Lecture Notes for

Logic and Computability

Course Number: IND04033UF

Contact

Bettina Könighofer
Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Austria
bettina.koenighofer@iaik.tugraz.at

Table of Contents

6 Predicate Logic

The Limitations of Propositional Logic. Propositional logic has limitations when encoding declarative sentences. It is quite easy to encode logical sentence components like *not*, *and*, *or*, *if ... then*. However, we would also like to express sentence components like *there exists ...* and *for all ...* which is not possible in propositional logic. To overcome this limitation, we need a more powerful type of logic like e.g. *predicate logic*, also called *first-order logic*.

Example. Model the following sentence in propositional logic:

"Every person who is 18 years or older is eligible to vote."

Solution. p with $p :=$"Every person who is 18 years or older is eligible to vote."

Note, that the above statement cannot be adequately expressed using only propositional logic, since propositional logic is not expressive enough to deal with quantified variables. Since the sentence is not referring to a specific person and the statement applies to all people who are 18 years or older, we are stuck. Therefore we need a richer logic like predicate logic.

6.1 Predicates and Quantifiers

Predicate logic is an extension of propositional logic. It adds the concept of predicates and quantifiers to better capture the meaning of statements that cannot be adequately expressed by propositional logic.

Predicates

A predicate is a function that takes one or more variables from a specific domain and returns true or false depending on the values of its variables. We denote predicates with *capital roman letters* such as P, Q, and R. A statement involving n variables $x_1, x_2, x_3, \ldots, x_n$ can be denoted by an n-ary predicate $P(x_1, x_2, x_3, \ldots, x_n)$. Once truth value has been assigned to all the variables $x_1, x_2, x_3, \ldots, x_n$, the statement $P(x_1, x_2, x_3, \ldots, x_n)$ becomes true or false.

Example. Write a formula φ in predicate logic that models the following sentence: "x is smaller than 5."

Solution. $\varphi \coloneqq STFive(x)$ using the predicate $STFive$ that returns true if $x < 5$ and false otherwise.

Example. What are the truth values of $STFive(4)$ and $STFive(6)$?

Solution. $STFive(4)$ is true, since it is the case that $4 < 5$ holds. $STFive(6)$ is false since it is not true that $6 < 5$.

Example. Let $I(x, y)$ denote the predicate that compares whether x is equivalent to $y + 1$, i.e., it returns true if $x \equiv y + 1$ and false otherwise. Give the truth values for $I(6, 5)$ and $I(1, 4)$?

Solution. $I(6, 5)$ is true since $6 = 5 + 1$. $I(1, 4)$ is false since $1 \neq 4 + 1$.

Quantifiers

Quantifiers are used to express the extent to which a predicate is true over a range of elements. Using quantifiers to create such propositions is called quantification. We distinguish between two types of quantification: the *universal quantification* and the *existential quantification.*

Universal Quantifier $\forall$

We can express statements which assert that *a property is true for all the values of a variable in a particular domain* using universal quantification. The notation $\forall x P(x)$ denotes the universal quantification of $P(x)$. $\forall x P(x)$ is read as "for all x P(x)". *The formula* $\forall x P(x)$ *evaluates to true if* $P(x)$ *is true for all values of* x *in a given domain.* It is very important to explicitly specify the domain when using universal quantification since the domain decides the possible values of x. Without the domain, the universal quantification has no meaning.

Example. Let $P(x) \coloneqq$ "x + 5 > x" with $x \in \mathbb{N}^+$. What is the truth value of the statement $\forall x P(x)$?

Solution. As $x + 5 > x$ for any positive natural number, $P(x) \equiv true$ for all x and therefore it holds that $\forall x P(x) \equiv true$.

Existential Quantifier $\exists$

We can express statements which assert that *there is an element with a certain property* by existential quantification. The notation $\exists x P(x)$ denotes the existential quantification of $P(x)$, i.e., $\exists x P(x)$ is true if and only if $P(x)$ is true

for at least one value of x in the domain. $\exists x P(x)$ is read as "There is at least one such x such that P(x)" and denotes the statement. *The formula $\exists x P(x)$ is true if there exists an element x in the domain such that $P(x)$."*

Example. Let $P(x)$:="x $>$ 10" with $x \in \mathbb{N}^+$. What is the truth value of the statement $\exists x P(x)$?

Solution. $P(x)$ is true for all natural numbers greater than 10 and false for all natural numbers less than 10. Therefore, the formula $\exists x P(x)$ is true.

