] \) iff \( M \models A[\overline{b}] \), where the induction hypothesis justifies the second equivalence.

If \( A \equiv (B \land C) \), then \( M \models A[\overline{a}] \) iff \( M \models B[\overline{a}] \) and \( M \models C[\overline{a}] \) iff \( M \models B[\overline{b}] \) and \( M \models C[\overline{b}] \) iff \( M \models A[\overline{b}] \), where the second equivalence is again justified by the IH. The argument for the \( \lor \) and \( \rightarrow \) cases are similar, and once we have considered the \( \exists \) case, the \( \forall \) case should be easy to complete as well.

If \( A \equiv (\exists v_i B) \), then \( M \models A[\overline{a}] \) iff for some \( i \)-variant \( \overline{a'} \), \( M \models B[\overline{a'}] \) iff \( M \models B[\overline{b}] \) for some \( i \)-variant \( \overline{b'} \) iff \( M \models A[\overline{b}] \). We can use the induction hypothesis to justify the second equivalence here because the free variables in \( B \) are just those in \( A \) plus \( v_i \), so if there’s some \( i \)-variant of \( \overline{a} \) that satisfies \( A \), there must be an \( i \)-variant of \( \overline{b} \) as well: By our assumption, \( \overline{a} \) and \( \overline{b} \) agree in all other (free) places that matter.

11 First-order Definability

**Definition 11.1.** If \( M \) is a structure for a language \( L \), we say that \( X \subseteq U^n \) is **definable** (with parameters) iff there is some formula \( A(\overline{x}, \overline{y}) \) in \( L \) and \( \overline{b} \) from \( U^m \) such that \( X = \{ (a_1, \ldots, a_n) \in U^n : M \models A(a_1, \ldots, a_n, \overline{b}) \} \).

We’ve mentioned “the language of arithmetic” several times so far without giving a precise definition of what language is intended. We’ll fix the meaning of the term now by taking the following language of Peano arithmetic (PA) to be the intended interpretation of “language of arithmetic” from now on. The alphabet for \( L_{PA} \) consists of

1. countably many variables: \( v_1, v_2, \ldots \);
2. a constant symbol 0;
3. function symbols $s, +, \times$;
4. for each $n$, countably many predicate symbols: $P_1^n, P_2^n, \ldots$; and
5. the logical symbols: $\neg, \lor, =, \forall$
6. the grouping symbols: $), ($.

The terms and formulas of $L_{PA}$ are defined as above.

The standard structure for this language, $N$, consists of the natural numbers; $s^N : \mathbb{N} \rightarrow \mathbb{N}$ as the successor function, which maps a number $n$ to $n + 1$; and the other symbols interpreted as expected.

Very many familiar sets of natural numbers are definable in $L_{PA}$. For example,

1. $\exists v_2 (2 \times v_2 = v_1)$
2. $v_1 \neq 1 \land \forall v_2 \forall v_3 (v_1 = v_2 \times v_3 \implies (v_2 = 1 \lor v_3 = 1))$
3. $\exists v_3 (v_1 + v_3 = v_2)$
4. $\exists v_3 (v_1 + v_3 = v_2 \land v_3 \neq 0)$

define the even numbers, the prime numbers, the relation $\leq$, and $<$ respectively. The sets definable in $L_{PA}$ are complex and have been studied extensively.

Later, when we discuss Gödel’s incompleteness theorems (see section 23), we’ll develop a notion of “provable definability” in $L_{PA}$. We’ll say, for example, that a formula $A(x)$ of $L_{PA}$ binumerates a property $P$ if $P(n)$ implies that $A(n)$ can be proved in PA and not $P(n)$ implies $\neg A(n)$ can be proved in PA.

The following proposition proved early in (Marker 2002) characterizes the definable sets in a structure for $L$ in terms of several closure properties.

**Definition 11.2.** If $f : A \rightarrow B$ is a function, the **graph of** $f$ is the following set of ordered pairs: $\{(a, f(a)) : a \in A\}$.

**Proposition 11.3.** If $M$ is a structure for $L$ and for all $n \geq 1$, each $D_n$ is a collection of subsets of $U^n_M$ such that
1. $U^n_M \in D_n$
2. For all $n$-ary $f$, the graph of $f^M$ is in $D_{n+1}$
3. For all $n$-ary $P$, $P^M \in D_n$
4. For all $i, j \leq n$, $\{(a_1, \ldots, a_n) \in U^n_M : a_i = a_j\} \in D_n$
5. If $X \in D_n$, then $U_M \times X \in D_{n+1}$
6. If $X \in D_{n+m}$ and $b \in U^n_M$, then $\{a \in U^n_M : (a, b) \in X\} \in D_n$
7. Each $D_n$ is closed under taking complements, unions, and intersections
8. If $X \in D_{n+1}$ and $\pi : U^{n+1}_M \to U^n_M$ projects an $(n+1)$-tuple onto its first $n$ coordinates, $\pi(X) \in D_n$

Then $X \subseteq U^n_M$ is definable iff $X \in D_n$.

**Proof.** We first want to show that all sets in the $D_n$ are definable. $U^n_M \in D_n$ for all $n$. This set is definable by $v_1 = v_1$ since any tuple from $U_M$ satisfies this formula. The graph of any $n$-ary function $f$ is defined by $f(v_1, \ldots, v_n) = v_{n+1}$. Similarly, an $n$-ary $P^M$ is defined by $P(v_1, \ldots, v_n)$. The set $\{(a_1, \ldots, a_n) \in U^n_M : a_i = a_j\}$ is defined by $v_i = v_j$. This takes care of (1)-(4).

For (5), suppose that $X \in D_n$ is definable by $A(v_1, \ldots, v_n, \overline{b})$, then $U_M \times X$ is definable by $A(v_2, \ldots, v_{n+1}, \overline{b})$ since the first coordinate in any assignment can now vary freely. The sets required by (6) are definable as well. If $X \subseteq U^{n+m}_M$ is defined by $A(v_1, \ldots, v_{n+m}, \overline{c})$ and $\overline{b} \in U^n_M$, then $A(v_1, \ldots, v_n, \overline{b}, \overline{c})$ defines $\{\overline{a} \in U^n_M : (\overline{a}, \overline{b}) \in X\}$. Complements, intersections, and unions of definable sets can be defined by negating, conjoining, and disjoining their defining formulas. For example, if $X, Y \subseteq U^n_M$ are defined by $A(\overline{x}, \overline{b})$ and $B(\overline{x}, \overline{c})$ respectively, $A(\overline{x}, \overline{b}) \land B(\overline{x}, \overline{c})$ defines $X \cap Y$. Projection onto the first $n$ coordinates is defined by binding the $(n + 1)^{st}$ variable in the set’s defining formula as, for example, in $\exists v_{n+1} C(\overline{x}, v_{n+1}, \overline{b})$. This takes care of (7) and (8).

We must show next that if $X \subseteq U^n_M$ is definable, it is in $D_n$. We first show that for any term $t$, $\{(\overline{a}, b) \in U^{n+1}_M : i^M[\overline{a}] = b\}$ is in $D_{n+1}$. 25
If \( t \) is a constant symbol \( c \), by (4) we have \( \{(c^M, c^M)\} \in D_2 \). By (6), \( \{c^M\} \in D_1 \). Applying (5) \( n \) times shows that \( \{(\overline{a}, c^M) : \overline{a} \in U^n_M\} \) is in \( D_{n+1} \). If \( t \) is a variable \( v_i \) and \( 1 \leq i \leq n \), then \( \{(\overline{a}, a_i)\} \in D_{n+1} \) by (4). If \( i > n \), then \( \{(\overline{a}, a_i)\} \in D_{n+1} \) by (1).

If \( t \) is an \( m \)-ary function symbol \( f \), we need to show that \( \{f^M : \overline{a} \in U^n_M\} \) is in \( D_n \). The graph of \( f^M \) is in \( D_{n+1} \) by (2). Call this set \( G \). By the IH, for all \( t_i \), the graph of \( t_i, G_i = \{(\overline{a}, b) : \overline{a} \in U^n_M\} \), is in \( D_{n+1} \). Then

\[
\{(\overline{a}, b) : \exists \overline{x} \left( (\overline{a}, x_1) \in G_1 \land \ldots \land (\overline{a}, x_m) \in G_m \land (\overline{x}, b) \in G \right)\}
\]

is in \( D_{n+1} \) by (7) and (8). This is the set of \( (\overline{a}, b) \) such that \( f^M \) applied to the \( t_i^M \) applied to \( \overline{a} \) give \( b \).

Now, if \( A \equiv t_1 = t_2 \), then \( A \) defines the set \( \{\overline{a} \in U^n_M : t_1^M[\overline{a}] = t_2^M[\overline{a}]\} \). This is the same set as \( \{\overline{a} \in U^n_M : \exists x \exists y (t_1^M[\overline{a}] = x \land t_2^M[\overline{a}] = y \land x = y)\} \), which is in \( D_n \) by the above induction and by properties (4), (7), and (8).

If \( A \) is an \( n \)-ary predicate symbol \( P(\overline{t}) \), \( A \) defines the set \( \{\overline{a} \in U^n_M : M \models A[\overline{a}]\} \). \( P^M \in D_m \) by (3). By the above induction and closure under unions and projections,

\[
\{\overline{a} : \exists \overline{x} (t_1^M[\overline{a}] = x_1 \land \ldots \land t_n^M[\overline{a}] = x_n \land \overline{x} \in P^M)\}
\]

is in \( D_n \).

So, the sets definable by atomic formulas are in the \( D_n \). Since these sets are closed under intersection, union, complementation, and projection, sets defined by any logical combination of atomic formulas are also in some \( D_n \). This shows that every \( X \subseteq U^n_M \) definable without parameters is in \( D_n \). Property (6) allows us to extend this conclusion to the subsets of \( U^n_M \) definable with parameters.

### 12 Adding Quantifier Rules to \( G_P \)

We’d now like to extend the deductive apparatus introduced in section 3 to allow us to prove things involving the quantifiers. Once again, our rules will come in
left-and-right pairs. We also add some new initial sequents to those allowed in $G_P$ to deal with identity. We’ll refer to the new proof system as $G^=$. When we don’t allow the use of the new identity axioms, we’ll call the system $G$.

The new axioms are of the following three forms.

\[ \Rightarrow \quad t = t \]
\[ s_1 = t_1, \ldots, s_n = t_n \quad \Rightarrow \quad f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n) \]
\[ s_1 = t_1, \ldots, s_n = t_n, P(s_1, \ldots, t_n) \quad \Rightarrow \quad P(t_1, \ldots, t_n) \]

Quantifier Rules:

\[
\begin{align*}
\text{L∀:} & \quad A(t), \Gamma \Rightarrow \Delta \\ & \quad \forall x A(x), \Gamma \Rightarrow \Delta \\
\text{R∀:} & \quad \Gamma \Rightarrow \Delta, A(y) \\ & \quad \Gamma \Rightarrow \Delta, \forall x A(x) \\
\text{L∃:} & \quad A(y), \Gamma \Rightarrow \Delta \\ & \quad \exists x A(x), \Gamma \Rightarrow \Delta \\
\text{R∃:} & \quad \Gamma \Rightarrow \Delta, A(t) \\ & \quad \Gamma \Rightarrow \Delta, \exists x A(x)
\end{align*}
\]

The variable $y$ occurring in $R∀$ and $L∃$ must be free and cannot occur in $\Gamma$ or $\Delta$. This variable is called the eigenvariable of the inference.

