Chapter 2
Derivations and Proofs in the Predicate Logic

2.1 Motivation

The propositional logic has several limitations for expressing ideas; mainly, it is not possible to quantify over sets of individuals and reason about them. These limitations can be better explained through examples

"Every prime number bigger than 2 is odd"
"There exists a prime number greater than any given natural number"

In the language of the propositional logic, this kind of properties can only be represented by a propositional variable because there is no way to split this information into simpler propositions joined by connectives and able to express the quantification over the natural numbers. In fact, the information in these sentences includes observations about sets of prime numbers, odd numbers, natural numbers, and quantification over them, and these relations cannot be straightforwardly captured in the language of propositional logic.

In order to overcome these limitations of the expressive power of the propositional logic, we extend its language with variables which range over individuals, and quantification over these variables. Thus, in this chapter we present the *predicate logic*, also known as *first-order logic*. In order to obtain a language with abilities to identify the required additional information, we need to extend the propositional language and provide a more expressive deductive calculus.

2.2 Syntax of the Predicate Logic

The language of the first-order predicate logic has two kinds of expressions: *terms* and *formulas*. While in the language of propositional logic formulas that are built up from propositional variables, in the predicate logic they are built from atomic formulas, that are relational formulas expressing properties of terms such as "prime(2)", "prime(x)",

M. Ayala-Rincón and F.L.C. de Moura, *Applied Logic for Computer Scientists*,
Undergraduate Topics in Computer Science, DOI 10.1007/978-3-319-51653-0_2

"x is bigger than 2", etc. Formulas are built from relational formulas using the logical connectives as in the case of propositional logic, but in predicate logic also quantifiers over variables will be possible. Terms and basic relational formulas are built out of variables and two sets of symbols $\mathbb{F}$ and $\mathbb{P}$. Each function symbol in $\mathbb{F}$ and each predicate symbol in $\mathbb{P}$ come with its fixed arity (that is, the number of its arguments). *Constants* can be seen as function symbols of arity zero. No predicate symbols with arity zero are allowed. This is the part of the language that is flexible since the sets $\mathbb{F}$ and $\mathbb{P}$ can be chosen arbitrarily.

Intuitively, predicates are functions that represent properties of terms. In order to define predicate formulas, we first define terms, and to do so, we assume an enumerable set $\mathbb{V}$ of *term variables*.

Definition 13 (*Terms*) A *term t* is defined inductively as follows:

1. Any variable $x \in \mathbb{V}$ is a term;
2. If $t_1, t_2, \ldots, t_n$ are terms, and $f \in \mathbb{F}$ is a function symbol with arity $n \geq 0$ then $f(t_1, t_2, \ldots, t_n)$ is a term. A function of arity zero is a constant.

Notation 1 We follow the usual notational convention for terms. Constant symbols, function symbols, arbitrary terms, and variables are denoted by Roman lower case letters, respectively, of the first, second, third, and fourth quarters of the alphabet: $a, b, \ldots$, for constant symbols; $f, g, \ldots$, for function symbols; $s, t, \ldots$ for arbitrary terms and; x, y, z, for variables.

Terms, as given in the previous definition, could be equivalently presented by the following syntax:

$$t ::= x \;\;||\;\; f(t, \ldots, t)$$

Definition 14 (*Variable occurrence*) The set of variables occurring in a term t, denoted by $\mathtt{var}(t)$, is inductively defined as follows:

- If $t = x$ then $\mathtt{var}(t) = \{x\}$
- If $t = f(t_1, \ldots, t_n)$ then $\mathtt{var}(t) = \mathtt{var}(t_1) \cup \cdots \cup \mathtt{var}(t_n)$

We define the substitution of the term u for x in the term t, written $t[x/u]$, as the replacement of all occurrences of x in t by u. Formally, we have the following definition.

Definition 15 (*Term Substitution*) Let t, u be terms, and x, a variable. We define $t[x/u]$ inductively as follows:

- $x[x/u] = u$;
- $y[x/u] = y$, for $y \neq x$;
- $f(t_1, \ldots, t_n)[x/u] = f(t_1[x/u], \ldots, t_n[x/u])$ $(n \geq 0)$.

Now we are a ready to define the formulas of the predicate logic:

Definition 16 (*Formulas*) The set of formulas of the first-order predicate logic over a variable set $\mathbb{V}$ and a symbol set $S = (\mathbb{F}, \mathbb{P})$ is inductively defined as follows:

1. $\perp$ and $\top$ are formulas;
2. If $p \in \mathbb{P}$ with arity $n > 0$, and $t_1, t_2, \ldots, t_n$ are terms then $p(t_1, t_2, \ldots, t_n)$ is a formula;
3. If φ is a formula then so is $(\neg\varphi)$;
4. If φ_1 and φ_2 are formulas then so are $(\varphi_1 \wedge \varphi_2)$, $(\varphi_1 \vee \varphi_2)$ and $(\varphi_1 \rightarrow \varphi_2)$;
5. If $x \in \mathbb{V}$ and φ is a formula then $(\forall_x \varphi)$ and $(\exists_x \varphi)$ are formulas.

The symbol $\forall_x$ (resp. $\exists_x$) means "for all x" (resp. "there exists a x"), and the formula φ is the *body* of the formula $(\forall_x \varphi)$ (resp. $(\exists_x \varphi)$). Since quantification is restricted to variable terms, the defined language corresponds to a so-called *first-order language*.

The set of formulas of the predicate logic have the following syntax:

$$\varphi ::= p(t, \ldots, t) \;||\; \perp \;||\; \top \;||\; (\neg\varphi) \;||\; (\varphi \wedge \varphi) \;||\; (\varphi \vee \varphi) \;||\; (\varphi \rightarrow \varphi) \;||\; (\forall_x \varphi) \;||\; (\exists_x \varphi)$$

Formulas of the form $p(t_1, \ldots, t_n)$ are called *atomic formulas* because they cannot be decomposed into simpler formulas. As usual, parenthesis are used to avoid ambiguities and the external ones will be omitted. The quantifiers $\forall_x$ and $\exists_x$ bind the variable x in the body of the formula. This idea is formalized by the notion of *scope of a quantifier*:

Definition 17 (*Scope of quantifiers, free and bound variables*) The scope of $\forall_x$ (resp. $\exists_x$) in the formula $\forall_x \varphi$ (resp. $\exists_x \varphi$) is the body of the quantified formula: φ. An occurrence of a variable x in the scope of $\forall_x$ or $\exists_x$ is called *bound*. An occurrence of a variable that is not bound is called *free*.

Since the body of a quantified formula can have occurrences of other quantified formulas that abstract the same variable symbol, it is necessary to provide more precise mechanisms to build the sets of free and bound variables of a predicate formula. This can be done inductively according to the following definitions:

Definition 18 (*Construction of the set of free variable*) Let φ be a formula of the predicate logic. The set of free variables of φ, denoted by $\mathtt{fv}(\varphi)$, is inductively defined as follows:

1. $\mathtt{fv}(\perp) = \mathtt{fv}(\top) = \emptyset$;
2. $\mathtt{fv}(p(t_1, \ldots, t_n)) = \mathtt{var}(t_1) \cup \ldots \cup \mathtt{var}(t_n)$;
3. $\mathtt{fv}(\neg\varphi) = \mathtt{fv}(\varphi)$;
4. $\mathtt{fv}(\varphi \Box \psi) = \mathtt{fv}(\varphi) \cup \mathtt{fv}(\psi)$, where $\Box \in \{\wedge, \vee, \rightarrow\}$;
5. $\mathtt{fv}(Q_x \varphi) = \mathtt{fv}(\varphi) \setminus \{x\}$, where $Q \in \{\forall, \exists\}$.

A formula without occurrences of free variables is called a *sentence*.

Definition 19 (*Construction of the set of bound variables*) Let φ be a formula of the predicate logic. The set of bound variables of φ, denoted by $\mathtt{bv}(\varphi)$, is inductively defined as follows:

1. $\mathtt{bv}(\bot) = \mathtt{bv}(\top) = \emptyset$;
2. $\mathtt{bv}(p(t_1, \ldots, t_n)) = \emptyset$;
3. $\mathtt{bv}(\neg\varphi) = \mathtt{bv}(\varphi)$;
4. $\mathtt{bv}(\varphi \Box \psi) = \mathtt{bv}(\varphi) \cup \mathtt{bv}(\psi)$, where $\Box \in \{\wedge, \vee, \rightarrow\}$;
5. $\mathtt{bv}(Q_x\varphi) = \mathtt{bv}(\varphi) \cup \{x\}$, where $Q \in \{\forall, \exists\}$.

Informally, the name of a bound variable is not important in the sense that it can be *renamed* to any *fresh* name without changing the semantics of the term. For instance, the formulas $\forall_x(x \leq x)$, $\forall_y(y \leq y)$ and $\forall_z(z \leq z)$ represent the very same object. The sole restriction that needs to be considered is that *variable capture* is forbidden, i.e., no free variable can become bound after a renaming of a variable. For instance, if p denotes a binary predicate then $\forall_x p(x, y)$ is a renaming of $\forall_z p(z, y)$, while $\forall_y p(y, y)$ is not. The next definition will formalize the notion of substitution. The capture of free variables by a substitution is also forbidden, and we assume that a renaming of bound variables is always performed when necessary to avoid capture.

Definition 20 (*Substitution*) Let φ be a formula of the predicate logic. The substitution of x by t in φ, written $\varphi[x/t]$, is inductively defined as follows:

1. $\bot[x/t] = \bot$ and $\top[x/t] = \top$;
2. $p(t_1, \ldots, t_n)[x/t] = p(t_1[x/t], \ldots, t_n[x/t])$;
3. $(\neg\psi)[x/t] = \neg(\psi[x/t])$;
4. $(\psi \Box \gamma)[x/t] = (\psi[x/t]) \Box (\gamma[x/t])$, where $\Box \in \{\wedge, \vee, \rightarrow\}$;
5. $(Q_y\psi)[x/t] = Q_y(\psi[x/t])$, where $Q \in \{\exists, \forall\}$, and renaming of bound variables is assumed to avoid capture of variables.

Example 6 Consider the following applications of substitution:

- $(\forall_x p(y))[y/x] = \forall_z p(y)[y/x] = \forall_z p(y[y/x]) = \forall_z p(x)$ and
- $(\forall_x p(x))[x/t] = \forall_y p(y)[x/t] = \forall_y p(y[x/t]) = \forall_y p(y)$.

Notice that in the second application, renaming x as y was necessary to avoid capture.

The necessary renamings to avoid capture of variables in substitutions can be implemented in several ways. For instance, it can be done by modifying item 5 in the definition of substitution in such a way that before propagating the substitution inside the scope of a quantified formula of the form $(Q_x\varphi)[x/t]$, where $Q \in \{\forall, \exists\}$, it is checked whether $x = y$ or $x \in \mathtt{fv}(t)$: whenever $x = y$ or $x \in \mathtt{fv}(t)$ renaming the quantified variable name x as a fresh variable name z is applied, in other case no renaming is needed

$$(Q_x\varphi)[y/t] = \begin{cases} (Q_z\varphi[x/z][y/t]), & \text{if } x = y \text{ or } x \in \mathtt{fv}(t), \\ (Q_x\varphi[y/t]), & \text{otherwise.} \end{cases}$$

The size of predicate expressions (terms and formulas) is defined in the usual manner.