Example. Model the following sentence in predicate logic:

"Every person who is 18 years or older is eligible to vote."

Solution. Using the domain $x \in People$ and the predicates $P(x)$:="x is 18 years", $R(x)$:="x is older than 18 years" and $Q(x)$:="x is eligible to vote" we get:

$$\forall x((P(x) \vee R(x)) \leftrightarrow Q(x)).$$

Example. Model the following sentence in predicate logic:

"Every lecturer is older than some student."

Solution. We define the following predicates for $x, y \in People$:

$$L(x) := \text{"}x \text{ is a lecturer"}$$
$$S(x) := \text{"}x \text{ is a student"}$$
$$O(x, y) := \text{"}x \text{ is older than } y\text{"}$$

Using these predicates, we can model the sentence as follows:

$$\forall x(L(x) \rightarrow \exists y(S(y) \wedge O(x, y)))$$

Example. Formalize the following reasoning using propositional logic and show whether the sequent is valid:

"*Every child has a mother. Maria is a child. Therefore, Maria has a mother.*"

Solution. We define the following atomic propositions:

$$p := \text{"}Every\ child\ has\ a\ mother.\text{"}$$
$$q := \text{"}Maria\ is\ a\ child.\text{"}$$
$$r := \text{"}Maria\ has\ a\ mother.\text{"}$$

Using propositional logic results in the following sequent:

$$p, q \vdash r$$

The sequent can easily disproven by the following counterexample: $\mathcal{M} := \{p = \top, q = \top, r = \bot\}$.

Example. Formalize the reasoning from above using predicate logic.

Solution. We define the following predicates:

$$C(x) := \text{“}x \textit{ is a child}\text{”}$$
$$M(x) := \text{“}x \textit{ has a mother}\text{”}$$

Using the domain of $x, y \in People$, we can model the sequent as follows:

$$\forall x(C(x) \leftrightarrow M(x)), \quad C(maria) \quad \vdash \quad M(maria)$$

Using predicate logic, the meaning of the statements is preserved such that reasoning is possible. In the next chapter, we will extend the natural deduction calculus to predicate logic such that we are able to prove such sequents.

Example. Model the following sentence in predicate logic:

”Niki and Ben have the same maternal grandmother.“

Solution. We use the binary predicate $M(x, y) :=$*“x is the mother of y.”* with $x, y \in People$ and get the following formula:

$$\forall x \forall y \forall u \forall v(M(x, y) \land M(y, niki) \land M(u, v) \land M(v, ben) \rightarrow x = u).$$

The formula states that, if y and v are Niki's and Ben's mothers, respectively, and x and u are their mothers (i.e. Ben's and Niki's maternal grandmothers, respectively), then x and u are the same person. Note, that we use the infix notation for the equality predicate instead of the prefix notation. Whenever it feels more natural you can use the infix notation. You can always use the equality predicate without defining it explicitly.

What are functions?

A function is an expression of one or more variables determined on some specific domain and returns a value from that domain. We denote functions with lowercase capital roman letters such as f, g, and h.

Example. Express the statement from before using the function symbol $m(x)$ that returns the mother of x.

Solution. Using the function m, the sentences from above can be encoded as follows:

$$m(m(niki)) = m(m(ben))$$

6.2 Syntax of Predicate Logic

In this section we define the syntactic rules that define well-formed formulas in predicate logic.

First, note that in predicate logic, we have two types of *sorts*. First, we have *terms* that talk about objects. Terms include individual objects (e..g, *ben*, *niki*),

variables since they represent objects (e.g., x, y), and function symbols since they refer to objects (e.g., $m(x)$).

Second, we have *formulas* that have a truth value. For example, $P(x, f(y)$ is a formula and x, y, and $f(x)$ are terms.

$$\overbrace{\underbrace{P}_{\text{Predicate}}\ (\ \underbrace{x}_{\text{Term}}\ ,\underbrace{f(y)}_{\text{Term}})}^{\text{Formula}}$$

To define the syntax of predicate logic, we use the following notation:

- V: Defines the set of variable symbols, e.g., x, y, z.
- $\mathcal{F}$: Defines the set of function symbols, e.g., f, g, h.
- $\mathcal{P}$: Defines the set of predicate symbols, e.g., P, Q, R.

Each predicate symbol and each function symbol comes with an arity, the number of arguments it expects. *Constants* are functions which don't take any arguments, i.e., nullary functions.