13 Soundness of $G^=$

We can now move on to proving the soundness theorem for our first-order system.

**Theorem 13.1. (Soundness Theorem)** \(\vdash \Gamma \Rightarrow \Delta\) implies \(\Gamma \vDash \Delta\).

**Proof.** The proof is again by induction on the length of the derivation of \(\Gamma \Rightarrow \Delta\). For the base case, any axiom is easily seen to be valid. The induction step for the inference rules also in \(G_P\) is essentially the same as in the proof of soundness given in section 4 except that we must now pay attention to assignments.

For example, suppose that our \((n + 1)^{st}\) step derives \(A \land B, \Gamma \Rightarrow \Delta\) by $L\land$. Then we must have \(A, B, \Gamma \Rightarrow \Delta\) earlier in the proof. So, any structure \(M\) and any variable assignment \(\overline{a}\) that makes all the sentences in $\Gamma, A$, and $B$ true also does so for some formula in $\Delta$. Since $A \land B$’s truth in a structure for $\overline{a}$ implies the

---

13 $\Gamma \vDash \Delta$ says that any structure $M$ and assignment $\overline{a}$ that satisfies every member of $\Gamma$ also satisfies some member of $\Delta$. 

27
truth of $A$ and $B$, some formula in $\Delta$ must be true as well for this structure and assignment.

Suppose that our $(n+1)^{st}$ step derives $\forall v_i A(v_i), \Gamma \Rightarrow \Delta$ by $L\forall$. Then the sequent $A(t), \Gamma \Rightarrow \Delta$ must also appear in the proof. If $\forall v_i A(v_i)$ is satisfied by an assignment $\bar{a}$ in some structure $M$, then so must be $A(t)$, since every $i$-variant of $\bar{a}$ satisfies $A(v_i)$. So, if by the IH, we must have some formula in $\Delta$ satisfied by $\bar{a}$.

Suppose that our $(n+1)^{st}$ step derives $\Gamma \Rightarrow \Delta, \forall v_i A(v_i)$ by $R\forall$. Then we must have $\Gamma \Rightarrow \Delta, A(y)$ earlier in our proof, where $y$ is free and does not occur in $\Gamma$ or $\Delta$. Suppose that an assignment $\bar{a}$ satisfies all the formulas in $\Gamma$ in some structure $M$. The only case we need to consider is if $A(y)$ is the only formula in $\Delta \cup \{A(y)\}$ that is satisfied by $\bar{a}$. Since $y$ does not occur in $\Gamma$ or $\Delta$, by proposition 10.8 any variant of $\bar{a}$ assigning a different object as the denotation of $y$ will still satisfy all the formulas in $\Gamma$. So, by the IH, any such variant of $\bar{a}$ must satisfy $A(y)$. Therefore, $\forall v_i A(v_i)$ is also satisfied by $\bar{a}$ in $M$. The cases for the $\exists$-rules are proved similarly.

We can use this theorem to prove non-derivability results.

**Proposition 13.2.** The sequent $\Rightarrow \forall v_1 \exists v_2 p^2_1(v_1, v_2) \iff \exists v_1 \forall v_2 p^2_1(v_1, v_2)$ is not derivable in $G$.

**Proof.** Consider a structure with domain $\{1, 2\}$. And let $p^2_1 = \{(1, 2), (2, 1)\}$. If this sequent were provable, it would contradict the soundness theorem, so it must not be. 

14 A Henkin-style Completeness Proof

We’d now like to show by a method expanding on the one used in section 7 that if $\Theta$ implies $\Gamma \Rightarrow \Delta$, then $\Gamma \Rightarrow \Delta$ is $\Theta$-provable. We’ll again show this by demonstrating the contrapositive. So, we’ll assume $\Gamma \Rightarrow \Delta$ does not have a $\Theta$-proof and build a structure and assignment that satisfies all the formulas in $\Theta \cup \Gamma$ but fails to satisfy any formula in $\Delta$. Again, for this proof we will supplement $G^\omega$ with the cut rule.
The cut elimination theorem proved below will show that its use is not necessary to prove the completeness of $G$ when $\Theta$ is empty.

**Lemma 14.1.** If $\Gamma \Rightarrow \Delta$ is not $\Theta$-provable, then $\Theta \cup \Gamma \cup \neg \Delta$ is consistent. If $\Gamma \Rightarrow \neg \Delta$ is not $\Theta$-provable, then $\Theta \cup \Gamma \cup \Delta$ is consistent.

*Proof.* See the proof of lemma 7.4.

Our goal, once again, will be to show that any consistent set of sentences is satisfiable. Since, on the supposition that $\Gamma \Rightarrow \Delta$ is not $\Theta$-provable, $\Theta \cup \Gamma \cup \neg \Delta$ is consistent, we’ll be in position to show that this set formulas is satisfiable. This will show that $\Theta \not\models \Gamma \Rightarrow \Delta$.

The idea here is to build a model for a consistent set of formulas out of the syntactic materials of the language itself. We’ll take all the closed terms of our language and use them as the entities making up our structure. So, $P(c)$ will be interpreted in our model as saying that the symbol $c$ itself has some property $P^M$, and so on. The plan then is to proceed as we did in section 7: We will expand $\Theta \cup \Gamma \cup \neg \Delta$ to something that’s negation complete and then find a way to satisfy that enlarged set of formulas, which will thereby satisfy the original set. There are some obstacles that make this process of model-building not quite as straightforward as we may have hoped however.

The first problem is that we may have sentences such as $\exists x P(x)$ that we want to make true in our model, yet there’s nothing that guarantees that there is some closed term $c$ of our language such that $P(c)$ is also around whenever $\exists x P(x)$ is. If there is no such $c$, however, there won’t be any object in our domain that verifies the sentence $\exists x P(x)$ in our structure. We must, therefore, find a way of ensuring that there’s some $c$—called a *witnessing constant*—such that $P(c)$ is also a sentence involved in our construction whenever we have a formula like $\exists x P(x)$ involved.

The second problem is that our language will have the $=$ symbol and we would like to interpret it as real identity, but we may have sentences that we need to make true in our structure of the form $a = b$, where $a$ and $b$ are distinct closed terms. If we take our structure to consist of the closed terms of our language, $a$, the denotation of $a$, won’t be really the same thing as $b$—they’re *distinct* closed terms. We’ll
solve this problem by defining an equivalence relation that identifies closed terms whenever we can prove \( a = b \) of them. We will then mod out by this relation and be left with a new structure where such an \( a \) and \( b \) have been identified.

We’ll begin by solving the first of our problems. We want to add enough constant symbols to \( \Sigma \) to ensure we have a language rich enough to ensure that there are constants able to serve as witnesses. Take a list of all the formulas of \( L \) with one free variable \( P(v_1), P(v_2), \ldots \). The idea is to add a new constant symbol to \( \Sigma \) for each one of these formulas. So, for \( P(v_1) \), we’ll add a constant symbol \( d_{P(v_1)} \), for \( P(v_2) \) we’ll add \( d_{P(v_2)} \), and so on. Every time we add one of these constant symbols for a formula with one free variable, we’ll add a witnessing axiom for that constant symbol to another set \( H \). The witnessing axiom for \( d \) is \( \exists x P(x) \rightarrow P(d) \). The idea here is that we’ll be able to take any of the axioms from \( H \) and use them as axioms in proofs when we try to prove things in our language built from the supplemented \( \Sigma \).

By adding these constant symbols and axioms, we will have gone from our original language \( L = L_0 \) to a new language, call it \( L_1 \). But now in \( L_1 \) we can have new formulas like \( P(x, d) \) which once again contain one free variable, but have no witnessing constant. \( P(x, d) \) doesn’t get a witnessing constant in the course of our previous construction because \( P(x, d) \) wasn’t part of the language \( L_0 \). However, since the formulas with one free variable in \( L_1 \) can also be enumerated and we can go through each of them and add witnessing constants and axioms for each of these formulas as well to construct the language \( L_2 \).

In order to ensure that every formula with one free variable has both a witnessing constant and axiom for it, we need to reiterate this process countably many times. Each time, we move from \( L_n \) to \( L_{n+1} \) by adding in witnessing constants to \( \Sigma \) for the formulas of \( L_n \) with one free variable and axioms for these constants to \( H \).

To take all these languages together, we let

\[
L_H = \bigcup_i L_i,
\]

and let \( H \) be the set of all witnessing axioms for constants in \( L_H \).
Definition 14.2. If \(d\) is a witnessing constant for a formula of language \(L_n\), we say that \(d\) is of level \(n + 1\). Similarly, the witnessing axiom for \(d\), \(\exists x P(x) \rightarrow P(d)\), is called a witnessing axiom of level \(n + 1\).

Proposition 14.3. If \(d\) is a witnessing axiom of level \(n + 1\), and \(P\) is a formula from language \(L_k\) where \(k \leq n\), then \(d\) does not appear in \(P\).

Proof. This is clear. \(d\) does not enter any language until \(L_{n+1}\), so it can’t appear in any formula from an earlier language.

Proposition 14.4. If \(d_1\) and \(d_2\) are witnessing constants such that the level of \(d_1\) is less than or equal to the level of \(d_2\), then \(d_2\) does not appear in the witnessing axiom for \(d_1\).

Proof. If \(d_2\) is of level strictly greater than \(d_1\), which is of level \(n + 1\), then it can’t appear in \(d_1\)’s witnessing axiom because this axiom is introduced in the move from language \(L_n\) to language \(L_{n+1}\) and is an axiom for some formula from language \(L_n\). But \(d_2\) is of a strictly greater level, so it can’t appear in any \(P\) from \(L_n\). If the level of \(d_1\) and \(d_2\) are both, say, \(n + 1\), then they are both introduced in the move from \(L_n\) to \(L_{n+1}\), but, again, the witnessing axiom for \(d_1\) is introduced for a formula from \(L_n\), so \(d_2\) can’t be a part of the formula from \(L_n\) in the antecedent of the witnessing axiom that \(d_1\) is introduced for. It can’t be in the consequent either because distinct formulas from \(L_n\) get distinct witnessing constants.

We need to make sure that in adding all these extra constant symbols and witnessing axioms we haven’t given ourselves the resources to prove new theorems in our original language \(L\). If we could prove new things, our original assumption that \(\Gamma \Rightarrow \Delta\) is not \(\Theta\)-provable may no longer be true. So, we want to show that if a sequent \(\Gamma \Rightarrow \Delta\) whose formulas are all from \(L\) can be \(\Theta\)-proved in \(G\) supplemented by some set of witnessing axioms from \(H\) (in language \(L_H\)), then \(\Gamma \Rightarrow \Delta\) can also be \(\Theta\)-proved in \(G\) without these axioms.

