

First Order Logic

The last chapter presented the general concept of a logic. In this chapter we consider a particular logic — first order predicate calculus, also known as simply first order logic. First order logic is by far the most widely studied logic. It was first formulated by Guttlob Frege in 1879. However, the fundamental semantic properties of first order logic were not really understood until the work of Skolem, Herbrand, and Godel in the late 1920s and early 1930s [Davis, 1983].

The semantic properties of first order logic contain an apparent paradox. On the one hand first order logic can be shown to be expressively weak — there are a great many common mathematical concepts, such as the concept of transitive closure, or of a finite set, that can not expressed in first order logic. On the other hand first order logic is often viewed as a universal language which can be used to represent arbitrary mathematical statements. Both of these statements are in some sense true. To avoid, or at least ameliorate, the confusion brought on by this apparent paradox a few words can be said about how both statements can be true simultaneously.

Although there are many concepts not expressible in first order logic, first order logic can be viewed as an adequate (Turing universal) programming language. Like first order logic, programming lanuages are expressively week. For example, no programming language can express the concept of halting, i.e., one can not write a predicate `Halts` which takes another program as input determines whether the input program always halts. So expressively weak languages like first order logic can still be adequate programming languages. Mathematicians have used first order logic as a programming language in which to encode all the known acceptable principles of mathematical inference. The result is axiomatic set theory. Any mathematical proof can, in principle, be expressed as a proof in first order set theory. In this sense axiomatic set theory is an adequate foundation for mathematics. However, it is known that these principles of inference are not complete, e.g., there are true statements about integers which can not be proven. This implies that there are true statements that can never be proven by mathematicians (unless new principles are adopted). It is also known that axiomatic set theory does not semantically express standard concepts, e.g., there are models of set theory in which there is an infinite integer. But set theory does encode all the principles of inference accepted by mathematicians.

This chapter, like preceding chapters, emphasizes semantics — the meaning of expressions in models. From a semantic point of view the most important aspect of first order logic is its expressive weakness — its inability to express common concepts. This chapter presents the syntax and semantics of first order logic plus some discussion of its proof theory (inference relations).

1 Syntax and Semantics

Like any logic, first order logic has propositions, models, and a way of assigning truth values to propositions in models. First order logic also has terms where each term has a well defined value in each model. The terms are compositional — the value of a term is determined by the values of its subterms. To define the propositions and terms of first order logic we assume the existence of certain symbols. We assume infinite sets of constants, variables, function symbols, proposition symbols, and predicate symbols. We also assume that each function and predicate symbol is associated with a positive integer, called the *arity* of the symbol, that specifies the number of arguments accepted by the function or predicate symbol. These symbols are used to generate the terms and propositions of first order logic according the following definition.

> **Definition:** An first order term is either a constant symbol, a variable, or an expression of the form $f(t_1, \cdots, t_n)$ where f is a function symbol of arity n and $t_1, \cdots, t_n$ are first order terms.
>
> An first order proposition is any one of the following.

- A proposition symbol.
- A predicate application of the form $P(t_1, \cdots, t_n)$ where P is a predicate symbol of arity n and $t_1, \cdots, t_n$ are first order terms.
- An equation of the form $t_1 = t_2$ where t_1 and t_2 are first order terms.
- Any Boolean combination of first order propositions.
- A proposition of the form $\forall x \Phi(x)$ (read as "for all x $\Phi(x)$") where x is a variable and $\Phi(x)$ is a first order proposition (that may or may not contain the variable x).

First order propositions are often defined to include formulas of the form $\exists x \Phi(x)$ (read "exists x $\Phi(x)$") in addition to formulas of the form $\forall x \Phi(x)$. We will treat the proposition $\exists x \Phi(x)$ as an abbreviation for $\neg \forall x \neg \Phi(x)$. Treating existential quantification as an abbreviation for propositions involving universal quantification simplifies the construction of a sound and complete inference system. In practical proof verification systems it is convenient to have separate inference rules for existential quantification.

A logic must have models as well as propositions. A model of first order logic is called a first order structure. A first order structure provides a meaning for each constant, variable, function symbol, proposition symbol, and predicate symbol.[1]

Definition: A first order structure contains the following.

- A set D called the domain of the structure (this can be any set whatsoever).