Definition 21 (*Size of predicate expressions*) Let t be a predicate term and φ a predicate formula. The size of t, denoted as $|t|$, is recursively defined as follows:

- $|x| = 1$, for $x \in \mathbb{V}$;
- $|f(t_1, \dots, t_n)| = 1 + |t_1| + \cdots + |t_n|$, for $n \geq 0$.

The size of φ, denoted as $|\varphi|$, is recursively defined as follows:

- $|\bot| = |\top| = 1$;
- $|p(t_1, \dots, t_n)| = 1 + |t_1| + \cdots + |t_n|$, for $n \geq 1$;
- $|(\neg\psi)| = 1 + |\psi|$;
- $|(\psi \Box \gamma)| = 1 + |\psi| + |\gamma|$, where $\Box \in \{\wedge, \vee, \rightarrow\}$;
- $|(Q_y \psi)| = 1 + |\psi|$, where $Q \in \{\exists, \forall\}$.

Exercise 23

a. Consider a predicate formula φ and a term t. Prove that there are no bound variables in the new occurrences of t in the formula $\varphi[x/t]$. For doing this use induction on the structure of φ. Of course, occurrences of the term t in the original formula φ might be under the scope of quantifiers and consequently variables occurring in these subterms would be bound.
b. Let k be the number of free occurrences of the variable x in the predicate formula φ. Prove, also by induction on φ, that the size of the term $\varphi[x/t]$ is given by $k|t| + |\varphi| - k$.
c. For $x \neq y$, prove also that:

   i. $\varphi[x/s][x/t] = \varphi[x/s[x/t]]$;
   ii. $\varphi[x/s][y/t] = \varphi[x/s[y/t]][y/t]$, if $y \notin \mathtt{var}(t)$;
   iii. $\varphi[x/s][y/t] = \varphi[y/t][x/s]$, if $x \notin \mathtt{var}(t)$ and $y \notin \mathtt{var}(s)$.

2.3 Natural Deduction in the Predicate Logic

The set of rules of natural deduction for the predicate logic is an extension of the set presented for the propositional logic. The rules for conjunction, disjunction, implication, and negation have the same shape, but note that now the formulas are the ones of predicate logic. In this section, we also discuss the minimal, intuitionistic, and classical predicate logic. Thus the rules are those in Table 1.2, without the rule $(\bot_e)$ for the minimal predicate logic and with this rule for the intuitionistic predicate logic, and in Table 1.3 for the classical predicate logic, plus four additional rules for dealing with quantified formulas.

We start by expanding the set of natural deduction rules with the ones for quantification. The first one is the elimination rule for the universal quantifier:

$$\frac{\forall_x \varphi}{\varphi[x/t]}\ (\forall_e)$$

The intuition behind this rule is that from a proof of $\forall_x \varphi$, we can conclude $\varphi[x/t]$, where t is any term. This transformation is done by the substitution operator previously defined that replaces every free occurrence of x by an arbitrary term t in φ. According to the substitution operator, "every" occurrence of x in φ is replaced with the "same" term t. The following example shows an application of $(\forall_e)$ in a derivation:

Example 7 $\forall_x p(a, x), \forall_x \forall_y (p(x, y) \to p(f(x), y)) \vdash p(f(a), f(a))$.

$$\dfrac{\dfrac{\forall_x p(a,x)}{p(a, f(a))}\,(\forall_e) \qquad \dfrac{\dfrac{\forall_x \forall_y (p(x,y) \to p(f(x), y))}{\forall_y p(a,y) \to p(f(a), y)}\,(\forall_e)}{p(a, f(a)) \to p(f(a), f(a))}\,(\forall_e)}{p(f(a), f(a))}\,(\to_e)$$

Note that the application of $(\to_e)$ is identical to what is done in the propositional calculus, except for the fact that now it is applied to predicate formulas.

The introduction rule for the universal quantifier is more subtle. In order to prove $\forall_x \varphi$ one needs first to prove $\varphi[x/x_0]$ in such a way that no open assumption in the derivation of $\varphi[x/x_0]$ can contain occurrences of x_0. This restriction is necessary to guarantee that x_0 is general enough and can be understood as "any" term, i.e., nothing has been assumed concerning x_0. The $(\forall_i)$ rule is given by

$$\dfrac{\varphi[x/x_0]}{\forall_x \varphi}\,(\forall_i)$$

where x_0 is a fresh variable not occurring in any open assumption in the derivation of $\varphi[x/x_0]$.

Example 8 $\forall_x (p(x) \wedge q(x)) \vdash \forall_x (p(x) \to q(x))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\forall_x (p(x) \wedge q(x))}{p(x_0) \wedge q(x_0)}\,(\forall_e)}{q(x_0)}\,(\wedge_e)}{p(x_0) \to q(x_0)}\,(\to_i)\emptyset}{\forall_x (p(x) \to q(x))}\,(\forall_i)$$

Note that the formula $p(x_0) \to q(x_0)$ depends only on the hypothesis $\forall_x (p(x) \wedge q(x))$, which does not contain x_0. Thus x_0 might be considered arbitrary, which allows the generalization through application of rule $(\forall_i)$. In fact, note that the above proof of $p(x_0) \to q(x_0)$ could be done for any other term, say t instead x_0, which explains the generality of x_0 in the above example.

The introduction rule for the existential quantifier is as follows:

$$\frac{\varphi[x/t]}{\exists_x\, \varphi}\ (\exists_i)$$

where t is any term.

Example 9 $\forall_x\, q(x) \vdash \exists_x\, q(x)$.

$$\cfrac{\cfrac{\forall_x\, q(x)}{q(x_0)}\ (\forall_e)}{\exists_x\, q(x)}\ (\exists_i)$$

Similarly to $(\forall_i)$, the elimination rule for the existential quantifier is more subtle:

$$\frac{\exists_x\, \varphi \qquad \begin{array}{c}[\varphi[x/x_0]]^u\\ \vdots\\ \chi\end{array}}{\chi}\ (\exists_e)\ u$$

This rule requires the variable x_0 be a fresh variable neither occurring in any other open assumption than in $[\varphi[x/x_0]]^u$ itself nor in the conclusion $\mathtt{fv}(\chi)$. The intuition of this rule might be explained as follows: knowing that $\exists_x\, \varphi$ holds, if assuming that an arbitrary x_0 witnesses the property φ, i.e., assuming $[\varphi[x/x_0]]^u$, one can infer χ, then χ holds in general. This kind of analysis is done, for instance, when properties about numbers are inferred from the knowledge of the existence of prime numbers of arbitrary size, or (good/bad) properties about institutions are inferred from the knowledge of the existence of the (good/bad) qualities of some individuals in their staffs. These general properties are inferred without knowing specific prime numbers or without knowing who are specifically the (good/bad) individuals in the institutions.

Example 10 This example attempts to bring a little bit intuition about the use of these rules. Let p, q, and r be predicate symbols with the intended meanings: $p(z)$ means "z is a planet different from the earth with similar characteristics"; $q(y)$ means "country y adopts action to mitigate global warming" and $r(x, y)$ means "x is a leader, who works in the ministry of agriculture or environment of country y and who is worried about climate change". Thus, from the hypotheses $\forall_y \exists_x r(x, y)$, $\forall_y \forall_x (r(x, y) \to q(y))$ and $\forall_z(\forall_y q(y) \to \neg p(z))$, we can infer that we do not need a "Planet B" as follows:

$$
\dfrac{\dfrac{\dfrac{\forall_y \exists_x r(x,y)}{\exists_x r(x,c_0)}\,(\forall_e) \qquad \dfrac{[r(l_0,c_0)]^u \qquad \dfrac{\dfrac{\dfrac{\forall_y \forall_x (r(x,y) \to q(y))}{\forall_x (r(x,c_0) \to q(c_0))}\,(\forall_e)}{r(l_0,c_0) \to q(c_0)}\,(\forall_e)}{q(c_0)}\,(\to_e)}{q(c_0)}\,(\exists_e)\ u}{\forall_y q(y)}\,(\forall_e) \qquad \dfrac{\forall_z (\forall_y q(y) \to \neg p(z))}{\forall_y q(y) \to \neg p(B)}\,(\forall_e)}{\neg p(B)}\,(\to_e)
$$

Example 11 The use of substitution in natural deduction rules for quantifiers is illustrated in this example. Initially, consider a unary predicate p. Below, it is depicted a derivation for $\exists_x\, p(x) \vdash \neg\forall_x \neg p(x)$.

$$
\dfrac{\exists_x p(x) \qquad \dfrac{\dfrac{[p(x_0)]^u \qquad \dfrac{[\forall_x \neg p(x)]^v}{\neg p(x_0)}\,(\forall_e)}{\bot}\,(\neg_e)}{\neg\forall_x \neg p(x)}\,(\neg_i)\ v}{\neg\forall_x \neg p(x)}\,(\exists_e)\ u
$$

Now, consider a predicate formula φ and a variable x that might or might not occur free in φ. The next derivation, denoted as ∇_3, proofs that $\vdash \exists_x \varphi \to \neg\forall_x \neg\varphi$. Despite the proof for φ appears to be the same than the one above for the unary predicate p, several subtle points should be highlighted. In the application of rule $(\exists_e)$ in the derivation ∇_3, it is forbidden the selection of a witness variable "y", to be used in the witness assumption $[\varphi[x/y]]^w$, such that y belongs to the set of free variables occurring in φ. Indeed, y should be a *fresh* variable. To understand this restriction, consider $\varphi = q(y, x)$ and suppose the intended meaning of q is "x is the double of y". If the existential formula is $\exists_x p(y, x)$ the witness assumption cannot be $p(y, x)[x/y] = p(y, y)$, since this selection of "y" is not arbitrary.

$$
\dfrac{\dfrac{\dfrac{[\exists_x \varphi]^u \qquad \dfrac{[\varphi[x/y]]^w \qquad \dfrac{[\forall_x \neg\varphi]^v}{\neg\varphi[x/y]}\,(\forall_e)}{\bot}\,(\neg_e)}{\bot}\,(\exists_e)\ w}{\neg\forall_x \neg\varphi}\,(\neg_i)\ v}{\exists_x \varphi \to \neg\forall_x \neg\varphi}\,(\to_i)\ u
$$

The rules for quantification discussed so far are summarized in Table 2.1. These rules together with the deduction rules for introduction and elimination of the con-

Table 2.1 Natural deduction rules for quantification

Introduction rules	Elimination rules
$\dfrac{\varphi[x/x_0]}{\forall_x\varphi}\ (\forall_i)$ where x_0 cannot occur free in any open assumption.	$\dfrac{\forall_x\varphi}{\varphi[x/t]}\ (\forall_e)$
$\dfrac{\varphi[x/t]}{\exists_x\varphi}\ (\exists_i)$	$\dfrac{\exists_x\varphi \qquad \begin{matrix}[\varphi[x/x_0]]^u\\ \vdots\\ \chi\end{matrix}}{\chi}\ (\exists_e)\ u$ where x_0 cannot occur free in any open assumption on the right and in χ.

nectives: $\wedge$, $\vee$, $\neg$, and $\to$, conform the set of natural deduction rules for the minimal predicate logic (that is, rules in Tables 2.1 and 1.2 except rule $(\bot_e)$). If in addition, we include the intuitionistic absurdity rule, we obtain the natural deduction calculus for the intuitionistic predicate logic (that is all rules in Tables 2.1 and 1.2). The classical predicate calculus is obtained from the intuitionistic one, changing the intuitionistic absurdity rule by the rule PBC (that is, rules in Tables 2.1 and 1.3).