Lemma 14.5. Suppose \(d\) doesn’t appear in \(\Gamma\) or \(\Delta\) and \(\Gamma \Rightarrow \Delta\) can be \(\Theta\)-proved possibly using \(\Rightarrow \exists x P(x) \rightarrow P(d)\) as an axiom, then \(\Gamma \Rightarrow \Delta\) can be \(\Theta\)-proved without using this axiom.
Proof. By the soundness of $\mathcal{G}^= + \text{Cut}$ and the completeness theorem for $\mathcal{G}_P + \text{Cut}$, for some $A \in \Delta$, each of the following is $\Theta$-provable: $\Gamma \Rightarrow (\exists x P(x) \rightarrow P(d)) \rightarrow A; \neg \exists x P(x), \Gamma \Rightarrow A$; and $P(d), \Gamma \Rightarrow A$. Since $d$ does not occur anywhere in $\Gamma$ or $A$, the proof of $P(d), \Gamma \Rightarrow A$ can be transformed into a proof of $P(y), \Gamma \Rightarrow A$ by replacing $d$ everywhere by a new variable $y$. Then by $R\exists$ we have a $\Theta$-proof of $\exists x P(x), \Gamma \Rightarrow A$. $\Gamma \Rightarrow A$ is a propositional consequence of $\exists x P(x), \Gamma \Rightarrow A, \neg \exists x P(x), \Gamma \Rightarrow A$, and $\Rightarrow \exists x P(x), \neg \exists x P(x)$, so by the completeness of $\mathcal{G}_P + \text{Cut}$, $\Gamma \Rightarrow A$ is $\Theta$-provable. Therefore, $\Gamma \Rightarrow \Delta$ is $\Theta$-provable by weakening. The axiom $\Rightarrow \exists x P(x) \rightarrow P(d)$ was not used in this proof. \(\exists\)

Theorem 14.6. (Elimination Theorem) If $\Gamma$ and $\Delta$ are from $\mathcal{L}$ and $\Gamma \Rightarrow \Delta$ is $(\Theta \cup H)$-provable, then $\Gamma \Rightarrow \Delta$ is $\Theta$-provable.

Proof. The proof is by induction on the number of axioms from $H$ used in the proof. For the base case, there are no witnessing axioms used, so clearly $\Gamma \Rightarrow \Delta$ is $\Theta$-provable.

Suppose for the induction hypothesis that $n$ uses of axioms from $H$ can be eliminated. Suppose now that there are $n + 1$ uses of axioms in the proof of $\Gamma \Rightarrow \Delta$. Find the axiom with the greatest possible level, and suppose it is the witnessing axiom for the witnessing constant $d$. By proposition 14.4, $d$ doesn’t occur in any of the other axioms used in the proof of $\Gamma \Rightarrow \Delta$. Further, since $d$ is not in $\mathcal{L}$ it doesn’t occur in $\Gamma$ or $\Delta$ either. Therefore, by lemma 14.5, $\Gamma \Rightarrow \Delta$ can be proved without appealing to this witnessing axiom. Thus, this axiom can be eliminated from the proof of $\Gamma \Rightarrow \Delta$. By the IH, the other $n$ witnessing axioms can be eliminated from the proof as well. Therefore, $\Gamma \Rightarrow \Delta$ is $\Theta$-provable. \(\exists\)

The Elimination Theorem shows that our extension of $\mathcal{L}$ to $\mathcal{L}_H$ and the addition of axioms from $H$ is conservative. That is, we can’t prove new things in our old language using pieces of the new language and witnessing axioms. This implies that $\Theta \cup \Gamma \cup \neg \Delta \cup H$ is consistent.

We now know how to extend a consistent set of formulas to a set of formulas in a new language with the addition of witnessing axioms to solve our first problem. We’ll refer to this process as extending our formulas to a Henkin Theory. Before
tackling the problem of dealing with the interpretation of $=$ in our structure, we’ll show that this set of formulas can be extended to a negation complete set.

The following lemma will simplify things a bit by allowing us to only work with sentences.

**Proposition 14.7.** If $A$ is any formula, let $A^c$ (called the *closure* of $A$) be $A$ with universal quantifiers binding any free variables in $A$. Then $\vdash \Gamma \Rightarrow A$ iff $\vdash \Gamma^c \Rightarrow A^c$.

*Proof.* Suppose that $\vdash \Gamma \Rightarrow A$. In the proof of $\Gamma \Rightarrow A$, simultaneously replace all free variables in $A$ with the eigenvariables $y_1, \ldots, y_n$. This yields a proof of $\Gamma \Rightarrow A(y_1, \ldots, y_n)$. $n$ applications of $\forall$ gives $\Gamma \Rightarrow A^c$. $\forall$ can be used on each of the formulas in $\Gamma$ to yield a proof of $\Gamma^c \Rightarrow A^c$. Suppose now that $\vdash \Gamma^c \Rightarrow A^c$. By repeated applications of the inversion lemma, $\Gamma \Rightarrow A(y_1, \ldots, y_n)$ is derivable. $\Gamma^c$ can be reduced to $\Gamma$ similarly. By simultaneously replacing the eigenvariables $y_1, \ldots, y_n$, with the free variables in $A$, we obtain a proof of $\Gamma \Rightarrow A$. $\dashv$

Until the final proof of the completeness theorem for $G + Cut$ below, we’ll treat $\Theta \cup \Gamma \cup \neg \Delta \cup H$ as $(\Theta \cup \Gamma \cup \neg \Delta \cup H)^c$.

**Lemma 14.8.** *(Lindenbaum’s Lemma)* If $\Theta \cup \Gamma \cup \neg \Delta \cup H$ is a consistent set of formulas, then it can be extended to a consistent negation complete set $\Phi^*$.  

*Proof.* Let $A_1, A_2, \ldots$ be an enumeration of the sentences of $L_n$ and $H$, and let $\Phi_0 = \Theta \cup \Gamma \cup \neg \Delta \cup H$. Then if $A_{n+1}$ or $\neg A_{n+1}$ is $\Phi_n$-provable, let $\Phi_{n+1} = \Phi_n$. Otherwise, let $\Phi_{n+1} = \Phi_n \cup \{A_{n+1}\}$. Let $\Phi^* = \bigcup_i \Phi_i$.

$\Phi^*$ is consistent and negation complete by the same reasoning we used in the proof of proposition 7.7. (It is negation complete by construction, and is consistent because each of the $\Phi_n$ are consistent and proofs can only be finitely many steps long.) $\dashv$

We’ve now prepared our set of sentences enough to have an effective means for constructing a model of $\Theta \cup \Gamma \cup \neg \Delta$ through building one for for $\Phi^*$. We must finally deal with the problem noted above about identity, however. Again, since the objects making up our structure are the closed terms of $L_n$ and we may need to make the

33
a sentence like $a = b$ true in our model, we have to find a way to identify these distinct closed terms in our structure.

It is perhaps easiest to see how this kind of identification works by looking at a very similar construction in a more familiar setting, so we’ll start there.

**Definition 14.9.** If $n$ is a natural number and $a$ is some integer, $n$ divides $a$ if there is some $k$ such that $a = n \cdot k$. “$n$ divides $a$” is often written as $n|a$.

Given some natural number $n$, we can define the following relation for any pair of integers $a$ and $b$

$$a \sim b \iff n|(a - b).$$

**Proposition 14.10.** $\sim$ is an equivalence relation; that is, it is reflexive ($a \sim a$), symmetric ($a \sim b$ implies $b \sim a$), and transitive (if $a \sim b$ and $b \sim c$, then $a \sim c$).

**Proof.** Reflexive: $(a - a) = 0 = n \cdot 0$, so $a \sim a$.

Symmetric: If $a \sim b$, then for some $k$, $(a - b) = n \cdot k$. But then, $n \cdot k = -(a - b) = (b - a)$. So, $n|(b - a)$ and $b \sim a$.

Transitive: If $a \sim b$ and $b \sim c$, then $(a - b) = n \cdot k_1$ and $(b - c) = n \cdot k_2$ for integers $k_1$ and $k_2$. $(a - c) = (a - b + b - c) = (a - b) + (b - c) = n \cdot k_1 + n \cdot k_2 = n \cdot (k_1 + k_2)$.

So, $n|(a - c)$ and $a \sim c$.

When the relation $\sim$ holds between $a$ and $b$, we usually write $a \equiv b \pmod{n}$. The idea is that dividing either $a$ or $b$ by $n$ leaves the same remainder, so once you “mod out” by $n$, the numbers are essentially equivalent.

If we let $[a] = \{b : a \equiv b \pmod{n}\}$, in a sense, we can reduce all the integers to $n$ distinct equivalence classes. For example, consider the case where $n$ is 3.

$$[0] = [3] = [6] = \ldots = \{\ldots, -6, -3, 0, 3, 6, 9 \ldots\}$$

$$[1] = [4] = [7] = \ldots = \{\ldots, -5, -2, 1, 4, 7, 10 \ldots\}$$

$$[2] = [5] = [8] = \ldots = \{\ldots, -4, -1, 2, 5, 8, 11 \ldots\}$$

34
Every integer belongs to one and only one of these equivalence classes. This is sometimes written as $\mathbb{Z}/ \sim = \{[0], [1], [2]\}$. When we take the integers and mod out by the relation that tells us which things are identical, we’re left with just these three distinct classes.

We can further define operations of addition and multiplication on $\mathbb{Z}/ \sim$.

$$[a] + [b] = [a + b]$$

$$[a] \cdot [b] = [a \cdot b]$$

So, for example, $[1] + [2] = [1 + 2] = [3] = [0]$, and $[2] \cdot [2] = [2 \cdot 2] = [4] = [1]$.

Since $[1] = [4] = [16] = [28] = \ldots$ it’s important to make sure that the results of additions and multiplications using these definitions don’t depend on which representative of the equivalence class one choses to work with. This is called showing these operations to be well-defined. We can prove that the definitions above are well-defined as follows.

**Proposition 14.11.** Suppose $a \sim a'$ and $b \sim b'$. Then $[a] + [b] = [a'] + [b']$. In other words, addition yields the same result regardless of which representatives of the equivalence classes are chosen.

*Proof.* $a \sim a'$ means that $a - a' = n \cdot k_1$ for some $k_1$, so $a = a' + n \cdot k_1$. Similarly, $b \sim b'$ implies $b = b' + n \cdot k_2$ for some $k_2$. But then, $a + b = a' + n \cdot k_1 + b' + n \cdot k_2 = a' + b' + n(k_1 + k_2)$. So, $(a + b) - (a' + b') = n(k_1 + k_2)$. That is, $[a+b] = [a'+b']$. Therefore, $[a] + [b] = [a+b] = [a'+b'] = [a'] + [b']$. ⊢

A very similar proof shows that the definition of $\cdot$ on our equivalence classes is also well-defined. That is, if we suppose that $a \sim a'$ and $b \sim b'$, then we can show that $[a] \cdot [b] = [a'] \cdot [b']$.

For our purposes, we’ll be interested in defining the following relation and showing it to be an equivalence relation. Let $a$ and $b$ be closed terms of $\text{L}_{\text{rt}}$, then we say

$$a \sim b \text{ iff } \Rightarrow a = b \text{ is } \Phi^*\text{-provable.}$$
**Proposition 14.12.** \( \sim \) is an equivalence relation.

*Proof.* \( \sim \) is reflexive: \( \Rightarrow a = a \) is an identity axiom. So, \( a \sim a \).

