

Lecture Notes for

Logic and Computability

Course Number: IND04033UF

by

Bettina Könighofer

Bernd Grabner

Belinda Uhl

Contact

Bettina Könighofer

Institute for Applied Information Processing and Communications (IAIK)

Graz University of Technology, Austria

bettina.koenighofer@iaik.tugraz.at

March, 2021



Graz University of Technology

Table of Contents

6	Predicate Logic	1
6.1	Predicates and Quantifiers	1
6.2	Syntax of Propositional Logic	5
6.3	Free and Bound Variables	7
6.3.1	Substitution	8
6.4	Semantics of Predicate Logic	9
6.4.1	Models	10
6.4.2	Evaluate a Formula under a Model	10
6.5	Exercises	12
6.5.1	Examples	12
6.5.2	Solutions	14
6.5.3	Self-Evaluation	17

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.

What is a predicate?

Informally, a predicate is a statement that may be true or false *depending on the values of its variables*. We denote predicates with *capital roman letters* such as P , Q , and R .

Definition - Predicate. *A predicate is an expression of one or more variables determined on some specific domain. A predicate with variables can be made a proposition (evaluate to true or to false) by either authorizing a value to the variable or by quantifying the variable.*

A statement involving n variables $x_1, x_2, x_3, \dots, x_n$ can be denoted by an n -ary predicate $P(x_1, x_2, x_3, \dots, x_n)$.

Example. Consider the following statement:

“ x is smaller than 5.”

Solution. The statement has two parts. The first part, the variable x , is the subject of the statement. The second part, “is smaller than 5”, is the predicate. *It refers to a property that the subject of the statement can have.* We denote the statement “ x is smaller than 5” with $P(x)$ where P denotes the predicate “is smaller than 5” and x is the variable. The predicate P can be considered as a function with Boolean return value. It tells the truth value of the statement $P(x)$ at x . Once a value has been assigned to the variable x , the statement $P(x)$ becomes true or false.

Example. Let $P(x)$ denote the statement “ $x < 5$ ”. What are the truth values of $P(4)$ and $P(6)$?

Solution. $P(4)$ is equivalent to the statement “ $4 < 5$ ”, which is true. $P(6)$ is equivalent to the statement “ $6 < 5$ ”, which is false.

Example. Let $I(x, y)$ denote the statement “ $x = y + 1$ ”. What is the truth value of $I(6, 5)$ and $I(1, 4)$?

Solution. $I(6, 5)$ is the statement “ $6 = 5 + 1$ ”, which is true. $I(1, 4)$ is the statement “ $1 = 4 + 1$ ”, which is false.

What are 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 $\forall xP(x)$ denotes the universal quantification of $P(x)$. $\forall xP(x)$ is read as “for all x $P(x)$ ” and denotes the statement

“ $P(x)$ for all values of x in the domain.”

Therefore, the formula $\forall xP(x)$ 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 $P(x) := “x + 5 > x”$ with $x \in \mathbb{N}^+$. What is the truth value of the statement $\forall xP(x)$?

Solution. As $x + 5$ is greater than x for any positive natural number, $P(x) \equiv T$ for all x and $\forall xP(x) \equiv T$.

Existential Quantifier \exists

We can express statements which assert that *there is an element with a certain property* by existential quantification. The notation $\exists xP(x)$ denotes the existential quantification of $P(x)$, i.e., $\exists xP(x)$ is true if and only if $P(x)$ is true for at least one value of x in the domain. $\exists xP(x)$ is read as “There is at least one such x such that $P(x)$ ” and denotes the statement

“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 $\exists xP(x)$?

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

Example. Model the following sentence in predicate logic:

“Every person who is 18 years or older, is eligible to vote.”

Solution. Using the domain $x \in \text{People}$ and the predicates $P(x) := “x$ is 18 years or older” and $Q(x) := “x$ is eligible to vote” we get:

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

Example. Model the following sentence in predicate logic:

“Every lecturer is older than some student.”