Example 12 The sequent $\vdash \exists_x\neg\varphi \to \neg\forall_x\varphi$ has the following intuitionistic proof ∇_1:

$$\dfrac{\dfrac{\dfrac{[\exists_x\neg\varphi]^u \qquad \dfrac{\dfrac{[\forall_x\varphi]^v}{\varphi[x/y]}\ (\forall_e) \qquad [\neg\varphi[x/y]]^w}{\bot}\ (\neg_e)}{\bot}\ (\exists_e)\ w}{\neg\forall_x\varphi}\ (\neg_i)\ v}{\exists_x\neg\varphi \to \neg\forall_x\,\varphi}\ (\to_i)\ u$$

The proof ∇_1 can be used to prove the sequent $\vdash \forall_x\varphi \to \neg\exists_x\neg\varphi$ as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\nabla_1\\ \exists_x\neg\varphi \to \neg\forall_x\varphi\end{array} \qquad [\exists_x\neg\varphi]^v}{\neg\forall_x\varphi}\,(\to_e) \qquad [\forall_x\varphi]^w}{\bot}\,(\neg_e)}{\neg\exists_x\neg\varphi}\,(\neg_i)\ v}{\forall_x\varphi \to \neg\exists_x\neg\varphi}\,(\to_i)\ w$$

Exercise 24 Prove intuitionistically that $\neg\exists_x\varphi \dashv\vdash \forall_x\neg\varphi$.

Exercise 25 Prove that:

a. if x does not occur free in ψ then prove that $(\exists_x\phi) \to \psi \vdash \forall_x(\phi \to \psi)$; and
b. if x does not occur free in ψ then prove that $(\forall_x\phi) \to \psi \vdash \exists_x(\phi \to \psi)$.

Exercise 26 Prove that

a. $(\forall_x\phi) \wedge (\forall_x\psi) \dashv\vdash \forall_x(\phi \wedge \psi)$; and
b. $(\exists_x\phi) \vee (\exists_x\psi) \dashv\vdash \exists_x(\phi \vee \psi)$.

Exercise 27 Prove that $\forall_x(p(x) \to \neg q(x)) \vdash \neg(\exists_x(p(x) \wedge q(x)))$.

The interpretation of formulas in the classical logic is different from the one in the intuitionistic logic. While in the intuitionistic logic the goal is to "have a constructive proof" of a formula φ, in the classical logic the goal is to "establish a proof of the truth" of φ. For instance, a classical proof admits the truth of a formula of the form $\exists_x\varphi$ without having an explicit witness for x. Such kind of proof (without an explicit witness for the existential) is not accepted in the intuitionistic logic. As an example, suppose that one wants to prove that there exists two irrational numbers x and y such that x^y is rational. If $r(x)$ means that "x is a rational number" then one aims to prove the sequent $\vdash \exists_x\exists_y(\neg r(x) \wedge \neg r(y) \wedge r(x^y))$. In order to do so, we assume some obvious facts in algebra, such as $\neg r(\sqrt{2})$ and $r((\sqrt{2}^{\sqrt{2}})^{\sqrt{2}})$.

$$\dfrac{\text{(LEM)}\,\dfrac{}{r(\sqrt{2}^{\sqrt{2}}) \vee \neg r(\sqrt{2}^{\sqrt{2}})} \qquad \nabla_1 \qquad \nabla_2}{\exists_x\exists_y(\neg r(x) \wedge \neg r(y) \wedge r(x^y))}\,(\vee_e)\ a, b$$

where ∇_1 is given by

$$\dfrac{\dfrac{\dfrac{\neg r(\sqrt{2}) \qquad \dfrac{\neg r(\sqrt{2}) \qquad [r((\sqrt{2})^{\sqrt{2}})]^a}{\neg r(\sqrt{2}) \wedge r((\sqrt{2})^{\sqrt{2}})}\,(\wedge_i)}{\neg r(\sqrt{2}) \wedge \neg r(\sqrt{2}) \wedge r((\sqrt{2})^{\sqrt{2}})}\,(\wedge_i)}{\exists_y(\neg r(\sqrt{2}) \wedge \neg r(y) \wedge r((\sqrt{2})^y))}\,(\exists_i)}{\exists_x\exists_y(\neg r(x) \wedge \neg r(y) \wedge r(x^y))}\,(\exists_i)$$

and ∇_2 is given by

$$\dfrac{\dfrac{\dfrac{[\neg r(\sqrt{2}^{\sqrt{2}})]^b \qquad \dfrac{\neg r(\sqrt{2}) \qquad r((\sqrt{2}^{\sqrt{2}})^{\sqrt{2}})}{\neg r(\sqrt{2}) \wedge r((\sqrt{2}^{\sqrt{2}})^{\sqrt{2}})}\,(\wedge_i)}{\neg r(\sqrt{2}^{\sqrt{2}}) \wedge \neg r(\sqrt{2}) \wedge r((\sqrt{2}^{\sqrt{2}})^{\sqrt{2}})}\,(\wedge_i)}{\exists_y(\neg r(\sqrt{2}^{\sqrt{2}}) \wedge \neg r(y) \wedge r((\sqrt{2}^{\sqrt{2}})^y))}\,(\exists_i)}{\exists_x \exists_y(\neg r(x) \wedge \neg r(y) \wedge r(x^y))}\,(\exists_i)$$

In the proof above, the witnesses depend on whether $\sqrt{2}^{\sqrt{2}}$ is rational or not. In the positive case, taking $x = y = \sqrt{2}$ allows us to conclude that x^y is rational, and in the negative case, this conclusion is achieved by taking $x = \sqrt{2}^{\sqrt{2}}$ and $y = \sqrt{2}$. So we proved the "existence" of an object without knowing explicitly the witnesses for x and y. This is acceptable as a proof in the classical logic, but not in the intuitionistic one.

Analogously to the intuitionistic case, the rules of the classical predicate logic are given by the rule schemes for the connectives ($\wedge$, $\vee$, $\neg$ and $\rightarrow$), the classical absurdity rule (PBC) (see Table 1.3) and the rules for the quantifiers (Table 2.1).

Example 13 While the sequents $\vdash \exists_x\varphi \rightarrow \neg\forall_x\neg\varphi$ and $\vdash \forall_x\varphi \rightarrow \neg\exists_x\neg\varphi$ have intuitionistic (indeed minimal) proofs as shown in Examples 11 and 12, the sequents $\vdash \neg\exists_x\neg\varphi \rightarrow \forall_x\varphi$ and $\vdash \neg\forall_x\neg\varphi \rightarrow \exists_x\varphi$ *have only* classical proofs. A proof for the former is given below.

$$\dfrac{\dfrac{\dfrac{\dfrac{[\neg\exists x\neg\varphi]^u \qquad \dfrac{[\neg\varphi[x/y]]^v}{\exists_x\neg\varphi}\,(\exists_i)}{\bot}\,(\neg_e)}{\varphi[x/y]}\,(\text{PBC})\ v}{\forall_x\varphi}\,(\forall_i)}{\neg\exists_x\neg\varphi \rightarrow \forall_x\varphi}\,(\rightarrow_i)\ u$$

Moreover, note that the above proof jointly with the one given in Example 11 shows that $\forall_x\varphi \dashv\vdash \neg\exists_x\neg\varphi$.

A proof of the sequent $\vdash \neg\forall_x\neg\varphi \rightarrow \exists_x\varphi$ is given below.

$$
\begin{prooftree}
\AxiomC{$[\varphi[x/y]]^v$}
\RightLabel{$(\exists_i)$}
\UnaryInfC{$\exists_x\varphi$}
\AxiomC{$[\neg\exists_x\varphi]^u$}
\RightLabel{$(\neg_e)$}
\BinaryInfC{$\bot$}
\RightLabel{$(\neg_i)\ v$}
\UnaryInfC{$\neg\varphi[x/y]$}
\RightLabel{$(\forall_i)$}
\UnaryInfC{$\forall_x\neg\varphi$}
\AxiomC{$[\neg\forall_x\neg\varphi]^w$}
\RightLabel{$(\neg_e)$}
\BinaryInfC{$\bot$}
\RightLabel{(PBC) u}
\UnaryInfC{$\exists_x\varphi$}
\RightLabel{$(\rightarrow_i)\ w$}
\UnaryInfC{$\neg\forall_x\neg\varphi \rightarrow \exists_x\varphi$}
\end{prooftree}
$$

Finally, this proof jointly with the one given in Example 12 shows that $\exists_x\varphi \dashv\vdash \neg\forall_x\neg\varphi$.

To verify that there are no possible intuitionistic derivations, notice that $\neg\exists_x\neg\varphi \rightarrow \forall_x\varphi$ and $\neg\forall_x\neg\varphi \rightarrow \exists_x\varphi$ together with the intuitionistic (indeed minimal) deduction rules allows derivation of non-intuitionistic theorems such as $\neg\neg\varphi \vdash \varphi$ (see next Exercise 28).

Exercise 28 Prove that there exist derivations for $\neg\neg\varphi \vdash \varphi$ using only the minimal natural deduction rules and each of the assumptions

a. $\neg\exists_x\neg\varphi \rightarrow \forall_x\varphi$ and
b. $\neg\forall_x\neg\varphi \rightarrow \exists_x\varphi$.

Hint: you can choose the variable x as any variable that does not occurs in φ. Thus, the application of rule $(\exists_e)$ over the existential formula $\exists_x\varphi$ has as witness assumption $[\varphi[x/x_0]]^w$ that has no occurrences of x_0.

In Exercise 24 we prove that there are intuitionistic derivations for $\neg\exists_x\varphi \dashv\vdash \forall_x\neg\varphi$. Also, in Example 12 we give an intuitionistic derivation for $\exists_x\neg\varphi \vdash \neg\forall_x\varphi$. Indeed, one can obtain minimal derivations for these three sequents.

Exercise 29 To complete $\neg\forall_x\varphi \dashv\vdash \exists_x\neg\varphi$ (see Example 12), prove that $\neg\forall_x\varphi \vdash \exists_x\neg\varphi$.

2.4 Semantics of the Predicate Logic

As done for the propositional logic in Chap. 1, here we present the standard semantics of first-order classical logic. The semantics of the predicate logic is not a direct extension of the one of propositional logic. Although this is not surprising, since the predicate logic has a richer language, there are some interesting points concerning the differences between propositional and predicate semantics that will be examined in this section. In fact, while a propositional formula has only finitely many interpretations, a predicate formula can have infinitely many ones.

We start with an example: let p be a unary predicate symbol, and consider the formula $\forall_x p(x)$. The variable x ranges over a domain, say the set of natural numbers $\mathbb{N}$. Is this formula true or false? Certainly, it depends on how the predicate symbol p is interpreted. If one interprets $p(x)$ as "x is a prime number", then it is false, but if $p(x)$ means that "x is a natural number" then it is true. Observe that the interpretation depends on the chosen domain, and hence the latter interpretation of p will be false over the domain of integers $\mathbb{Z}$.