\( \sim \) is symmetric: Suppose \( a \sim b \). Then \( \Rightarrow a = b \) has a \( \Phi^* \)-proof. \( a = b, b = b \Rightarrow b = a \) is an identity axiom, so by two uses of the cut rule we have a \( \Phi^* \)-proof of \( \Rightarrow b = a \).

\( \sim \) is transitive: Suppose \( a \sim b \) and \( b \sim c \). Then \( \Rightarrow a = b \) and \( \Rightarrow b = c \) are \( \Phi^* \)-provable. \( a = b, b = c \Rightarrow a = c \) is an identity axiom. So, by two uses of the cut rule again we have a \( \Phi^* \)-proof of \( \Rightarrow a = c \).

We can now define a structure for \( L_H \) that will allow us to show that \( \Phi^* \) has a model. Let \( T_c \) be the set of closed terms of \( L_H \). Then \( M \) is defined as follows.

1. \( U = T_c / \sim \)

2. For each constant symbol \( c \) let \( c^M = [c] \)

3. For \( n \)-ary \( f \), \( f(a_1, \ldots, a_n)^M = f^M([a_1], \ldots, [a_n]) = [f(a_1, \ldots, a_n)] \)

4. For \( n \)-ary \( P \), let \( P^M = \{([a_1], \ldots, [a_n]) : \Rightarrow P(a_1, \ldots, a_n) \text{ is } \Phi^* \text{-provable}\} \).

Just as was required in the case of defining the operations of addition and multiplication on equivalence classes of integers in the modular arithmetic case, we must make sure that the definitions of the interpretations of \( f \) and \( P \) in this structure are well-defined. That is, we must prove the following.

**Proposition 14.13.** Suppose that \( a_i \sim b_i \) for \( i \in \{1, \ldots, n\} \). Then

(i) \( f^M([a_1], \ldots, [a_n]) = f^M([b_1], \ldots, [b_n]) \)

(ii) \( ([a_1], \ldots, [a_n]) \in P^M \text{ iff } ([b_1], \ldots, [b_n]) \in P^M \).

*Proof.* \( a_i \sim b_i \) means that we can \( \Phi^* \)-prove \( \Rightarrow a_i = b_i \) for all \( i \). The sequent \( a_1 = b_1, \ldots, a_n = b_n \Rightarrow f(a_1, \ldots, a_n) = f(b_1, \ldots, b_n) \) is an identity axiom. So, by repeated applications of the cut rule, we have a \( \Phi^* \)-proof of \( \Rightarrow f(a_1, \ldots, a_n) = f(b_1, \ldots, b_n) \). Therefore, \( [f(a_1, \ldots, a_n)] = [f(b_1, \ldots, b_n)] \).

\(^{14}\) So, the entities in \( U \) are equivalence classes of closed terms that are provably equivalent. We’ll write, as is standard, \([a]\) for the equivalence class containing \( a \).
\( a_1 = b_1, \ldots, a_n = b_n, P(a_1, \ldots, a_n) \Rightarrow (b_1, \ldots, b_n) \) is also an identity axiom.

By repeated cuts with the \( \Phi^* \)-provable sequents \( \Rightarrow a_i = b_i \) and \( \Rightarrow P(a_1, \ldots, a_n) \), the sequent \( \Rightarrow P(b_1, \ldots, b_n) \) is provable. Therefore, \( ([b_1], \ldots, [b_n]) \in P^M \). The other direction is symmetric.

This proposition tells us that the pieces of our structure are well-defined. We can now show that \( M \) is a model for \( \Phi^* \).

**Theorem 14.14.** Suppose \( \Phi^* \) is a consistent negation complete Henkin Theory. Then \( M \models A \) iff \( A \) is \( \Phi^* \) provable.

**Proof.** Suppose \( A \) is atomic, then it is either of the form (i) \( P(a_1, \ldots, a_n) \) or (ii) \( a_i = a_j \). If \( A \equiv P(a_1, \ldots, a_n) \) and \( M \models A \). Then by the definition of our structure, \( A \) is \( \Phi^* \)-provable. If \( A \) is \( \Phi^* \)-provable, then again by definition, \( M \models A \). If \( A \equiv a_i = a_j \) and \( M \models a_i = a_j \), then \( a_i^M = a_j^M \). That is, \( [a_i] = [a_j] \).

By definition of \( \sim \), this implies \( \Rightarrow a_i = a_j \) is \( \Phi^* \)-provable. Suppose now that \( a_i = a_j \) is \( \Phi^* \)-provable, then by the definition of \( M \), \( [a_i] = [a_j] \) and, therefore, \( M \models a_i = a_j \).

The argument for each of the Boolean connectives is identical to the ones given in the proof of the corresponding lemma in section 7.

Suppose \( A \equiv \exists x B(x) \) and \( M \models A \). Then there is some object \([c]\) such that if we take \( x \) to be a name for \([c]\), \( B(x) \) is satisfied. Each object in our structure is an equivalence class of closed terms, and each closed term names its own equivalence class. Therefore everything in \( U \) has a name. Let \( c \) be the name of the object \([c]\) that satisfies \( B(x) \). Then \( M \models B(c) \). By the IH, \( B(c) \) is \( \Phi^* \)-provable. By \( \exists \), \( \exists x B(x) \) has a \( \Phi^* \)-proof. Suppose \( \exists x B(x) \) is \( \Phi^* \)-provable. By its construction, \( \Phi^* \) contains a witnessing axiom of the form \( \exists x B(x) \rightarrow B(d) \) for a witnessing constant \( d \). By propositional logic, \( B(d) \) is provable. So, by the IH, \( M \models B(d) \). Therefore, there is some object \( d^M \) in \( U \) such that if we take \( x \) to be a name for \( d^M \), \( B(x) \) is satisfied. Therefore, \( M \models \exists x B(x) \). \( \square \)

**Theorem 14.15.** (Extended Completeness Theorem) \( \Theta \models \Gamma \Rightarrow \Delta \) implies \( \Gamma \Rightarrow \Delta \) is \( \Theta \)-provable.
Proof. Suppose $\Gamma \Rightarrow \Delta$ is not $\Theta$-provable. By proposition 14.7, $\Gamma \Rightarrow \Delta$ is $\Theta$-provable iff $\Gamma^c \Rightarrow \Delta^c$ is $\Theta^c$-provable. Therefore, $(\Theta \cup \Gamma \cup \neg \Delta)^c$ is consistent by lemma 14.1. By adding witnessing constants and axioms and using the Elimination Theorem, we can conservatively extend this set to a Henkin Theory. By Lindenbaum’s Lemma, we can again extend this set of formulas to a consistent negation complete set $\Phi^*$. By theorem 14.14, we can find a structure $M$ that is a model for $\Phi^*$. Therefore, $\Theta^c \not\models \Gamma^c \Rightarrow \Delta^c$. Therefore, $\Theta \not\models \Gamma \Rightarrow \Delta$.

15 The Compactness Theorem

An important, but almost immediate, corollary of the completeness theorem is the first-order version of the compactness theorem.\(^{15}\)

**Theorem 15.1.** (Compactness Theorem) If $\Gamma$ is a set of sentences in a first-order language $L$ and every finite subset of $\Gamma$ has a model, then $\Gamma$ has a model as well.

**Proof.** Suppose $\Gamma$ has no models. Our proof of the completeness theorem showed how to construct a model of any consistent set of sentences, so $\Gamma$ must not be consistent. Therefore, $\Gamma' \Rightarrow$ for some finite $\Gamma' \subseteq \Gamma$. By the soundness theorem, $\Gamma'$ must not have a model.

The following are some standard examples of applications of the compactness theorem. First, we’ll show that there are nonstandard models of Peano Arithmetic (PA); that is, models that makes all the axioms of PA true, but that are not isomorphic to the standard structure $N$ of the natural numbers. The language of PA was introduced in section 11. The axioms that characterize the theory of PA are as follows. As is standard, I’ll often write $n \times m$ as $nm$.

**Definition 15.2.** These axioms aren’t the exact ones Peano originally formulated (he actually treated $+$ and $\times$ as defined symbols), but these axioms have become (approximately) the standard form.

1. $\forall x (sx \neq 0)$

\(^{15}\) This theorem has purely model-theoretic and topological proofs as well.
0 is not the successor of any number.

2. \( \forall x \forall y (sx = sy \rightarrow x = y) \)

If \( x \) and \( y \) have the same successor, \( x \) and \( y \) are identical.

3. \( \forall x \ (x + 0 = x) \)

4. \( \forall x \forall y (x + sy = s(x + y)) \)

These axioms force + to behave like ordinary addition.

5. \( \forall x (x \times 0 = 0) \)

6. \( \forall x \forall y (x \times sy = (xy) + x) \)

These axioms force \( \times \) to behave like ordinary multiplication.

7. \( (P(0) \land \forall x (P(x) \rightarrow P(sx))) \rightarrow \forall x P(x) \)

This axiom schema is the formal version of induction in PA.

We don’t need to make use of these axioms at the moment, but they will be examined closely when we move on to study Gödel’s first incompleteness theorem.

**Proposition 15.3.** PA has a nonstandard model.

*Proof.* First define, \( x < y \) to be an abbreviation of \( \exists z (x + sz = y) \). Now, let \( L_{PA^+} \) be the language generated by the alphabet of \( L_{PA} \) with a new constant symbol \( \omega \) added. Further, add to the axioms of PA countably many new axioms of the form \( n < \omega \)—one for each \( n \). Call this set of axioms \( PA^+ \). Any finite \( T \subseteq PA^+ \) is modeled by the structure \( N \) where we let \( \omega^N \) be a number greater than any \( m \) that is named in one of the finitely many axioms of the form \( n < \omega \) in \( T \). By compactness, \( PA^+ \) has a model. There must be an object \( \omega^N \) in this model that is greater than anything named by one of our expressions \( n \). Since the axioms of PA are a subset of the axioms of \( PA^+ \), PA is also modeled by this structure. But since there is no natural number greater than every natural number, this structure must not be (isomorphic to) the natural numbers; i.e., the model is nonstandard. \( \neg \)
Another common application of the compactness theorem is to show that there is no first-order sentence that is true in all and only finite structures; that is, the finite structures are not first-order axiomatizable.

For any \( n \), we can find a sentence that is true in all and only structures with at least \( n \) elements. So, for example, \( \exists v_1 \exists v_2 (v_1 \neq v_2) \) is only true in a structure with 2 elements; and if a structure has 2 elements, this sentence is true in that structure. Our next proposition shows that we can’t come up with a similar kind of sentence to characterize finite structures.