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

$$\begin{aligned} L(x) &:= \text{“}x \text{ is a lecturer”} \\ S(x) &:= \text{“}x \text{ is a student”} \\ O(x, y) &:= \text{“}x \text{ is older than } y\text{”} \end{aligned}$$

Using this predicates, we can model the sentence in predicate logic as following:

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

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:

$$\begin{aligned} p &:= \text{“Every child has a mother.”} \\ q &:= \text{“Maria is a child.”} \\ r &:= \text{“Maria has a mother.”} \end{aligned}$$

Using propositional logic results in the following sequent:

$$p, q \vdash r$$

The sequent can easily be proven to be not valid by the following counterexample: $\mathcal{M} := \{p = \top, q = \top, r = \perp\}$.

Example. Formalize the reasoning from above using predicate logic.

Solution. We define the following predicates:

$$\begin{aligned} C(x) &:= \text{“}x \text{ is a child”} \\ M(x) &:= \text{“}x \text{ has a mother”} \end{aligned}$$

Using this predicates, we can model the sequent using predicate logic as following over the domain of *People*:

$$\forall x (C(x) \leftrightarrow M(x)), C(\text{maria}) \vdash M(\text{maria})$$

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:

“Niko and Ben have the same maternal grandmother.”

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

$$\forall x \forall y \forall u \forall v (M(x, y) \wedge M(y, niko) \wedge M(u, v) \wedge M(v, ben) \rightarrow x = u).$$

The formula states that, if y and v are Niko's and Ben's mothers, respectively, and x and u are their mothers (i.e. Ben's and Niko's maternal grandmothers, respectively), then x and u are the same person. Note, that we used the **equality** predicate. Unlike other predicates, it is usually written in between its arguments rather than before them; that is, we write $x = y$ instead of (x, y) to say that x and y are equal.

What are functions?

A function is an expression of one or more variables determined on some specific domain and returns a value from the domain. The function symbols of predicate logic give us a way of avoiding ugly encodings. 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 m , the sentences from above has a simpler encoding than it had using M :

$$m(m(niko)) = m(m(ben))$$

6.2 Syntax of Propositional 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 , $niko$), 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.

$$\begin{array}{c} \text{Formula} \\ \overbrace{P(x, f(y))} \\ \underbrace{P}_{\text{Predicate}} \quad \underbrace{x}_{\text{Term}} \quad \underbrace{f(y)}_{\text{Term}} \end{array}$$

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, \dots, t_n are terms and $f \in \mathcal{F}$ has arity $n > 0$, then $f(t_1, t_2, \dots, t_n)$ 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, \dots, t_n are terms over \mathcal{F} , then $P(t_1, t_2, \dots, t_n)$ 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 propositional logic the convention that $\forall x$ and $\exists x$ binds 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 $P(t_1, t_2, \dots, t_n)$ have the symbol P as a node with n subtrees, namely the parse trees of the terms t_1, t_2, \dots, t_n .

Example. Draw the syntax tree to the following formula:

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

Solution.

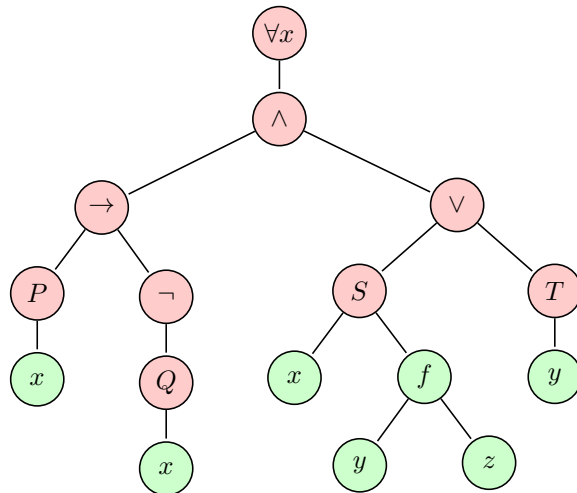


Figure 6.1: A syntax tree of a predicate logic formula.
Red nodes are formulas, green nodes are terms.