This situation is similar in the propositional logic: according to the interpretation, some formulas can be either true or false. So what do we need to determine the truth value of a predicate formula? First of all, we need a domain of concrete individuals, i.e., a nonempty set $\mathtt{D}$ that represents known individuals (e.g., numbers, people, organisms, etc.). Function symbols (and constants) are associated to functions in the so called *structures*:

Definition 22 (*Structure*) A structure of a first-order language L over the set $S = (\mathbb{F}, \mathbb{P})$, also called an S-structure, is a pair $\langle \mathtt{D}, \mathtt{m} \rangle$, where $\mathtt{D}$ is a nonempty set and $\mathtt{m}$ is a map defined as follows:

1. if f is a function symbol of arity $n \geq 0$, then $\mathtt{m}(f)$ is a function from $\mathtt{D}^n$ to $\mathtt{D}$. A function from $\mathtt{D}^0$ to $\mathtt{D}$ is simply an element of $\mathtt{D}$.
2. if p is a predicate symbol of arity $n > 0$, then $\mathtt{m}(p)$ is a subset of $\mathtt{D}^n$.

Intuitively, the set $\mathtt{m}(p)$ contains the tuples of elements that satisfy the predicate p. As an example, consider the formula $q(a)$, where a is a constant, and the structure $\langle \{0, 1\}, \mathtt{m} \rangle$, where $\mathtt{m}(a) = 0$ and $\mathtt{m}(q) = \{0\}$. The formula $q(a)$ is true in this structure because the set $\mathtt{m}(q)$ contains the element 0, the image of the constant a by the function $\mathtt{m}$. But $q(a)$ would be false in other structures; for instance, it is false in the structure $\langle \{0, 1\}, \mathtt{m}' \rangle$, where $\mathtt{m}'(a) = 0$ and $\mathtt{m}'(q) = \emptyset$.

If a formula contains (free) variables, such as the formula $q(x)$, then a special mechanism is needed to interpret variables. Variables are associated to elements of the domain $\mathtt{D}$ through *assignments* that are functions from the set of variables $\mathbb{V}$ to the domain $\mathtt{D}$. So, if d is an assignment such that $d(x) = 0$ then $q(x)$ is true in the structure $\langle \{0, 1\}, \mathtt{m} \rangle$ above, and if $d'(x) = 1$ then $q(x)$ is false.

Definition 23 (*Interpretation of terms*) An *interpretation* I is a pair $\langle \langle \mathtt{D}, \mathtt{m} \rangle, d \rangle$ containing a structure and an assignment. Given an interpretation I and a term t, the interpretation of t by I, written t^I, is inductively defined as follows:

1. For each variable x, $x^I = d(x)$;
2. For each function symbol f with arity $n \geq 0$, $f(t_1, \ldots, t_n)^I = \mathtt{m}(f)(t_1^I, \ldots, t_n^I)$.

Thus, based on the interpretations of terms, the semantics of predicate formulas concerns the truth value of a formula that can be either T (true) or F (false). This notion is formalized in the following definition.

Definition 24 (*Interpretation of Formulas*) The truth value of a predicate formula φ according to a given interpretation of terms $I = \langle \langle \mathtt{D}, \mathtt{m} \rangle, d \rangle$, denoted as φ^I, is inductively defined as:

1. $\bot^I = F$ and $\top^I = T$;
2. $p(t_1, \dots, t_n)^I = \begin{cases} T, & \text{if } (t_1^I, \dots, t_n^I) \in \mathtt{m}(p), \\ F, & \text{if } (t_1^I, \dots, t_n^I) \notin \mathtt{m}(p); \end{cases}$
3. $(\neg\psi)^I = \begin{cases} T, & \text{if } \psi^I = F, \\ F, & \text{if } \psi^I = T; \end{cases}$
4. $(\psi \wedge \gamma)^I = \begin{cases} T, & \text{if } \psi^I = T \text{ and } \gamma^I = T, \\ F, & \text{otherwise;} \end{cases}$
5. $(\psi \vee \gamma)^I = \begin{cases} T, & \text{if } \psi^I = T \text{ or } \gamma^I = T, \\ F, & \text{otherwise;} \end{cases}$
6. $(\psi \to \gamma)^I = \begin{cases} F, & \text{if } \psi^I = T \text{ and } \gamma^I = F, \\ T, & \text{otherwise;} \end{cases}$
7. $(\forall_x \psi)^I = \begin{cases} T, & \text{if } \psi^{I\frac{x}{a}} = T \text{ for every } a \in \mathtt{D}, \\ F, & \text{otherwise;} \end{cases}$
8. $(\exists_x \psi)^I = \begin{cases} T, & \text{if } \psi^{I\frac{x}{a}} = T \text{ for at least one } a \in \mathtt{D}, \\ F, & \text{otherwise.} \end{cases}$

where $I\frac{x}{a}$ denotes the interpretation I modifying its assignment d, in such a way that it maps x to a, and any other variable y to $d(y)$.

Definition 25 (*Models*) An interpretation I is said to be a *model* of φ if $\varphi^I = T$. We write $I \models \varphi$ to denote that I is a model of φ.

The notion of Model is extended to sets of formulas in a straightforward manner: If Γ is a set of predicate formulas then I is a model of Γ, denoted by $I \models \Gamma$, whenever I is a model of each formula in Γ.

Example 14 Let I be an interpretation with domain $\mathbb{N}$ and $\mathtt{m}(p) = \{(m, n) \in \mathbb{N} \times \mathbb{N} \mid m < n\}$. Then I is a model of $\forall_x \exists_y\, p(x, y)$, denoted as $I \models \forall_x \exists_y\, p(x, y)$, because for every natural x one can find another natural y bigger than x. With similar arguments, one can conclude that I is not a model of $\exists_x \forall_y\, p(x, y)$.

Definition 26 (*Satisfiability*) Let φ be a predicate formula. If φ has a model then it is said to be *satisfiable*; otherwise, it is *unsatisfiable*. This notion is also extended to sets of formulas: Γ is satisfiable if and only if there exist an interpretation I such that for all $\varphi \in \Gamma$, $I \models \varphi$.

Definition 27 (*Logical consequence and Validity*) Let $\Gamma = \{\phi_1, \dots, \phi_n\}$ be a finite set of predicate formulas, and φ a predicate formula. We say that φ is a logical consequence of Γ, denoted as $\Gamma \models \varphi$, if every model of Γ is also a model of φ, i.e. $I \models \Gamma$ implies $I \models \varphi$, for every interpretation I. When Γ is empty then φ is said to be *valid*, which is denoted as $\models \varphi$.

Example 15 We claim that $\forall_x(p(x) \to q(x)) \models (\forall_x p(x)) \to (\forall_x q(x))$. In fact, let $I = \langle\langle \mathtt{D}, \mathtt{m}\rangle, d\rangle$ be a model of $\forall_x(p(x) \to q(x))$, i.e., $I \models \forall_x(p(x) \to q(x))$. If there exists an element in the domain of I that does not satisfy the predicate p then $\forall_x p(x)$ is false in I and hence, $(\forall_x(p(x)) \to (\forall_x q(x))$ would be true in I. Otherwise, $I \models \forall_x p(x)$, and hence $I\frac{x}{a} \models p(x)$, for all $a \in \mathtt{D}$. Since $I \models \forall_x(p(x) \to q(x))$, we conclude that $I\frac{x}{a} \models q(x)$, for all $a \in \mathtt{D}$. Therefore, $I \models \forall_x q(x)$.

The study of models can be justified by the fact that validity in a model is an invariant of provability in the sense that a sequent is provable exactly when all its interpretations are also models. This suggests a way to prove when a sequent is not provable: it is enough to find an interpretation that is not a model of the sequent. In the next section, we formalize this for the predicate logic.

2.5 Soundness and Completeness of the Predicate Logic

2.5.1 Soundness of the Predicate Logic

The soundness of predicate logic can be proved following the same idea used for the propositional logic. Therefore, we need to prove the following theorem:

Theorem 6 (Soundness of the predicate logic) *Let Γ be a set of predicate formulas, if $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$. In other words, if φ is provable from Γ then φ is a logical consequence of* Γ.

Proof The proof is by induction on the derivation of $\Gamma \vdash \varphi$ similarly to the propositional case, and hence we focus just on the new rules: $(\forall_e)$, $(\forall_i)$, $(\exists_e)$, $(\exists_i)$.

If the last rule applied in the proof $\Gamma \vdash \varphi$ is $(\boldsymbol{\forall_e})$, then $\varphi = \psi[x/t]$ and the premise of the last rule is $\forall_x \psi$ as depicted in the following figure, where $\{\gamma_1, \ldots, \gamma_n\}$ is the subset of formulas in Γ used in the derivation.

$$\dfrac{\begin{array}{c}\gamma_1 \quad \cdots \quad \gamma_n \\ \hline \forall_x \psi\end{array}}{\psi[x/t]}\ (\forall_e)$$

The subtree rooted by the formula $\forall_x \psi$ and with open leaves labeled by formulas in Γ, corresponds to a derivation for the sequent $\Gamma \vdash \forall_x \psi$ that by induction hypothesis implies $\Gamma \models \forall_x \psi$. Therefore, for all interpretations that make the formulas in Γ true, also $\forall_x \psi$ would be true: $I \models \Gamma$ implies $I \models \forall_x \psi$. The last implies that for all $a \in \mathtt{D}$, where $\mathtt{D}$ is the domain of I, $I\frac{x}{a} \models \psi$, and in particular, $I\frac{x}{t^I} \models \psi$. Consequently,

$I \models \psi[x/t]$. Therefore, one has that for any interpretation I, such that $I \models \Gamma$, $I \models \psi[x/t]$, which implies $\Gamma \models \psi[x/t]$.

If the last rule applied in the proof of $\Gamma \vdash \varphi$ is $(\forall_i)$, then $\varphi = \forall_x \psi$ and the premise of the last rule is $\psi[x/x_0]$ as depicted in the following figure:

$$\dfrac{\dfrac{\gamma_1 \quad \cdots \quad \gamma_n}{\psi[x/x_0]}}{\forall_x \psi}\ (\forall_i)$$

The subtree rooted by the formula $\psi[x/x_0]$ and with open leaves labeled by formulas in $\{\gamma_1, \ldots, \gamma_n\} \subset \Gamma$, corresponds to a derivation for the sequent $\Gamma \vdash \psi[x/x_0]$, in which no open assumption contains the variable x_0. This variable can be selected in such a manner that it does not appear free in any formula of Γ. By induction hypothesis, we have that $\Gamma \models \psi[x/x_0]$. This implies that all interpretations that make the formulas in Γ true, also make $\psi[x/x_0]$ true: $I \models \Gamma$ implies $I \models \psi[x/x_0]$. Since x_0 does not occurs in Γ, for all $a \in \mathtt{D}$, where $\mathtt{D}$ is the domain of I, $I\frac{x}{a} \models \Gamma$ and also $I\frac{x_0}{a} \models \psi[x/x_0]$ or, equivalently, $I\frac{x}{a} \models \psi$. Hence $\Gamma \models \forall_x \psi$.