Terms

Terms are defined as follows:

- Any variable is a term.
- If $c \in \mathcal{F}$ is a nullary function, then c is a term.
- If $t_1, t_2, ..., t_n$ are terms and $f \in \mathcal{F}$ has arity $n > 0$, then $f(t_1, t_2, ..., t_n)$ is a term.
- Nothing else is a term.

Note that the first building blocks of terms are constants (nullary functions) and variables. More complex terms are built from function symbols using as many previously built terms as required by such function symbols.

Formulas

Formulas are defined as follows:

- If $P \in \mathcal{P}$ is a predicate with arity $n > 0$ and t_1, t_2, ..., t_n are terms over $\mathcal{F}$, then $P(t_1, t_2, ..., t_n)$ is a formula.
- If ϕ is a formula, then $\neg\phi$ is a formula.
- If ϕ and ψ are formulas, then $(\phi \wedge \psi)$, $(\phi \vee \psi)$, and $(\phi \rightarrow \psi)$ are formulas.
- If ϕ is a formula and x is a variable, then $(\forall x\ \phi)$ and $(\exists x\ \phi)$ are formulas.
- Nothing else is a formula.

Note, that arguments, that are given to a predicate, are always terms.

Binding Priorities

We add to the binding priorities that we defined for porpositional logic the convention that $\forall x$ and $\exists x$ bings like $\neg$. Therefore, the precedence rules can be given by:

1. $\forall$, $\exists$, $\neg$ bind most tightly;
2. then $\wedge$;
3. then $\vee$;
4. then $\rightarrow$ which is right-associative.

Parse Tree

The parse tree is constructed in the same way as for formulas in propositional logic, but with two additional sorts of nodes:

- Nodes for the quantifiers $\forall x$ and $\exists x$ have one subtree.
- Predicates of the form $P(t_1, t_2, \ldots, t_n)$ have the symbol P as a node with n subtrees, namely the parse trees of the terms $t_1, t_2, \ldots, t_n$.

Example. Draw the syntax tree to the following formula:

$$\forall x \Big(\big(P(x) \rightarrow \neg Q(x)\big) \wedge \big(S(x, f(y, z)) \vee T(y)\big) \Big).$$

Solution.

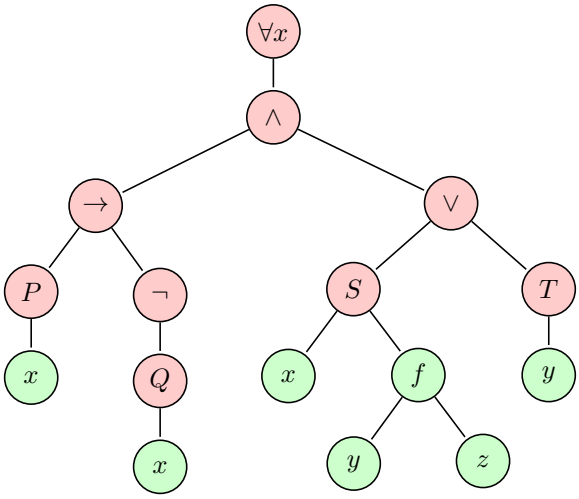

Figure 6.1: A syntax tree of a predicate logic formula.
Red nodes are formulas, green nodes are terms.

6.3 Free and Bound Variables

With the introduction of $\forall$ and $\exists$, we also have to think about the scope of a these quantifiers and whether a variable is free or bound.

Definition - Free and Bound Variables. *Let ϕ be a formula in predicate logic. An occurrence of x in ϕ is free if it is a leaf node in the syntax tree of ϕ such that there is no path upwards from that node x to a node $\forall x$ or $\exists x$. Otherwise, that occurrence of x is called bound.*

Definition - Scope of a Quantifier. *For $\forall x\phi$, or $\exists x\phi$, we say that ϕ is the scope of $\forall x$, respectively $\exists x$.*

In other words, if x occurs in ϕ, then it is bound if, and only if, it is in the scope of some $\exists x$ or some $\forall x$; otherwise it is free. In terms of parse trees, the scope of a quantifier is its subtree.

Example. Construct a parse tree for the following formula ϕ and determine the scope of its quantifiers and which occurrences of the variables are free and which are bound:

$$\phi \coloneqq \forall x\ (P(x) \vee Q(y, x)) \wedge R(x)$$

Solution. The syntax tree with labels denoting free and bound variables can be seen in Figure 6.2.