[1] In most treatments of first order logic, a first order structure does not provide meanings for variables. There are three motivations for the nonstandard treatment of variables given here. First, the treatment given here is formally simpler than the classical treatment. Second, the treatment given here allows first order logic to be viewed as a simple special case of the general definition of a logic — each (possibly open) formula has a well defined value in each model. Finally, this nonstandard treatment emphasizes the formal similarity between constants and variables — from a formal perspective there is no difference between constants and variables.

- An interpretation of each constant symbol as an element of D.
- An interpretation of each variable as an element of D.
- An interpretation of each function symbol f of arity n as a function that takes n elements of D and returns an element of D, i.e., as function from D^n to D.
- An interpretation of each proposition symbol as either **T** or **F**.
- An interpretation of each predicate symbol P of arity n as a predicate on n elements of D, i.e., as a function from D^n to the set $\{\mathbf{T},\ \mathbf{F}\}$.

For example, let $\mathcal{R}$ be a first order structure whose domain consists of the real numbers and which assigns the standard meanings to the constant symbols 0 and 1, the binary function symbols $+$ and $\times$, and the binary predicate symbol $<$. Let $\mathcal{N}$ be a first order structure whose domain consists of the integers and which assigns each of the above symbols their standard meaning over the integers. The first order structures $\mathcal{R}$ and $\mathcal{N}$ are quite different. For example, in $\mathcal{R}$ there exist domain elements that are strictly between zero and one while in $\mathcal{N}$ there are no such domain elements. We will see that the proposition $\exists x(<(0,\ x) \wedge <(x,\ 1))$ is true in $\mathcal{R}$ but false in $\mathcal{N}$.

We now defined terms, propositions, and models. All that remains is to define the semantic value of each term and proposition in each model. To define the semantic value of quantified propositions we need the following auxiliary definition.

> **Definition:** For any element d of the domain of a first order structure $\mathcal{M}$, and for any variable x we define $\mathcal{M}[x := d]$ to be the first order structure that is identical to $\mathcal{M}$ except that it interprets the variable x as the domain element d.

Now let $\mathcal{M}$ be a particular first order structure. The semantic value of first order terms and propositions is defined as follows.

Definition: For any first order term t, the value $\mathcal{V}(t,\ \mathcal{M})$ is an element of the semantic domain of $\mathcal{M}$. The domain element $\mathcal{V}(t,\ \mathcal{M})$ is defined as follows.

- For a constant symbol c we have that $\mathcal{V}(c,\ \mathcal{M})$ is the value that $\mathcal{M}$ assigns to the symbol c.
- For a variable x we have that $\mathcal{V}(x,\ \mathcal{M})$ is the value that $\mathcal{M}$ assigns to the variable x.
- For a term of the form $f(t_1, \cdots, t_n)$ we have that $\mathcal{V}(f(t_1, \cdots t_n), \mathcal{M})$ equals the value of the function that $\mathcal{M}$ assigns to f applied to the domain elements $\mathcal{V}(t_1,\ \mathcal{M}), \cdots, \mathcal{V}(t_n,\ \mathcal{M})$.

For any first order proposition Φ, the value $\mathcal{V}(\Phi,\ \mathcal{M})$ is either $\mathbf{T}$ or $\mathbf{F}$. The value $\mathcal{V}(\Phi,\ \mathcal{M})$ is defined as follows.

- For a proposition symbol Q we have that $\mathcal{V}(Q,\ \mathcal{M})$ is the truth value that $\mathcal{M}$ assigns to Q.
- For a proposition of the form $P(t_1, \cdots, t_n)$ we have that $\mathcal{V}(P(t_1, \cdots t_n), \mathcal{M})$ equals the value of the predicate that $\mathcal{M}$ assigns to P applied to the domain elements $\mathcal{V}(t_1,\ \mathcal{M}), \cdots, \mathcal{V}(t_n,\ \mathcal{M})$.
- Boolean combinations have their standard meaning, e.g., $\Phi \vee \Psi$ is true if Φ is true or Ψ is true.
- For propositions of the form $\forall x \Phi(x)$, we have $\mathcal{V}(\forall x \Phi(x)) = \mathbf{T}$ if and only if, for all elements d of the domain of $\mathcal{M}$, we have $\mathcal{V}(\Phi(x), \mathcal{M}[x := d]) = \mathbf{T}$.

We have now completely defined first order logic. We have specified a set of terms, a set of propositions, a set of models, and a way of assigning meaning to each term and proposition in each model.

2 First Order Inference

Consider a set Σ of first order propositions and a particular first order proposition Φ. We would like to determine whether or not $\Sigma \models \Phi$. The following is one of the basic results of mathematical logic and computer science.