If the last rule applied in the proof of $\Gamma \vdash \varphi$ is $(\exists_i)$, then $\varphi = \exists_x \psi$ and the premise of the last rule is $\psi[x/t]$ as depicted in the following figure, where again $\{\gamma_1, \ldots, \gamma_n\}$ is the subset of formulas of Γ used in the derivation:

$$\dfrac{\dfrac{\gamma_1 \quad \cdots \quad \gamma_n}{\psi[x/t]}}{\exists_x \psi}\ (\exists_i)$$

The subtree rooted by the formula $\psi[x/t]$ and with open leaves labeled by formulas of Γ, corresponds to a derivation of the sequent $\Gamma \vdash \psi[x/t]$ that by induction hypothesis implies $\Gamma \models \psi[x/t]$. Therefore, any interpretation I that makes the formulas in Γ true, also makes $\psi[x/t]$ true. Thus, since $I \models \psi[x/t]$ implies $I\frac{x}{t^I} \models \psi$, one has that $I \models \exists_x \psi$. Therefore, $\Gamma \models \exists_x \psi$.

Finally, for a derivation of the sequent $\Gamma \vdash \varphi$ that finishes with an application of the rule $(\exists_e)$, one has as premises the formulas $\exists_x \psi$ and φ. The former labels a root of a subtree with open leaves labeled by assumptions in $\{\gamma_1, \ldots, \gamma_n\} \subset \Gamma$ that corresponds to a derivation for the sequent $\Gamma \vdash \exists_x \psi$; the later labels a subtree with open leaves in $\{\gamma_1, \ldots, \gamma_n\} \cup \{\psi[x/x_0]\}$ and corresponds to a derivation for the sequent $\Gamma, \psi[x/x_0] \vdash \varphi$, where x_0 is a variable that does not occur free in $\Gamma \cup \{\varphi\}$, as depicted in the figure below:

$$\dfrac{\begin{array}{c}\gamma_1 \quad \cdots \quad \gamma_n\\ \hline \vdots \\ \exists_x \psi\end{array} \qquad \begin{array}{c}[\psi[x/x_0]]^u \; \gamma_1 \quad \cdots \quad \gamma_n\\ \hline \vdots \\ \varphi\end{array}}{\varphi}\ (\exists_e)\ u$$

By induction hypothesis, one has $\Gamma \models \exists_x \psi$ and $\Gamma, \psi[x/x_0] \models \varphi$. The first means that for any interpretation I such that $I \models \Gamma$, $I \models \exists_x \psi$. Thus, there exists some $a \in \mathrm{D}$, the domain of I, such that $I\frac{x}{a} \models \psi$. Notice also that since x_0 does not occur in Γ, one has that $I\frac{x_0}{a} \models \Gamma$. From the second, since $I\frac{x_0}{a} \models \Gamma, \psi[x/x_0]$, one has that $I\frac{x_0}{a} \models \varphi$. But, since x_0 does not occurs in φ, one concludes that $I \models \varphi$. $\square$

Exercise 30 Complete all other cases of the proof of the Theorem 6 of soundness of predicate logic.

2.5.2 Completeness of the Predicate Logic

The completeness proof for the predicate logic is not a direct extension of the completeness proof for the propositional logic. The completeness theorem was first proved by Kurt Gödel, and here we present the general idea of a proof due to Leon Albert Henkin (for nice complete presentations see references mentioned in the chapter on suggested readings).

The kernel of the proof is based on the fact that *every consistent set of formulas is satisfiable*, where consistency of the set Γ means that the absurd is not derivable from Γ:

Definition 28 A set Γ of predicate formulas is *consistent* if not $\Gamma \vdash \bot$.

Note that if we assume that **every consistent set is satisfiable** then the completeness can be easily obtained as follows:

Theorem 7 (Completeness) *Let Γ be a set of predicate formulas. If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$.*

Proof We prove that not $\Gamma \vdash \varphi$ implies not $\Gamma \models \varphi$. From not $\Gamma \vdash \varphi$ one has that $\Gamma \cup \{\neg\varphi\}$ is consistent because if $\Gamma \cup \{\neg\varphi\}$ were inconsistent then $\Gamma \cup \{\neg\varphi\} \vdash \bot$ by definition, and one could prove φ as follows:

$$\dfrac{\begin{array}{c}\Gamma, [\neg\varphi]^a\\ \vdots \\ \bot\end{array}}{\varphi}\ (\mathrm{PBC})\ a$$

Therefore, $\Gamma \vdash \varphi$, which contradicts the supposition that not $\Gamma \vdash \varphi$. Now, since $\Gamma \cup \{\neg\varphi\}$ is consistent, by the assumption that consistent sets are satisfiable, we have that $\Gamma \cup \{\neg\varphi\}$ is satisfiable. Therefore, we conclude that not $\Gamma \models \varphi$. □

Our goal from now on is to prove that **every consistent set of formulas is satisfiable**. The idea is, given a consistent set of predicate formulas Γ, to build a model I for Γ, and since the sole available information is its consistency, this must be done by purely syntactical means that is using the language to build the desired model.

The key concepts in Henkin's proof are the notion of *witnesses* of existential formulas and extension of consistent sets of formulas to *maximally consistent* sets.

Definition 29 (*Witnesses and maximally consistency*) Let Γ be a set of formulas

Γ *contains witnesses* if and only if for every formula of the form $\exists_x \varphi$ in Γ, there exists a term t such that $\Gamma \vdash \exists_x \varphi \to \varphi[x/t]$.

Γ is *maximally consistent* if and only if for each formula φ, $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$.

Notice that from the definition, for any possible extension of a maximally consistent set Γ, say Γ' such that $\Gamma \subseteq \Gamma'$, $\Gamma' = \Gamma$. Maximally consistent sets are also said to be *closed for negation*.

The proof is done in two steps, and uses the fact that every subset of a satisfiable set is also satisfiable:

1. every consistent set can be extended to a maximally consistent set containing witnesses;
2. every maximally consistent set containing witnesses has a model.

If Γ does not contain witnesses, these formulas cannot be built in a straightforward manner, since one cannot choose any arbitrary term t to be witness of the existential formula without changing the semantics. Nevertheless, any consistent set can be extended to another consistent set containing witnesses. The simplest case is when the language is countable and the set Γ uses only a finite set of free variables that is $\mathtt{fv}(\Gamma)$ is finite. Since the set of existential formulas is also countable and there are infinite unused variable (those that do not appear free in Γ). Then these variables can be used as witnesses without any conflict. The other cases are more elaborated and are left as research exercises to the reader (Exercises 32 and 33): the case in which the language is countable, but Γ uses infinitely many free variables and the case in which the language is not countable.

In the sequel we will treat the simplest case in which the set of constant, function, and predicate symbols occurring in Γ is at most countable and there are only finitely many variables occurring in Γ. The next two lemmas complete the first part of the proof: a consistent set might be extended to a maximally consistent set with witnesses. This is done proving first how variables might be used to include witnesses and then how a consistent set with witnesses can be extended to a maximally consistent set.

Lemma 4 (Construction of witnesses) *Let Γ be a consistent set over a countable language such that $\texttt{fv}(\Gamma)$ is finite. There exists an extension $\Gamma' \supseteq \Gamma$ over the same language, such that Γ' is consistent and contains witnesses.*

Proof Let $\exists_{x_1}\varphi_1, \exists_{x_2}\varphi_2, \ldots$ be an enumeration of all the existential formulas built over the language. Let $y_1, y_2, \ldots$ be an enumeration of the variables not occurring free in Γ, and consider the formulas below, for $i > 0$:

$$(\exists_{x_i}\varphi_i) \to \varphi_i[x_i/y_i]$$

Let Γ_0 be defined as Γ, and Γ_n, for $n > 0$ be defined as shown below:

$$\Gamma_n = \Gamma_{n-1} \cup \{(\exists_{x_n}\varphi_n) \to \varphi_n[x_n/y_n]\}$$

We will prove the consistence of Γ' defined as $\Gamma' = \bigcup_{n\in\mathbb{N}} \Gamma_n$ by induction on n. The base case is trivial since Γ is consistent by hypothesis. For $k > 0$, suppose Γ_{k-1} is consistent, but Γ_k is not, i.e.

$$\Gamma_k = \Gamma_{k-1} \cup \{(\exists_{x_k}\varphi_k) \to \varphi_k[x_k/y_k]\} \vdash \bot \tag{2.1}$$

Now consider the following derivation:

$$\dfrac{(\text{LEM})\ (\exists_{x_k}\varphi_k) \vee \neg(\exists_{x_k}\varphi_k) \qquad \begin{array}{c}\Gamma_{k-1}\ [\exists_{x_k}\varphi_k]^a\\ \nabla_1\\ \bot\end{array} \qquad \begin{array}{c}\Gamma_{k-1}\ [\neg\exists_{x_k}\varphi_k]^b\\ \nabla_2\\ \bot\end{array}}{\bot}\ (\vee e)\ a\ b$$

where

$$\nabla_1: \quad \dfrac{[\exists_{x_k}\varphi_k]^a \qquad \dfrac{\Gamma_{k-1} \qquad \dfrac{[\varphi_k[x_k/y_k]]^u}{\exists_{x_k}\varphi_k \to \varphi_k[x_k/y_k]}\ (\to i)\emptyset}{\bot}\ (2.1)}{\bot}\ (\exists e)u$$

and

$$\nabla_2: \quad \dfrac{\Gamma_{k-1} \qquad \dfrac{\dfrac{[\neg\exists_{x_k}\varphi_k]^b}{\neg\varphi_k[x_k/y_k] \to \neg\exists_{x_k}\varphi_k}\ (\to i)\emptyset}{\exists_{x_k}\varphi_k \to \varphi_k[x_k/y_k]}\ (\text{CP})}{\bot}\ (2.1)$$

But this is a proof of $\Gamma_{k-1} \vdash \bot$ which contradicts the assumption that Γ_{k-1} is consistent. Therefore, Γ_k is consistent. $\square$

In the previous proof, note that if $\Gamma_{i-1} \vdash \exists_{x_i} \varphi_i$ then it must be the case that $\Gamma_i \vdash \varphi_i[x_i/y_i]$ in order to preserve the consistency. Therefore, $\varphi_i[x_i/y_i]$ might be added to the set of formulas, but not its negation, as will be seen in the further construction of maximally consistent sets.

Now we prove that every maximally consistent set containing witnesses has a model.

Lemma 5 (Lindenbaum) *Each consistent set of formulas* Γ *over a countable language is contained in a maximally consistent set* Γ^* *over the same language.*

Proof Let $\delta_1, \delta_2, \ldots$ be an enumeration of the formulas built over the language. In order to build a consistent expansion of Γ we recursively define the family of indexed sets of formulas Γ_i as follows:

- $\Gamma_0 = \Gamma$
- $\Gamma_i = \begin{cases} \Gamma_{i-1} \cup \{\delta_i\}, & \text{if } \Gamma_{i-1} \cup \{\delta_i\} \text{ is consistent;} \\ \Gamma_{i-1}, & \text{otherwise.} \end{cases}$

Now let $\Gamma^* = \bigcup\limits_{i \in \mathbb{N}} \Gamma_i$. We claim that Γ^* is maximally consistent. In fact, if Γ^* is not maximally consistent then there exists a formula $\gamma \notin \Gamma^*$ such that $\Gamma^* \cup \{\gamma\}$ is consistent. But by the above enumeration, there exists $k \geq 1$ such that $\gamma = \delta_k$, and since $\Gamma_{k-1} \cup \{\gamma\}$ should be consistent, $\delta_k \in \Gamma_{k+1}$. Hence $\delta_k = \gamma \in \Gamma^*$. $\square$

From the previous Lemmas 4 and 5, one has that every consistent set of formulas built over a countable set of symbols and with finitely many free variables can be extended to a maximally consistent set which contains witnesses. In this manner we complete the first step of the prove.