6.3 Free and Bound Variables

With the introduction of \forall and \exists , we also have to think about the scope of 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 $\forall x$ and $\exists x$. 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:

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

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

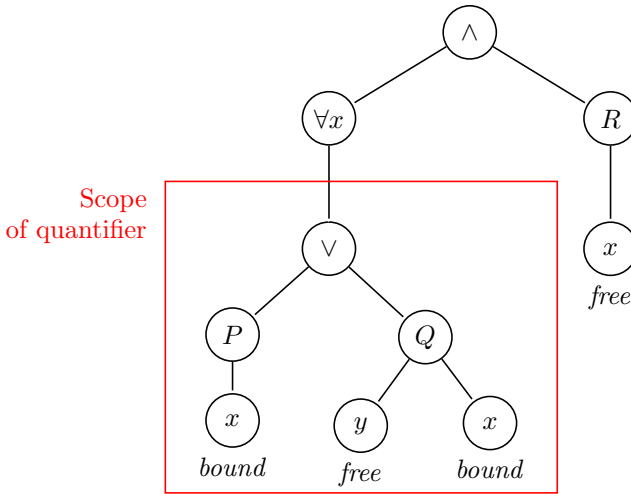


Figure 6.2: A syntax tree illustrating free and bound variables.

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 leaf 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 .

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

In writing $\phi[t/x]$, we really mean this to be the formula *obtained* by performing the replacement operation $[t/x]$ on ϕ .

First note that if all occurrences of x are bound in ϕ , none of them gets substituted by t . Furthermore note that 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:

$$\phi := \forall y (P(x) \wedge Q(y)) \vee (R(y) \wedge Q(x))$$

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

$$\phi [f(z)/x] = \forall y (P(f(z)) \wedge Q(y)) \vee (R(y) \wedge Q(f(z)))$$

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

$$\psi = \forall x (P(x) \wedge Q(y)) \vee (R(y) \wedge Q(x))$$

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

$$\psi [f(z)/x] = \forall x (P(x) \wedge Q(y)) \vee (R(y) \wedge Q(f(z)))$$

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

$$\phi := \forall y (P(x) \wedge Q(y)) \vee (R(y) \wedge Q(x))$$

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.

$$\phi [f(y)/x] := \forall y (P(x) \wedge Q(y)) \vee (R(y) \wedge Q(x))$$

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

In propositional logic, we used truth tables in order to show a certain outcome of a formula. Unfortunately, this is not possible in predicate logic. A model in predicate logic needs to define a concrete meanings to all predicate and function symbols involved. For example, terms could denote real numbers and P could denote 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 $f^{\mathcal{M}} : \mathcal{A}^n \rightarrow \mathcal{A}$;
- 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 $l : \text{var} \rightarrow \mathcal{A}$.

We use the $f^{\mathcal{M}}$ and $P^{\mathcal{M}}$ when we talk about a *concrete* function and predicate in a model \mathcal{M} , whereas we use f and P denote just function and predicate symbols. $P^{\mathcal{M}}$ is defined by tuples, which make P true.

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

$$\phi := \forall x \exists y P(x, y)$$

Solution. In this example, the model consists of a domain \mathcal{A} and a concrete predicate $P^{\mathcal{M}}$. For example, we can define \mathcal{M} :

- $\mathcal{A} = \{a, b\}$
- $P^{\mathcal{M}} = \{(a, b), (b, a)\}$ (meaning $P(a, b) = \text{true}$, $P(b, a) = \text{true}$, 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 $\mathcal{M} \models \phi$ for each logical formula ϕ by structural induction on ϕ .

- P : If ϕ is of the form $P(t_1, t_2, \dots, t_n)$, then we interpret the terms t_1, t_2, \dots, 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, \dots, a_n of \mathcal{A} for each of these terms. Now $\mathcal{M} \models P(t_1, t_2, \dots, t_n)$ holds if and only if (a_1, a_2, \dots, a_n) is in the set $P^{\mathcal{M}}$.
- $\forall x$: The relation $\mathcal{M} \models \forall x \psi$ holds if and only if $\mathcal{M} \models_{l[x \rightarrow a]} \psi$ holds **for all** $a \in \mathcal{A}$.