> **Undecidability Theorem:** If $\mathcal{L}$ contains at least one predicate symbol and at least one function symbol of more than one argument, then the entailment relation $\models$ of the logic defined by $\mathcal{L}$ is undecidable, i.e., there can not exist any procedure which terminates on every input and which, given any Σ and Φ, determines whether or not $\Sigma \models \Phi$.

The proof of this theorem will not be given here. However, the basic idea behind this theorem is a simple reduction of the halting problem to the problem of determining first order entailment. For any particular computer program one can construct a finite set of "axioms" Σ and a particular proposition Φ such that $\Sigma \models \Phi$ if and only if the given program halts. Since no procedure can determine whether an arbitrary program halts, no procedure can determine questions of first order entailment.

Although first order entailment is not decidable, there exists a set of inference rules that is both sound and complete for first order entailment. Recall that a sequent proof system is a pair of a set of ordinary inference rules and a set of sequent rules. Also recall that a sequent system is *acceptable* if the inference relation generated by the ordinary rules is polynomial time decidable. It is possible to give a sequent proof system for first order logic whose ordinary rule set consists of the Boolean constraint propagation rules plus the rules for reasoning about equality. The acceptability of the sequent system described here for first order logic is given by the following theorem.

> **Theorem:** The inference relation $\vdash_{\mathrm{BCP} \cup C}$, generated by the union of the rule sets BCP and C, is decidable in $O(nlogn)$ time (assuming that hash table operations are unit time).

The proof of this theorem is similar to, although somewhat more elaborate than, the proof of that $\vdash_{\mathrm{BCP}}$ is linear time decidable. The proof is not given here.

To construct a sequent system that is complete for first order logic, the rule sets BCP and C must be extended with sequent rules. First, we include the sequent rule of case analysis discussed earlier.

Case Analysis

$$\frac{\Sigma \cup \{\Psi\} \vdash \Phi \qquad \Sigma \cup \{\neg\Psi\} \vdash \Phi}{\Sigma \vdash \Phi}$$

When combined with the rule set BCP that defines Boolean constraint propagation, the case analysis rule provides for complete Boolean inference. When combined with the rule set C for equality reasoning, the case analysis rule provides for complete reasoning about *quantifier-free* first order logic. In other words, if no quantifiers appear in Σ of Φ, then $\Sigma \models \Phi$ if and only if Φ is provable from Σ in the sequent system consisting of the ordinary rules of BCP and C plus the single case analysis sequent rule. To reason about quantifiers we add more sequent rules. The first is the rule of universal generalization given below. In this rule x must be a variable that does not appear in Σ, y is any variable, and $\Phi(y)$ is the result of replacing all free occurrences of x in $\Phi(x)$ by the variable y.[2]

Universal Generalization

$$\frac{\Sigma \vdash \Phi(x)}{\Sigma \vdash \forall y \Phi(y)}$$

The soundness of this rule follows from the semantics of first order logic and the (assumed) soundness of $\vdash$. Consider a model $\mathcal{M}$ of Σ. If x does not occur in Σ, then for any element d of the domain of $\mathcal{M}$, the $\mathcal{L}$-structure $\mathcal{M}[x := d]$ must also be a model of Σ. If $\Sigma \models \Phi(x)$, and x does not occur in Σ, then in any model of Σ, the proposition $\Phi(x)$ must be true for all values of x. So we must have $\Sigma \models \forall x \Phi(x)$. The formula $\forall x \Phi(x)$ is equivalent to any formula

[2] An occurrence of x in the proposition $\Phi(x)$ is called *free* if it is not inside a subproposition of the form $\forall x \Psi(x)$. For example, in the proposition $P(x) \rightarrow \forall x P(x)$ only the first occurrence of x is free.

of the form $\forall y \Phi(y)$. The above rule allows one to deduce $\Sigma \vdash \forall y \Phi(y)$ even when y occurs in Σ.

Our last rule says that if we know $\forall \Phi(x)$ then we can conclude $\Phi(t)$ for any term t. The proposition $\Phi(t)$ is the result of replacing all free occurrence of x in $\Phi(x)$ by the term t with appropriate renaming of bound variables in $\Phi(x)$ to avoid the capture of free variables in t.[3] The sequent rule of universal instantiation can be stated as follows.