Now, we will complete the second step of the proof that is that any maximally consistent set that contain witnesses is satisfiable. We start with two auxiliary definitional observations.

Lemma 6 *Let* Γ *be a maximally consistent set of formulas. Then for any formula* φ *either* $\varphi \in \Gamma$ *or* $\neg\varphi \in \Gamma$.

Lemma 7 *Let* Γ *be a maximally consistent set. For any formula* φ, $\Gamma \vdash \varphi$ *if, and only if* $\varphi \in \Gamma$.

Proof Suppose $\Gamma \vdash \varphi$. From Lemma 6, either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$. If $\neg\varphi \in \Gamma$ then Γ would be inconsistent:

$$\dfrac{\begin{array}{c}\Gamma \\ \nabla \\ \varphi\end{array} \qquad \begin{array}{c}\Gamma \\ \nabla \\ \neg\varphi\end{array}}{\bot}\ (\neg_e)$$

Therefore, $\varphi \in \Gamma$. $\square$

We now define a model that is called the *algebra* or *structure of terms* for the set Γ which is assumed to be maximally consistent and containing witnesses. The model, denoted as I_Γ, is built from Γ by taking as domain, the set $\mathtt{D}$ of all terms built over the countable language of Γ as given in the definition of terms Definition 13. The designation d for each variable is the same variable and the interpretation of each non-variable term is itself too: $t^{I_\Gamma} = t$. Notice that since our predicate language does not deal with equality symbol, different terms are interpreted as different elements of $\mathtt{D}$. The map $\mathtt{m}$ of I_Γ maps each n-ary function symbol in the language, f, in the function f^{I_Γ} such that for all terms $t_1, \ldots, t_n$, $(f(t_1, \ldots, t_n))^{I_\Gamma} = f^{I_\Gamma}(t_1^{I_\Gamma}, \ldots, t_n^{I_\Gamma}) = f(t_1, \ldots, t_n)$, and for each n-ary predicate symbol p, p^{I_Γ} is the relation defined as

$$(p(t_1, \ldots, t_n))^{I_\Gamma} = p^{I_\Gamma}(t_1^{I_\Gamma}, \ldots, t_n^{I_\Gamma}) \text{ if and only if } p(t_1, \ldots, t_n) \in \Gamma$$

With these definitions we have that for any atomic formula φ, $\varphi \in \Gamma$ if and only if $I_\Gamma \models \varphi$. In addition, according to the interpretation of quantifiers, for any atomic formula $\forall_{x_1} \ldots \forall_{x_n} \varphi \in \Gamma$ if and only if $I_\Gamma \models \forall_{x_1} \ldots \forall_{x_n} \varphi$ and $\exists_{x_1} \ldots \exists_{x_n} \varphi \in \Gamma$ if and only if $I_\Gamma \models \exists_{x_1} \ldots \exists_{x_n} \varphi$.

Using the assumptions that Γ has witnesses and is maximally consistent, formulas can be correctly interpreted in I_Γ as below.

1. $\bot^{I_\Gamma} = F$ and $\top^{I_\Gamma} = T$
2. $\varphi^{I_\Gamma} = T$, iff $\varphi \in \Gamma$, for any atomic formula φ
3. $(\neg\varphi)^{I_\Gamma} = T$, iff $\varphi^{I_\Gamma} = F$
4. $(\varphi \wedge \psi)^{I_\Gamma} = T$, iff $\varphi^{I_\Gamma} = T$ and $\psi^{I_\Gamma} = T$
5. $(\varphi \vee \psi)^{I_\Gamma} = T$, iff $\varphi^{I_\Gamma} = T$ or $\psi^{I_\Gamma} = T$
6. $(\varphi \to \psi)^{I_\Gamma} = T$, iff $\varphi^{I_\Gamma} = F$ or $\psi^{I_\Gamma} = T$
7. $(\exists_x \varphi)^{I_\Gamma} = T$, iff $(\varphi[x/t])^{I_\Gamma} = T$, for some term $t \in \mathtt{D}$
8. $(\forall_x \varphi)^{I_\Gamma} = T$, iff $(\varphi[x/t])^{I_\Gamma} = T$, for all $t \in \mathtt{D}$.

Indeed, this interpretation is well-defined only under the assumption that Γ has witnesses and is maximally consistent. For instance, the item 3 is well-defined since $\neg\varphi \in \Gamma$ if and only if not $\varphi \in \Gamma$. For the item 5, if $(\varphi \vee \psi) \in \Gamma$ and not $\varphi \in \Gamma$, by maximally consistency one has that $\neg\varphi \in \Gamma$; thus, from $(\varphi \vee \psi)$ and $\neg\varphi$, it is possible to derive ψ (by simple application of rules $(\vee_e)$ and $(\neg_e)$ and $(\bot_e)$). Similarly, if we assume $(\varphi \vee \psi) \in \Gamma$ and not $\psi \in \Gamma$, we can derive φ. For the item 6, suppose $(\varphi \to \psi) \in \Gamma$ and $\varphi \in \Gamma$, then one can derive ψ (by application of $(\to_e)$); otherwise, if $(\varphi \to \psi) \in \Gamma$ and not $\psi \in \Gamma$, by maximally consistency, $\neg\psi \in \Gamma$, from which one can infer $\neg\varphi$ (by application of contraposition). For the item 7, if we assume $\exists_x \varphi \in \Gamma$, by the existence of witnesses, there is a term t such that $\exists_x \varphi \to \varphi[x/t] \in \Gamma$, and from these two formulas we can derive $\varphi[x/t]$ (by a simple application of rule $(\to_e)$).

Exercise 31 Complete the analysis well-definedness for all the items in the interpretation of formulas I_Γ, for a set Γ that contains witnesses and is maximally complete.

Theorem 8 (Henkin) *Let* Γ *be a maximally consistent set containing witnesses. Then for all* φ,

$$I_\Gamma \models \varphi, \text{ if, and only if } \Gamma \vdash \varphi.$$

Proof The proof is done by induction on the structure of φ. If φ is an atomic formula then $\varphi \in \Gamma$ iff $(\varphi)^{I_\Gamma} = T$, by definition.
If $\varphi = \neg\varphi_1$ then

$$\begin{array}{ll}
\neg\varphi_1 \in \Gamma & \Longleftrightarrow \text{(because } \Gamma \text{ is maximally consistent)} \\
\varphi_1 \notin \Gamma & \Longleftrightarrow \text{(by induction hypothesis)} \\
\text{not } I_\Gamma \models \varphi_1 & \Longleftrightarrow \text{(by definition)} \\
I_\Gamma \models \neg\varphi_1. &
\end{array}$$

If $\varphi = \varphi_1 \wedge \varphi_2$ then:

$$\begin{array}{ll}
\varphi_1 \wedge \varphi_2 \in \Gamma & \Longleftrightarrow \text{(by definition)} \\
\varphi_1 \in \Gamma \text{ and } \varphi_2 \in \Gamma & \Longleftrightarrow \text{(by induction hypothesis for both } \varphi_1 \text{ and } \varphi_2\text{)} \\
I_\Gamma \models \varphi_1 \text{ and } I_\Gamma \models \varphi_2 & \Longleftrightarrow \text{(by definition)} \\
I_\Gamma \models \varphi_1 \wedge \varphi_2. &
\end{array}$$

If $\varphi = \varphi_1 \vee \varphi_2$ then:

$$\begin{array}{ll}
\varphi_1 \vee \varphi_2 \in \Gamma & \Longleftrightarrow \text{(by definition)} \\
\varphi_1 \in \Gamma \text{ or } \varphi_2 \in \Gamma & \Longleftrightarrow \text{(by induction hypothesis for both } \varphi_1 \text{ and } \varphi_2\text{)} \\
I_\Gamma \models \varphi_1 \text{ or } I_\Gamma \models \varphi_2 & \Longleftrightarrow \text{(by definition, no matter the condition holds for } \varphi_1 \text{ or } \varphi_2\text{)} \\
I_\Gamma \models \varphi_1 \vee \varphi_2. &
\end{array}$$

If $\varphi = \varphi_1 \to \varphi_2$ then we split the proof into two parts. First, we show that $\varphi_1 \to \varphi_2 \in \Gamma$ implies $I_\Gamma \models \varphi_1 \to \varphi_2$. We have two subcases

1. $\varphi_1 \in \Gamma$: In this case, $\varphi_2 \in \Gamma$. In fact, if $\varphi_2 \notin \Gamma$ then $\neg\varphi_2 \in \Gamma$ by the maximality of Γ, and Γ becomes contradictorily inconsistent:

$$\dfrac{\dfrac{\varphi_1 \to \varphi_2 \quad \varphi_1}{\varphi_2}\,(\to_e) \qquad \neg\varphi_2}{\bot}\,(\neg_e)$$

Thus, by induction hypothesis one has

$$\begin{array}{ll}
\varphi_1 \in \Gamma \text{ and } \varphi_2 \in \Gamma & \Longleftrightarrow \text{(by induction hypothesis for both } \varphi_1 \text{ and } \varphi_2\text{)} \\
I_\Gamma \models \varphi_1 \text{ and } I_\Gamma \models \varphi_2 & \Longrightarrow \text{(by definition)} \\
I_\Gamma \models \varphi_1 \to \varphi_2. &
\end{array}$$

2. $\varphi_1 \notin \Gamma$: In this case, $\neg\varphi_1 \in \Gamma$ by the maximality of Γ. Therefore,

$$\begin{array}{lll} \neg\varphi_1 \in \Gamma & \Longleftrightarrow & \text{(by induction hypothesis)} \\ I_\Gamma \models \neg\varphi_1 & \Longleftrightarrow & \text{(by definition)} \\ \text{not } I_\Gamma \models \varphi_1 & \Longrightarrow & \text{(by definition)} \\ I_\Gamma \models \varphi_1 \rightarrow \varphi_2. & & \end{array}$$

Now we prove that $I_\Gamma \models \varphi_1 \rightarrow \varphi_2$ implies $\varphi_1 \rightarrow \varphi_2 \in \Gamma$. By definition of the semantics of implication, there are two cases

1. $\varphi_1^{I_\Gamma} = F$: In this case, we have that $(\neg\varphi_1)^{I_\Gamma} = T$, and hence $\neg\varphi_1 \in \Gamma$, by induction hypothesis. We can now derive $\varphi_1 \rightarrow \varphi_2$ as follows, and conclude by Lemma 7:

$$\cfrac{\cfrac{\cfrac{\neg\varphi_1 \qquad [\varphi_1]^a}{\perp}\,(\neg_e)}{\varphi_2}\,(\perp_e)}{\varphi_1 \rightarrow \varphi_2}\,(\rightarrow_i)\ a$$