**Proposition 15.4.** There is no set of first-order formulas that characterizes finite structures.

*Proof.* Suppose that the set of formulas \( \Gamma \) is true in all and only finite structures. Let \( \#_n \) be the sentence like the one indicated above that is true in all and only structures with at least \( n \) elements. And let \( \Gamma^+ \) be the set \( \Gamma \cup \bigcup_i \#_i \). We can easily show that, for any finite \( T \subseteq \Gamma^+ \), \( T \) has a model. This is because \( T \) is made up of sentences from \( \Gamma \) along with a finite number of axioms of the form \( \#_n \). If \( m \) is the greatest number such that \( \#_m \) is in \( T \), then we must find a structure with at least \( m \) elements to model \( T \). But, by assumption, \( \Gamma \) has a model in a structure with any finite number of elements. Therefore, it has, in particular, a model in a structure with \( m \) elements. So, since any finite subset of \( \Gamma^+ \) has a model, \( \Gamma^+ \) has a model as well by compactness. But, by construction, \( \Gamma^+ \) can only have a model in an infinite structure. \( \Gamma \subseteq \Gamma^+ \), so \( \Gamma \) also has a model in an infinite structure. \( \dashv \)

Another important application of the compactness theorem is to prove one direction of the Löwenheim-Skolem Theorem. This theorem shows, as it’s sometimes put, that first-order logic can’t distinguish between infinite cardinalities.

16 The Löwenheim-Skolem Theorem

The Löwenheim-Skolem Theorem is a theorem about the *sizes* of models we can expect to find for sets of first-order formulas. The theorem is generally presented as being composed of a theorem for each of two “directions,” an upward and a downward version. The upward version says that if some set of formulas \( \Gamma \) has a
model of infinite cardinality \( \kappa \), i.e., with a domain \( U \) such that \( |U| = \kappa \), then it also has models of each cardinality greater than \( \kappa \). The downward version says that if \( \Gamma \) has a model at all, it has a countable model.

**Definition 16.1.** \( M \) is an elementary substructure of \( N \) if \( M \) and \( N \) are both structures for a language \( L \) such that any first-order formula \( P(\overline{x}) \) and variable assignment \( \overline{a} \), where the \( a_i \) are in \( M \), \( M \models P(\overline{a}) \) if \( N \models P(\overline{a}) \). If \( M \) is an elementary substructure of \( N \), we write \( M \preceq N \). In such a case, we also say that \( N \) is an elementary extension of \( M \).

**Proposition 16.2.** *(Tarski-Vaught Criterion)* If \( M \) and \( N \) are structures for \( L \) and \( M \) is a substructure of \( N \), then \( M \preceq N \) iff for any formula \( A(x, \overline{y}) \) and \( \overline{b} \in U^n_M \), if \( N \models \exists x A(x, \overline{b}) \), there is an \( a \in U_M \) such that \( N \models A(a, \overline{b}) \).

*Proof.* If \( M \preceq N \), \( N \models \exists x A(x, \overline{b}) \) iff \( M \models \exists x A(x, \overline{b}) \) iff there is some \( a \in U_M \) such that \( M \models A(a, \overline{b}) \) iff \( N \models A(a, \overline{b}) \). The other direction is proved by induction on the complexity of \( A(x, \overline{y}) \).

For \( A \) atomic, \( M \models A(a, \overline{b}) \) iff \( N \models A(a, \overline{b}) \) since \( M \subseteq N \). If \( A \equiv \neg B \), \( M \models \neg B(a, \overline{b}) \) iff \( M \not\models B(a, \overline{b}) \) iff \( N \not\models B(a, \overline{b}) \) iff \( N \models \neg B(a, \overline{b}) \). \( M \models A(a, \overline{b}) \) iff \( N \models A(a, \overline{b}) \) is proved similarly for other quantifier-free \( A \).

If \( M \models \exists x B(x, \overline{b}) \), then there is some \( a \in U_M \) such that \( M \models B(a, \overline{b}) \). By the IH, \( N \models B(a, \overline{b}) \). So, \( N \models \exists x B(x, \overline{b}) \). If \( N \models \exists x B(x, \overline{b}) \), by hypothesis, there is some \( a \in U_M \) such that \( N \models B(a, \overline{b}) \). Since \( M \subseteq N \), \( M \models B(a, \overline{b}) \). Therefore, \( M \models \exists x B(x, \overline{b}) \). 

**Definition 16.3.** If \( A(x, \overline{y}) \in L \), the function symbol \( f^n_A \), where \( n \) is the length of \( \overline{y} \), is called a Skolem function for \( A(x, \overline{y}) \). The formula

\[
\forall \overline{y} (\exists x A(x, \overline{y}) \rightarrow A(f^n_A(\overline{y}), \overline{y}))
\]

is called the Skolem axiom for \( f^n_A \).

If \( M \models \exists x A(x, \overline{b}) \) for any \( \overline{b} \), we can always interpret \( f^n_A \) in such a way that \( M \models A(f^n_A(\overline{b}), \overline{b}) \). Just define \( f^n_A(\overline{b})^M \) to be one of the things in \( U \) that satisfies \( A(x, \overline{b}) \).
If there is no such object, we can define $f^n_A(b) = b_1$. (Any other object would do just as well in this case, however.) If we interpret $f^n_A$ in this way, the Skolem axiom for $f^n_A$ will be modelled by any structure for a language containing $f^n_A$.

**Lemma 16.4. (Skolemization Lemma)** If $M$ is a structure for $L$ that models a set of formulas $\Gamma$, we can define a structure $M^S$ for a language $L^S$ that contains Skolem functions for all formulas of the form $A(x, \overline{y})$ such that $M^S$ is a model of $\Gamma \cup S$, where $S$ is the set of Skolem axioms for Skolem functions in $L^S$.

**Proof.** We construct $L^S$ and $S$ as a union of smaller languages and sets of axioms, expanding $M$ along the way. Let $L_0 = L$, $S_0 = \emptyset$, and $M_0 = M$. Let $L_{n+1}$ be the language obtained by adding Skolem functions for each formula of the form $A(x, \overline{y})$ to the alphabet of $L_n$. Let $S_{n+1}$ be $S_n$ along with all the Skolem axioms for Skolem functions added to $L_{n+1}$. At each stage of this construction, we expand $M_n$ to $M_{n+1}$ a structure for $L_{n+1}$ by interpreting the Skolem functions as explained above. Clearly, $M_{n+1}$ is a model of $\Gamma \cup S_{n+1}$. Let $L^S$, $S$, and $M^S$ be the union of these languages, sets, and structures respectively.

**Theorem 16.5. (Downward Löwenheim-Skolem Theorem)** Let $\Gamma$ be a set of formulas in a countable first-order language $L$. Then if $\Gamma$ has a model, $\Gamma$ has a model in an elementary substructure with a countable domain.

**Proof.** Suppose that $M$ is a model of $\Gamma$. Expand $M$ to $M^S$ in the language $L^S$, and let $X$ be a countable subset of $M^S$’s domain. $X$ can be expanded to a substructure of $M^S$ by adding in constants to serve as denotations of the constant symbols in $L^S$ if necessary, and by adding in objects to form a set closed under the functions of $M^S$. Since $L$ and $L^S$ are countable, this expansion of $X$ to a substructure $X'$ keeps $X'$ countable.

Suppose $\overline{b} \in X'$ and $M^S \models \exists x A(x, \overline{b})$, then since $M^S$ is a model of $S$, there is some $f^n_A(\overline{b})^{M^S}$ in $U_{M^S}$ that satisfies $A(x, \overline{b})$. This object is also in the domain of $X'$, since $X'$ is closed under the functions of $M^S$ by construction. Therefore, $X' \preceq M^S$ by the Tarski-Vaught criterion. If we call $X'$ with all the added Skolem functions forgotten $N$, forgetting about these functions in $M^S$ allows us to conclude $N \preceq M$.
Skolem himself thought it paradoxical that the theory of an uncountable structure, say, the real numbers could have countable models. This surprising situation is known as Skolem’s Paradox.\textsuperscript{16}

**Definition 16.6.** If $M$ and $N$ are structures for $L$ and $\varphi$ is an injective function from $U_M$ to $U_N$ with the following properties, $\varphi$ is called an *embedding* of $M$ into $N$.

1. for each constant symbol $c$, $\varphi(c^M) = c^N$
2. for each $n$-ary $P$ and $\bar{a} \in U^n_M$, $\bar{a} \in P^M$ iff $\varphi(\bar{a}) \in P^N$
3. for each $n$-ary $f$ and $\bar{a} \in U^n_M$, $\varphi(f^M(\bar{a})) = f^N(\varphi(\bar{a}))$

If $\varphi$ satisfies the further condition below, it is said to be an *elementary* embedding.

4. for all $A(\bar{x})$ and $\bar{a} \in U^n_M$, $M \models A(\bar{a})$ iff $N \models A(\varphi(\bar{a}))$

**Definition 16.7.** If $M$ is a structure for $L$, we let $L_M$ be the language obtained by extending $L$’s alphabet by a constant symbol for each element of $U_M$.

**Definition 16.8.** The following set of closed-formulas is called the *elementary diagram* of a structure $M$.

$$\text{eldiag}(M) = \{A(c_1, \ldots, c_n) : A \text{ is prime and } M \models A(c_1, \ldots, c_n)\}$$

**Lemma 16.9.** (*Elementary Diagram Lemma*) If $N$ is a structure for $L_M$ such that $N \models \text{eldiag}(M)$, then there is an elementary embedding of $M$ into $N$.

*Proof.* Define $\varphi : U_M \rightarrow U_N$ as follows. Every $a \in U_M$ has a name $\bar{a}$ in $L_M$. Let $\varphi(a) = a^N$. If $a \neq b$, $a \neq b$ is in $\text{eldiag}(M)$. Therefore, $N \models a \neq b$, which implies $a^N \neq b^N$; that is, $\varphi$ is injective.

If $\bar{a} \in P^M$, $P(\bar{a}) \in \text{eldiag}(M)$. Therefore, $N \models P(\bar{a})$ and $\varphi(\bar{a}) \in P^N$. If $\bar{a} \notin P^M$, $\neg P(\bar{a}) \in \text{eldiag}(M)$ and by similar reasoning, $\varphi(\bar{a}) \notin P^N$. If $f^M(\bar{a}) = b$, $f(\bar{a}) = b \in \text{eldiag}(M)$ and, therefore, $N \models f(\bar{a}) = b$. Therefore,

\textsuperscript{16} The article by Tim Bays in the Stanford Encyclopedia of Philosophy, which can be found at plato.stanford.edu/entries/paradox-skolem provides a thorough discussion of this "paradox."
\( f^N(\varphi(\overline{a})) = \varphi(b) \). So, \( \varphi \) is an embedding. The embedding can be shown to be elementary by induction on the complexity of formulas. The base case has already been established.

\textbf{Theorem 16.10. (Upward Löwenheim-Skolem Theorem)} Let \( \Gamma \) be a set of formulas in a countable first-order language \( L \). Then if \( \Gamma \) has a model with an infinite domain, then \( \Gamma \) has a model in elementary extensions of any greater cardinality.

\textit{Proof.} Suppose \( \Gamma \) has a model \( M \) such that \( |U| \) is at least countable. We need to show that if \( \kappa \) is a cardinal greater than \( \aleph_0 \), \( \Gamma \) has a model with a domain of cardinality \( \kappa \).

Expand the alphabet of \( L \) to \( L_M \). Let \( |I| = \kappa \) and for every \( i \in I \) add a constant symbol \( d_i \) to the alphabet of \( L_M \). Further, add to \( \Gamma \) the set \( \text{eldiag}(M) \) and the following set of axioms: \( \{d_i \neq d_j : i, j \in I \text{ and } i \neq j\} \). Call this new set \( \Gamma' \). Any model of \( \Gamma' \), must have cardinality \( \kappa \) since it contains axioms that say that each of the \( \kappa \)-many constant symbols we added to \( L \) are all not identical to each other in the structure.