- $\exists x$: Dually, $\mathcal{M} \models \exists x\psi$ holds if and only if $\mathcal{M} \models_{\{x \rightarrow a\}} \psi$ 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 $\phi := \forall x\exists yP(x, y)$. Does it hold that $\mathcal{M} \models \phi$?

Solution. We need to show that for any possible value for x , there exists a value for y , such that P evaluates to true. By replacing the variable x with the term a , and by replaying the variable y with the term b , we get $P(a, b)$ which is *true*. Furthermore, the substitution from x by b and y by a results in $P(b, a)$ which is again true. Therefore, it holds that $\mathcal{M} \models \phi$.

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 $\mathcal{M} \models \psi$?

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 : $P(a, a) = \perp$. 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 : $P(b, b) = \perp$

Therefore, there is no such x for which P evaluates to true under all possible values for y . Therefore, $\mathcal{M} \not\models \psi$.

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

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

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

6.5 Exercises

6.5.1 Examples

Example 1. Using the domain $\mathcal{A} := \textit{people}$ and the predicates

$$L(x, y) : x \textit{ loves } y$$

$$P(x, y) : x \textit{ plays } y$$

$$G(x) : x \textit{ plays a game}$$

$$P(x) : x \textit{ is a professor}$$

$$S(x) : x \textit{ is a student}$$

translate the following sentences into formulas in predicate logic:

- (a) Some students love Alice.
- (b) Every Professor loves Bob.
- (c) Some students love every professor.
- (d) Some students and Bob play a game.
- (e) Not every student plays games.
- (f) Some professors play no games.

Example 2. Using the domain $\mathcal{A} := \textit{people}$, translate the following sentences into predicate logic. Give the meaning of any function and predicate symbols you use.

- (a) Alice has no sister.
- (b) Not everybody likes everybody.
- (c) A person who wears a crown is either a king or a queen.
- (d) Not all students like both partying and dancing.
- (e) No animal is both a cat and a dog.

Example 3. Draw a syntax tree and state the scope of all quantifiers and all free and bound variables using this formulas:

$$(a) P(x, y) \vee \exists y \forall x (Q(x, y) \wedge R(y, z))$$

$$(b) \forall x \left((P(x, y) \rightarrow P(x, x)) \vee (Q(y, z) \wedge \exists y R(x, y, z)) \right)$$

Example 4. Consider the following formula: $\phi := \forall x \exists y (P(x, y) \wedge Q(y, x))$. Which of the following models satisfy ϕ ?

(a) \mathcal{M}_1 :

- $A = \{a, b\}$
- $P^M = \{(a, b), (b, a)\}$
- $Q^M = \{(a, a), (b, b), (b, a)\}$

(b) \mathcal{M}_2 :

- $A = \{a, b\}$
- $P^M = \{(a, b), (b, a), (b, b)\}$
- $Q^M = \{(a, a), (b, b), (b, a)\}$

(c) \mathcal{M}_3 :

- $A = \mathbb{N}$
- $P^M = \{(m, n) \mid m > n\}$
- $Q^M = \{(m, n) \mid m \neq n\}$

Example 5. For each of the formulas below, find a model that satisfies the formula and one that does not:

(a) $\forall x (P(x, x) \vee P(x, y)) \rightarrow \exists y (Q(y, y) \wedge Q(y, x))$