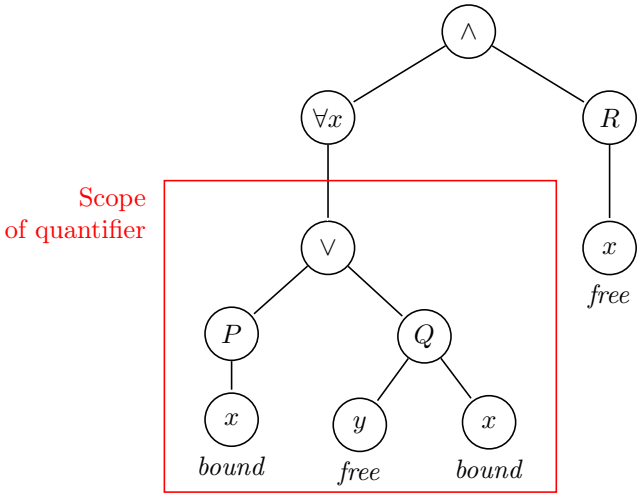

Figure 6.2: A syntax tree illustrating free and bound variables.

The scope of $\forall x$ is $(P(x) \vee Q(y, x))$. Thus, the first two occurrences of x are bound to $\forall x$. Starting for the y leave node, the only quantifier we run into is $\forall x$ but that x has nothing to do with y. So y is free in this formula. The third occurrence of x is also free.

6.3.1 Substitution

Variables are place holders and can be *replaced* with more concrete information.

Definition - Substitution. *Given a variable x, a term t and a formula ϕ we define $\phi[t/x]$ to be the formula obtained by replacing* ***each free occurrence*** *of variable x in ϕ with t.*

$$\phi[\underbrace{t}_{\text{Term}} / \underbrace{x}_{\text{Variable}}]$$

Therefore, if all occurrences of x are bound in ϕ, none of them gets substituted by t. Furthermore, it is *not allowed* to perform substitutions such that *a variable gets* captured *by a quantifier*: meaning that the variable was free before and by carrying out the substitution it gets bound by a quantifier.

Example. Compute $\phi[f(z)/x]$ for the following formula:

$$\phi \coloneqq \forall y\big(P(x) \land Q(y)\big) \lor \big(R(y) \land Q(x)\big)$$

Solution. All occurrences of x are free and can be replayed by $f(z)$.

$$\phi\left[f(z)/x\right] = \forall y\ \big(P(f(z)) \land Q(y)\big) \lor \big(R(y) \land Q(f(z))\big)$$

Example. Compute $\psi[f(z)/x]$ for the following formula:

$$\psi = \forall x\big(P(x) \land Q(y)\big) \lor \big(R(y) \land Q(x)\big)$$

Solution. We can only substitute the x in $Q(x)$, as the other occurrence x is bound to the $\forall x$ quantifier.

$$\psi\left[f(z)/x\right] = \forall x\ \big(P(x) \land Q(y)\big) \lor \big(R(y) \land Q(f(z))\big)$$

Example. Compute $\phi[f(y)/x]$ for the following formula:

$$\phi \coloneqq \forall y\big(P(x) \land Q(y)\big) \lor \big(R(y) \land Q(x)\big)$$

Solution. We are not allowed to replace x with $f(y)$, since x is free before the substitution and the term $f(y)$ contains a y which is in the scope of the $\forall y$ quantifier. Therefore, the variable would be captured which is not allowed.

$$\phi\left[f(y)/x\right] \coloneqq \forall y\big(P(x) \land Q(y)\big) \lor \big(R(y) \land Q(x)\big)$$

6.4 Semantics of Predicate Logic

We will extend the notion of *models* that we discussed for propositional logic to predicate logic. In propositional logic, a model defined an assignment of truth values to all variables such that the formula evaluated to true or to false.

A model in predicate logic differs from a model in propositional logic in the treatment of predicates and functions.

6.4.1 Models

A model in predicate logic needs to define a concrete meaning to all predicate and function symbols involved. For example, the predicate P is defined in the model to be the relation *"greater than"* on the set of real numbers.

Definition - Model in Predicate Logic. *A model $\mathcal{M}$ consists of the following set of data:*

- *A non-empty set $\mathcal{A}$, the universe/domain of concrete values*;
- *for each nullary function symbol $f \in \mathcal{F}$, a concrete element $f^{\mathcal{M}} \in \mathcal{A}$*;
- *for each nullary predicate symbol $P \in \mathbb{P}$, a truth value*;
- *for each function symbol $f \in \mathcal{F}$ with arity $n > 0$, a concrete function $f^{\mathcal{M}} : \mathcal{A}^n \to \mathcal{A}$*;
- *for each predicate symbol $P \in \mathbb{P}$ with arity $n > 0$: subset $P^{\mathcal{M}} \subseteq \mathcal{A}^n$.*
- *for any free variable* `var`: a lookup-table $l : \texttt{var} \to \mathcal{A}$.