For any finite \( \Delta \subseteq \Gamma' \), \( \Delta \) has a model in \( M \). This is because \( \text{eldiag}(M) \) is clearly satisfied by \( M \) and \( \Gamma \subseteq \Gamma' \) has a model in some infinite domain. There are only finitely many axioms of the form \( d_i \neq d_j \) in \( \Delta \). Therefore, we can assign an object to each of the \( d_i \) and \( d_j \) in these axioms in such a way that they name different objects in the structure—there are infinitely many objects to chose from.

By compactness, \( \Gamma' \) has a model \( N \). The domain of this model has cardinality \( \kappa \). \( \Gamma \) is a subset of \( \Gamma' \), so \( \Gamma \) also has a model with a domain of cardinality \( \kappa \). By the elementary diagram lemma, there is an elementary embedding of \( M \) into \( N \).

\textbf{17 Inversion and Cut Elimination}

The last three sections have dealt with the completeness of \( G + \text{Cut} \) and some consequences of this theorem in a model-theoretic spirit. We’ll return now to pursue a more concrete, proof-theoretic approach to completeness and some related subjects.
Definition 17.1. The multisets $\Gamma$ and $\Delta$ that appear in each of the proof rules of $G$ are called the context of the rule. The formula in the conclusion of each rule that is not part of the context is referred to as the principal formula.

Definition 17.2. The length of a branch in a derivation tree $D$ is the number of nodes in the branch minus 1. The height of $D$ is the maximum length of its branches.

Theorem 17.3. (Inversion Theorem) Let $\vdash_n \Gamma \Rightarrow \Delta$ mean that the derivation of $\Gamma \Rightarrow \Delta$ in $G$ has height at most $n$.

1. If $\vdash_n A \land B$, $\Gamma \Rightarrow \Delta$, then $\vdash_n A$, $B$, $\Gamma \Rightarrow \Delta$.
2. If $\vdash_n \Gamma \Rightarrow \Delta$, $A \land B$, then $\vdash_n \Gamma \Rightarrow \Delta$, $A$ and $\vdash_n \Gamma \Rightarrow \Delta$, $B$.
3. If $\vdash_n \Gamma \Rightarrow \Delta$, $A \lor B$, then $\vdash_n \Gamma \Rightarrow \Delta$, $A$, $B$.
4. If $\vdash_n A \lor B$, $\Gamma \Rightarrow \Delta$, then $\vdash_n \Gamma \Rightarrow \Delta$, $A$ and $\vdash_n \Gamma \Rightarrow \Delta$, $B$.
5. If $\vdash_n \Gamma \Rightarrow \Delta$, $A \rightarrow B$, then $\vdash_n A$, $\Gamma \Rightarrow \Delta$, $B$.
6. If $\vdash_n A \rightarrow B$, $\Gamma \Rightarrow \Delta$, then $\vdash_n \Gamma \Rightarrow \Delta$, $A$ and $\vdash_n B$, $\Gamma \Rightarrow \Delta$.
7. If $\vdash_n \Gamma \Rightarrow \Delta$, $\neg A$, then $\vdash_n A$, $\Gamma \Rightarrow \Delta$.
8. If $\vdash_n \neg A$, $\Gamma \Rightarrow \Delta$, then $\vdash_n \Gamma \Rightarrow \Delta$, $A$.
9. If $\vdash_n \Gamma \Rightarrow \Delta$, $\forall x A(x)$, then $\vdash_n \Gamma \Rightarrow \Delta$, $A(y)$ for any $y \notin \Gamma \cup \Delta$.
10. If $\vdash_n \exists x A(x)$, $\Gamma \Rightarrow \Delta$, then $\vdash_n A(y)$, $\Gamma \Rightarrow \Delta$ for any $y \notin \Gamma \cup \Delta$.

Proof. The proof is by induction on $n$. Each case is proved in essentially the same way. Consider, for example, (6). Suppose that there is a derivation $D$, with height $n + 1$, of $A \rightarrow B$, $\Gamma \Rightarrow \Delta$. Since $A \rightarrow B$ is not atomic, $D$ is not an axiom. If the derivation concludes with the sequent $A \rightarrow B$, $\Gamma \Rightarrow \Delta$ after the application of a rule for which $A \rightarrow B$ is not principal, $D$ ends with an inference of one of the following two forms.

$$r_1 \quad \frac{A \rightarrow B, \Gamma' \Rightarrow \Delta'}{A \rightarrow B, \Gamma \Rightarrow \Delta} \quad r_2 \quad \frac{A \rightarrow B, \Gamma' \Rightarrow \Delta', A \rightarrow B, \Gamma'' \Rightarrow \Delta''}{A \rightarrow B, \Gamma \Rightarrow \Delta}$$
In the second case, the induction hypothesis is that

(ii) \( n \in \emptyset \Rightarrow \emptyset \); \( A \) and

(iii) \( n \in B; \emptyset \Rightarrow \emptyset \) and

(iv) \( n \in C; \emptyset \Rightarrow \emptyset \); \( A \) and

Applying the inference \( r_2 \) to (i) and (iii) yields a proof of \( \emptyset \Rightarrow \emptyset \) of height \( n + 1 \). Applying \( r_2 \) to (ii) and (iv) gives

If \( A \) is principal in the final inference of \( D \), there are three possibilities. The final step is an application of \( L \Rightarrow \), \( LW \), or \( LC \). In the \( L \Rightarrow \) case, the rule can only apply if \( \emptyset \Rightarrow \emptyset \) and \( B; \emptyset \Rightarrow \emptyset \) have already been derived. In the \( LW \) case, we must have \( n \in \emptyset \Rightarrow \emptyset \) and our desired sequents can be derived by weakening. In the \( LC \) case, the induction hypothesis applied to \( A \Rightarrow B; A \Rightarrow B; \emptyset \Rightarrow \emptyset \) gives us \( n \in A \Rightarrow B, \emptyset \Rightarrow \emptyset \) and \( \emptyset \Rightarrow \emptyset \). Applying the \( \text{IH} \) again, we have \( n \in \emptyset \Rightarrow \emptyset \), \( A \) and \( \emptyset \Rightarrow \emptyset \). Whence, \( \emptyset \Rightarrow \emptyset \) by contraction.

Since we’ve already shown \( G^\emptyset + \text{Cut} \) to be complete by the Henkin construction, the following theorem implies that \( G \) itself is complete. That is, if \( \emptyset \) and \( \emptyset \) are finite multisets, \( \emptyset \mid \emptyset \) implies \( \emptyset \Rightarrow \emptyset \).

**Definition 17.4.** The *depth* of a formula \( A \) of \( L \) is defined as follows.

1. If \( A \) is atomic, \( \text{dp}(A) = 1 \).
2. If \( A \equiv \neg B \) or \( \exists x \ B \) or \( \forall x \ B \), \( \text{dp}(A) = 1 + \text{dp}(B) \).
3. If \( A \equiv B \lor C \) or \( B \land C \) or \( B \Rightarrow C \), \( \text{dp}(A) = 1 + \max \{ \text{dp}(B), \text{dp}(C) \} \).

The *depth of a cut* is the depth of the cut formula.

**Definition 17.5.** The *degree* of a derivation \( D \) is the maximum depth of the cuts it employs.

**Theorem 17.6.** (Cut Elimination) If \( \emptyset \Rightarrow \emptyset \) is provable in \( G + \text{Cut} \), it is provable in \( G \) without the cut rule.

**Proof.** Suppose \( D \) is a degree \( d \) proof of \( \emptyset \Rightarrow \emptyset \). We’ll show that there is a proof of \( \emptyset \Rightarrow \emptyset \) of degree \( < d \) by induction on the height of \( D \).

Suppose \( D \) has one of the following two forms, where neither \( r_i \) is a cut.\(^{17}\)

\(^{17}\) When a sequent is located directly below some variant of \( D \) (i.e., when there is no line separating them), that sequent is taken to be part of, and the final step in, the deduction \( D \).
Then by the IH, \( D_1 \) in the first case and \( D_1 \) together with \( D_2 \) in the second have degree less than \( d \), and applying \( r_i \) yields a \(< d\)-degree proof of \( \Gamma \Rightarrow \Delta \).

Suppose now that \( D \) is a degree \( d \) proof of has the form below; \( i.e. \), it ends with a cut.

\[
\begin{array}{ccc}
D_1 & D_2 \\
\Gamma \Rightarrow \Delta, C & C, \Gamma \Rightarrow \Delta
\end{array}
\]

By the cut-elimination lemma below, there is a \(< d\)-degree proof of \( \Gamma, \Gamma \Rightarrow \Delta \), which can be contracted to a \(< d\)-degree proof of \( \Gamma \Rightarrow \Delta \).

The degree of any proof of \( \Gamma \Rightarrow \Delta \) is finite, so repeatedly applying the above procedure eventually produces a proof of degree 0; that is, a cut-free proof.

**Lemma 17.7.** *(Cut-Elimination Lemma)* If \( dp(C) = d \) and

\[
\begin{array}{ccc}
D_1 & D_2 \\
\Gamma \Rightarrow \Delta, C^m & C^n, \Gamma' \Rightarrow \Delta'
\end{array}
\]

where \( D_1 \) and \( D_2 \) each have degree less than \( d \), then there is a \(< d\)-degree proof of \( \Gamma, \Gamma' \Rightarrow \Delta, \Delta', \) where \( C^m \) abbreviates a non-zero number of occurrences of \( C \).

**Proof.** The proof is by induction on the sum of the heights of \( D_1 \) and \( D_2 \), and there are many cases to consider. First, suppose that one of \( D_1 \) or \( D_2 \) is an axiom. Take \( D_1 \) for instance. Then \( \Gamma' \Rightarrow \Delta \) and \( D_1 \) is just \( C \Rightarrow C \). In this case, \( \Gamma' = \{C\} \) and \( \Delta \) is empty. The sequent we’re trying to find a degree zero proof for is, therefore, \( C, \Gamma' \Rightarrow \Delta' \). \( D_2 \) is a degree 0 derivation of \( C^n, \Gamma' \Rightarrow \Delta' \) by hypothesis. Contraction yields \( C, \Gamma' \Rightarrow \Delta' \). The case for \( D_2 \) is symmetric.

Suppose the final rule applied in \( D_1 \) is a structural rule. In each case, the argument for \( D_2 \) is symmetric. If the rule is a form of weakening and \( C \) is not principal, then \( D_1 \) ends in one of the following two ways.

\[
\begin{array}{ccc}
D_1' & D_1' \\
\Gamma' \Rightarrow \Delta, C^m & \Gamma \Rightarrow \Delta, C^m
\end{array}
\]
In the first case, the sum of the heights of $D_2$ and the derivation of $\Gamma' \Rightarrow \Delta, C^m$ is less than the sum of $D_1$ and $D_2$, so by the IH there is a proof of $\Gamma, \Gamma'' \Rightarrow \Delta, \Delta'$ of degree less than $d$. In the second case, the IH yields the $\alpha < d$-degree proof of $\Gamma, \Gamma' \Rightarrow \Delta, \Delta''$. Structural rules can then be applied to derive $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$ in each case by assumption. If $C$ is principal in a weakening inference that produces $\Gamma \Rightarrow \Delta, C^{m+1}$, then $\Gamma \Rightarrow \Delta, C^m$ is $\alpha$-degree-provable and the IH yields the desired proof of $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$.