Universal Instantiation $$\frac{\Sigma \vdash \forall x \Phi(x)}{\Sigma \vdash \Phi(t)}$$

It is possible to state this last inference rule as an ordinary rule rather than a sequent rule. However, if the rule of universal instantiation were added to the ordinary rule set, then the inference relation generated by the ordinary rules would be undecidable and the sequent system would be unacceptable.

Recall that propositions of the form $\exists x \Phi(x)$ are being treated as abbreviations for propositions of the form $\neg \forall x \neg \Phi(x)$, so no additional inference rules are needed for existential quantification.

> **Definition:** We define FOL to be the sequent proof system whose ordinary rule set is the union of the rule sets BCP and C, and whose sequent rules consist of case analysis, universal generalization, and universal instantiation.

[3]A proposition of the form $\forall y \Psi(y)$ can be renamed to the equivalent proposition $\forall z \Psi(z)$. This may be necessary to avoid the capture of variables. For example, consider instantiating the proposition

$$\forall x (P(x) \rightarrow \forall y Q(x, y))$$

with the term $f(y)$. This will produce a proposition of the form

$$P(f(y)) \rightarrow \forall z Q(f(y),\ z).$$

Note that the bound variable y has been renamed to z to avoid the capture of the occurrence of y in the term $f(y)$.

We write $\Sigma \;\;\vdash_{\mathrm{FOL}}\;\; \Phi$ if there exists some finite subset Γ of Σ such that there is a derivation of $\Gamma \vdash \Phi$ in the sequent system FOL. This definition allows the inference relation to be defined even when Σ is infinite.

Refutation Theorem: For any (possibly infinite) set of propositions Σ, if $\Sigma \;\;\not\vdash_{\mathrm{FOL}}\;\; \mathbf{F}$ then Σ is satisfiable.

The proof of the refutation theorem is discussed in a later section.

Completeness Corollary: The inference relation $\vdash_{\mathrm{FOL}}$ is sound and complete for first order logic.

The soundness of $\vdash_{\mathrm{FOL}}$ follows directly from its definition. The completeness of $\vdash_{\mathrm{FOL}}$ follows directly from the refutation theorem. To prove completeness, suppose that $\Sigma \models \Phi$. This implies that $\Sigma \cup \{\neg\Phi\}$ is not satisfiable. By the refutation theorem, this implies that $\Sigma \cup \{\neg\Phi\} \;\;\vdash_{\mathrm{FOL}}\;\; \mathbf{F}$ (if $\Sigma \cup \{\neg\Phi\} \;\;\not\vdash_{\mathrm{FOL}}\;\; \mathbf{F}$ then $\Sigma \cup \{\neg\Phi\}$ would be satisfiable). But, by the inference rules of BCP, this implies that $\Sigma \cup \{\neg\Phi\} \;\;\vdash_{\mathrm{FOL}}\;\; \Phi$. Since $\Sigma \cup \{\Phi\} \;\;\vdash_{\mathrm{FOL}}\;\; \Phi$, the case analysis sequent rule gives $\Sigma \;\;\vdash_{\mathrm{FOL}}\;\; \Phi$.

Consider a fixed finite set of propositions Σ. It is possible to write a computer program that runs for ever generating all proofs and which lists all of the formulas Ψ such that $\Sigma \;\;\vdash_{\mathrm{FOL}}\;\; \Psi$.

Observation: For any finite set of propositions Σ, the set of propositions Ψ such that $\Sigma \;\;\vdash_{\mathrm{FOL}}\;\; \Psi$ can be recursively enumerated.

The above corollary does not imply that is possible to determine whether or not $\Sigma \;\;\vdash_{\mathrm{FOL}}\;\; \Phi$ for a given Σ and Φ. Although one can enumerate all of the infinitely many consequences of Σ, at no finite time will one ever be sure that Φ is not provable.

3 The Compactness Theorem

The fact that the refutation theorem holds for an infinite set Σ has an important corollary — the compactness of first order logic.

> **Definition:** A logic is called *compact* if for any infinite set Σ of propositions, if every finite subset of Σ is satisfiable (has a model) then the infinite set Σ is also satisfiable.

> **Compactness Theorem:** First order logic is compact.

To prove the compactness theorem, consider an infinite set of propositions Σ such that every finite subset of Σ is satisfiable. By the definition of $\models$ and $\mathbf{F}$, for any set propositions Γ we have that $\Gamma \models \mathbf{F}$ if and only if Γ is unsatisfiable. By assumption, there is no finite subset Γ of Σ such that $\Gamma \models \mathbf{F}$. By the soundness of $\vdash_{\mathrm{FOL}}$, there is no finite subset Γ of Σ such that $\Gamma \vdash_{\mathrm{FOL}} \mathbf{F}$. So $\Sigma \nvdash_{\mathrm{FOL}} \mathbf{F}$. So by the refutation theorem, Σ must be satisfiable.