2. $\varphi_2^{I_\Gamma} = T$: By induction hypothesis $\varphi_2 \in \Gamma$, and we derive $\varphi_1 \rightarrow \varphi_2$ as follows, and conclude by Lemma 7

$$\cfrac{\varphi_2}{\varphi_1 \rightarrow \varphi_2}\,(\rightarrow_i)\ \emptyset$$

If $\varphi = \exists_x \varphi_1$ then

$$\begin{array}{lll} \exists_x\varphi_1 \in \Gamma & \Longleftrightarrow & \text{(for some } t \in \mathtt{D}\text{, since } \Gamma \text{ contains witnesses)} \\ \varphi_1[x/t] \in \Gamma & \Longleftrightarrow & \text{(by induction hypothesis)} \\ I_\Gamma \models \varphi_1[x/t] & \Longleftrightarrow & \text{(by definition)} \\ I_\Gamma \models \exists_x\varphi_1. & & \end{array}$$

If $\varphi = \forall_x \varphi_1$ then

$$\begin{array}{lll} \forall_x\varphi_1 \in \Gamma & \Longleftrightarrow & \text{(otherwise } \Gamma \text{ becomes inconsistent as shown below)} \\ \varphi_1[x/t] \in \Gamma,\ \text{for all } t \in \mathtt{D} & \Longleftrightarrow & \text{(by induction hypothesis)} \\ I_\Gamma \models \varphi_1[x/t],\ \text{for all } t \in \mathtt{D} & \Longleftrightarrow & \text{(by definition)} \\ I_\Gamma \models \forall_x\varphi_1. & & \end{array}$$

For the first equivalence, note that if $\forall_x\varphi_1 \in \Gamma$ then $\varphi_1[x/t] \in \Gamma$, for all term $t \in \mathtt{D}$, otherwise Γ becomes contradictorily inconsistent

$$\dfrac{\neg\varphi_1[x/t] \qquad \dfrac{\forall_x \varphi_1}{\varphi_1[x/t]}\,(\forall_e)}{\perp}\,(\perp_e)$$ □

Using as a model I_Γ, it is possible to conclude, in this case, that consistent sets are satisfiable.

Corollary 2 (Consistency implies satisfiability) *If* Γ *is a consistent set of formulas over a countable language with a finite set of free variables then* Γ *is satisfiable.*

Proof Initially, Γ is consistently enlarged obtaining the set Γ' including witnesses according to the construction in Lemma 4; afterwards, Γ' is closed maximally obtaining the set $(\Gamma')^*$according to the construction in Lindenbaum's Lemma 5. This set contains witnesses and is maximally consistent; then, by Henkin's Theorem 8, I_Γ is a model of $(\Gamma')^*$, hence a model of Γ too. □

Exercise 32 **(*)** Research in the suggested related references how a consistent set built over a countable set of symbols, but that uses infinite free variables can be extended to a maximal consistent set with witnesses. The problem is that in this case there are no new variables that can be used as witnesses. Thus, one needs to extend the language with new constant symbols that will act as witnesses, but each time a new constant symbol is added to the language the set of existential formulas change.

Exercise 33 **(*)** Research the general case in which the language is not restricted, that is the case in which Γ is built over a non-countable set of symbols.

2.5.3 Compactness Theorem and Löwenheim-Skolem Theorem

The connections between $\models$ and $\vdash$ as well as between consistence and satisfiability provided in this section, give rise to other additional important consequences that relate semantic and syntactic elements of the predicate logic. Here we present two important theorems that are related with the scope and limits of the expressiveness of predicate logic.

Theorem 9 (Compactness) *Given a set* Γ *of predicate formulas and a formula* φ*, the following holds:*

i. $\Gamma \models \varphi$ *if and only if there is a finite set* $\Gamma_0 \subseteq \Gamma$ *such that* $\Gamma_0 \models \varphi$
ii. Γ *is satisfiable if and only if for all finite set* $\Gamma_0 \subseteq \Gamma$*,* Γ_0 *is satisfiable.*

Proof i. For necessity, if $\Gamma \models \varphi$, by completeness there exists a derivation ∇ for $\Gamma \vdash \varphi$. The derivation ∇ uses only a finite subset of assumptions, say $\Gamma_0 \subseteq \Gamma$. Thus, $\Gamma_0 \vdash \varphi$ and, by correctness, one concludes that $\Gamma_0 \models \varphi$. For sufficiency, suppose that $\Gamma_0 \models \varphi$, for a finite set $\Gamma_0 \subseteq \Gamma$. By completeness there exists a derivation ∇ for $\Gamma_0 \vdash \varphi$. But ∇ is also a derivation for $\Gamma \vdash \varphi$; hence, by correctness one concludes that $\Gamma \models \varphi$.

ii. Necessity is proved by contraposition: if Γ_0 were unsatisfiable for some finite set $\Gamma_0 \subseteq \Gamma$, then Γ_0 would be inconsistent, since consistency implies satisfiability (Corollary 2); thus, $\Gamma_0 \vdash \bot$, which implies also that $\Gamma \vdash \bot$ and by correctness that $\Gamma \models \bot$. Hence, Γ would be unsatisfiable. Sufficiency is proved also by contraposition: if we assume that Γ is unsatisfiable, then since there exists no model for Γ, $\Gamma \models \bot$ holds. By completeness also, $\Gamma \vdash \bot$ and hence, there exists a finite set $\Gamma_0 \subseteq \Gamma$, such that $\Gamma_0 \vdash \bot$, which by correctness implies that $\Gamma_0 \models \bot$. Thus, we conclude that Γ_0 is unsatisfiable.

□

The compactness theorem has several applications that are useful for restricting the analysis of consistency and satisfiability of arbitrary sets of predicate formulas to only finite subsets. This also has important implications in the possible cardinality of models of sets of predicate formulas such as the one given in the following theorem.

Theorem 10 (Löwenheim–Skolem) *Let Γ be a set of formulas such that for any natural $n \in \mathbb{N}$, there exists a model of Γ with a domain of cardinality at least n. Then Γ has also infinite models.*

Proof Consider an additional binary predicate symbol E and the formulas φ_n for $n > 0$, defined as

$$\forall_x \; E(x, x) \wedge \exists_{x_1,\ldots,x_n} \bigwedge_{i \neq j; i,j=1}^{n} \neg E(x_i, x_j)$$

For instance, the formulas φ_1 and φ_3 are given respectively as $\forall_x E(x, x)$ and $\forall_x \; E(x, x) \wedge \exists_{x_1} \exists_{x_2} \exists_{x_3} \; (\neg E(x_1, x_2) \wedge \neg E(x_1, x_3) \wedge \neg E(x_2, x_3))$.

Notice that φ_n has models of cardinality at least n. It is enough to interpret E just as a the reflexive relation among the elements of the domain of the interpretation. Thus, pairs of different elements of the domain do not belong to the interpretation of E.

Let Φ be the set of formulas $\{\varphi_n \mid n \in \mathbb{N}\}$. We will prove that all finite subsets of the set of formulas $\Gamma \cup \Phi$ are satisfiable and then by the compactness theorem conclude that $\Gamma \cup \Phi$ is satisfiable too. An interpretation $I \models \Gamma \cup \Phi$ should have an infinite model, since also $I \models \Phi$ and all formulas in Φ are true in I only if there are infinitely many elements in the domain of I.

To prove that any finite set $\Gamma_0 \subset \Gamma \cup \Phi$ is satisfiable, let k be the maximum k such that $\varphi_k \in \Gamma_0$. Since Γ has models of arbitrary finite cardinality, let I' be a model of Γ with at least k elements in its domain $\mathtt{D}$. I' can be extended in such a manner that the binary predicate symbol E is interpreted just as the reflexive relation over $\mathtt{D}$. Let I be the extended interpretation. It is clear that $I \models \Gamma$ since E is a new symbol and also $I \models \Gamma_0 \cap \Phi$ since the domain has at least k different elements. Also, since $I \models \Gamma$, we have that $I \models \Gamma \cap \Gamma_0$. Hence, $I \models \Gamma_0$ and so we conclude that Γ_0 is satisfiable. □

Exercise 34 Prove that there is no predicate formula φ that holds exclusively for all finite interpretations.

Exercise 35 Let E be a binary predicate symbol, e a constant and $\cdot$ and -1 be binary and unary function symbols, respectively. The theory of groups is given by the models of the set of formulas Γ_G

$$\begin{gathered}
\forall_x\ E(x, x) \\
\forall_{x,y}\ (E(x, y) \to E(y, x)) \\
\forall_{x,y,z}\ (E(x, y) \wedge E(y, z) \to E(x, z)) \\
\forall_x\ E(x \cdot e, x) \\
\forall_x\ E(x \cdot x^{-1}, e) \\
\forall_{x,y,z}\ E((x \cdot y) \cdot z, x \cdot (y \cdot z))
\end{gathered}$$

Notice that according to the three first axioms the symbol E should be interpreted as an equivalence relation such as the equality. Indeed, the three other axioms are those related with group theory itself: the fourth one states the existence of an identity element, the fifth one the inverse function and the sixth one the associativity of the binary operation.

Prove the existence of infinite models by proving that for any $n \in \mathbb{N}$, the structure of arithmetic modulo n is a group of cardinality n. The elements of this structure are all integers modulo n (i.e., the set $\{0, 1, \ldots, n-1\}$), with addition and identity element 0.

Exercise 36 A graph is a structure of the form $G = \langle V, E \rangle$, where V is a finite set of vertices and $E \subset V \times V$ a set of edges between the vertices. The problem of reachability in graphs is the question whether there exists a finite path of *consecutive* edges, say $(u, u_1), (u_1, u_2), \ldots, (u_{n-1}, v)$, between two given nodes $u, v \in V$.

Prove that there is no predicate formula that expresses reachability in graphs. Hint: the key observation to conclude is that the problem of reachability between two nodes might be answered positively whenever there exists a path of arbitrary length.

2.6 Undecidability of the Predicate Logic

The gain of expressiveness obtained in predicate logic w.r.t. to the propositional logic comes at a price. Initially, remember that for a given propositional formula φ, one can always answer whether φ is valid or not by analyzing its truth table. This means that there is an algorithm that receives an arbitrary propositional formula as input and **always** answers after a finite amount of time `yes`, if the given formula is valid; or `no`, otherwise. The algorithm works as follows: build the truth table for φ and check whether it is true for all interpretations. Note that this algorithm is not efficient because the (finite) number of possible interpretations grows exponentially w.r.t. the number of propositional variables occurring in φ.

In general, a computational question with a `yes` or `no` answer depending on the parameters is known as a *decision problem*. A decision problem is said to be *decidable* whenever there exists an algorithm that correctly answers `yes` or `no` for each instance of the problem, and when such algorithm does not exist the decision problem is said to be *undecidable*. Therefore, we conclude that that *validity* is decidable in propositional logic.

The natural question that arises at this point is whether validity is decidable or not in predicate logic. Note that the truth table approach is no longer possible because the number of different interpretations for a given predicate formula φ is not finite. In fact, as stated in the previous paragraph the gain of expressiveness of the predicate logic comes at a price: validity is undecidable in predicate logic. This fact is usually known as the *undecidability of predicate logic*, and has several important consequences. In fact, it is straightforward from the completeness of predicate logic that provability is also undecidable, i.e., there is no algorithm that receives a predicate formula φ as input and returns `yes` if $\vdash \varphi$, or `no` if not $\vdash \varphi$.