If $C$ is not principal in a contraction inference ending $D_1$, the same argument as above yields a $\alpha < n$-degree proof of $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. If $C$ is principal, $D_1$ ends as follows.

\[
\frac{D_1'}{
\bar{r} \quad \frac{D_1'}{\Gamma \Rightarrow \Delta, C^{m+1}} \qquad \frac{D_1''}{\Gamma''' \Rightarrow \Delta'', C^m} \qquad \frac{\bar{r}}{r}}{\Gamma \Rightarrow \Delta, C^m}
\]

By the IH, $D_1'$ and $D_2$ yield a $\alpha < d$-degree proof of $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$.

Consider the cases where (i) $C$ is not principal in the final connective-inference of $D_1$ or (ii) $C$ not principal in the final inference of $D_2$. Suppose (i). There are two subcases to consider depending on whether the final rule used is a two- or one-premise rule. We’ll consider only the two-premise case. In that case, $D_1$ has the form below.

\[
\frac{D_1'}{r \quad \frac{D_1'}{\Gamma'' \Rightarrow \Delta'', C^m} \qquad \frac{D_1''}{\Gamma''' \Rightarrow \Delta'', C^m} \qquad \frac{\bar{r}}{r}}{\Gamma \Rightarrow \Delta, C^m}
\]

By the IH (applied to $D_1'$ and $D_2$ as well as to $D_1'$ and $D_1'^{18}$), $D_1'$ along with $D_2$ and $D_1''$ along with $D_2$ can be used to construct $\alpha < d$-degree proofs of $\Gamma, \Gamma'' \Rightarrow \Delta, \Delta''$ and $\Gamma, \Gamma''' \Rightarrow \Delta, \Delta'''$. $r$ can now be used on these sequents to derive the sequent $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. The case where $C$ is not principal in $D_2$ is symmetric.

Finally, suppose that $C$ is principal in the final inference of $D_1$ and $D_2$. This means that the final rules applied in $D_1$ and $D_2$ respectively are $R\land$ and $L\land$, or $R\lor$ and $L\lor$, etc. Consider first the case where $C$ is of the form $\neg A$. Then $D_1$ and $D_2$ are of the following forms.

\[18\] This “criss-crossing” pattern will repeat often in what follows.
By the IH, there are $< d$-degree proofs of $A;€;€$, $⇒;¬A$ and $¬A$. Therefore, $⇒;¬A$ has a $< d$-degree proof because it can be derived from these sequents by a $< d$-depth cut.

If $C$ is of the form $A \land B$, $D_1$ and $D_2$ are as below.

\[
\begin{array}{ccc}
D_1' & D_1'' & D_2' \\
A, \Gamma \Rightarrow \Delta, (A \land B)^m, A & \Gamma \Rightarrow \Delta, (A \land B)^m, B & A, (A \land B)^n, \Gamma' \Rightarrow \Delta' \\
\Gamma \Rightarrow \Delta, (A \land B)^{m+1} & \Gamma \Rightarrow \Delta, (A \land B)^{m+1} & (A \land B)^{n+1}, \Gamma' \Rightarrow \Delta'
\end{array}
\]

By the IH, we can find $< d$-degree proofs of the sequents $\Gamma, \Gamma' \Rightarrow \Delta, \Delta', A$ and $\Gamma, \Gamma' \Rightarrow \Delta, \Delta', B$ and $A, B, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. Two $< d$-depth cuts yields a $< d$-degree proof of $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. The case for $C \equiv A \lor B$ is symmetric.

If $C$ is $A \rightarrow B$, $D_1$ and $D_2$ have the following form.

\[
\begin{array}{ccc}
D_1' & D_1'' & D_2' \\
A, \Gamma \Rightarrow \Delta, (A \rightarrow B)^m, B & \Gamma \Rightarrow \Delta, (A \rightarrow B)^m, B & A, (A \rightarrow B)^n, \Gamma' \Rightarrow \Delta' \\
\Gamma \Rightarrow \Delta, (A \rightarrow B)^{m+1} & \Gamma \Rightarrow \Delta, (A \rightarrow B)^{m+1} & (A \rightarrow B)^{n+1}, \Gamma' \Rightarrow \Delta'
\end{array}
\]

By the IH, we can find $< d$-degree proofs of the sequents $A, \Gamma, \Gamma' \Rightarrow \Delta, \Delta', B$ and $\Gamma, \Gamma' \Rightarrow \Delta, \Delta', A$ and $B, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. Again, two $< d$-degree cuts yield $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$.

If $C$ is $\exists x A$, $D_1$ and $D_2$ are as below, where $y$ does not occur in $\Gamma' \cup \Delta'$.

\[
\begin{array}{ccc}
D_1' & D_1'' & D_2' \\
\Gamma \Rightarrow \Delta, (\exists x A(x))^m, A(t) & \Gamma \Rightarrow \Delta, (\exists x A(x))^m, A(t) & A(y), (\exists x A(x))^n, \Gamma' \Rightarrow \Delta' \\
\Gamma \Rightarrow \Delta, (\exists x A(x))^{m+1} & \Gamma \Rightarrow \Delta, (\exists x A(x))^{m+1} & (\exists x A(x))^{n+1}, \Gamma' \Rightarrow \Delta'
\end{array}
\]

By the IH, it is possible to find $< d$-degree proofs of $A(y), \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$ and $\Gamma, \Gamma' \Rightarrow \Delta, \Delta', A(t)$. Since $y$ does not occur in $\Gamma'$ or $\Delta'$, we can substitute $t$ for $y$ everywhere in the proof of the first sequent to arrive at a $< d$-degree proof of $A(t), \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. A $< d$-degree cut then yields $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. The case for $C \equiv \forall x A$ is similar.
18 Interpolation and Definability Theorems

Definition 18.1. If $\Gamma$ is a formula or set of formulas, let $\mathcal{L}(\Gamma)$ be the set of non-logical symbols and free variables with occurrences in $\Gamma$.

Theorem 18.2. (Craig Interpolation Theorem) Suppose that $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$ is valid. Then there is a formula $C$ such that $\mathcal{L}(C) \subseteq (\mathcal{L}(\Gamma + \Delta) \cap \mathcal{L}(\Gamma' + \Delta')) + \{\top\}$ and $\Gamma \Rightarrow \Delta, C$ and $C, \Gamma' \Rightarrow \Delta'$ are valid. A $C$ as in this theorem is called an interpolant.

Proof. By the completeness of $G$ and the cut elimination theorem, we may consider a cut-free derivation of $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. The proof of the theorem is by induction on the number $n$ of inferences in this derivation. If $n = 0$, the sequent proved is of the form $A \Rightarrow A$ for $A$ atomic. We must consider the following cases.

1. $\Gamma = \Delta = \{A\}$
2. $\Gamma' = \Delta' = \{A\}$
3. $\Gamma = \Delta' = \{A\}$
4. $\Gamma' = \Delta = \{A\}$

In the first, let $C$ be $\neg \top$; in the second, let $C$ be $\top$; in the third, let $C$ be $A$; in the fourth, let $C$ be $\neg A$.

If the final inference is a structural rule, the induction hypothesis immediately yields an interpolant. The other cases will be illustrated by the following two examples.

Suppose now that the final in our proof is $L \lor$. Let $\Theta = \Gamma + \Gamma' - \{A \lor B\}$. Then the proof ends as below.

$$
\frac{D \quad D'}{A, \Theta \Rightarrow \Delta, \Delta'} \quad \frac{D'}{B, \Theta \Rightarrow \Delta, \Delta'}
\frac{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}{\frac{A \lor B, \Theta \Rightarrow \Delta, \Delta'}{\frac{A. \Theta \Rightarrow \Delta, \Delta'}{A, \Theta \Rightarrow \Delta, \Delta'}}}
$$

By the IH, there are interpolants $C_1$ and $C_2$ such that each of (i) $A \Rightarrow \Delta, C_1$; (ii) $C_1, \Theta \Rightarrow \Delta'$; (iii) $B \Rightarrow \Delta, C_2$; and (iv) $C_2, \Theta \Rightarrow \Delta'$ is provable. Structural rules
followed by \( R \lor \) and \( L \lor \) applied to \((i)\) and \((iii)\) allow \( A \lor B, \Theta \Rightarrow \Delta, C_1 \lor C_2 \) to be proved. Similarly, \((ii)\) and \((iv)\) can be used to prove \( C_1 \lor C_2, A \lor B, \Theta \Rightarrow \Delta, \Delta' \). \( C_1 \lor C_2 \) may, therefore serve as the required interpolant.

Suppose the final inference is \( R \forall \). Let \( \Theta = \Delta + \Delta' - \{ \forall x A(x) \} \). Then the proof ends as below.

\[
\begin{array}{c}
D \\
\Gamma, \Gamma' \Rightarrow \Theta, A(y) \\
\hline
\Gamma, \Gamma' \Rightarrow \Delta, \Delta'
\end{array}
\]

By the IH, the following sequents are provable where \( C' \) is an interpolant: \((i)\) \( \Gamma \Rightarrow \Theta, C' \); \((ii)\) \( C', \Gamma' \Rightarrow A(y) \). Since \( y \) cannot occur in \( \Gamma \) or \( \Theta \), \( \forall y \) applied to \((i)\) gives a proof of \( \Gamma \Rightarrow \Theta, \forall x C'. \) \( \forall y \) applied to \((ii)\) gives \( \forall x C', \Gamma' \Rightarrow A(y) \), from which \( \forall x C', \Gamma' \Rightarrow \forall x A(x) \) can be derived. Therefore, \( \forall x C' \) is the required interpolant.

---

**Definition 18.3.** If \( P \) and \( P' \) are \( n \)-ary predicate symbols and \( \Gamma \) is a set of sentences with no occurrences of \( P' \) and \( \Gamma(P') \) is \( \Gamma \) with any occurrence of \( P \) replaced by \( P' \), then \( \Gamma \) explicitly defines \( P \) if there is a formula \( A \) such that

\[
\Gamma \vdash \forall \overline{x}(A(\overline{x}) \leftrightarrow P(\overline{x})).
\]

\( \Gamma \) implicitly defines \( P \) if

\[
\Gamma \cup \Gamma(P') \models \forall \overline{x}(P(\overline{x}) \leftrightarrow P'(\overline{x})).
\]

**Theorem 18.4.** (Beth Definability Theorem) \( \Gamma \) (as indicated above) implicitly defines \( P \) iff it explicitly defines \( P \).