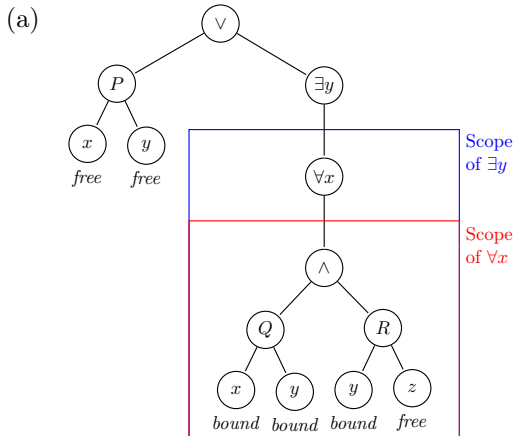
(b) $\exists x \forall y ((P(x, y) \wedge P(y, x)) \vee (Q(x, y) \wedge R(y, x)))$

(c) $\forall x (P(f(x)) \rightarrow Q(f(x)))$

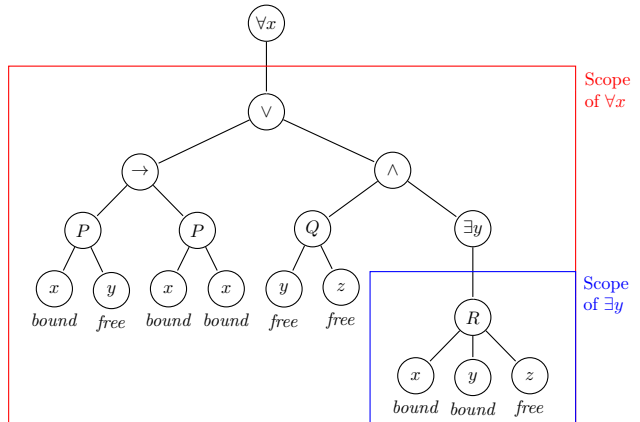
6.5.2 Solutions

- Solution 1.**
- (a) $\exists x (S(x) \wedge L(x, \text{alice}))$
 - (b) $\forall x (P(x) \wedge L(x, \text{bob}))$
 - (c) $\exists x \forall y (S(x) \wedge P(y) \wedge L(x, y))$
 - (d) $\exists x (S(x) \rightarrow (G(x) \wedge G(\text{bob})))$
 - (e) $\neg \forall x (S(x) \wedge G(x))$
 - (f) $\exists x (P(x) \wedge \neg G(x))$

- Solution 2.**
- (a) $S(x, y) : x$ is y 's sister
 $\neg \exists x S(\text{alice}, x)$
 - (b) $P(x) : x$ is a person; $L(x, y) : x$ likes y
 $\neg \forall x \forall y (P(x) \wedge P(y) \wedge L(x, y))$
 - (c) $P(x) : x$ is a person; $C(x) : x$ wears a crown;
 $K(x) : x$ is a king; $Q(x) : x$ is a queen
 $\forall x ((P(x) \wedge C(x)) \rightarrow (K(x) \vee Q(x)))$
 - (d) $S(x) : x$ is a student; $P(x) : x$ likes partying;
 $D(x) : x$ likes dancing
 $\neg \forall x (P(x) \rightarrow (S(x) \wedge P(x)))$
 - (e) $A(x) : x$ is an animal; $C(x) : x$ is a cat;
 $D(x) : x$ is a dog
 $\forall x (A(x) \rightarrow \neg (C(x) \wedge D(x)))$

Solution 3.

(b)

**Solution 4.**

$$\begin{aligned}
 \text{(a)} \quad x = a, y = a &: P(a, a) \wedge Q(a, a) && \perp \\
 x = a, y = b &: P(a, b) \wedge Q(b, a) && \top \\
 x = b, y = a &: P(b, a) \wedge Q(a, b) && \perp \\
 x = b, y = b &: P(b, b) \wedge Q(b, b) && \perp \\
 \mathcal{M}_1 \not\models \phi
 \end{aligned}$$

$$\begin{aligned}
 \text{(b)} \quad x = a, y = a &: P(a, a) \wedge Q(a, a) && \perp \\
 x = a, y = b &: P(a, b) \wedge Q(b, a) && \top \\
 x = b, y = a &: P(b, a) \wedge Q(a, b) && \perp \\
 x = b, y = b &: P(b, b) \wedge Q(b, b) && \top \\
 \mathcal{M}_2 \models \phi
 \end{aligned}$$