The standard technique for proving the undecidability of the predicate logic consists in reducing a known undecidable problem to the validity of the predicate logic in such a way that decidability of validity of the predicate logic entails the decidability of the other problem leading to a contradiction. In what follows, we consider the word problem for a specific monoid introduced by G.S. Tseitin, and that is well-known to be undecidable.

A semigroup is an algebraic structure with a binary associative operator $\cdot$ over a given set A. When in addition the structure has an identity element id which is called a monoid. By associativity, one understands that for all x, y, z in A, $x \cdot (y \cdot z) = (x \cdot y) \cdot z$, and for all $x \in A$ the identity satisfies the properties $id \cdot x = x$ and $x \cdot id = x$. In general, the word problem in a given semigroup with a given set of equations E (between pairs of elements of A), is the problem of answering whether two words are equal *applying* these equations.

By an application of an equation, say $u = v$ in E, one can understand an equational transformation of the form below, where x y are any elements of A.

$$x \cdot (u \cdot y) = x \cdot (v \cdot y)$$

Hence, the word problem consists in answering for any pair of elements $x, y \in A$ if there exists a finite chain, possibly of length zero, of applications of equations that transform x in y:

$$x \equiv x_0 \overset{u_1=v_1}{=} x_1 \overset{u_2=v_2}{=} x_2 \overset{u_3=v_3}{=} \dots \overset{u_n=v_n}{=} x_n \equiv y \tag{2.2}$$

In the chain above, the notation $\equiv$ is used for syntactic equality and $\overset{u_i=v_i}{=}$ for highlighting that the equation applied in the application step is $u_i = v_i$.

Tseitin's monoid is given by the set Σ^* of words freely generated by the quinary alphabet $\Sigma = \{a, b, c, d, e\}$. In this structure, the binary associative operator is the concatenation of words and the empty word plays the role of the identity. The set of equations is given below. For simplicity, we will omit parentheses and the concatenation operator.

$$\begin{aligned} ac &= ca \\ ad &= da \\ bc &= cb \\ bd &= db \\ ce &= eca \\ de &= edb \\ cdca &= cdcae \end{aligned} \tag{2.3}$$

As previously mentioned, Tseitin introduced this specific monoid with the congruence generated by this set of equations and proved that the word problem in this structure is undecidable.

In order to reduce the above problem to the validity of the predicate logic, we choose a logical language with a constant symbol $\Box$, five unary function symbols f_a, f_b, f_c, f_d, and f_e, and a binary predicate P. The constant $\Box$ will be interpreted as the empty word, and each function symbol, say $f_\star$ for $\star \in \Sigma$, as the concatenation of the symbol $\star$ to the left of the term given as argument of $f_\star$. For example, the word $baaecde$ will be encoded as $f_b(f_a(f_a(f_e(f_c(f_d(f_e(\Box)))))))$, which for brevity will be written simply as $f_{baaecde}(\Box)$. The binary predicate P will play the role of equality, i.e., $P(x, y)$ is interpreted as x is equal to y (modulo the congruence induced by the set of equations above, which would be assumed as axioms).

Our goal is, given an instance of the word problem $x, y \in \Sigma^*$ specified above, to build a formula $\varphi_{x,y}$ such that x equals y in this structure if and only if $\models \varphi_{x,y}$. The formula $\varphi_{x,y}$ is of the form

$$\varphi' \to P(f_x(\Box), f_y(\Box)) \tag{2.4}$$

where φ' is the following formula:

$$\begin{aligned} \forall_x (P(x, x)) \wedge \\ \forall_x \forall_y (P(x, y) \to P(y, x)) \wedge \\ \forall_x \forall_y, \forall_z (P(x, y) \wedge P(y, z) \to P(x, z)) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_{ac}(x), f_{ca}(y))) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_{ad}(x), f_{da}(y))) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_{bc}(x), f_{cb}(y))) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_{bd}(x), f_{db}(y))) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_{ce}(x), f_{eca}(y))) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_{de}(x), f_{edb}(y))) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_{cdca}(x), f_{cdcae}(y))) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_a(x), f_a(y))) \wedge \\ \forall_x \forall_y (P(x, y) \to P(f_b(x), f_b(y))) \wedge \end{aligned} \tag{2.5}$$

$$\forall_x \forall_y (P(x, y) \to P(f_c(x), f_c(y)))\wedge$$
$$\forall_x \forall_y (P(x, y) \to P(f_d(x), f_d(y)))\wedge$$
$$\forall_x \forall_y (P(x, y) \to P(f_e(x), f_e(y)))$$

Suppose $\models \varphi_{x,y}$. Our goal is to find a model for $\varphi_{x,y}$ which tells us if there is a solution to the instance $x, y \in \Sigma^*$. Consider the interpretation I with domain Σ^* and such that

- the constant $\square$ is interpreted as the empty word;
- each unary function symbol $f_\star$, for $\star \in \Sigma$, is interpreted as the function $f_\star^I : \Sigma^* \to \Sigma^*$ that appends the symbol $\star$ to the word $x \in \Sigma^*$ given as argument, i.e., $f_\star^I(x) = \star x$;
- and the binary predicate P is interpreted as follows:
$P(x, y)^I$ if and only if there exists a chain, possibly of length zero, of applications of the Eqs. (2.3) that transform x into the word y.

We claim that $I \models \varphi'$. Let us consider each case

- $I \models \forall x(P(x, x))$: take the empty chain.
- $I \models \forall_x \forall_y (P(x, y) \to P(y, x))$: for any x, y such that $I \models P(x, y)$, take the chain given for $P(x, y)$ in reverse order.
- $I \models \forall_x \forall_y \forall_z (P(x, y) \wedge P(y, z) \to P(x, z))$: for any x, y, z such that $I \models P(x, y)$ and $I \models P(y, z)$, append the chains given for $P(x, y)$ and $P(y, z)$.
- $I \models \forall_x \forall_y (P(x, y) \to P(f_{ac}(x), f_{ca}(y)))$: for any x, y such that $I \models P(x, y)$, take the chain given for $P(x, y)$ and use this for the chain of equations for $acx = acy$; then add an application of the equation $ac = ca$ to obtain cay. A similar justification is given for all other cases related with Eqs. (2.3), but the last.
- $I \models \forall_x \forall_y (P(x, y) \to P(f_\star(x), f_\star(y)))$ where $\star \in \Sigma$: for any x, y such that $I \models P(x, y)$, take the chain given for $P(x, y)$ and use it for the chain for the equation $\star x = \star y$.

Since $I \models \varphi_{x,y}$ and $I \models \varphi'$, we conclude that $I \models P(f_x(\square), f_y(\square))$. Therefore, the instance x, y of the word problem has a solution.

Conversely, suppose the instance x, y of the word problem has a solution in Tseitin's monoid; i.e., there is a chain of applications of the Eqs. (2.3) from x resulting in the word y as given in the chain (2.2). We will suppose that this chain is of length n.

We need to show that $\varphi_{x,y}$ is valid; i.e., that $\models \varphi_{x,y}$. Let us consider an arbitrary interpretation I' over a domain $\mathtt{D}$ with an element $\square^{I'}$, five unary functions $f_a^{I'}$, $f_b^{I'}$, $f_c^{I'}$, $f_d^{I'}$, $f_e^{I'}$ and a binary relation $P^{I'}$. Since $\varphi_{x,y}$ is equal to $\varphi' \to P(f_u(\square), f_v(\square))$, we have to show that if $I' \models \varphi'$ then $I' \models P(f_u(\square), f_v(\square))$.

We proceed by induction in n, the length of the chain of applications of Eqs. (2.3) for transforming x in y.

IB: case $n = 0$, we have that $x \equiv y$ and if $I' \models \varphi'$, $I' \models \forall_x P(x, x)$ which also implies that $I' \models P(x, x)$.

IS: case $n > 0$, the chain of applications of equations to transform x in y is of the form

$$x \equiv x_0 \overset{u_1=v_1}{=} x_1 \overset{u_2=v_2}{=} x_2 \overset{u_3=v_3}{=} \dots x_{n-1} \overset{u_n=v_n}{=} x_n \equiv y$$

By induction hypothesis we have that $I' \models P(x, x_{n-1})$. If we prove that $I' \models P(x_{n-1}, y)$, we can conclude that $I' \models P(x, y)$, since $I' \models \forall_x \forall_y \forall_z P(x, y) \wedge P(y, z) \to P(x, z)$ because we are assuming that $I' \models \varphi'$.

Thus, the proof resumes to prove that equalities obtained by one step of application of equations in (2.3) hold in I': in particular if we suppose that $u_n = v_n$ is the equation $u = v$ in (2.3), $x_{n-1} \equiv wuz$ and $y = wvz$, we need to prove that $I' \models P(f_{wuz}(\Box), f_{wvz}(\Box))$, which is done by the following three steps:

1. First, one has that $I' \models P(f_z(\Box), f_z(\Box))$, since $I' \models \forall_x P(x, x)$.
2. Second, since $u = v$ in (2.3), $I' \models \forall_x \forall_y (P(x, y) \to P(f_u(x), f_v(y)))$. Thus, by the previous item one has that $I' \models P(f_{uz}(\Box), f_{vz}(\Box))$;
3. Third, $I' \models P(f_{wuz}(\Box), f_{wvz}(\Box))$ is obtained from the last item, inductively on the length of w, since $I' \models \forall_x \forall_y (P(x, y) \to P(f_\star(x), f_\star(y)))$, for all $\star \in \Sigma$.

To conclude the undecidability of validity of the predicate logic, if we suppose the contrary, we will be able to answer for any $x, y \in \Sigma^*$ if $\models P(f_x(\Box), f_y(\Box))$ answering consequently if x equals y in Tseitin's monoid, which is impossible since the word problem in this structure is undecidable.

Theorem 11 (Undecidability of the Predicate Logic) *Validity in the predicate logic that is answering whether for a given formula φ, $\models \varphi$ is undecidable.*

Notice that by Gödel completeness theorem undecidability of validity immediately implies undecidability of derivability in the predicate logic. Indeed, in the above reasoning one can use the completeness theorem to alternate between validity and derivability.

Exercise 37 Accordingly to the three steps above to prove $I' \models P(f_{wuz}(\Box), f_{wvz}(\Box))$, build a derivation for the sequent $\vdash P(f_{wuz}(\Box), f_{wvz}(\Box))$. Concretely, prove that

a. $\varphi' \vdash P(f_z(\Box), f_z(\Box))$, for $z \in \Sigma^*$;
b. $\varphi', P(f_z(\Box), f_z(\Box)) \vdash P(f_{uz}(\Box), f_{vz}(\Box))$, for $u = v$ in the set of equations (2.3);
c. $\varphi', P(f_{uz}(\Box), f_{vz}(\Box)) \vdash P(f_{wuz}(\Box), f_{wvz}(\Box))$, for $w \in \Sigma^*$;
d. $\varphi' \vdash P(f_{wuz}(\Box), f_{wvz}(\Box))$.

Springer

http://www.springer.com/978-3-319-51651-6

Applied Logic for Computer Scientists
Computational Deduction and Formal Proofs
Ayala-Rincón, M.; de Moura, F.L.C.
2017, XVIII, 150 p. 4 illus., Softcover
ISBN: 978-3-319-51651-6