Compactness has severe ramifications for first order logic. The fact that first order logic is compact can be used show that a wide variety of natural and intuitive concepts can not be expressed in first order logic. For example, consider the notion of transitive closure.

> **Lemma:** The concept of transitive closure can not be expressed in first order logic.

To clarify the statement of the above lemma we must precisely define what it would mean to express the concept of transitive closure. consider a directed graph. A directed graph is a set of directed arcs between nodes. Any directed graph can be encoded as a binary predicate — we let A be the predicate such that $A(n_1, n_2)$ is true if there is an arc in the graph from n_1 to n_2. In the same way, any binary predicate can be viewed as a directed graph. Now consider two binary predicates A and T. We say that the predicate (or directed graph) T is the transitive closure of the predicate (or directed graph) A if $T(n_1,\ n_2)$ is true if and only if there is a path (of length zero or more) from n_1 to n_2 in the directed graph represented by A. In other words, T is the transitive closure of A if for any n_1 and n_2 we have that $T(n_1,\ n_2)$ if and only if there exists some sequence $m_1, m_2, \cdots, m_k$ such that m_1 is n_1, m_k is n_2, and $A(m_j,\ mj + 1)$ for each $j < k$.

Now suppose that one wishes to use first order logic to express the statement that T is the transitive closure of A. One might start by writing down the proposition

$$\forall x, y(T(x, y) = (x = y \vee A(x,\ y) \vee \exists z A(x,\ z) \wedge T(z,\ y))).$$

This propositions looks like a recursive definition of the predicate T in terms of the predicate A. Unfortunately, the above proposition is not a "definition" at all — for a given predicate A there are many different ways of interpreting T so that the above proposition is true.[4] Let $\mathcal{M}$ be the first order structure whose domain is the set of natural numbers, that interprets A as the predicate such that A is true of two numbers x and y if and only if $x = y$, and which interprets T as the universally true predicate, i.e., T is true of any pair of numbers x and y. One can verify that the above proposition is true in the structure $\mathcal{M}$ even though T is not the transitive closure of A. So the above proposition does not express the statement that T is the transitive closure of A.

Using the compactness theorem one can prove that the statement that T is the transitive closure of A can not be expressed by any set of first order propositions. Let Γ be a (possibly infinite) set of propositions. Suppose that Γ expresses the statement that T is the transitive closure of A, i.e., for any first order model $\mathcal{M}$, we have that $\mathcal{M}$ satisfies Γ if and only if, in the model $\mathcal{M}$, T is the transitive closure of A. We can now use the compactness theorem to derive a contradiction. Let b and c be any two constant symbols. We now define an infinite series of propositions $\Psi^0, \Psi^1, \Psi^2, \cdots$ where Ψ^n expresses the statement that there is no path of length n from b to c. More specifically, Ψ^0 is the proposition $b \neq c$, Ψ^1 is the proposition $\neg A(b,\ c)$ and, in general, Ψ^n

[4] In computer programming languages, recursive definitions are based on least fixed point semantics. In the "definition" of T in terms of A in first order logic there is no least fixed point requirement — any fixed point is acceptable.

is the proposition

$$\forall x_1, x_2, \cdots x_{n-1} \neg(A(b, x_1) \wedge A(x_1,\ x_2) \wedge \cdots \wedge A(x_{n-1},\ c)).$$

Now let Γ' be the infinite set of propositions

$$\Gamma \cup \{T(b,\ c), \Psi^0, \Psi^1, \Psi^2, \cdots\}.$$

Assuming that Γ expresses the statement that T is the transitive closure of A, every finite subset of Γ' must be satisfiable. To see this, consider any non-negative number n and the set

$$\Gamma \cup \{T(b,\ c), \Psi^0, \Psi^1, \Psi^2, \cdots \Psi^n\}.$$

If Γ represents the statement that T is the transitive closure of A then this finite extension of Γ must be satisfiable. To see this it suffices to observe that there exists a first order structure in which T is the transitive closure of A (so Γ is satisfied) and in which $T(b,\ c)$ is true and yet the shortest path from b to c has length greater than n. So every finite subset of Γ' is satisfiable. The compactness theorem now implies that the infinite set Γ' is satisfiable. Let $\mathcal{M}$ be any model of Γ'. In the model $\mathcal{M}$ we have that $T(b,\ c)$ is true, but for any number n we also have that Ψ^n is true. So, although $T(b,\ c)$ is true in $\mathcal{M}$, there is no path in $\mathcal{M}$ in the graph defined by the predicate A from b to c. So, even though $\mathcal{M}$ is a model of Γ, the predicate T is not the transitive closure of the predicate A. So Γ can not express the statement that T is the transitive closure of A.