**Proof.** From right-to-left is clear. Suppose now that \( \Gamma \) implicitly defines \( P \); that is,

\[
\Gamma \cup \Gamma(P') \models \forall \overline{x}(P(\overline{x}) \leftrightarrow P'(\overline{x})).
\]

By compactness, for finite subsets \( \Gamma' \) and \( \Gamma'(P') \),

\[
\Gamma' \cup \Gamma'(P') \models \forall \overline{x}(P(\overline{x}) \leftrightarrow P'(\overline{x})).
\]
This implies
\[ \bigwedge \Gamma' \land P(\bar{y}) \models \bigwedge \Gamma'(P') \rightarrow P'(\bar{y}). \]

By the Craig interpolation theorem, there is some interpolant \( A(\bar{y}) \) such that.
\[ \models \left( \bigwedge \Gamma' \land P(\bar{y}) \right) \rightarrow A(\bar{y}) \]
\[ \models A(\bar{y}) \rightarrow \left( \bigwedge \Gamma'(P') \rightarrow P'(\bar{y}) \right) \]

The first of these valid formulas along with the extended completeness theorem shows part of what is required for \( \Gamma \) to explicitly define \( P \) by \( A(\bar{y}) \). The second, along with the assumption that \( \Gamma \) implicitly defines \( P \) (and again by completeness), gives the other part.

19 Herbrand’s Theorem

If \( S \equiv \Gamma \Rightarrow \Delta \) is a sequent with \( \Gamma = \{A_1, \ldots, A_m\} \) and \( \Delta = \{B_1, \ldots, B_n\} \), we will write \( S \) as a multiset of signed formulas: \( \{-A_1, \ldots, -A_n, +B_1, \ldots, +B_m\} \).

Multiple sequents \( S, S', S'', \text{ etc.} \) can then be represented as the union these signed multisets. We’ll write this simply as \( S, S', S'', \text{ etc.} \).\(^{19}\)

**Definition 19.1.** The formula(s) in the premise(s) that are used to derive the principal formula in a sequent inference are said to be *active* in that inference.

**Definition 19.2.** Two inferences are said to be *adjacent* if they are separated by (zero or more) structural inferences.

**Definition 19.3.** A proof in \( G^= \) is said to comply with the *pure variable convention* if the sets of free and bound variables occurring in the deduction are disjoint and the eigenvariable in any quantifier inference only occurs in the subtree generated by the conclusion of that inference.

By renaming variables, any deduction in \( G^= \) can be transformed into a deduction that complies with the pure variable convention.

\(^{19}\) This notation is essentially the same as the one used in (Troelstra and Schwichtenberg 2000). The symbols + and − are chosen here because in \( \bigwedge A_i \rightarrow \bigvee B_i \), the \( A_i \) are said to occur negatively and the \( B_i \) positively.
Lemma 19.4. (Permutation Lemma) If \( r_1 \) and \( r_2 \) are adjacent inferences in a pure variable deduction such that \( r_1 \) is a quantifier inference, \( r_2 \) is not, and \( r_1 \) is above \( r_2 \), then if the principal formula in \( r_1 \) is not active in \( r_2 \), \( r_1 \) can be permuted below \( r_2 \).

Proof. Let \( QxA(x) \) be the principal formula and \( \Gamma \) be the sequents active in \( r_1 \), and let \( B \) be principal and \( \Delta \) active in \( r_2 \). \( \Theta \) will consist of the context. If one of these symbols is modified by a prime (’), this will indicate that it has been modified by the use of structural rules.

The quantifier rules are all one-premise rules, so if we can show how to permute these rules over a general (non-quantifier) one-premise and two-premise rule, the lemma will be proved. A one-premise quantifier rule can be permuted below a non-quantifier one-premise rule as follows, where the first inferences are transformed into the second.

\[
\frac{\Gamma, \Delta, \Theta}{\pm QxA(x), \Delta, \Theta} \quad \frac{\Gamma, \Delta, \Theta}{\Gamma, \Delta', \Theta'} \quad \frac{\Gamma, \pm B, \Theta'}{\pm QxA(x), \pm B, \Theta'}
\]

A two-premise non-quantifier rule can also be permuted below a quantifier inference as follows.

\[
\frac{\Gamma, \Delta_1, \Theta}{\pm QxA(x), \Delta_1, \Theta} \quad \frac{\Gamma, \Delta_1, \Theta}{\Gamma, \Delta_1', \Theta'} \quad \frac{\Gamma, \Delta_2, \Theta'}{\Gamma, \pm B, \Theta'} \quad \frac{\pm QxA(x), \pm B, \Theta'}{\pm QxA(x), \pm B, \Theta'}
\]

Example 19.5. \( \forall \) can be permuted below \( \land \). The inferences below on the left can be transformed into those on the right where the use of \( \land \) comes before \( \forall \).

\[
\frac{+A(y), -B, -C, \Theta}{+\forall xA(x), -B, -C, \Theta} \quad \frac{+A(y), -B, -C, \Theta}{+A(y), -B, -C, \Theta'} \quad \frac{+A(y), -B, -C, \Theta'}{+A(y), -(B \land C), \Theta'} \quad \frac{+\forall xA(x), -(B \land C), \Theta'}{+\forall xA(x), -(B \land C), \Theta'}
\]
Because the deduction on the left can be assumed to meet the pure variable convention, \( y \) only occurs above \( +A(y), -B, -C, \Theta \). This implies that \( y \) does not occur in \( \Theta' \). \( y \) also must not occur in \( -B \) or \( -C \) for \( \forall \) to be applicable in the left-hand deduction, and therefore it does not occur in \( -(B \land C) \) either.

**Theorem 19.6. (Herbrand’s Theorem)** Let \( B \) be a formula in prenex form; i.e., \( B \) is of the form \( Q_1x_1 \ldots Q_nx_nA(\overline{x}) \) with \( A \) quantifier-free. Then \( B \) is provable in \( G \) iff there is a propositionally-provable sequent \( \Rightarrow \Gamma \) made up of substitution instances of \( A(\overline{x}) \) such that \( \Rightarrow B \) can be derived from \( \Gamma \) using only structural and quantifier inferences.

**Proof.** From right-to-left the theorem is trivial. For the left-to-right direction, suppose that \( B \) is provable in \( G \). Then it has a cut-free proof. By the permutation lemma, this proof of \( B \) can be translated into one where all the propositional inferences come before all the quantifier inferences. Let \( \Rightarrow \Gamma \) be the sequent derived by propositional inferences before any quantifier inferences have been made.\(^{20}\) Since no further logical connectives can be introduced by the quantifier inferences, this sequent must consist of substitution instances of \( A(\overline{x}) \).

Since any formula of \( L \) is provably equivalent to a formula in prenex form, Herbrand’s theorem can be seen as a reduction of the question of a first-order formula’s provability to a question about propositional provability.

[(Herbrand 1931) consistency towards recursive functions and then \( \Sigma_1 \)-completeness of \( Q \) in next sections.]

## 20  Recursive Functions and the URM

(\textit{Cutland 1980})

**Theorem 20.1.** There is a total function of one variable that is not computable.

**Proof.**

\(^{20}\) This sequent—the last before only quantifier and structural inferences remaind—is often called the \textit{midsequent} of the deduction.
Theorem 20.2. (*s-m-n Theorem*)

Proof.

Theorem 20.3. Universal program

Proof.

21 Representing Functions and Relations in \( Q \)

[...]

22 Undecidability of Validity

Theorem 22.1. (*Undecidability Theorem*)

Proof.

Theorem 22.2. (*Rice’s Theorem*)

Proof.

23 Gödel’s First Incompleteness Theorem

[...]

24 Gentzen’s Transfinite Consistency Proof

[...]
25 The $\lambda$-Calculus

Since we want to consider terms that differ only by a renaming of bound variables as the same $\lambda$-term, we’ll first define a set of “pre-terms” and then mod out by an equivalence relation that identifies certain of these pre-terms.

**Definition 25.1.** Let $\Sigma = \{\lambda, , \, v_1, v_2, \ldots\}$. The pre-$\lambda$-terms are defined to be the smallest $X \subseteq \Sigma^*$ such that

1. the variables $v_1, v_2, \ldots$ are in $X$;
2. if $x$ is a variable and $M$ is in $X$, then $(\lambda x M)$ is in $X$; and
3. if $M$ and $N$ are in $X$, then so is $(MN)$.

The set of pre-terms will be denoted $\Lambda^{-}$.

**Definition 25.2.** The set of free variables in a pre-term $M$, denoted $FV(M)$, is defined inductively as follows.

1. if $M$ is a variable, $FV(M) = \{M\}$.
2. if $M$ and $N$ are pre-terms, $FV(MN) = FV(M) \cup FV(N)$.
3. if $M \equiv (\lambda x N)$ for some variable $x$ and pre-term $N$, $FV(M) = FV(N) - \{x\}$.

**Definition 25.3.** If $M$ and $N$ are pre-terms, the application of $M$ to $N$ is defined inductively as follows.

1. if $M$ is a variable, $MN \equiv N$.
2. 
3. 

**Definition 25.4.** The relation $=_{\alpha}$ of $\alpha$-conversion is the smallest transitive and reflexive relation on $\Lambda^{-}$ such that

1. if $M =_{\alpha} N$, then $MP =_{\alpha} NP$ and $PM =_{\alpha} PN$;
For any $N$, $P$, and $x$, result of substituting $N$ for free occurrences of $x$ in $P$ has the following inductive definition.

1. $x[x := N] \equiv N$
2. $y[x := N] \equiv y$ \quad \text{if } y \neq x
3. $(PQ)[x := N] \equiv (P[x := N])(Q[x := N])$
4. $(\lambda x. P)[x := N] \equiv \lambda x. P$
5. $(\lambda y. P)[x := N] \equiv \lambda y.(P[x := N])$ \quad \text{if } x \neq y \text{ and } x \notin \text{FV}(P)
6. $(\lambda y. P)[x := N] \equiv \lambda y.(P[x := N])$ \quad \text{if } x \neq y, x \in \text{FV}(P), \text{ and } y \notin \text{FV}(N)$
7. $(\lambda y. P)[x := N] \equiv \lambda z.((P[y := z])[x := N])$ \quad \text{if } x \neq y, x \in \text{FV}(P), y \in \text{FV}(N), z \notin \text{FV}(NP)$

**Definition 25.5.** A $\lambda$-term $P$ with $\text{FV}(P) = \emptyset$ is called a **combinator**.

**Definition 25.6.** Each of the following $\lambda$-terms is used commonly enough to warrant a special name.

1. $I \equiv \lambda x. x$
2. $K \equiv \lambda xy. x$
3. $S \equiv \lambda yz.xz(yz)$
4. $T \equiv K$
5. $F \equiv \lambda xy.y$

(4) and (5) are called **truth-values**.

**Definition 25.7.**

**Proposition 25.8.** is a fixed-point combinator.
Definition 25.9. The following closed $\lambda$-terms are called the standard numerals

A numeral system like the standard numerals is said to be adequate if it can be used to define all recursive functions on $\mathbb{N}$. We’ll introduce recursive functions, $\lambda$-definability, and show that the standard numerals are adequate presently.

26 Recursive Functions and $\lambda$-Definability

Theorem 26.1. Every recursive function is $\lambda$-definable.

Proof.

Theorem 26.2. Every $\lambda$-definable function is recursive.

Proof.

Theorem 26.3. Every partial recursive function is $\lambda$-definable.

Proof.

Theorem 26.4. Every $\lambda$-definable function is partial recursive.

Proof.

References