To denote a *concrete* instance of a function f or a predicate P in a model $\mathcal{M}$, we use the notation $f^{\mathcal{M}}$ and $P^{\mathcal{M}}$. We often define $P^{\mathcal{M}}$ as tuples which make P true and use function tables to define $f^{\mathcal{M}}$.

Example. Give a model $\mathcal{M}$ for the following formula:

$$\phi \coloneqq \forall x \exists y P(x, y)$$

Solution. The model consists of a domain $\mathcal{A}$ and a concrete predicate instance $P^{\mathcal{M}}$. We give one possible model for ϕ:

- $\mathcal{A} = \{a, b\}$
- $P^{\mathcal{M}} = \{(a, b), (b, a)\}$ (meaning $P(a, b) = true$, $P(b, a) = true$, for all other cases, P evaluates to $false$)

6.4.2 Evaluate a Formula under a Model

Given a model $\mathcal{M}$, we define the satisfaction relation $\mathcal{M} \vDash \phi$ for each logical formula ϕ by structural induction on ϕ.

- P : If ϕ is of the form $P(t_1, t_2, \ldots, t_n)$, then we interpret the terms $t_1, t_2, \ldots, t_n$ in our set $\mathcal{A}$ by replacing all variables with their values according to the look-up table l and interpret any function symbols $f \in \mathcal{F}$ by $f^{\mathcal{M}}$. In this way we compute concrete values $a_1, a_2, ..., a_n$ of $\mathcal{A}$ for each of these terms. Now $\mathcal{M} \vDash P(t_1, t_2, \ldots, t_n)$ holds if and only if $(a_1, a_2, ..., a_n)$ is in the set $P^{\mathcal{M}}$.
- $\forall x$: The relation $\mathcal{M} \vDash \forall x \psi$ holds if and only if $\mathcal{M} \vDash_{l[x \to a]} \psi$ holds **for all** $a \in \mathcal{A}$.
- $\exists x$: Dually, $\mathcal{M} \vDash \exists x \psi$ holds if and only if $\mathcal{M} \vDash_{l[x \to a]} \psi$ holds **for some** $a \in \mathcal{A}$.

Example. Given a model $\mathcal{M}: \ \mathcal{A} = \{a, b\}, \ P^{\mathcal{M}} = \{(a, b), (b, a)\}$ and a formula $\phi \coloneqq \forall x \exists y P(x, y)$. Does it hold that $\mathcal{M} \vDash \phi$?

Solution. We need to show that for any possible value for x, there exists a value for y, such that P evaluates to true. Since $P(a, b)$ and $P(b, a)$ are both true, this is the case and it holds that $\mathcal{M} \vDash \phi$.

Example. Given a model $\mathcal{M}: \ \mathcal{A} = \{a, b\}, \ P^{\mathcal{M}} = \{(a, b), (b, a)\}$ and a formula $\psi = \exists x \forall y \ P(x, y)$. Does it hold that $\mathcal{M} \vDash \psi$?

Solution. We need to show that there is value for x, such that for all possible values for y, P evaluates to true. We perform the following substitutions:

- x and y substituted with a: $P(a, a) = \bot$. Therefore, we try the next substitution for x.
- x with b and y with a: $P(b, a) = \top$
- x with b and y with b: $P(b, b) = \bot$

Therefore, there is no such x for which P evaluates to true under all possible values for y. Therefore, $M \nvDash \psi$.

Example. Give a formula $\psi = \exists x \forall y \ P(x, y)$ and a model $\mathcal{M}: \ A = \mathbb{N}, \ P^{\mathcal{M}} = x \leq y \mid P^M = \{(1, 1), (1, 2) \dots (2, 2), \dots\}$. Does it hold that $\mathcal{M} \vDash \psi$?

Solution. Let us substitute x with 1. We need to show that P evaluates to true for all values of y.

- y substituted with 1: $P(1, 1) = \top$
- y substituted with 2: $P(1, 2) = \top$
- $\dots$
- y substituted with n: $P(1, n) = \top$ for any $n > 1$

Therefore, we can conclude that $\mathcal{M} \vDash \psi$.