4 Proof of the Refutation Theorem

The proof involves a Herbrand construction. Given an (infinite) set of propositions Σ such that $\Sigma \not\vdash_{\mathrm{FOL}} \mathbf{F}$ we construct a "Herbrand model" of Σ. The Herbrand model is a first order structure whose domain consists of equivalence classes of terms. The construction of the Herbrand model will be included in a later version of the notes.

5 Problems

1. Reduce the STRIPS planning problem to the problem of first order validity. More specifically, given a set $o_1, \cdots o_b$ of STRIPS operators, a set Σ of initial propositions, and a set Γ of goal propositions, show how to construct a first order formula Φ such that Φ is valid if and only if there exists a solution to the given STRIPS planning problem. (You can assume that the planning problem is ground, i.e., all propositions involved are simple proposition symbols.)

2. Let $\mathcal{L}$ be the first order language with function symbols $+$ and $\times$, constant symbols 0 and 1, and binary predicate $<$ (and, of course, equality). Let $\mathcal{R}$ be the first order model whose domain consists of the real numbers and which gives the standard interpretation to all of the symbols of $\mathcal{L}$. Let Γ be the theory of $\mathcal{R}$, i.e., the set of first order formulas that are true in $\mathcal{R}$. Prove that there exists a model of the infinite set Γ in which there exists an element ∞ that is greater than every integer.

3. Consider a function f of one argument. I will say that there exists "an f-path" from x to y if y can be derived from x by applying f some finite number of times, i.e., y equals x, or y equals $f(x)$, or y equals $f(f(x))$, ... If S is a set then the "f-closure" of S is the set of all objects y such that there is an f-path from some member of S to y. Alternatively, we could define $f(S)$ to be the set of all objects of the form $f(x)$ for x in S. We could then define the f-closure of S to be S union $f(S)$ union $f(f(S))$ union, ...

Give a proof that the notion of f-closure can not be defined in first order logic. More specifically, let P and Q be monadic (i.e., one argument) predicate symbols and let f be a monadic function symbol. Consider a first order structure $\mathcal{M}$. The predicates P and Q identify subsets of the domain of $\mathcal{M}$ — the set of elements on which P and Q are true respectively. Give a proof that there is no first order proposition Φ such that Φ is true in a first order structure $\mathcal{M}$ if and only if the set of domain elements on which Q is true is exactly the f-closure of the set of elements on which P is true.

4. Consider a first order structure $\mathcal{M}$ (a model of first order logic) with an infinite domain and such that for every (semantic) element d of the domain of

$\mathcal{M}$ there exists a term t such that $d = \mathcal{V}(t,\ \mathcal{M})$, i.e., d is the semantic value of t. In other words, every element of the domain of $\mathcal{M}$ is the value of some term. We assume that there is a countably infinite set of constants, variables, function symbols and predicate symbols. Let Γ be a set of formulas such that each formula in Γ is true in the structure $\mathcal{M}$ (Γ might be the infinite set of all formulas true in $\mathcal{M}$). Prove that there exists some other first order structure $\mathcal{M}'$ such that every formula in Γ is also true in $\mathcal{M}'$ and such that there is an element of the domain of $\mathcal{M}'$ that is not the value of any term.

You should use the fact that one can enumerate all terms in an infinite sequence $t_1,\ t_2,\ t_3,\ \ldots$ (the set of all terms is countable). You can also introduce a "new" constant symbol c that does not appear in any of the "old" terms $t_1,\ t_2,\ t_3,\ \ldots$. It is sufficient to find a structure $\mathcal{M}'$ such that there is an element of the domain of $\mathcal{M}'$ which is not the value of any of the old terms. Use the back of the page if necessary.

References

[Davis, 1983] Martin Davis. The history and early history of automated deduction. In *Automation of Reasoning I, Classical Papers on Computational Logic 1955-1966*, pages 1–28. Springer-Verlag, 1983.