$$\begin{aligned}
 \text{(c)} \quad x = 1, y = 1 &: P(1, 1) \wedge Q(1, 1) && \perp \\
 x = 1, y = 2 &: P(1, 2) \wedge Q(2, 1) && \perp \\
 x = 1, y = n &: P(1, n) \wedge Q(n, 1) && \perp \\
 x = 2, y = 1 &: P(2, 1) \wedge Q(2, 1) && \top \\
 x = n, y = 1 &: P(n, 1) \wedge Q(n, 1) && \top \\
 x = n, y = n - 1 &: P(n, n - 1) \wedge Q(n, n - 1) && \top \\
 \mathcal{M}_3 \not\models \phi
 \end{aligned}$$

Solution 5.(a) Model \mathcal{M}_1 satisfies the formula:

$$\mathcal{A} = \{a, b\}$$

$$P^{\mathcal{M}} = \{(a, a), (b, b)\}$$

$$Q^{\mathcal{M}} = \{(a, a), (b, b)\}$$

Model \mathcal{M}_2 does not satisfy the formula:

$$\mathcal{A} = \{a, b\}$$

$$P^{\mathcal{M}} = \{(a, a), (b, b)\}$$

$$Q^{\mathcal{M}} = \{\} = \perp$$

(b) Model \mathcal{M}_1 satisfies the formula:

$$\mathcal{A} = \{a, b\}$$

$$P^{\mathcal{M}} = \{(a, a), (a, b), (b, a)\}$$

$$Q^{\mathcal{M}} = \{\} = \perp$$

$$R^{\mathcal{M}} = \{\} = \perp$$

Model \mathcal{M}_2 does not satisfy the formula:

$$\mathcal{A} = \{a, b\}$$

$$P^{\mathcal{M}} = \{\} = \perp$$

$$Q^{\mathcal{M}} = \{\} = \perp$$

$$R^{\mathcal{M}} = \{\} = \perp$$

(c) Model \mathcal{M}_1 satisfies the formula:

$$\mathcal{A} = \{a, b\}$$

$$f(a) = a; f(b) = b$$

$$P^{\mathcal{M}} = \{a\}$$

$$Q^{\mathcal{M}} = \{a\}$$

Model \mathcal{M}_2 does not satisfy the formula:

$$\mathcal{A} = \{a, b\}$$

$$f(a) = a; f(b) = b$$

$$P^{\mathcal{M}} = \{a, b\}$$

$$Q^{\mathcal{M}} = \{\}$$

6.5.3 Self-Evaluation

These examples will be solved online during class.

Example 1. Translate the following sentences into predicate logic. Give the meaning of any functions and predicate symbols you use.

- All humans need water.
- Not everybody likes coffee.

Example 2. For each of the formulas below, find a model that satisfies the formula and one that does not. Please draw a syntax tree and state all free variables while solving this task.

$$(a) \exists x(P(x, x)) \wedge \forall y(P(x, y))$$

$$(b) \forall x \exists y(P(f(y)) \wedge P(x)) \rightarrow Q(f(f(y)))$$

Example 3. In the following list, tick all items that are required for a complete model of a formula φ in predicate logic.

- A non-empty, possibly infinite set of values for terms.
- A concrete value for every bound variable in φ .
- A definition for each predicate in φ , detailing for which values/tuples the predicate returns *true*.
- A look-up table, assigning *true* or *false* to every term in φ .

Example 4. Consider the formula $\phi := \exists x \forall y(P(x, y) \rightarrow (Q(x, y) \vee R(x, y)))$. Does the following model \mathcal{M} satisfy the formula?

$$\mathcal{A} = \{a, b\}$$

$$P^{\mathcal{M}} = \{(a, a), (a, b)\}$$

$$Q^{\mathcal{M}} = \{(m, a) | m \in \mathcal{A}\}$$

$$R^{\mathcal{M}} = \{(a, a), (b, b)\}$$